

# Tecnologie di Sicurezza in Internet: livello rete

Gianluca Mazzini  
Rev 0.83

# Elementi caratterizzanti della sicurezza

- Confidenzialità
- Integrità
- Disponibilità
- Autenticità

# 1. Confidenzialità

- Solo le entità autorizzate possono leggere le informazioni
- Solo le entità autorizzate possono sapere che la comunicazione è in corso

# 2. Integrità

- Correttezza: deve essere possibile verificare che non siano stati introdotti errori dalla trasmissione e/o dal processamento
- Consistenza: il dato deve avere una consistenza semantica intrinseca e deve essere possibile effettuare una verifica
- E' possibile effettuare alcune modifiche, durante il trasporto. Queste devono rientrare all'interno di una tipologia predefinita. Le modifiche possono essere effettuate solo da entità autorizzate

### 3. Disponibilità

- Il dato deve rispettare determinate caratteristiche, opportunamente definite a priori
- Tempistiche: il dato può essere integro, ma se impiega troppo tempo ad essere consegnato (trasporto e processamento) può non essere più usabile
- Equità: il sistema deve essere scalabile rispetto al numero di utenti potenzialmente coinvolti, senza privilegi o sacrifici

### 4. Autenticità

- Occorre avere certezza sul mittente del dato e su chi, eventualmente, lo ha modificato
- Il mittente non deve poter negare la sua identità
- Il mittente non deve poter assumere l'identità di terzi

# Analisi dei costi

- Finanziari: valutazione dei costi in base alle esigenze
- Complessità: analisi dei compromessi tra memoria, banda a disposizione, tempi di processamento, disponibilità del dato

# Analisi dei rischi e piano di sicurezza

- Valore dei singoli oggetti: chi accede ai dati del progetto, costo dell'intero progetto, chi accede al servizio, valore dell'intero servizio, conseguenze economiche di una violazione, conseguenze giuridiche di una violazione
- Meccanismi gerarchici di attribuzione delle competenze. Esempio: demoni funzionanti come super-utente, demoni confinati allo spazio di uno username senza accesso alle restanti porzioni del filesystem
- Individuazione dei punti deboli
- Individuazione dei sistemi di monitoraggio per il controllo sistematico dei punti deboli

# Tipologie di attacco

- Interruzione: viene interrotto un flusso dati attivo tra sorgente e destinatario
- Intercettazione: si cerca di capire il contenuto dei dati
- Modifica: viene cambiato, completamente o parzialmente, il contenuto dei dati
- Fabbricazione: vengono creati dei nuovi dati, eventualmente emulando le fasi di connessione end-to-end. Occorre spesso avere il controllo di almeno un elemento della rete di trasporto

## Come garantire i punti chiave

- Confidenzialità: cifratura, controllo di accesso
- Integrità: controllo di accesso, controllo di consistenza
- Disponibilità: controllo di accesso, ridondanza, monitoring
- Autenticità: firma digitale

# Problemi/attacchi in Internet

- Bug
- Malware (software creato allo scopo di danneggiare computer su cui viene eseguito. Il termine deriva dalle parole inglesi malicious e software e ha dunque il significato letterale di "programma malvagio")
  - Cavalli di Troia
  - Virus
  - Worms
  - Backdoor
  - Spyware
  - Dialer

# Bug

- Di solito involontari
- Errori nel software: una serie di procedure sono state mal progettate e implementate e possono essere causa di malfunzionamenti del sistema. Possono essere gravi al punto da rendere vulnerabile ad attacchi il sistema che ospita il software stesso
- Il più famoso è buffer overflow: fornire una stringa di dati più lunga di quanto atteso, invadendo la zona di memoria dove è stato riposto il program counter (PC) per il ritorno da una procedura. Il PC viene fatto puntare alla stringa dati che contiene un piccolo programma in codice macchina. Al ritorno da una procedura viene eseguito il programma dell'attaccante, con varie possibili conseguenze.
- Meno comunemente bug può indicare un difetto di progettazione in un componente hardware

- Etimologia: il termine bug è legato ad un aneddoto; nel 1945 il tenente Grace Hopper stava cercando le cause del malfunzionamento di un computer Mark II quando si accorse che una falena si era incastrata tra i circuiti.





# Cavalli di Troia

- Tipo di malware le cui funzionalità sono nascoste all'interno di un programma o di una interfaccia utilizzata dall'utente. L'utente, eseguendo o comunque utilizzando le procedure contenenti il cavallo di troia, invoca procedure indesiderate con varie possibili conseguenze.
- Caso Internet: viene registrato un indirizzo molto simile a quello a cui l'utente vuole accedere, un generico utente effettua un errore di digitazione e si ritrova sul sito troiano, immette un identificativo di accesso, il troiano rimanda l'utente al sito reale effettuando il login. L'utente non si accorge di nulla. Il troiano detiene l'identificativo di accesso per successive operazioni.
- Caso software locale: il troiano lascia all'utente la capacità di utilizzare normalmente il programma installato, il troiano si comporta da server e consente accessi tramite Internet a client maliziosi, che possono effettuare varie operazioni sulla macchina dell'utente.
- I troiani non si diffondono autonomamente ma richiedono un intervento diretto dell'attaccante per diffondere l'eseguibile malizioso alla vittima. Spesso è la vittima stessa a ricercare e scaricare il troiano sul proprio computer, durante il download di software

# Virus 1/4

- Parti di codice che si diffondono copiandosi all'interno di: programmi, file eseguibili in diversi formati, aree di boot di memorie permanenti quali dischi fissi, memorie flash, etc.
- Un virus non è generalmente un programma eseguibile. Per essere attivato, deve infettare un programma ospite, o una sequenza di codice che viene lanciata automaticamente.
- Ad esempio: Il virus legge le intestazioni del file (lunghezza, checksum, etc) e le memorizza per riutilizzarle in seguito. Il virus si aggiunge al file e ne riscrive l'intestazione. Le dimensioni del virus e le modifiche che il virus provoca nell'intestazione rappresentano quella che solitamente viene chiamata firma del virus. Il virus inserisce una copia di se stesso nel file da infettare e pone tra le prime istruzioni un comando di salto alla prima linea della sua copia, alla fine della sua esecuzione tornerà ad eseguire la parte restante del file originale. Il virus, quindi, viene solitamente eseguito prima del programma.

# Virus 2/4

- Caratteristiche
  - Resistente all'individuazione
  - Robusto, difficile da disattivare ed eliminare
  - Facilmente propagabile
  - Relativamente semplice da realizzare
  - Di dimensioni contenute
  - Il più possibile indipendente dalla piattaforma
- Target
  - Boot sector (infetta e cambia il loader del SO)
  - Demoni di sistema
  - Librerie
  - Applicazioni
  - Documenti/Dati (Esempio: macro di word, windows meta file)

# Virus 3/4

- Percorsi di infezione
  - Supporti rimovibili
  - Attachment a Email
  - File sharing
  - Applicazioni P2P
- Virus signature (firma del virus)
  - Data in cui è avvenuta la modificata
  - Dimensione del file: per non cambiare la dimensione a causa della propria aggiunta è possibile effettuare una compressione di una parte del file originale e successivamente effettuare la decompressione in fase di esecuzione
  - Pattern: sequenza di codice eseguibile che caratterizza il virus. La ricerca viene solitamente impostata su pattern corti per motivi di velocità ed ampliata in caso di matching

# Virus 4/4

- Protezioni adottate dai virus
  - Compressione del virus per mascherare il pattern. Eventualmente può essere eseguita congiuntamente con parte di codice in modo da non avere un pattern noto del virus compresso.
  - Polimorfismo: forme di alterazione del pattern, eventualmente gestite in modo autonomo, ad esempio tramite l'introduzione di un numero arbitrario di comandi che non eseguono alcuna operazione.
  - Alterazione di alcune funzionalità del sistema: ad esempio alterando i comandi o le librerie di interfaccia per conoscere i nomi dei file o i processi in corso, in modo da mascherare le attività del virus
- Come proteggersi dai virus
  - Meccanismi di Backup e Restore periodici
  - Checksum dei file: realizzato solitamente dal SO e scambiato su canali secondari, di difficile accesso. Il virus cambia il checksum ed il SO genera degli allarmi.
  - Antivirus
  - Controllo di esecuzione: i programmi vengono fatti girare in uno spazio limitato, senza possibilità di accedere ad altre regioni

# Worm 1/3

- Un worm è una particolare categoria di malware in grado di autoreplicarsi.
  - È simile ad un virus, ma a differenza di questo non necessita di legarsi ad altri programmi eseguibili per diffondersi.
  - Modifica il computer che infetta, in modo da venire eseguito ogni volta che si avvia la macchina e rimanere attivo finché non si spegne il computer o non si arresta il processo corrispondente.
  - Non cerca di esser amministratore o scrivere sul boot sector ma si comporta come utente e cerca di propagarsi inviando pacchetti ad altre macchine e magari mandandole in crash.
  - Tenta di replicarsi sfruttando Internet in diverse maniere
- 
- Etimologia: Il termine deriva da un romanzo di fantascienza degli anni 1970 di Brunner; i ricercatori che stavano scrivendo uno dei primi studi sul calcolo distribuito notarono le somiglianze tra il proprio programma e quello descritto nel libro e ne adottarono il nome. Uno dei primi worm diffusi sulla rete fu Internet Worm, creato dal figlio di un alto dirigente della NSA nel 1988 e riuscì a colpire oltre un terzo dei computer collegati a quel tempo in rete.

# Worm 2/3

- Propagazione del Worm
  - Random: invia e-mail con attachment a qualsiasi server mail, ad esempio identificato tramite motore di ricerca
  - DNS: tramite l'analisi dell'albero dei domini si cerca di determinare l'indirizzo di server di posta o di server web da attaccare (red code, 1988)
  - Liste: vengono ricavate da motori di ricerca o server whois liste di particolari utenti o servizi. Attualmente vi sono sistemi di dissuasione tramite inserimento di stringhe da leggere su GIF quadrettati, su cui gli OCR non riescono ad operare.
- Accesso a demoni
  - meccanismi di autoriconoscimento (.rhosts, etc)
  - attacco a username/password mediante forza bruta oppure con dizionari. La protezione può essere fatta impostando un numero massimo di tentativi di accesso o un tempo minimo tra prove diverse.
  - meccanismi di guadagno dell'accesso tramite bug. Preventiva esplorazione dei demoni attivi e acquisizione della versione del demone per analizzarne le criticità.

# Worm 3/3

- Meccanismi di protezione dei Worm
  - root kit in cui vengono cambiati i comandi o le librerie di sistema per nascondere il nome del processo o i file generati
  - PID: viene cambiato frequentemente il numero univoco che identifica l'esecuzione del worm, in modo da renderne difficile il reperimento
  - Cambio periodo del nome e di tutti i file collegati al Worm
  - Cancellazione dei log che interessano le azioni del Worm



# Worm: modello biologico 1/3

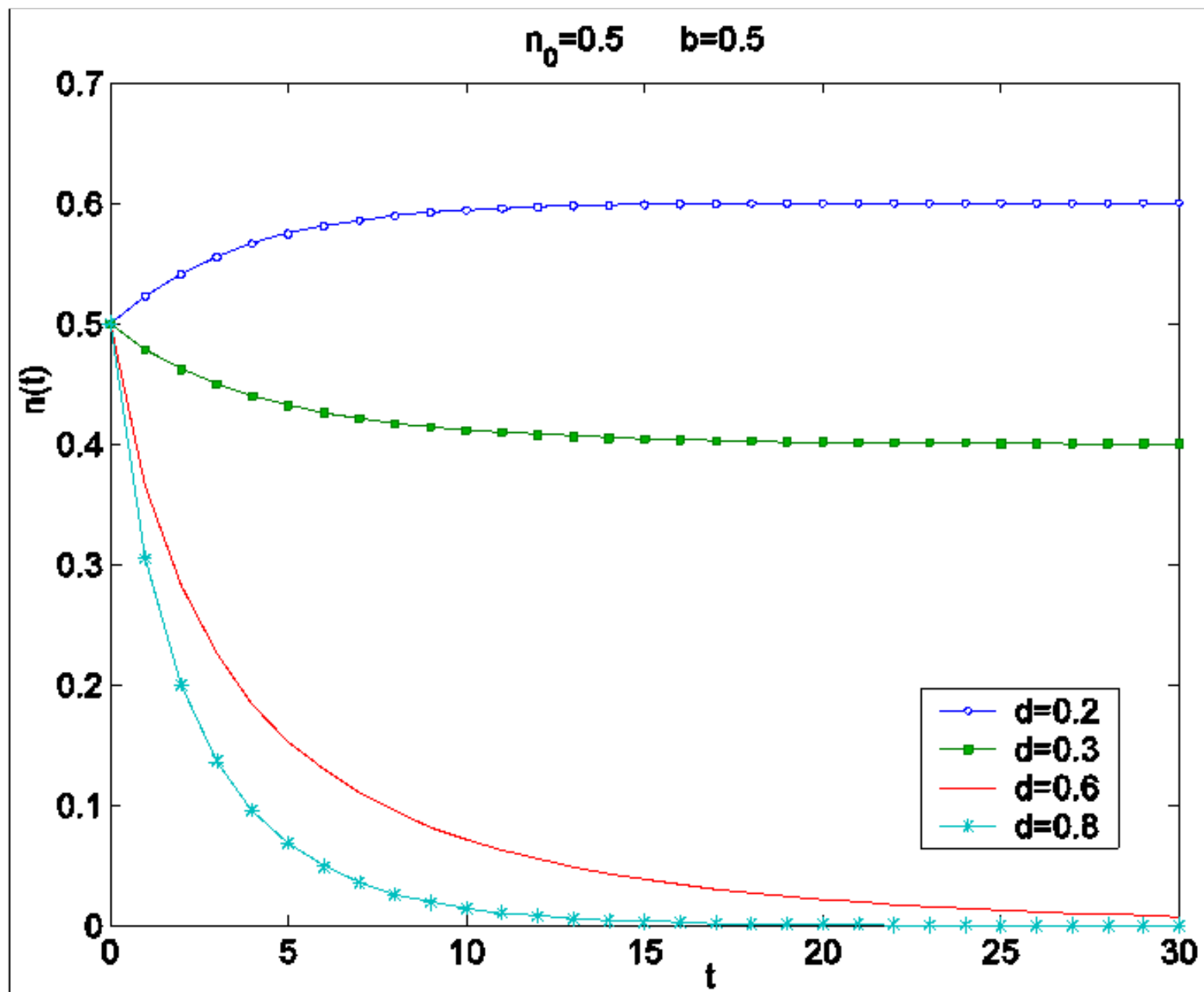
- Modello di propagazione dei worm basato sul modello biologico epidemico
- Ogni host può trovarsi in uno di due stati: infetto o suscettibile a infezione
- Un host suscettibile ad infezione può essere infettato da un host infetto e un host infetto può essere ripulito e diventare suscettibile a infezione; si ha quindi un'alternanza tra stati: susceptible, infected, susceptible (SIS model)
- Equazione differenziale non lineare che descrive la dinamica della popolazione infetta.
- $n(t)$  = frazione di host infetti rispetto al totale di host vulnerabili
- $b$  = tasso di nascita degli host infetti (infezione da host infetto su host suscettibile)
- $d$  = tasso di morte degli host infetti, velocità con cui un host infetto ritorna suscettibile

$$\frac{\partial n}{\partial t} = bn(1-n) - dn$$

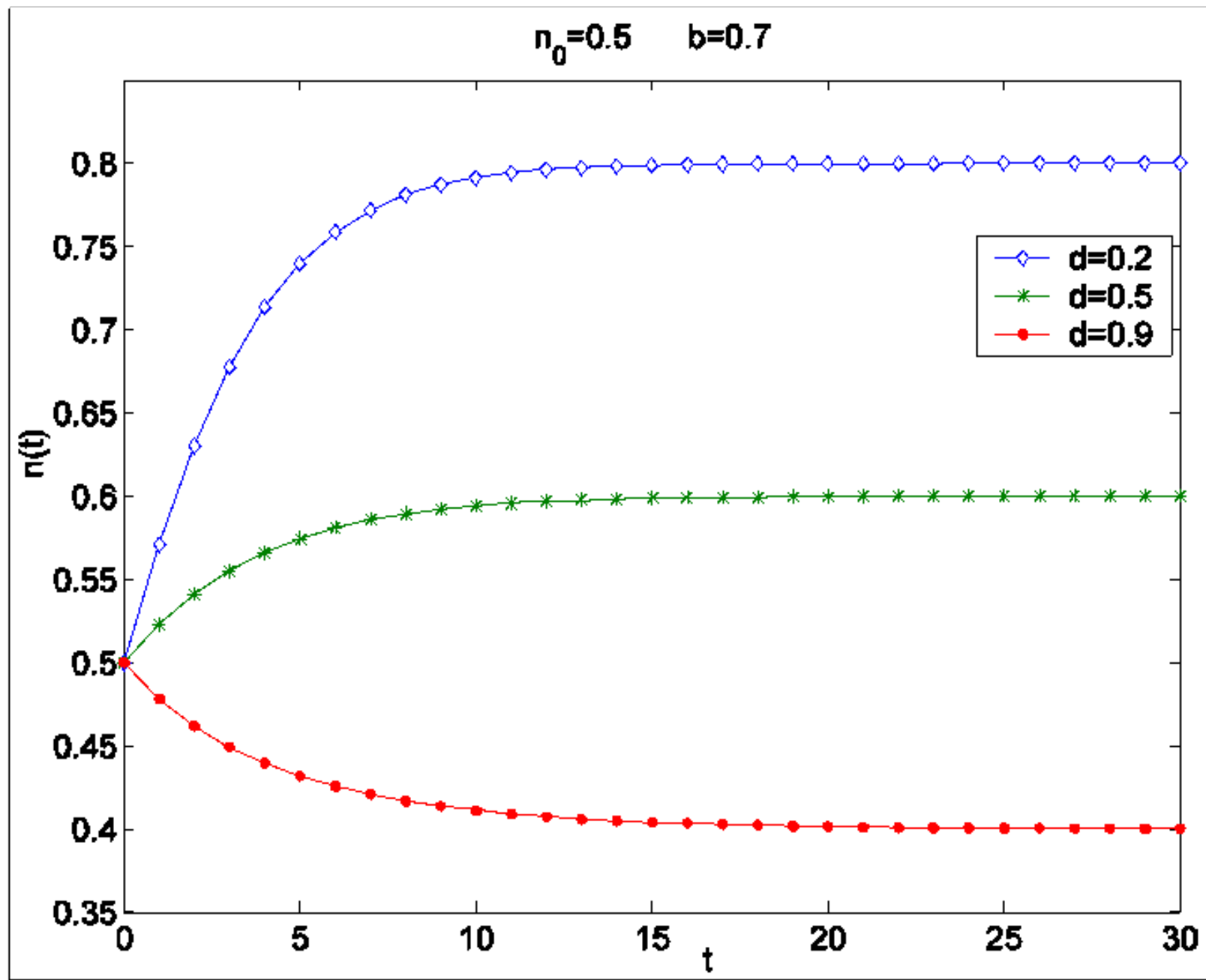
$$n(t) = \frac{n_0 \left(1 - \frac{d}{b}\right)}{n_0 + \left(1 - \frac{d}{b} - n_0\right) e^{-(b-d)t}}$$

- Soluzione stazionaria dipendente da rapporto tra tassi di nascita e morte
- Analisi della velocità di convergenza del transitorio

## Worm: modello biologico 2/3



## Worm: modello biologico 3/3



# Backdoor

- Accesso non noto all'amministratore del sistema realizzato dal programmatore per scopi di manutenzione e controllo del software. Molto spesso consente l'accesso all'intero sistema e a tutti i dati presenti, bypassando tutti i livelli di sicurezza impostati.

# Spyware

- Sistemi in grado di monitorare le azioni dell'utente, residenti sul computer stesso dell'utente o collegati alla stessa rete locale. L'analisi dei dati può portare a collezionare coppie di username e password, chiavi crittografiche, indirizzi di e-mail, etc.

# Dialer

- Sistemi in grado di modificare l'impostazione del sistema per forzare l'utilizzo di particolari linee telefoniche per la connessione ad internet. Le conseguenze possono essere di utilizzare connessioni verso provider ad alto costo oppure verso provider che monitorizzano il traffico dell'utente.

# Crittografia

- Scienza che si occupa di nascondere il contenuto informativo di un messaggio in modo tale che possa venir compreso solo dal destinatario e risultare incomprensibile a terze parti
- Utilizzata anche per autenticazione di un generico messaggio
- La sorgente vuole inviare alla destinazione un messaggio  $M$  (messaggio in chiaro, plaintext)
  - Deve arrivare alla destinazione senza che nessun altro lo legga
  - Si applica una funzione di encoding e si invia il risultato cifrato  $C$ .  $C=E(M)$
  - Il destinatario applica una funzione di decodifica per ottenere il messaggio originale in chiaro  $M$ .  $M=D(C)$
  - La riservatezza del messaggio è legata alla segretezza di  $E$  e  $D$ , dunque occorrerebbero due diverse  $E$  e  $D$  per ogni diversa entità con cui  $S$  comunica
- Comunemente si utilizzano funzioni parametriche, dipendenti da una chiave  $K$ . La sicurezza non è basata sulla segretezza dell'algoritmo crittografico, disponibile a tutti, ma sulla segretezza della chiave  $K$ .

# Classificazione dei sistemi di crittografia

- CHIAVE SEGRETA (PRIVATA)
  - Si utilizza un'unica chiave durante codifica e decodifica
  - $C=E(M,K)$   $M=D(C,K)$
  - K deve essere mantenuta segreta, nota solo alla sorgente e al destinatario
  - Problema: fase di scambio della chiave tra sorgente e destinazione, potrebbe essere intercettata da terzi
- CHIAVE PUBBLICA
  - Si utilizzano due chiavi diverse, una pubblica e una segreta
  - $C=E(M,K1)$   $M=D(C,K2)$
  - K1 pubblica; K2 privata e segreta, nota solo alla destinazione che è l'unica in grado di recuperare M da C.
  - K1 privata e K2 pubblica. Solo la sorgente può codificare il messaggio che tutti possono leggere. Utilizzato per problematiche di firma
- BLOCK CIPHER
  - Il messaggio originale viene diviso in blocchi di lunghezza fissata e l'algoritmo di codifica viene applicato ad ogni singolo blocco.
- STREAM CIPHER
  - L'intero messaggio viene codificato un simbolo alla volta e viene fornito in uscita un simbolo codificato per ogni simbolo in chiaro entrato nel codificatore.

# Principali obiettivi

- Il testo in chiaro non deve poter essere ottenuto facilmente dal testo cifrato
- La chiave non deve poter esser ottenuta facilmente dal testo cifrato
- Lo spazio delle possibili chiavi deve essere grande abbastanza per resistere ad un attacco di forza bruta (ricerca esaustiva)
- Nel 1949 Shannon presentò due principi per fare una buona crittografia:
  - CONFUSIONE: non si può predire facilmente come le statistiche del messaggio cifrato dipendono da quelle del messaggio in chiaro.
  - DIFFUSIONE: estendere l'influenza di un carattere del messaggio in chiaro sulla maggior parte possibile del messaggio cifrato

# Cifratura: alcune definizioni

- PER TRASPOSIZIONE: sistema crittografico in cui l'occultamento del significato del messaggio e' ottenuto spostando le lettere in posizioni diverse da quelle originarie, senza mutare l'identità della lettera
- PER SOSTITUZIONE: ogni lettera di un messaggio è sostituita da un'altra lettera, ma conserva la sua posizione
- MONOALFABETICA: cifratura per sostituzione in cui l'alfabeto cifrante non è mai sostituito durante la generazione del messaggio cifrato. Ad ogni lettera dell'alfabeto viene associato sempre lo stesso carattere cifrato.
- POLIALFABETICA: cifratura per sostituzione, in cui l'alfabeto cifrante cambia durante la generazione del crittogramma. La chiave determina l'ordine in cui devono venire scambiate le lettere e il ciclo viene ripetuto con una periodicità pari alla lunghezza della chiave. Ad ogni lettera dell'alfabeto vengono associati caratteri diversi.




# Crittoanalisi

- Scopo: scoprire ed eliminare tali protezioni. La crittoanalisi cerca di rivelare le comunicazioni criptate senza conoscere la chiave.
- Esistono numerose tecniche di crittoanalisi e possono essere classificate in base alla quantità di conoscenza che l'analista ha a disposizione.
- Si suppone che l'algoritmo sia noto e che ci sia libero accesso ai dati cifrati trasmessi; la segretezza della comunicazione è affidata alla segretezza della chiave.
- Classifichiamo le tecniche in quattro gruppi:
  - Attacco solo su testo cifrato: l'avversario ha a disposizione solo C. Se un sistema è vulnerabile a tale tipo di attacco è considerato completamente insicuro.
  - Attacco su testo noto: dove si conosce una certa quantità, casuale, di messaggio M ed il corrispondente testo cifrato C.
  - Attacco con messaggio scelto: è possibile forzare la cifratura di un messaggio particolare M e verificare il corrispondente cifrato C.
  - Attacco con testo cifrato scelto: è possibile ottenere la decodifica di un particolare messaggio cifrato C.
- Attacco a distribuzione di frequenza, attacco polialfabetico con indice di coincidenza, crittoanalisi lineare, crittoanalisi differenziale

# Cifratore a trasposizione

- Il messaggio è diviso in gruppi di fissata lunghezza ai quali viene applicata la stessa permutazione di posizione delle lettere.
- Per violarli deve indovinare il periodo di trasposizione. Si procede provando, in modo esaustivo, raggruppamenti di varie dimensioni. Se applicando la stessa permutazione ad altri gruppi si ottengono parole sensate, si è risolto il codice.

Plaintext:	THEGA	TORSA	RETHE	BESTT	EAMIN	THESE	CGOGA	TORSX
Key:								
Ciphertext:	HAGTE	OASTR	EEHRE	ETTBE	ANIEH	HESTE	GAGCO	OXSTR

Plaintext: **THEGATORSARETHEBESTTEAMINTHESEC**

write in row-wise

T	H	E	G	A
T	O	R	S	A
R	E	T	H	E
B	E	S	T	T
E	A	M	I	N
T	H	E	S	E
C	X	X	X	X

read out column-wise

Ciphertext: **ttrbetchoeeahxertsmexgshtisxaaetnex**

# Cifratura One Time Pad

- Sistema stream cipher, inventato da Vernam nel 1917. Si basa sull'impiego di una chiave comune con le seguenti proprietà:
  - deve avere la stessa lunghezza del messaggio
  - deve essere una sequenza completamente casuale di caratteri
  - non deve essere riutilizzata
- la sicurezza di tale sistema non dipende particolarmente dalla complessità della funzione di cifratura utilizzata; sono sufficienti semplici operazioni, quali: somma e sottrazione
- Sistema teoricamente immune da attacco: un avversario che entra in possesso del messaggio non ha possibilità di capire il messaggio a meno che conosca la chiave.
- Limitazioni: necessità di scambiare ogni volta la chiave, tramite un canale sicuro
- Questo sistema rende impossibile la decodifica del messaggio ma ad un hacker resta sempre la possibilità di modificare il messaggio trasmesso e rendere dunque impossibile la decodifica del messaggio da parte del ricevente.

# Cifratore per sostituzione: cifrario di Cesare

- Svetonio nella “Vita dei dodici Cesari” racconta che Giulio Cesare usava per le sue corrispondenze riservate un codice di sostituzione molto semplice, nel quale la lettera chiara veniva sostituita dalla lettera che la segue di tre posti nell'alfabeto: la lettera A è sostituita dalla D, la B dalla E e così via.
- Esempio:
  - Chiaro A B C D E F
  - Cifrato D E F G H I
- Algoritmo
  - $C[i] = (M[i] + 3) \bmod 26$  M appartiene a  $\{0, \dots, 25\}$  lettere dell'alfabeto
  - Decifrazione:  $(C[i] - 3) \bmod 26 = M[i]$

# Rotazione

- Generalizzazione cifrario di Cesare
- $C[i] = (M[i] + k) \bmod N$
- $N$  alfabeto,  $k$  in  $[0 \dots N]$

# Moltiplicazione

- $C[i] = (k M[i]) \bmod N$
- $N$  alfabeto,  $k$  in  $[1 \dots N]$ , e  $k$  e  $N$  relativamente primi

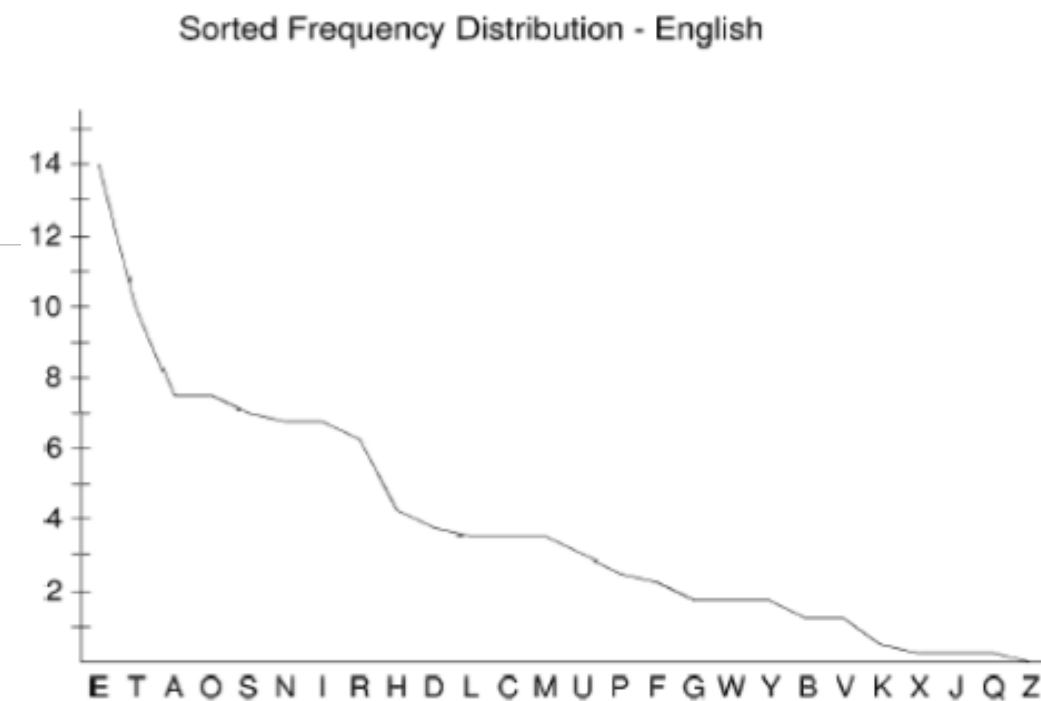
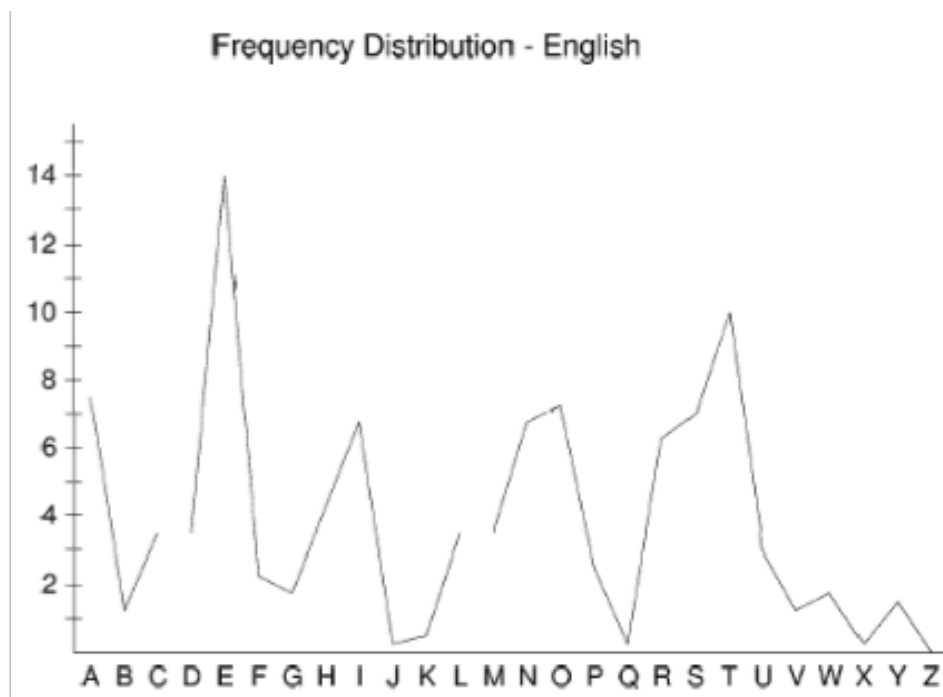
# Affine

- $C[i] = [(k M[i]) + h] \bmod N$
- $N$  alfabeto,  $(k, h)$  in  $[1 \dots N] \times [1 \dots N]$ , e  $k$  e  $N$  relativamente primi

# Attacco ad analisi della frequenza delle lettere

- Uno dei sistemi più utilizzati per la crittoanalisi di testi cifrati, specie quelli della famiglia dei metodi a sostituzione, è l'analisi della frequenza con cui si ripetono alcune lettere in una data lingua.
- Analizzando un testo scritto, ad esempio in italiano, anche di lunghezza limitata, si osserva che le lettere che compaiono più frequentemente sono nell'ordine: e,i,a,o. In generale, analizzando un qualsiasi testo, è possibile stabilire quale sono le frequenze di ogni lettera dell'alfabeto.
- Analizzando le frequenze del testo cifrato, si può determinare la lettera originale, associando le frequenze della lingua originale a quella del messaggio cifrato. Il confronto deve essere operato ordinando i diagramma delle frequenze in modo da renderlo decrescente.
- Ad esempio se in un testo italiano cifrato la frequenza più alta è della la lettera "x", la lettera non cifrata è la "e".
- Operando la sostituzione di tutte le lettere del testo cifrato, il testo sarà decrittato al minimo nel 60% delle lettere che lo compongono. Questo a causa dell'appiattimento nell'istogramma delle frequenze nella sua coda.

# Attacco ad analisi della frequenza delle lettere



# Cifratura a sostituzione polialfabetica: Cifrario di Vigenere 1/2

- Generalizzazione del codice di Cesare
- Utilizza 26 alfabeti cifranti per cifrare un solo messaggio
- Invece di spostare sempre dello stesso numero di posti la lettera da cifrare, questa viene spostata di un numero di posti variabile, determinato in base ad una parola chiave, da concordarsi tra mittente e destinatario, e da scriversi sotto il messaggio, carattere per carattere; essendo in genere molto più corta del messaggio, la parola chiave deve essere ripetuta molte volte sotto questo: è importante sottolineare che se la chiave ha lunghezza  $m$ , lo spazio delle chiavi ha dimensione  $26^m$
- Tavola di Vigenere

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z	A
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z	A	B
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z	A	B	C	D
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z	A	B	C	D	E
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z	A	B	C	D	E	F
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z	A	B	C	D	E	F	G
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z	A	B	C	D	E	F	G	H
J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z	A	B	C	D	E	F	G	H	I
K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	N	O	P	Q	R	S	T	U	V	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	O	P	Q	R	S	T	U	V	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	P	Q	R	S	T	U	V	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	Q	R	S	T	U	V	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	R	S	T	U	V	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	S	T	U	V	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	T	U	V	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	U	V	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	V	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X
Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y



# Cifrario di Vigenere 2/2

- L'algoritmo di cifratura:
  - considera la prima lettera del testo in chiaro e della chiave
  - le usa come coordinate cartesiane nella tavola
  - l'intersezione fornisce il carattere da sostituire nel testo cifrato
  - itera per tutta la lunghezza del testo
- L'algoritmo di decifrazione:
- considera la prima lettera della chiave
- nella riga corrispondente alla lettera della chiave individua la lettera del testo cifrato
- il carattere che contrassegna tale colonna è la lettera in chiaro
- itera per tutta la lunghezza del testo
- Esempio: ARRIVANO chiave=VERME

VERMEVER

VVIUZVFR

# Attacco polialfabetico: metodo Kasiski

- Occorre determinare la lunghezza della chiave  $T$
- Porzioni ripetute di messaggio  $M$  con la stessa porzione di chiave risultano parti di  $C$  identiche
- Il numero di caratteri compresi tra gli inizi di due segmenti ripetuti sarà un multiplo di  $T$
- La ricerca deve essere basata su pattern di tre o più caratteri in modo da ridurre la probabilità di falso allarme, con eventuali dovuti a eventuali ripetizioni accidentali.
- Tra tutti i potenziali multipli di  $T$  si effettua il massimo comun divisore per ottenere  $T$  candidato
- Noto  $T$  si procede con  $T$  differenti analisi di frequenza, una su ogni singolo elemento della chiave

# Attacco polialfabetico: indice di coincidenza 1/2

- Nel 1920 Friedman propone un'alternativa al metodo Kasiski per determinare il periodo di ripetizione  $T$ , basato sull'indice di coincidenza (IC)
- IC = probabilità che due caratteri scelti a caso dallo stesso testo cifrato siano uguali
- Per un testo uniforme, in cui le lettere hanno la stessa probabilità di apparizione,  $IC=0.038$
- Per un testo italiano  $IC=0.075$ , per un testo inglese  $IC=0.066$
- Per determinare IC di un testo cifrato si confrontano il testo cifrato originale ed una sua versione shiftata di  $k$  passi ottenendo  $IC(k)$
- Esempio:

a	a	s	a	d	a	c	s	d	a	s	d	d	d	c	v	e	s	d
a	c	s	d	a	s	d	d	d	c	v	e	s	d	a	a	s	a	d
*		*				*				*				*				*

- Si determina IC come il rapporto tra il numero di coincidenze trovate (lettere uguali nella stessa posizione) e il totale di lettere del segmento di testo in esame
- nell'esempio  $IC(k)=5/19=0.26$

# Attacco polialfabetico: indice di coincidenza 2/2

- Si ripete il meccanismo del calcolo dell'indice di coincidenza per diversi valori di  $k$ . La lunghezza della chiave  $T$  corrisponde al valore  $k$  per cui l'indice di coincidenza è più prossimo a quello della lingua in esame.
- Es:  $IC(1)=0.028$ ,  $IC(2)=0.045$ , ...,  $IC(6)=0.35$ ,  **$IC(7)=0.07$** ,  $IC(8)=0.032$ , selezionato  $k=7$
- Noto  $T$  si procede con  $T$  differenti analisi di frequenza, una su ogni singolo elemento della chiave

## Esempio

0	$T$	$2T$	...
1	$T+1$	$2T+1$	...
2	$T+2$	$2T+2$	...
3	$T+3$	$2T+3$	...

# Crittoanalisi lineare

- Attacco su testo noto (l'attaccante deve poter avere a disposizione un set casuale di coppie testo in chiaro e relativo cifrato) applicabile a sistemi di cifratura simmetrica a blocchi
- L'idea base consiste nell'approssimare l'operazione di una porzione di algoritmo cifrante con un'espressione lineare (XOR)
- Si considerano  $u$  bit dell'ingresso  $X=[X[1],X[2],\dots,X[u]]$  e  $v$  bit dell'uscita  $Y=[Y[1], Y[2],\dots,Y[v]]$ , si sommano modulo 2 e si va a valutare la probabilità di ottenere la seguente uguaglianza, per diversi set di bit
- $X[1]+X[2]+\dots+X[u]+Y[1]+Y[2]+\dots+Y[v]=0$
- Prendendo a caso gli  $u$  e i  $v$  bit tale probabilità dovrebbe essere circa 0.5. L'approccio dell'attacco lineare consiste nel cercare quei particolari set di bit che portano a valori di probabilità che maggiormente si discostano da 0.5 (vicino a 0 o a 1, sintomo di poca capacità del codice di generare casualità).
- Maggiore è lo scostamento dallo 0.5, meno coppie (M,C) servono ad effettuare l'attacco.

# Crittoanalisi differenziale

- Si tratta di un attacco a messaggio scelto, per sistemi di cifratura simmetrica a blocchi
- Sfrutta la probabilità di occorrenza di particolari differenze tra ingressi e uscite
- Si considera una coppia di ingressi  $X'$  e  $X''$  e coppia di relative uscite  $Y'$  e  $Y''$ , di  $n$  bit ciascuno, si valuta la differenza
- $DX = X' + X''$   $DY = Y' + Y''$
- L'attaccante può scegliere delle coppie di ingressi  $X'$  e  $X''$  che risultino in una particolare differenza  $DX$ , sapendo che questa porta ad una determinata differenza  $DY$  con un'elevata probabilità
- In un sistema idealmente casuale la probabilità che data una certa  $DX$  si verifichi una certa  $DY$  è  $2^{-n}$ . L'analisi differenziale cerca di sfruttare i casi in cui, data una  $DX$ , la  $DY$  si verifica con una probabilità molto elevata.

# Enigma

- Macchina per cifrare e decifrare elettromeccanica, al servizio dell'esercito tedesco.
  - Penetrato dagli inglesi soltanto successivamente al recupero di un esemplare da un sottomarino tedesco, nonostante gli sforzi di scienziati del calibro di Turing.
  - Aspetto estetico di una macchina da scrivere con due tastiere: una vera in cui battere il testo, una composta da lettere luminose per indicare il messaggio cifrato.
- 
- E' formata da 3 rotori; ogni rotore ha una serie di 26 contatti elettrici su entrambe le facce e i contatti su un lato sono collegati a quelli dell'altro in modo casuale. Quando l'operatore preme un tasto, un segnale elettrico passa da rotore a rotore fino a mostrare una lettera illuminata che è il carattere cifrato: la lettera del testo in chiaro viene sostituita dalla lettera del primo rotore, la quale viene poi sostituita da un altro alfabeto col secondo rotore ed infine da un terzo con l'ultimo rotore. Il primo rotore ruota di una lettera ad ogni pressione di tasto, il secondo ruota di una lettera ogni volta che il primo compie un giro e il terzo ruota di una lettera quando il secondo termina un giro. La chiave consiste nella disposizione iniziale dei rotori; questa chiave veniva cambiata ogni 24 ore secondo una regola prefissata. Anche i collegamenti interni dei rotori sono segreti.
  - Per decifrare un testo si dispongono i rotori nella posizione indicata dalla chiave e digitando il testo cifrato appare nei tasti luminosi il testo in chiaro



# Schema Feistel 1/3

- Esempio di iterated block cipher: cifratura a blocchi realizzata in più round di processamento
- Il messaggio in chiaro  $M$  è l'ingresso del primo round; il testo cifrato  $C$  è l'uscita dell'ultimo round
- La chiave  $K$  è utilizzata per generare le chiavi  $K[i]$  usate ai round  $R[i]$
- Ogni round utilizza una diversa chiave  $K[i]$  in ingresso alla funzione  $f$
- La funzione  $f$  non deve essere necessariamente invertibile; per la decodifica infatti viene utilizzata la funzione stessa e non la sua inversa
- Occorrono due round per diffondere il contributo di ogni bit su tutto il blocco
- Può essere impiegato un numero arbitrario di round
- Per decifrare si impiega lo stesso schema usato in cifratura, invertendo l'ordine delle chiavi utilizzate in cifratura

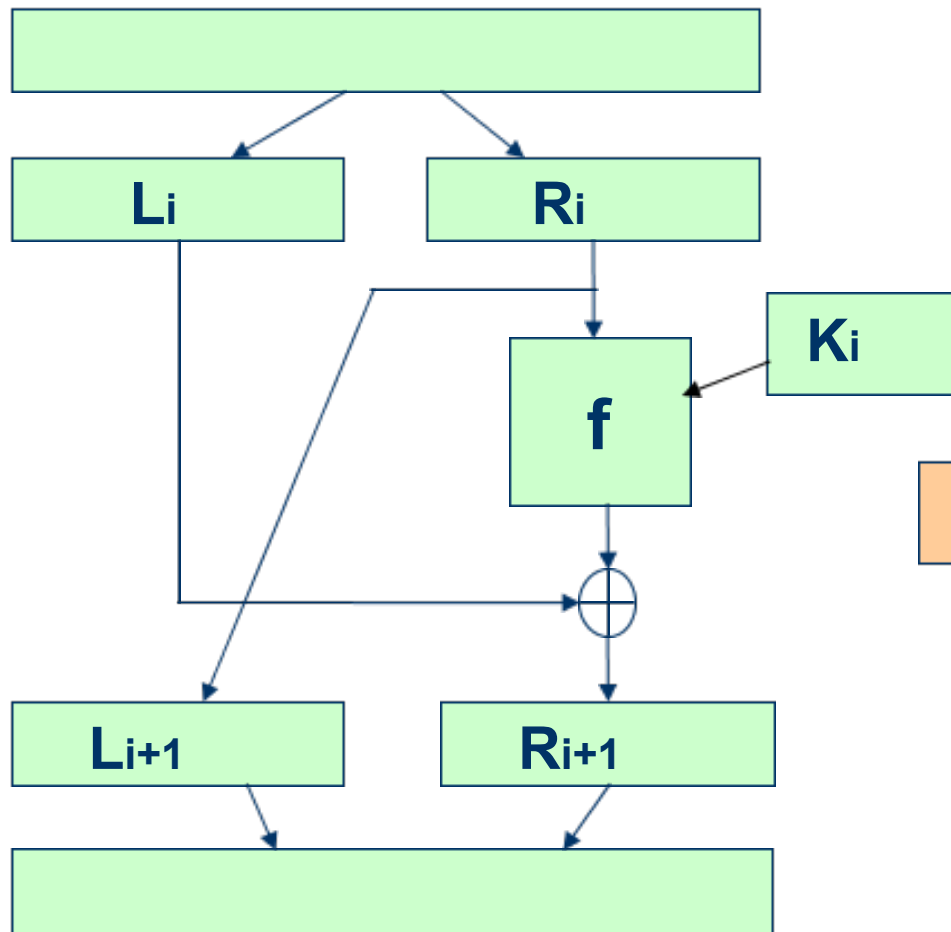


# Schema Feistel 2/3

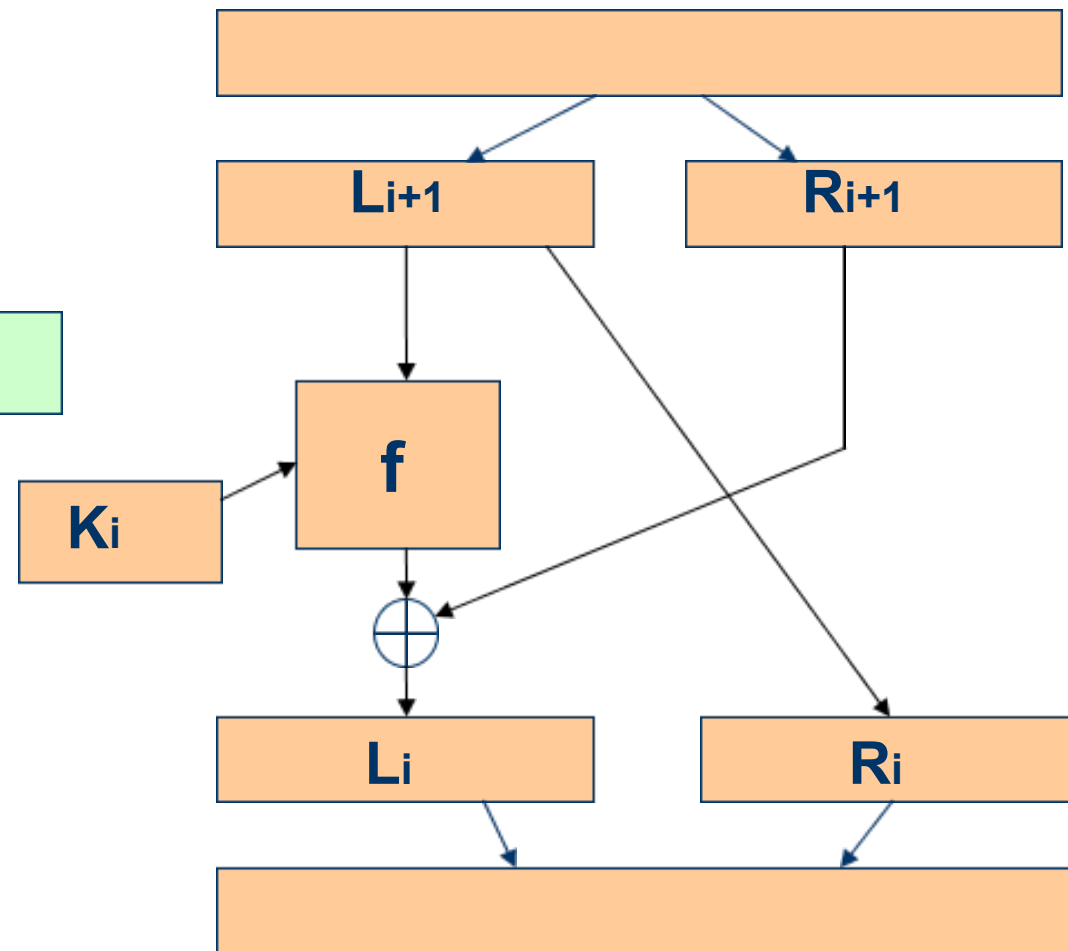
- Codifica:
  - il testo M è diviso in due metà
  - $L[i+1]=R[i]$
  - $R[i+1]=f(R[i],K[i])+L[i]$
- Decodifica:
  - utilizza le stese chiavi ma inserite nei vari round nell'ordine inverso rispetto alla codifica
  - $R[i]=L[i+1]$
  - $L[i]=f(R[i],K[i])+R[i+1]=f(L[i+1],K[i])+R[i+1]$

# Schema Feistel 3/3

**codifica**

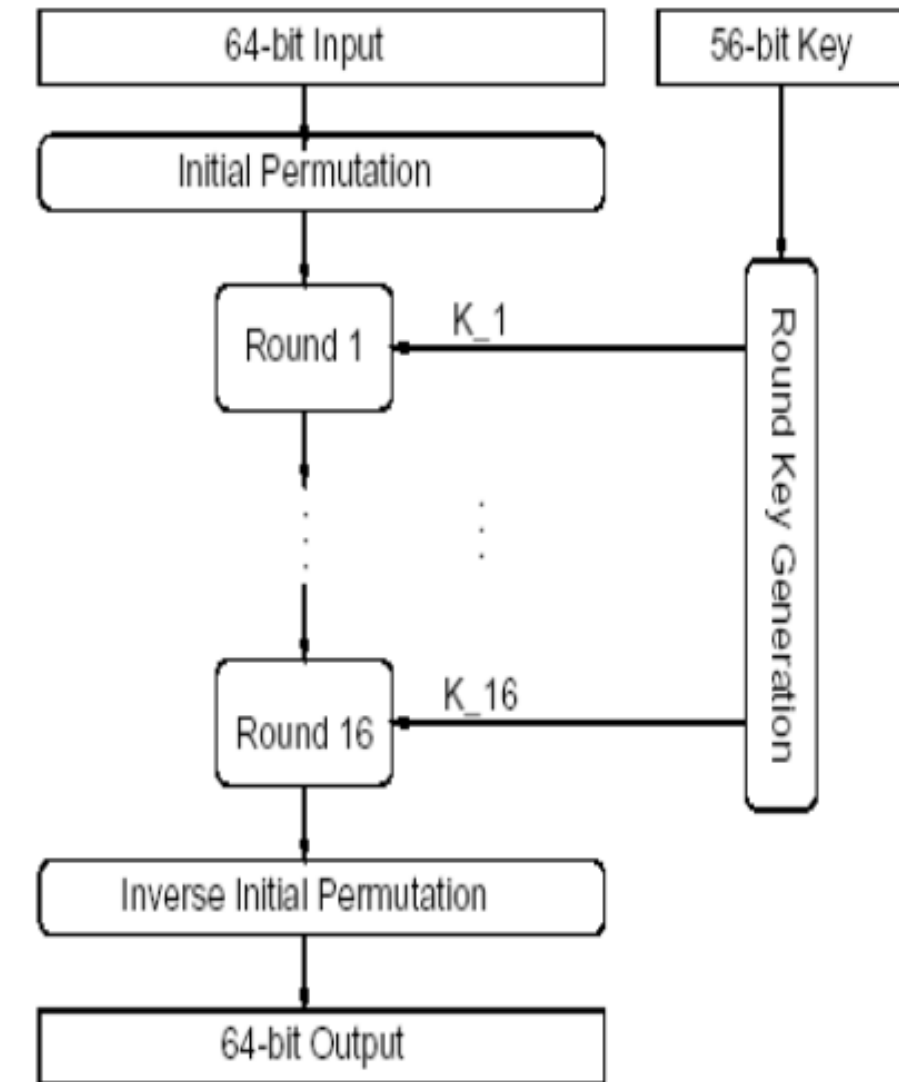


**decodifica**



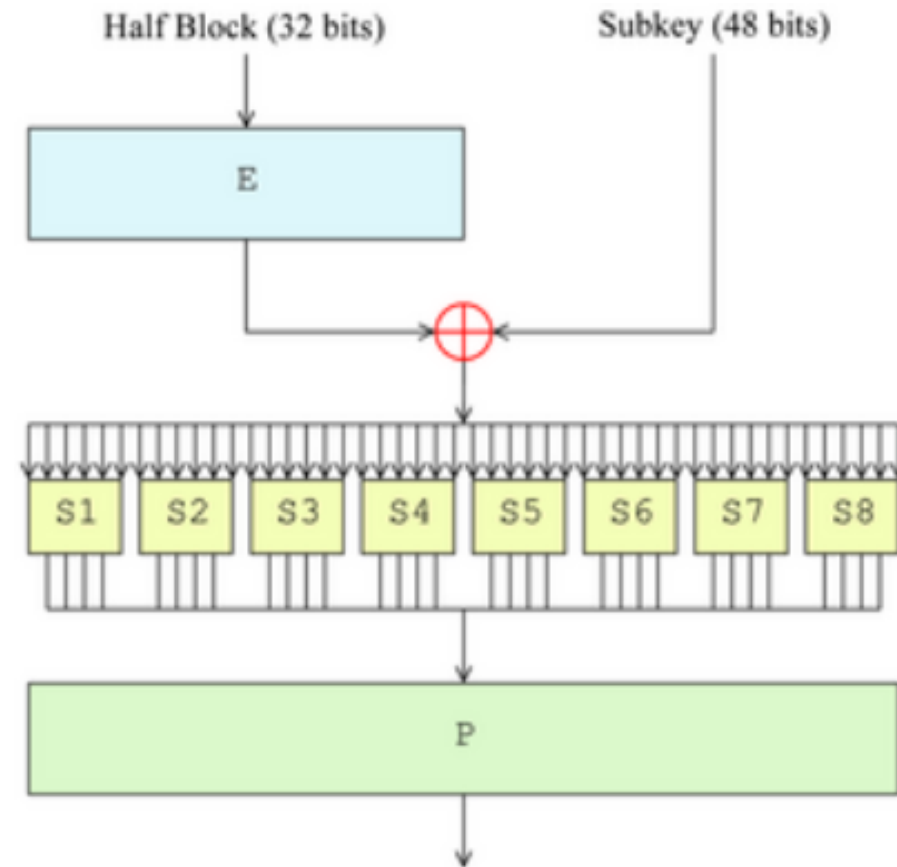
# DES: Data Encryption Standard 1/3

- Sistema di iterated block cipher a chiave segreta
- Struttura di tipo Feistel con 16 round, opera su blocchi di 64 bits ed utilizza una chiave di 56 bit, da cui si ricavano round keys di 48 bit
- Le permutazioni iniziale e finale sono tra di loro inverse e non hanno alcuna importanza per la cifratura ma sono state probabilmente aggiunte per facilitare il caricamento dei blocchi sull'hardware tipico degli anni '70.
- Il tipo più pratico di attacco a tutt'oggi è quello con forza bruta. E' possibile applicare altri tipi di attacco, con analisi lineare e differenziale ma richiedono una quantità di testo in chiaro conosciuto o scelto a priori tale che ne rende difficile l'utilizzo pratico.



# DES: Data Encryption Standard 2/3

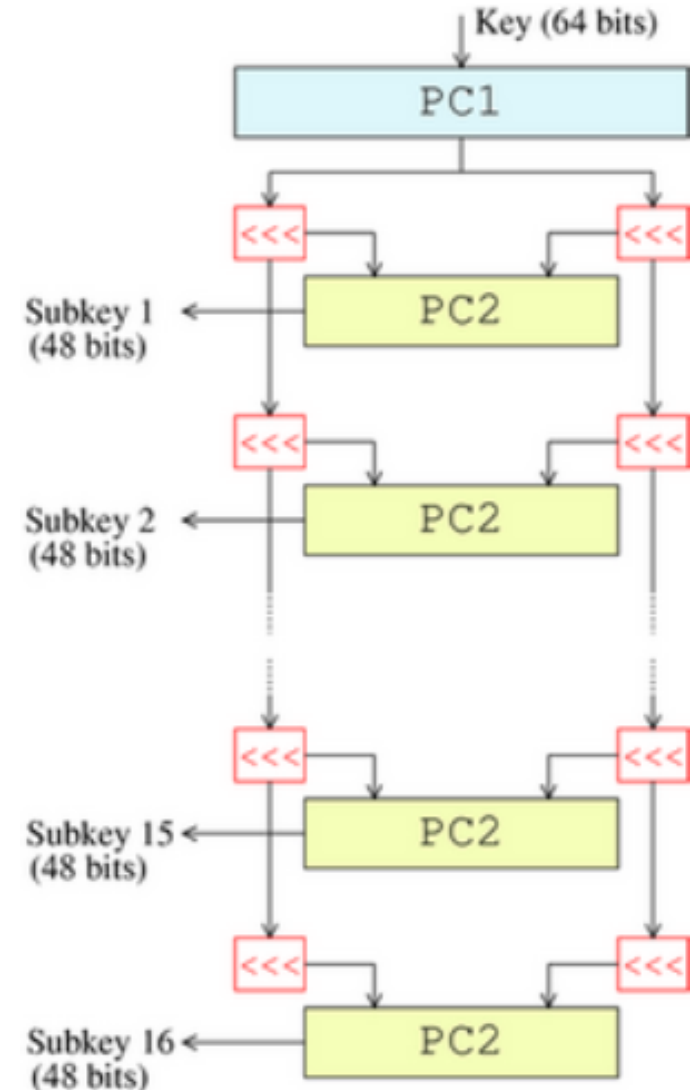
- **Feistel function f**
- **Espansione:** il mezzo blocco di 32-bit è espanso fino a 48 bit utilizzando la permutazione di espansione (E), che duplica alcuni bit.
- **Miscelazione con la chiave:** il risultato è combinato con una sottochiave.
- **Sostituzione:** il blocco viene diviso in 8 pezzi di 6 bit prima del processamento con le S-box (substitution box). Ognuna delle otto S-box sostituisce sei bit in input con quattro bit in output mediante una trasformazione descritta in modo tabellare. Le S-box forniscono il cuore della sicurezza del DES.
- **Permutazione:** i 32 risultati delle S-box sono riordinati in base alle permutazioni della P-box.
- **La confusione e diffusione** è ottenuta mediante l'alternanza di: sostituzioni mediante le S-box, permutazioni con la P-box ed espansioni.



# DES: Data Encryption Standard 3/3

- **Generazione sottochiavi**

- Inizialmente, vengono selezionati 56 bit della chiave dagli iniziali 64 bit mediante Permuted Choice 1 (PC-1); i rimanenti otto bit sono scartati o utilizzati come bit di controllo della parità.
- I 56 vengono suddivisi in due metà di 28 bit ciascuna. Nei cicli successivi entrambe le metà vengono ruotate verso sinistra di uno o due bit (specifico per ogni ciclo) e quindi vengono scelti 48 bit per la sottochiave mediante la Permuted Choice 2 (PC-2), 24 bit dalla metà di sinistra e 24 bit da quella di destra. La rotazione significa che in ogni sottochiave è usato un insieme differente di bit; ogni bit è usato circa in 14 delle 16 sottochiavi.
- Il gestore delle chiavi per la decifratura deve generare le chiavi nell'ordine inverso quindi la rotazione è verso destra invece che verso sinistra.



# Attacchi al DES

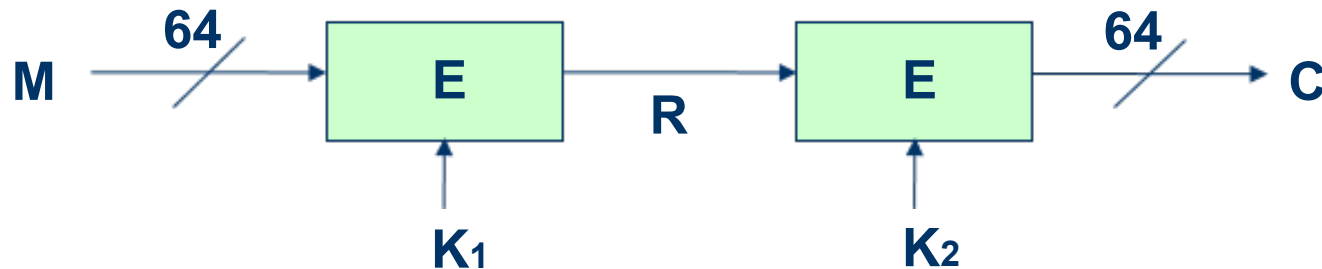
- Non è stato scoperto nessun attacco semplice per il DES
- Le chiavi possibili sono  $2^{56} \approx 7E16$
- Nel luglio 1998 la Electronic Frontier Foundation (EFF) con una macchina appositamente costruita (DES Cracker) ha forzato il DES in circa tre giorni
- Nel gennaio 1999 una rete composta da 100.000 PC e il DES Cracker ha forzato il DES in 22 ore e 15 minuti
- Un attacco migliore rispetto alla forza bruta consiste nell'attacco differenziale: questo richiede la cifratura di  $2^{47}$  messaggi scelti dall'attaccante e analizza l'evoluzione delle differenze tra i due messaggi, legati tra loro dal fatto che son stati cifrati con la stesa chiave
- In alternativa, tramite l'analisi lineare si può recuperare la chiave del DES dall'analisi di  $2^{43}$  messaggi noti

# DES multiplo

- Non è utile cifrare due volte con la stessa chiave, perché lo spazio di un attacco a forza bruta è uguale (si raddoppia solo il tempo del singolo tentativo).
- Non è detto che applicare più volte lo stesso algoritmo aumenti la sicurezza. Non avrebbe senso se date due chiavi qualsiasi  $K_1, K_2$  esistesse sempre una terza chiave  $K_3$  tale che  $E_{K_2}[E_{K_1}[M]] = E_{K_3}[M]$ . In questo caso la doppia cifratura (e quindi una catena arbitrariamente lunga di cifrature) sarebbe sempre equivalente ad una cifratura semplice
- Per il DES è stato dimostrato (1992) che non è così
- Evoluzioni: Doppio DES, 3DES

# Doppio DES

- L'idea più semplice è utilizzare due cifrature con due chiavi diverse:
- $C = EK_2[R] = EK_2[EK_1[M]]$

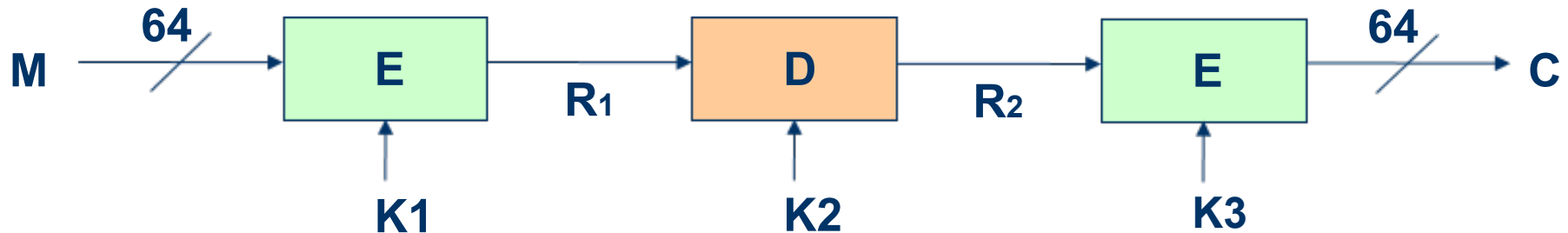


- Uno schema di questo tipo è però in generale (non solo per il DES) vulnerabile ad un attacco di tipo “meet-in-the-middle”.
- Data una coppia  $(M,C)$ ,  $EK_1[M] = DK_2[C] = R$ .
- Si cifra  $M$  con tutte le  $2^{56}$   $K_1$  chiavi possibili e si memorizzano i corrispondenti  $R$  in una tabella
- Si decifra  $C$  con tutte le  $2^{56}$   $K_2$  chiavi possibili, ottenendo  $R'$
- Si cerca valori di  $R$  già presenti nella tabella: la coppia di chiavi esatte  $(K_1, K_2)$  è infatti quella per cui si ottiene  $R=R'$



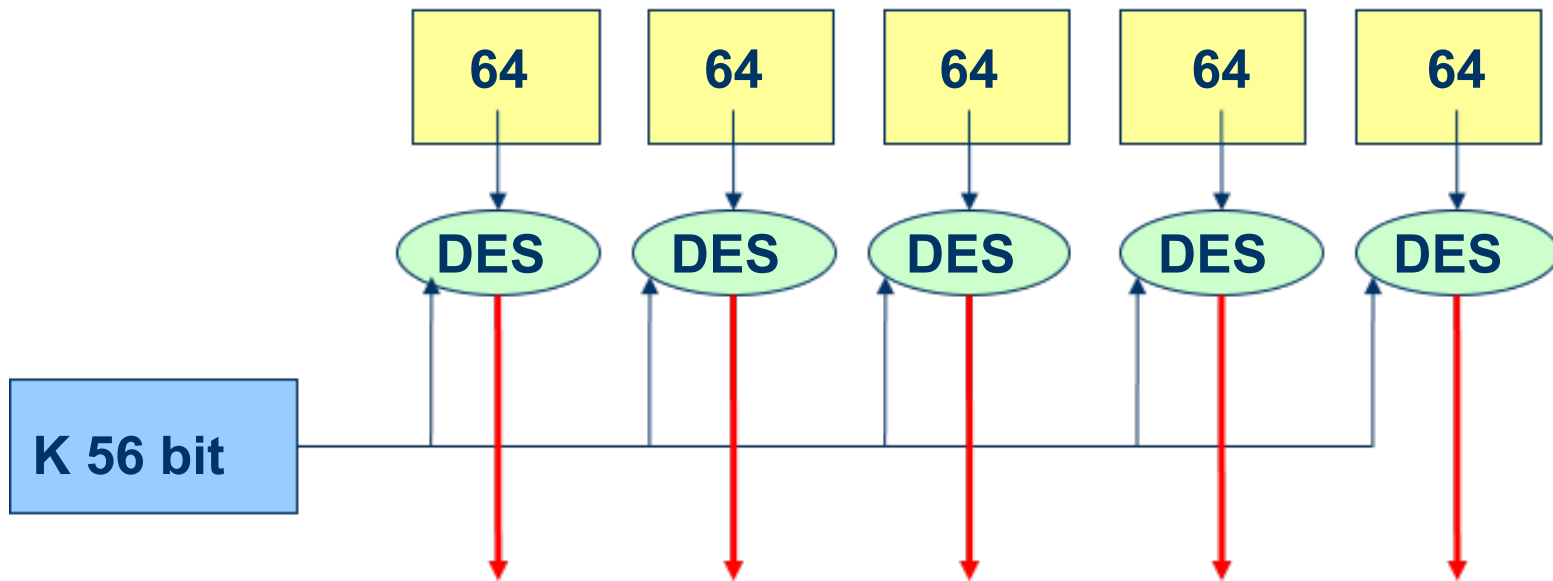
# Triplo DES (3DES)

- Con il triplo DES (3DES) si applica il DES tre volte: viene effettuata prima una cifratura con una chiave K1, poi una decifratura con una chiave K2, poi una cifratura con una chiave K3
- la dimensione della chiave totale è dunque pari a  $56 \times 3 = 168$  bit
- Una variante prevede  $K3 = K1$ , per cui la dimensione della chiave è in questo caso 112 bit.



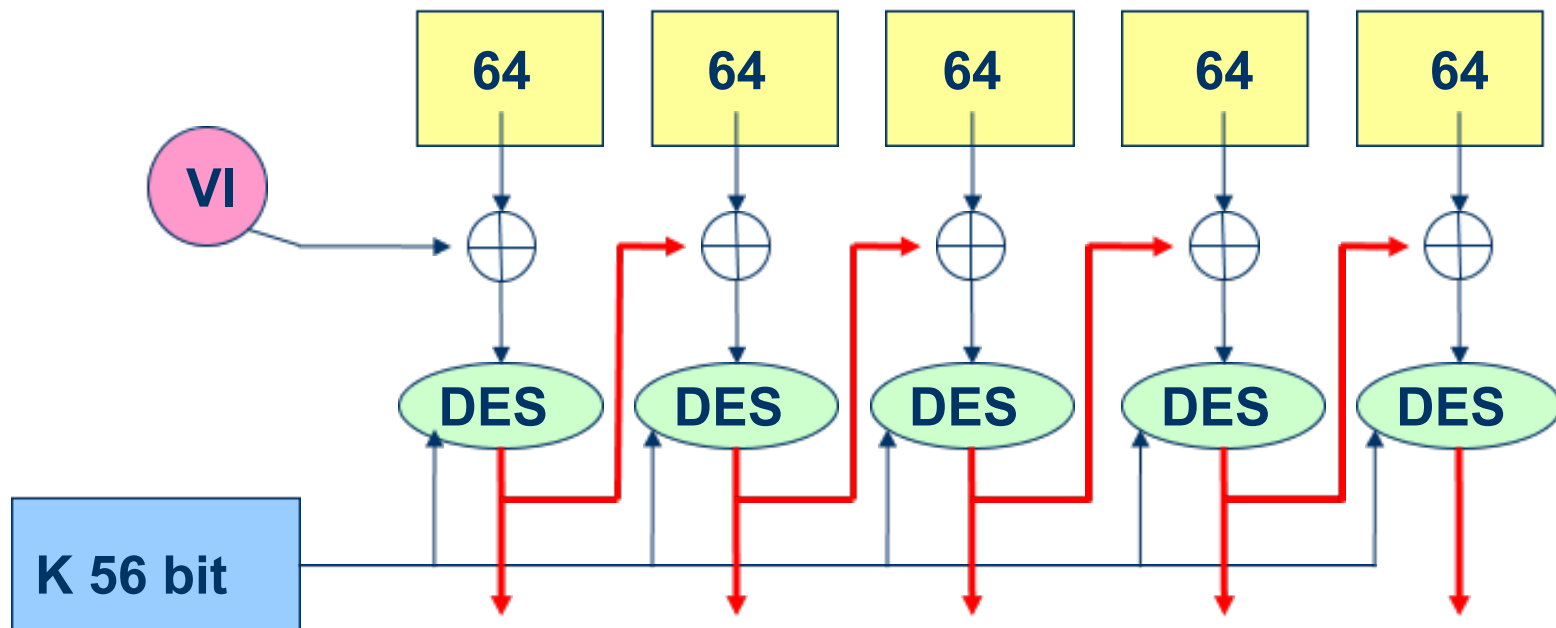
# Utilizzo degli schemi di codifica a blocchi simmetrica 1/4

- Electronic Code Book (ECB)
  - Si suddivide M in blocchi di 64 bit e si applica l'algoritmo di codifica ad ogni blocco, in modo indipendente da blocco a blocco



# Utilizzo degli schemi di codifica a blocchi simmetrica 2/4

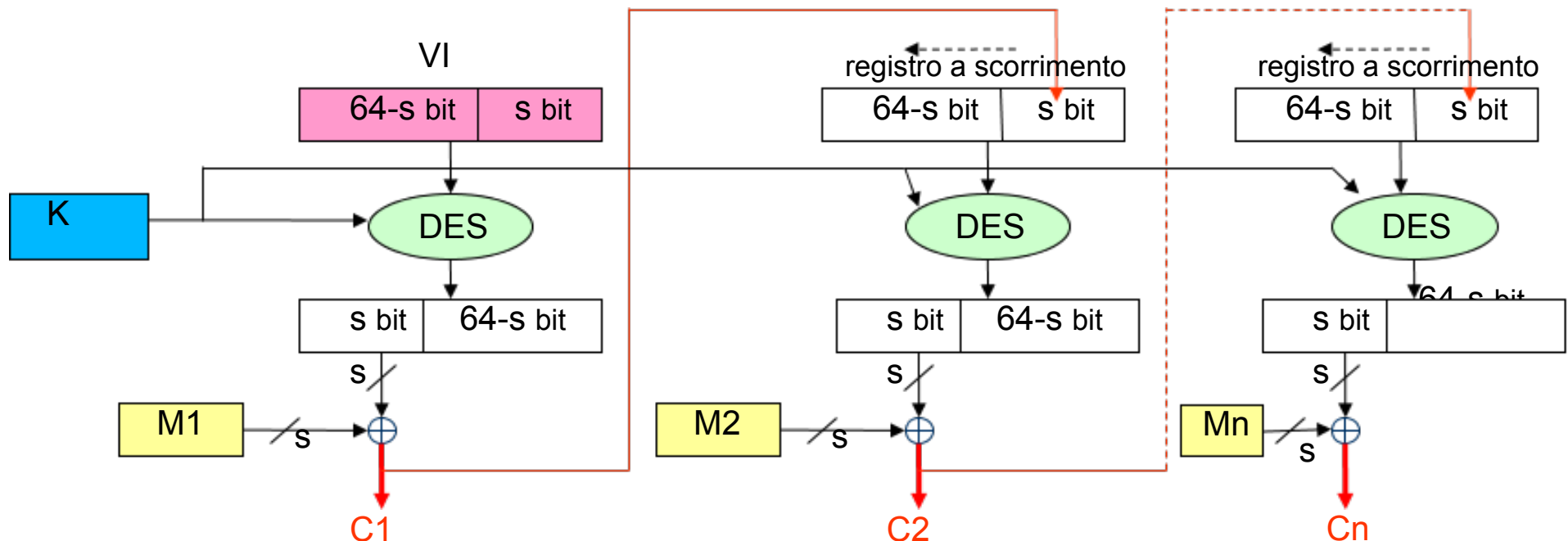
- Cipher Block Chaining (CBC)
  - M viene diviso in blocchi di 64 bit
  - Ogni blocco viene sommato modulo 2 (XOR) con il cifrato del blocco precedente
  - Il primo blocco viene sommato (XOR) con un vettore di inizializzazione (VI)
  - Errori su un bit si propagano nei blocchi seguenti
  - Decifrazione: ogni blocco deve essere decifrato con la chiave K e in seguito deve anche essere sommato (XOR) con il blocco cifrato precedente



# Utilizzo degli schemi di codifica a blocchi simmetrica 3/4

- Cipher Feedback (CFB)

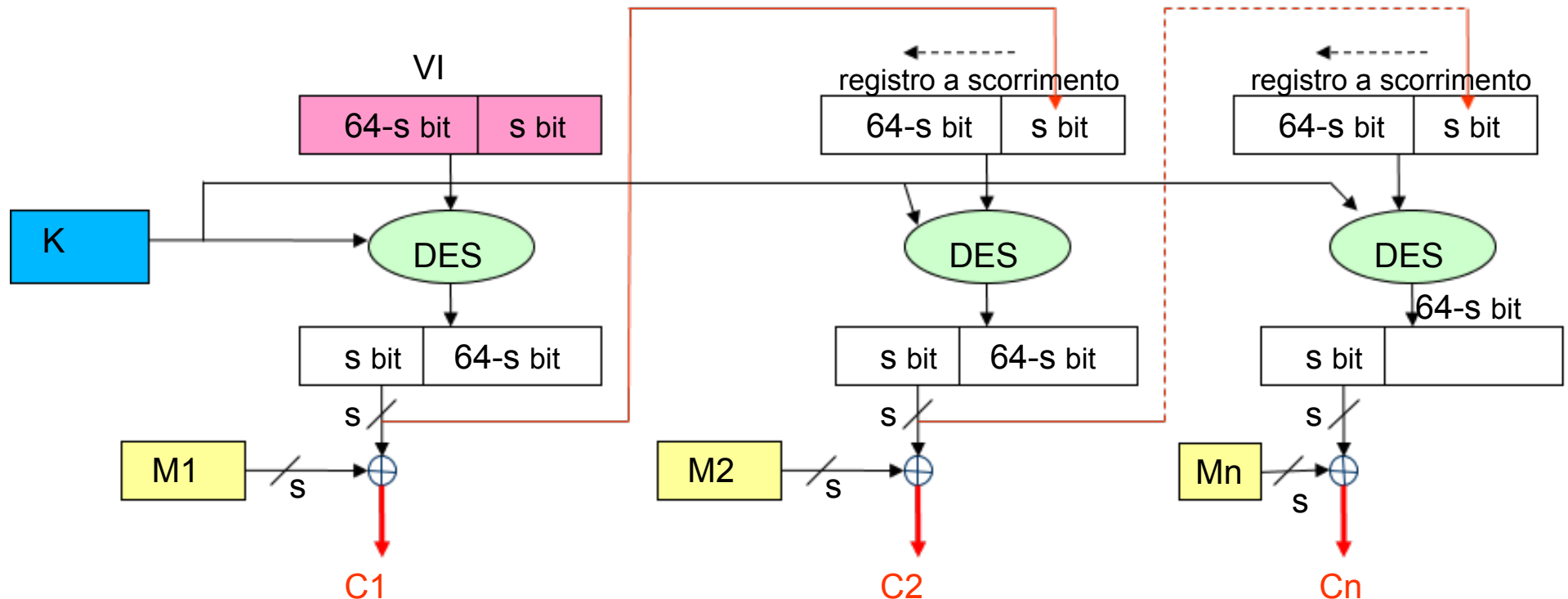
- È possibile convertire il DES a blocchi in una cifratura a flussi
- Non necessità di riempire il messaggio in modo da avere una lunghezza multipla delle dimensioni di un blocco e consente di operare in tempo reale. Viene cifrato e trasmesso immediatamente un carattere alla volta ( $s$  bit). Si lavora elaborando  $s$  bits alla volta.
- Come per CBC ci è concatenazione, quindi una parte di testo cifrato è funzione di tutto il testo in chiaro precedente
- VI = vettore di inizializzazione, 64 bit
- input del DES è dato da 64 –  $s$  bit dell'input precedente e da  $s$  bit ottenuti come XOR tra gli  $s$  bit del testo cifrato in precedenza e  $s$  bit del testo in chiaro



# Utilizzo degli schemi di codifica a blocchi simmetrica 4/4

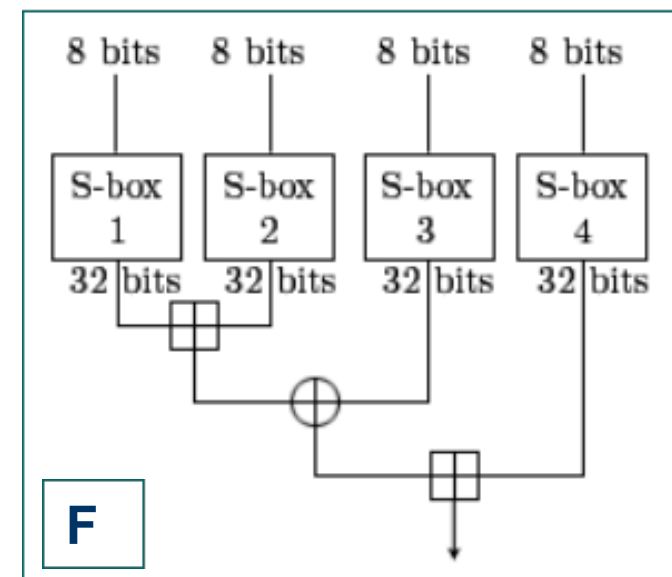
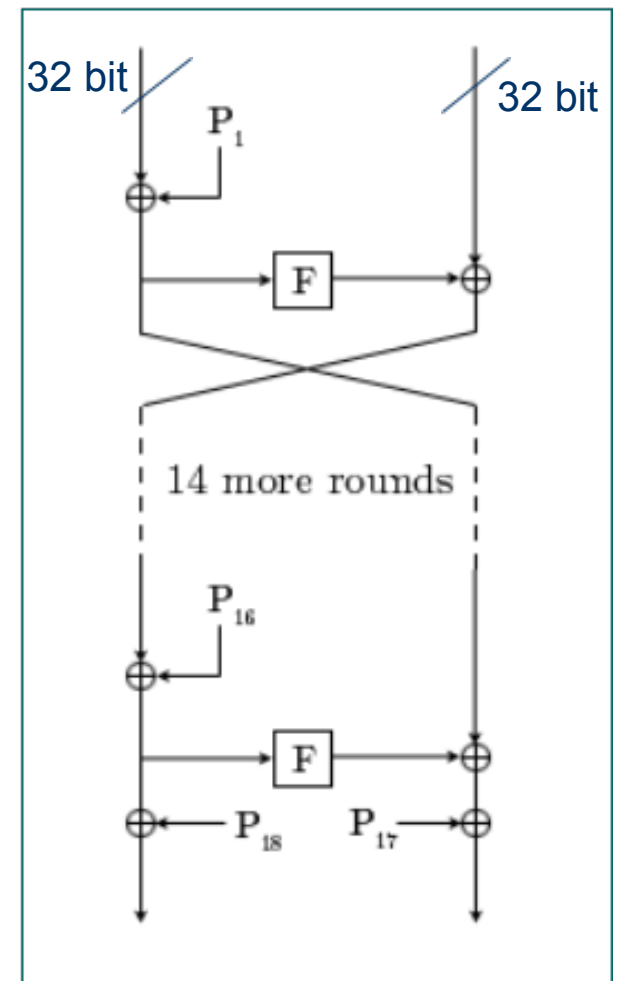
- Output Feedback (OFB)

- Struttura simile a CFB, ma al registro di scorrimento successivo viene inviato il blocco di  $s$  bits più significativi dell'output dell'algoritmo di crittografia
- Vantaggio rispetto a CFB: non propaga errore di trasmissione nei bit, un errore in  $C1$  implica un errore solo nel valore recuperato  $M1$ ; invece con CFB tale errore si propaga anche in successive parti del testo perchè è  $C1$  che va a riempire il registro di scorrimento
- Svantaggio rispetto a CFB: più vulnerabile ad un attacco a modifica el flusso informativo



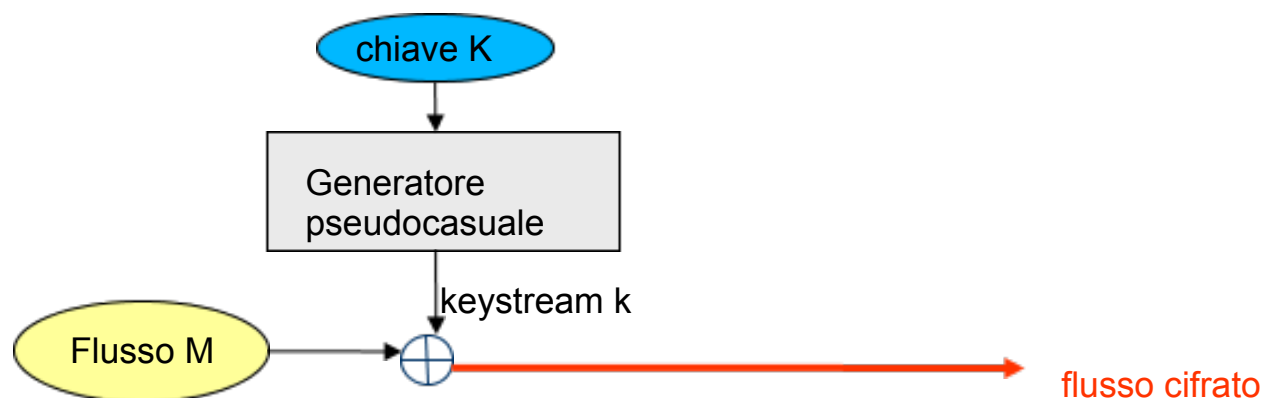
# Schemi avanzati: Blowfish

- Consiste in una iterazione di 16 cicli di Feistel
- Blowfish utilizza blocchi di 64 bit e una chiave di lunghezza variabile fra i 32 e i 448 bit, utilizzata per generare 18 sottochiavi a 32 bit detto P-array
- La funzione F divide i 32 bit in ingresso in quattro byte, utilizzati a loro volta come ingressi delle S-Box. Ciascuna S-Box porta 8 bit in 32. I risultati sono alternativamente sommati in modo algebrico e binario
- A ogni ripetizione del ciclo si usa un elemento diverso del P-array
- All'ultima iterazione si utilizza sia P17 che P18
- Dato che Blowfish è una rete di Feistel, può essere invertito utilizzando il P-array in modo inverso



# Schemi avanzati: RC4 1/2

- RC4 è uno schema di cifratura a flussi progettato da Rivest nel 1987; genera una sequenza pseudo-casuale utilizzata per cifrare e decifrare un flusso dati a partire da una chiave K lunga da 1 a 256 bytes
- È utilizzato nel protocollo TLS/SSL (Transport Layer Security / Secure Sockets Layer) per la trasmissione sicura di pagine web e nel protocollo WEP (Wired Equivalent Privacy) all'interno dello standard IEEE 802.11b
- Schema generale di funzionamento dei cifratori a flussi:
  - La chiave K viene utilizzata come inizializzazione per un generatore di numeri pseudocasuali, che genera una sequenza di bit chiamata keystream
  - La cifratura avviene eseguendo uno XOR bit a bit tra la keystream e il flusso di bit del messaggio in chiaro



# Schemi avanzati: RC4 2/2

- Vettore di stato S composto da 256 byte
- Chiave K può essere lunga da 1 a 256 byte
- Inizializzazione e Generazione

```
for i from 0 to 255
    S[i] := i
endfor
j := 0
for i from 0 to 255
    j := (j + S[i] + key[i mod
keylength]) mod 256
    swap(S[i],S[j])
endfor
```

```
i := 0
j := 0
while
    i := (i + 1) mod 256
    j := (j + S[i]) mod 256
    swap(S[i],S[j])
    output S[(S[i] + S[j]) mod
256]
endwhile
```



# Funzioni One-way

- $F: A \rightarrow B$
- Dato  $b$  appartenente a  $B$  è difficile trovare un  $a$  appartenente ad  $A$  per cui  $F(a)=b$
- Problema difficile, cioè non esiste metodo più veloce del tentativo a forza bruta
- Resistenza a collisione
- collisione = quando due ingressi diversi producono la stessa uscita

# Funzioni Hash

- Funzione che prende ingressi da un largo set A e li mappa in elementi (hash value, o message digest) di lunghezza fissa di un set finito B
- Funzione one-way che serve per trasformare un testo di lunghezza arbitraria in una stringa di lunghezza fissa, relativamente limitata.
- La lunghezza dei valori di hash varia a seconda della funzione usata.
- Non esiste corrispondenza biunivoca tra hash e testo, quindi può verificarsi che ad un hash corrispondano più testi (collisione). La qualità di un hash si misura in base alla difficoltà di individuare due testi che generino collisione. La probabilità di trovare collisione risulta più alta di quanto intuitivamente si possa pensare.
- In crittografia si utilizzano funzioni di hash one-way
  - Per verificare l'integrità di un messaggio (funzione di hash applicata a un testo anche minimamente diverso produce un diverso valore di hash)
  - Firme digitali (più veloce firmare la funzione di hash invece dell'intero documento)
- Caratteristiche:
  - Facili da calcolare
  - Resistenti alle collisioni

# Paradosso del compleanno 1/2

- Date di nascita: insieme di N elementi distinti ed equiprobabili
- $P_u$  probabilità che due persone appartenenti ad un gruppo di K compiano gli anni lo stesso giorni. Complementare  $P_d = 1 - P_u$ .
- Valutazione di  $P_d$ : data una qualunque persona del gruppo vi sono N-1 casi su N in cui il compleanno di una seconda persona avvenga in un giorno diverso; se si considera una terza persona, ci sono N-2 casi su N in cui compie gli anni in un giorno diverso dalle prime due persone, etc.

$$P_d = (N-1)/N \cdot (N-2)/N \cdot \dots \cdot (N-K+1)/N$$

$$\begin{aligned} P_d &= (1-1/N) \cdot (1-2/N) \cdot \dots \cdot (1-(K-1)/N) \\ &= \prod_{j=1, K-1} (1-i/N) \end{aligned}$$

$$\begin{aligned} P_d &= (N-1)(N-2)\dots(N-K+1)/N^{(K-1)} \\ &= (N-1)! / [(N-K)! N^{(K-1)}] \\ &= N! / [(N-K)! N^k] \end{aligned}$$

# Paradosso del compleanno 2/2

$$1-x \sim e^{(-x)}$$

$$\begin{aligned} P_d &\sim \prod_{j=1, K-1} e^{(-j/N)} \\ &= e^{[-\sum_{j=1, K-1} j/N]} \\ &= e^{[-K(K-1)/(2N)]} \end{aligned}$$

$$P_u \sim 1 - e^{[-K(K-1)/(2N)]}$$

$$K \sim \sqrt{2N \ln(1/(1-P_u))}$$

$$K \sim o(\sqrt{N})$$

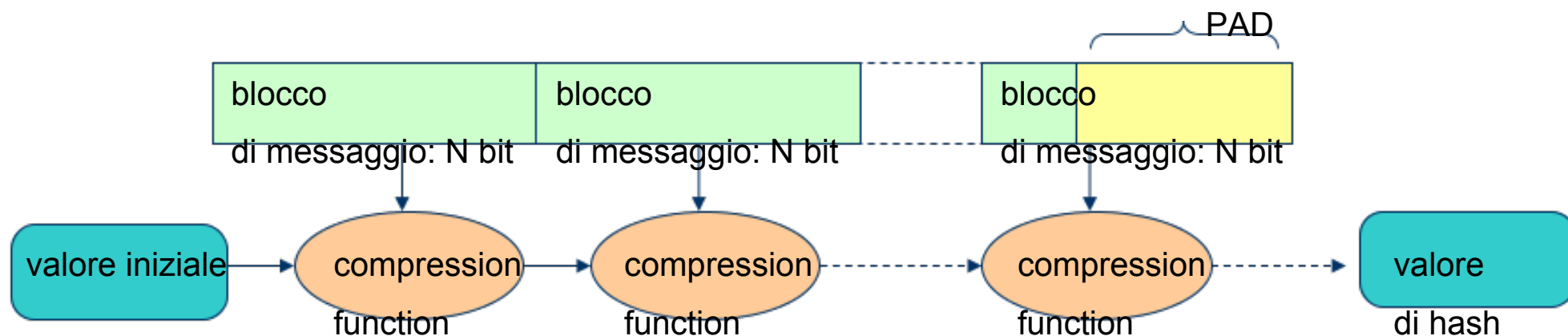
- Con  $P_u=0.5$  e valori di hash a  $N$  bit forniscono  $2^{(N/2)}$  gradi di libertà

- Dato un gruppo di 23 persone, ci sono più del 50% di probabilità che due di esse siano nate nello stesso giorno dell'anno. Con 30 persone la probabilità è maggiore del 70%. Se scegliamo una funzione che trasforma 23 chiavi in un indirizzo di una tabella di 365 elementi, la probabilità che due chiavi NON collidano è solo circa 0.5

# Caratteristiche delle funzioni hash

- Per trovare due messaggi, tra tutti i possibili, con lo stesso hash di  $N$  bit (quindi con collisione), occorre in media considerare  $2^{(N/2)}$
- Fissato un messaggio  $M$ , per trovare un messaggio  $M'$  che ha lo stesso valore di hash occorre considerare  $2^N$  diversi messaggi candidati
- Visto che la funzione di hash deve proteggere contro usi impropri, la lunghezza dell'hash deve essere il doppio di quanto necessario per mantenere sicurezza

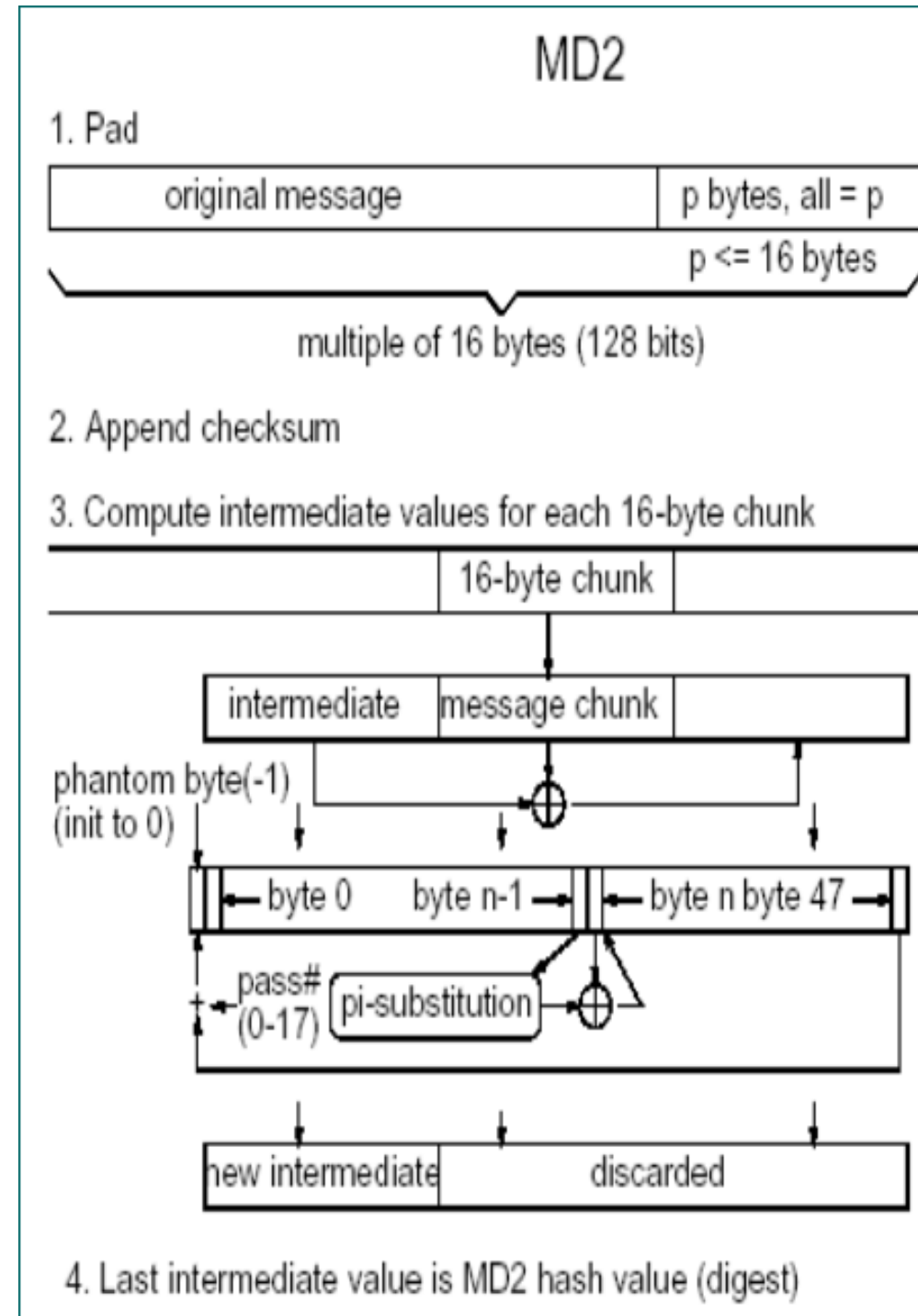
# Meccanismo funzioni hash



- In generale le funzioni di hash presentano una struttura iterativa. Il calcolo del valore di hash dipende da una variabile di concatenamento; questa variabile è inizializzata ad un valore specificato dall'algoritmo di hash e viene aggiornato iterativamente in funzione di blocchi successivi del messaggio originale
- Per poter suddividere il messaggio originale in blocchi di N bit, si aggiungono in fondo dei bit di PAD, per rendere la lunghezza totale un multiplo di N
- MD2, MD4 e MD5 sviluppate da Rivest (1989-90-91) prendono un messaggio di lunghezza arbitraria e producono hash di 128 bit. Dettagli in RFC 1319-1321.
- SHA Secure Hash Algorithm. Sviluppato dal NIST come specificato nel SHS (Secure Hash Standard). Produce hash di 160 bit. RFC 3174

# MD2

- Aggiunta ad M di PAD per avere messaggio multiplo di 16 byte
- Aggiunta in coda di un checksum di 16 byte
- Viene prodotto un digest di 128 bits (16 bytes)
- creazione di un blocco da 48 byte: digest precedente (A) | chunk (B) | A xor B
- $C[n] = C[n] + f(C[n-1])$  sui 48 byte con f permutazione guidata dalle cifre del pi-greco
- ripetizione dei due passi precedenti 18 volte
- output 16 byte



# MD4 1/2

- M + pad + 8 byte per indicare la lunghezza del messaggio prima del padding costituiscono un multiplo di 64 byte
- 4 buffer da 32 bit inizializzati a A=01234567, B=89abcdef, C=fedcba98, D=76543210
- 3 funzioni ausiliarie su blocchi di 32 bit
  - $F(X,Y,Z) = XY \text{ OR } \text{not}(X) Z$ ;  $G(X,Y,Z) = XY \text{ OR } XZ \text{ OR } YZ$ ;  $H(X,Y,Z) = X \text{ XOR } Y \text{ XOR } Z$
- F opera secondo: if X then Y else Z; G opera come maggioranza if at least two of X, Y, Z are on, then G has a "1" bit in that bit position; H è la parità
- Partizionamento in blocchi da 16 byte  $X[i]=M[16 i+j]$
- AA=A, BB=B, CC=C, DD=D
- Round 1 [abcd k s]
  - $a = (a + F(b,c,d) + X[k]) \lll s$
  - [ABCD 0 3] [DABC 1 7] [CDAB 2 11] [BCDA 3 19]
  - [ABCD 4 3] [DABC 5 7] [CDAB 6 11] [BCDA 7 19]
  - [ABCD 8 3] [DABC 9 7] [CDAB 10 11] [BCDA 11 19]
  - [ABCD 12 3] [DABC 13 7] [CDAB 14 11] [BCDA 15 19]



## MD4 2/2

- Round 2
  - $a = (a + G(b,c,d) + X[k] + 5A827999) \lll s$
  - [ABCD 0 3] [DABC 4 5] [CDAB 8 9] [BCDA 12 13]
  - [ABCD 1 3] [DABC 5 5] [CDAB 9 9] [BCDA 13 13]
  - [ABCD 2 3] [DABC 6 5] [CDAB 10 9] [BCDA 14 13]
  - [ABCD 3 3] [DABC 7 5] [CDAB 11 9] [BCDA 15 13]
- Round 3
  - $a = (a + H(b,c,d) + X[k] + 6ED9EBA1) \lll s$
  - [ABCD 0 3] [DABC 8 9] [CDAB 4 11] [BCDA 12 15]
  - [ABCD 2 3] [DABC 10 9] [CDAB 6 11] [BCDA 14 15]
  - [ABCD 1 3] [DABC 9 9] [CDAB 5 11] [BCDA 13 15]
  - [ABCD 3 3] [DABC 11 9] [CDAB 7 11] [BCDA 15 15]
- $A=A+AA; B=B+BB; C=C+CC; D=D+DD$
- output ABCD

# MD5 1/2

- Analogo a MD4
- 4 funzioni e 4 round
  - $F(X,Y,Z) = XY \text{ OR } \text{not}(X) Z$ ;  $G(X,Y,Z) = XZ \text{ OR } Y \text{ not}(Z)$ ;  $H(X,Y,Z) = X \text{ XOR } Y \text{ XOR } Z$ ;  $I(X,Y,Z) = Y \text{ XOR } (X \text{ OR } \text{not}(Z))$
- Round 1 [abcd k s i]
  - $a = b + ((a + F(b,c,d) + X[k] + T[i]) \lll s)$
  - [ABCD 0 7 1] [DABC 1 12 2] [CDAB 2 17 3] [BCDA 3 22 4]
  - [ABCD 4 7 5] [DABC 5 12 6] [CDAB 6 17 7] [BCDA 7 22 8]
  - [ABCD 8 7 9] [DABC 9 12 10] [CDAB 10 17 11] [BCDA 11 22 12]
  - [ABCD 12 7 13] [DABC 13 12 14] [CDAB 14 17 15] [BCDA 15 22 16]
- Round 2 [abcd k s i]
  - $a = b + ((a + G(b,c,d) + X[k] + T[i]) \lll s).$
  - [ABCD 1 5 17] [DABC 6 9 18] [CDAB 11 14 19] [BCDA 0 20 20]
  - [ABCD 5 5 21] [DABC 10 9 22] [CDAB 15 14 23] [BCDA 4 20 24]
  - [ABCD 9 5 25] [DABC 14 9 26] [CDAB 3 14 27] [BCDA 8 20 28]
  - [ABCD 13 5 29] [DABC 2 9 30] [CDAB 7 14 31] [BCDA 12 20 32]

## MD5 2/2

- Round 3 [abcd k s t]
  - $a = b + ((a + H(b,c,d) + X[k] + T[i]) \lll s)$
  - [ABCD 5 4 33] [DABC 8 11 34] [CDAB 11 16 35] [BCDA 14 23 36]
  - [ABCD 1 4 37] [DABC 4 11 38] [CDAB 7 16 39] [BCDA 10 23 40]
  - [ABCD 13 4 41] [DABC 0 11 42] [CDAB 3 16 43] [BCDA 6 23 44]
  - [ABCD 9 4 45] [DABC 12 11 46] [CDAB 15 16 47] [BCDA 2 23 48]
- Round 4 [abcd k s t]
  - $a = b + ((a + I(b,c,d) + X[k] + T[i]) \lll s)$
  - [ABCD 0 6 49] [DABC 7 10 50] [CDAB 14 15 51] [BCDA 5 21 52]
  - [ABCD 12 6 53] [DABC 3 10 54] [CDAB 10 15 55] [BCDA 1 21 56]
  - [ABCD 8 6 57] [DABC 15 10 58] [CDAB 6 15 59] [BCDA 13 21 60]
  - [ABCD 4 6 61] [DABC 11 10 62] [CDAB 2 15 63] [BCDA 9 21 64]

# SHA

- Secure Hash Algorithm
- Stessi principi di MD4 e MD5 ma più complesso e più sicuro
- Viene prodotto un digest di 160 bit
- 80 iterazioni con costanti e funzioni variabili rispetto alle iterazioni
- costanti
  - $K(t) = 5A827999$  (  $0 \leq t \leq 19$  )
  - $K(t) = 6ED9EBA1$  (  $20 \leq t \leq 39$  )
  - $K(t) = 8F1BBCDC$  (  $40 \leq t \leq 59$  )
  - $K(t) = CA62C1D6$  (  $60 \leq t \leq 79$  )
- funzioni
  - $f(t;B,C,D) = (B \text{ AND } C) \text{ OR } ((\text{NOT } B) \text{ AND } D)$  (  $0 \leq t \leq 19$  )
  - $f(t;B,C,D) = B \text{ XOR } C \text{ XOR } D$  (  $20 \leq t \leq 39$  )
  - $f(t;B,C,D) = (B \text{ AND } C) \text{ OR } (B \text{ AND } D) \text{ OR } (C \text{ AND } D)$  (  $40 \leq t \leq 59$  )
  - $f(t;B,C,D) = B \text{ XOR } C \text{ XOR } D$  (  $60 \leq t \leq 79$  )

# Autenticazione

- A vuole essere sicuro dell'identità di B
- A e B sono entrambi in possesso di una chiave segreta  $K$
- A invia a B un piccolo messaggio  $MA$ , che deve essere usato solo una volta da A
- B calcola una funzione di hash  $H(K|MA)$  e la invia ad A
- A calcola la stessa funzione e verifica se coincide con il valore di hash ricevuto da B
- Questo meccanismo di autenticazione si basa sull'osservazione che le funzioni di hash sono costruite in modo da evitare collisioni e in modo che sia difficile ricostruire il messaggio originale a partire dal valore di hash

# Controllo di integrità di messaggi

- $H(M)$  da solo non è sufficiente perché la funzione  $H$  è nota a tutti
- Con  $H(K|M)$  solo chi ha la chiave  $K$  può generare o controllare l'integrità di  $M$
- Questo però non può essere utilizzato perché in genere gli algoritmi di hash usano come valore intermedio la stessa quantità usata come hash output; questo significa che, dato un messaggio  $M$  e il suo hash  $H(M)$ , può essere aggiunta una parte  $A$  in fondo a  $M$  e calcolare un nuovo valore di hash valido, inizializzando l'algoritmo al valore  $H(M)$  ed eseguendo poi l'algoritmo con la parte che si vuole aggiungere
- Per risolvere si utilizza invece  $H(M|K)$

# Cifratura con funzioni di hash

- Modello OFB
  - $K[0] = VI$
  - $K[i] = H(K[i-1])$
  - $C[i] = M[i] + K[i]$
  - inversione:  $M[i] = C[i] + K[i]$
- Modello CFB
  - $C[0] = VI$
  - $C[i] = P[i] + H(K, C[i-1])$
  - Inversione  $P[i] = C[i] + H(K, C[i-1])$

# Meccanismo di Karn

- Schema di Feistel in cui la funzione  $f$  è rappresentata da una funzione di hash  $H$
- Plaintext  $M$ : diviso in due metà  $M = ML \parallel MR$
- Chiave  $K$ : divisa in due metà  $K = KL \parallel KR$
- Ciphertext  $C$ : diviso in due metà  $C = CL \parallel CR$
- Cifratura:
  - $CR = MR + H(ML \parallel KL)$
  - $CL = ML + H(CR \parallel KR)$
- Decifratura:
  - $ML = CL + H(CR \parallel KR)$
  - $MR = CR + H(ML \parallel KL)$



# Schema di Karn modificato

- Lo schema di Karn può essere attaccato nel caso in cui l'attaccante sia in possesso di messaggi  $M=ML|MR$  e  $M'=ML|MR'$  con la stessa parte sinistra  $ML=ML'$
- Modello più robusto:
  - $X=MR+H(ML|KL)$
  - $Y=ML+H(X|KR)$
  - $Z=X+H(Y|KL)$
  - $C=Z|Y$

# Algebra modulare

- Consideriamo due interi  $m, n$  con  $n > 0$
- Il resto  $r$  di  $a/m$  è il più piccolo intero non negativo che differisce da  $a$  per un multiplo di  $m$
- $r = a \bmod m$  (resto)
- Esiste un intero  $q$  per cui  $a - r = qm$ , cioè  $m$  divide  $a - r$
- Due interi  $a$  e  $b$  sono equivalenti modulo  $m$ , se divisi per  $m$  forniscono lo stesso resto
- Sistema completo di resti modulo  $m$ ,  $\mathbb{Z}_m$ ,  $\{a[1], a[2], \dots, a[m]\}$  tali che  $a[i] \bmod m \neq a[j] \bmod m$ 
  - sistema dei minimi resti positivi  $= \{0, 1, 2, 3, 4, 5, 6\}$
  - sistema dei resti a minimo modulo  $= \{-3, -2, -1, 0, 1, 2, 3\}$
- Congruenze lineari:  $b = ax \bmod m$ 
  - Data la congruenza lineare  $ax \bmod m = 1$ ,  $x$  si dice inverso moltiplicativo di  $a$  modulo  $m$

# Numeri primi

- Un numero intero si dice primo quando risulta divisibile solo per se stesso e per 1
- I numeri primi sono infiniti
- **Teorema fondamentale dell'aritmetica:** Ogni numero intero è esprimibile come prodotto di numeri primi; la fattorizzazione è unica.

# Massimo comune divisore (gcd)

- Il massimo comune divisore tra due interi  $a$  e  $b$ , indicato come  $\gcd(a,b)$ , è il più grande tra i divisori comuni ad  $a$  e  $b$
- Dal teorema fondamentale dell'aritmetica il  $\gcd(a,b)$  è il prodotto di tutti i primi comuni ad entrambe le fattorizzazioni di  $a$  e  $b$ , presi con il minimo esponente
- Vale  $\gcd(x,0)=x$  per ogni intero  $x$
- Gli interi  $a$  e  $b$ ,  $a>b>0$ , si dicono **relativamente primi** se  $a$  non è divisibile per  $b$ , cioè se  **$\gcd(a,b)=1$**
- Esempi:
  - $\gcd(12,8)=4$
  - $\gcd(12,24)=12$
  - $\gcd(12,25)=1 \Rightarrow 12$  e  $25$  sono relativamente primi tra loro

# Teorema di Euclide

- Algoritmo che consente di calcolare il  $\gcd(a,b)$ ,  $a,b$  interi,  $a>b>0$
- **$\gcd(a,b)=\gcd(b,a \bmod b)$** 
  - Se  $g=\gcd(a,b)$  abbiamo  $a \bmod g = 0$  e  $b \bmod g = 0$ , quindi anche  $(a-b) \bmod g = 0$ , quindi  $g(a,b)=g(a-b,b)$
  - $g(a,b)=g(b,a-b)=g(b,a-kb)=g(b,a \bmod b)$
  - Si procede con calcolo iterativo:  $\gcd(595,408)=\gcd(408,187)=\gcd(187,34)=\gcd(34,17)=\gcd(17,0)=17$
- Equazione di Bezout: esistono  $u$  e  $v$  in  $\mathbb{Z}$  per cui vale  **$\gcd(a,b)=ua+vb$** 
  - per calcolare  $u$  e  $v$  si parte dall'ultima equazione dell'algoritmo iterato di Euclide e si risale
  - $a=bQ[1]+R[1]$ ,  $b=R[1]Q[2]+R[2]$ ,  $R[1]=R[2]Q[3]+R[3]$ ,  $R[2]=R[3]Q[4]+R[4]$ , ...,  $R[k-2]=R[k-1]Q[k]+R[k]$  sino a  $R[k]=\gcd(a,b)$
  - A Contrario si ha il generico  $R[k]=R[k-2]-R[k-1]Q[k]$  per cui risalendo si ha sempre una struttura del tipo  $R[k]=R[z]P[z]-R[z-1]Q[z]$  sino ad arrivare a  $R[k]=aP[1]-bQ[1]$  che dimostra l'equazione

# Teorema di Euclide: Applicazioni

- | $\gcd(595, 408)$ | $x/y$     | quoziente | resto |
|------------------|-----------|-----------|-------|
|                  | $595/408$ | 1         | 187   |
|                  | $408/187$ | 2         | 34    |
|                  | $187/34$  | 5         | 17    |
|                  | $34/17$   | 2         | 0     |

- $\gcd(595, 187) = 17 = u \cdot 595 + v \cdot 408$  ( $u=11, v=-16$ )  
 $17 = 187 - 5 \cdot 34$ ,  $17 = 187 - 5 \cdot (408 - 2 \cdot 187) = 11 \cdot 187 - 5 \cdot 408$ ,  $17 = 11 \cdot (595 - 408) - 5 \cdot 408 = 11 \cdot 595 - 16 \cdot 408$

- Inverso moltiplicativo:  $ax \bmod m = 1$ ,  $ax \bmod m = ax - Qm$ ,  $\gcd(a, m) = 1$  quindi  $\gcd(a, m) = ax - Qm$

- Esempio  $a=3$ ,  $m=14$

- $1 = 3 - 2 \cdot 1$

- $1 = 3 - (14 - 3 \cdot 4) \cdot 1$

- $1 = 3 - 14 + 3 \cdot 4$

- $1 = 5 \cdot 3 - 14 \cdot 1$

$x/y$	quoziente	resto
$14/3$	4	2
$3/2$	1	1
$2/1$	2	0

# Teorema di Fermat

- Dato  $p$  primo e  $a > 0$  con  $\gcd(a, p) = 1$  si ha  $a^{(p-1)} \bmod p = 1$
- $\text{Comb}[p, 0] = \text{Comb}[p, p] = 1$ ,  $\text{Comb}[p, i] = \frac{p(p-1)!}{(i!(p-i)!)} \bmod p = 0$
- $(x+y)^p \bmod p = \sum_{i=0, p} \text{Comb}[p, i] x^i y^{(p-i)} \bmod p = (x^p + y^p) \bmod p$
- Induzione. Ipotesi:  $k^p \bmod p = k \bmod p$  Tesi:  $(k+1)^p \bmod p = (k^p + 1^p) \bmod p = (k+1) \bmod p$
- Esempi:  $p=3$ ,  $a=4$  quindi  $4^2 \bmod 3 = 16 \bmod 3 = 1$  errato  $p=3$ ,  $a=6$  quindi  $6^2 \bmod 3 = 36 \bmod 3 = 0$

## Funzione Toziente

- **Sistema ridotto di resti modulo  $m$** ,  $Z_m^*$ , definito con resti modulo  $m$  con  $\gcd(a[i], m) = 1$
- Esempio
  - $Z_{10} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
  - $Z_{10}^* = \{1, 3, 7, 9\}$
- Si chiama funzione **Toziente** di  $m$ , indicata con  $F(m)$ , la cardinalità di  $Z_m^*$ 
  - Se  $m$  è primo e  $a < m$  allora  $\gcd(a, m) = 1$ , quindi  $F(m) = m-1$
  - Se  $\gcd(m, n) = 1$  allora  $F(mn) = F(m)F(n)$
  - Se  $p, q$  primi allora  $F(pq) = (p-1)(q-1)$
- Esempio:  $F(10) = F(5 \cdot 2) = (5-1)(2-1) = 4$

# Teorema di Eulero

- Dato  $a$  e  $m$  interi,  $m > 0$ ,  $\gcd(a, m) = 1$  si ha  $a^{F(m)} \bmod m = 1$
- Fermat con  $m = p$  primo,  $a^{p-1} \bmod p = a^{F(p)} \bmod p = 1$
- Esempio:  $m = 10$ ,  $a = 3$ ,  $F(10) = 4$  •  $3^4 \bmod 10 = 81 \bmod 10 = 1$



# Corollari al teorema di Eulero

- Dati due primi  $p$  e  $q$  e gli interi  $n=pq$  e  $m$ ,  $0 < m < n$ , vale:  **$m^{(F(n)+1)} \bmod n = m$**
- se  $\gcd(m,n)=1$  segue dal teorema Eulero
- altrimenti  $m$  multiplo di  $p$  oppure di  $q$ 
  - esempio (non limitativo)  $m=cp$ ,  $c$  intero positivo,  $\gcd(m,q)=1$
  - $m^{F(q)} \bmod q = 1$  (Eulero)
  - $m^{F(n)} \bmod q = m^{(F(q)F(p))} \bmod q = (m^{F(q)} \bmod q)^{F(p)} \bmod q = 1^{F(p)} \bmod q = 1$
  - quindi  $m^{F(n)} = kq + 1$
  - $m \cdot m^{F(n)} = (kq + 1)cp = kcn + m = m^{(F(n)+1)}$
  - $m^{(F(n)+1)} \bmod n = m$
- Dati due primi  $p$  e  $q$  e gli interi  $n=pq$  e  $m$ ,  $0 < m < n$ , vale:  **$m^{(k F(n)+1)} \bmod n = m$** 
  - $m^{(k F(n)+1)} \bmod n = [(m^{F(n)})^k m] \bmod n = [(m^{F(n)} \bmod n)^k m] \bmod n = [1^k m] \bmod n = m$

# RSA

- Sistema di crittografia a chiave pubblica, inventato nel 1978 da Rivest, Shamir e Adleman
- Sicurezza basata sull'intrattabilità del problema della fattorizzazione di numeri interi molto grandi
  - GENERAZIONE DELLE CHIAVI
  - CIFRATURA
  - DECIFRATURA
  - ASPETTI COMPUTAZIONALI
  - SICUREZZA

# RSA: generazione delle chiavi

- Scelta di  $p$  e  $q$  primi, molto grandi
- $n=pq$  e  $F(n)=(p-1)(q-1)$
- Scelta in modo casuale di  $e$  con  $1 < e < F(n)$  e  $\gcd(e, F(n))=1$ . Nota che se  $e$  è primo ed  $e > p$ ,  $e > q$ ,  $e < F(n)$  la condizione è soddisfatta infatti  $\gcd(e, F(n)) = \gcd(e, (p-1)(q-1)) = \gcd((p-1)(q-1), (p-1)(q-1) \bmod e) = \gcd((p-1)(q-1), 1) = 1$
- $d$  tale che  $ed \bmod F(n) = 1$ , cioè inverso moltiplicativo di  $e$

# RSA: cifratura

- Chiave di cifratura  $(n,e)$
- $M$  rappresentato come un numero intero nel range  $0, \dots, n-1$  eventualmente con meccanismi a blocchi successivi
- $C = M^e \bmod n$

# RSA: decifratura

- Chiave di decifratura  $(n,d)$
- $M = C^d \bmod n$
- Il ruolo di  $e$  e di  $d$  sono completamente invertibili, si può dare a  $(n,e)$  il ruolo di chiave pubblica o di chiave privata
- Motivazioni per il funzionamento:
  - $ed \bmod \phi(n) = 1$  cioè  $ed = k\phi(n) + 1$
  - $(M^e \bmod n)^d \bmod n = M^{ed} \bmod n = M^{k\phi(n)+1} \bmod n = M$  (Corollario Eulero)

# RSA: aspetti computazionali

- Se l'esponenziazione viene realizzata nel campo degli interi e poi ridotta modulo  $n$  occorre lavorare con numeri molto elevati. Invece, grazie ad una proprietà dell'algebra modulare, si possono ridurre i risultati intermedi modulo  $n$ :
- $[(a \bmod n) (b \bmod n)] \bmod n = (ab) \bmod n$
- Efficienza: approccio in grado di calcolare  $a^m$  in  $k$  passi, con  $k$  numero di cifre binarie che compongono  $m$ , piuttosto che in  $m$  passi

$$m = c[0]2^0 + c[1]2^1 + \dots + c[k]2^k$$

$$a^m = a^{(c[0]2^0 + c[1]2^1 + \dots + c[k]2^k)}$$

$$a^m = (a)^{c[0]} (a^2)^{c[1]} \dots (a^{2^k})^{c[k]}$$

$$b[0] = 1$$

$$a[j] = a[j-1]^2$$

$$b[j+1] = \text{If}[c[j]=0; b[j]; b[j]a[j]]$$

# RSA: sicurezza

- $d$  può essere ricavato una volta nota la fattorizzazione di  $n=pq$ ; infatti, noti  $p$  e  $q$  si può calcolare  $F(n)$  e quindi ricavare  $d$  risolvendo la congruenza lineare  $ex \bmod F(n) = 1$
- La sicurezza di RSA dipende dal problema di fattorizzazione dell'intero  $n$
- Violare RSA è equivalente al problema di fattorizzare un intero grande; non è impossibile violare RSA ma è computazionalmente difficile; il prodotto di due interi grandi viene eseguito rapidamente, la fattorizzazione di  $n=pq$  in  $p$  e  $q$  è molto difficile
- Gli algoritmi di fattorizzazione più efficienti hanno complessità  $O(e^{(\ln(n)^{1/3}) \ln(\ln(n))^{2/3}})$  dove  $n$  è il numero da fattorizzare
  - Esempio: la fattorizzare un intero  $n$  di 664 bit richiede almeno  $10^{23}$  passi; una rete di  $10^6$  computer ciascuno che esegue  $10^6$  passi al sec impiegherebbe 4000 anni a fattorizzare  $n$
- Un altro attacco consiste nel calcolo di  $F(n)$ : ma si dimostra che per interi del tipo  $n=pq$ , con  $p$  e  $q$  primi il calcolo di  $F(n)$  e la fattorizzazione di  $n$  sono equivalenti dal punto di vista computazionale

# Distribuzione delle chiavi

- Distribuzione delle chiavi: insieme delle precauzioni che permettono a due entità che vogliono comunicare per mezzo di un crittosistema simmetrico, di disporre di una chiave segreta comune, evitando che venga intercettata da possibili attaccanti
- Metodo di Diffie-Hellman, basato sul problema del logaritmo discreto (DLP), permette a due entità di concordare una chiave segreta per mezzo di uno scambio di informazioni non segreto
- Alternativa: inviare la chiave segreta tramite un algoritmo di codifica a chiave pubblica
- Una volta scambiata la chiave segreta tramite una di queste alternative, la si può utilizzare in un sistema di crittografia simmetrica => i sistemi simmetrici sono più veloci rispetto a quelli asimmetrici, possono essere utilizzati anche per criptare informazioni in tempo reale, come ad esempio una telefonata

# Ordine

- Dati due interi  $a$  e  $m$ ,  $m > 0$ ,  $\gcd(a, m) = 1$ . Si dice ordine di  $a$  modulo  $m$ ,  **$\text{ord}(a, m)$** , il più piccolo intero positivo  $k$  per cui vale  **$a^k \bmod m = 1$**
- Se  $p$  è primo allora  $\text{ord}(a, p) = F(p) = p - 1$  infatti  $a^{(p-1)} \bmod p = 1$  direttamente dal teorema di Fermat

## Radice primitiva

- Dati due interi  $a$  e  $m$ ,  $m > 0$ ,  $\gcd(a, m) = 1$ . Si dice che  $a$  è una radice primitiva modulo  $m$  se  **$\text{ord}(a, m) = F(m)$**  e quindi  $a^{F(m)} \bmod m = 1$
- $a$  è radice primitiva se e solo se l'insieme  $\{a, a^2, \dots, a^{F(m)}\}$  forma un insieme ridotto di resti modulo  $m$

## Indice o logaritmo discreto

- Dati due interi  $a$  e  $m$ ,  $m > 0$ ,  $\gcd(a, m) = 1$ . Sia  $g$  una radice primitiva modulo  $m$ . Esiste un unico intero  $v(a)$ , con  $0 \leq v(a) < F(m)$ , per cui vale:  **$a \bmod m = g^{v(a)} \bmod m$**
- Tale  $v(a)$  si dice indice di  $a$  rispetto a  $g$  modulo  $m$



# Problema del logaritmo discreto

- Siano dati un primo  $p$ , una radice primitiva  $g$  modulo  $p$  e un elemento  $h$  dell'insieme ridotto di resti  $\mathbb{Z}_p^*$ ; trovare l'intero  $x$ ,  $0 \leq x < p-1 = \phi(p)$  tale che  **$h = g^x \bmod p$**
- La fattorizzazione di un intero e il calcolo di un logaritmo discreto sono problemi di equivalente difficoltà
- IL problema di Diffie-Hellman è in relazione con il problema del logaritmo discreto e la sua difficoltà computazionale garantisce la sicurezza dello scambio delle chiavi secondo l'algoritmo di Diffie-Hellman

# Scambio delle chiavi secondo lo schema Diffie-Hellman

- A e B vogliono concordare una chiave segreta  $K$
- A sceglie un primo  $p$  elevato ed un intero  $g$  tale che  $1 < g < p$  e li comunica a B,  $(p, g)$  sono considerati pubblici
- A sceglie un testo  $M_A$  che mantiene segreto ed invia a B  $g^{M_A} \bmod p$
- B sceglie un testo  $M_B$  che mantiene segreto ed invia a A  $g^{M_B} \bmod p$
- Dato che  $g^{(M_A M_B)} = (g^{M_A})^{M_B} = (g^{M_B})^{M_A}$  sia A che B possono calcolare  $K = g^{(M_A M_B)} \bmod p$
- Un attaccante riesce ad essere in possesso di  $p$ ,  $g$ ,  $g^{M_A} \bmod p$ ,  $g^{M_B} \bmod p$ , ma non conosce né  $M_A$  né  $M_B$  per cui non può calcolare  $K$  se non risolvendo il problema del logaritmo discreto e quindi ottenendo  $M_A$  oppure  $M_B$  dalle trasmissioni tra A e B

# Crittosistema di El Gamal: generazione delle chiavi

- Ogni utente genera una chiave pubblica e la corrispondente chiave privata
- Sceglie un primo  $p$  elevato e calcola una radice primitiva  $g$
- sceglie un intero  $a$  con  $1 \leq a < p-1$  e calcola  $z = g^a \bmod p$
- CHIAVE PUBBLICA:  $(p, g, z)$
- CHIAVE PRIVATA:  $a$

## Crittosistema di El Gamal: cifratura

- B vuole inviare un messaggio  $M$  ad A
- Si procura la chiave pubblica di A:  $(p, g, z)$
- Rappresenta  $M$  come un intero nell'intervallo  $0, 1, \dots, p-1$
- Sceglie a caso un intero  $k$ ,  $1 \leq k < p-1$
- $c_1 = g^k \bmod p$
- $c_2 = (M z^k) \bmod p$
- $C = (c_1, c_2)$

# Crittosistema di El Gamal: decifratura

- $(c_1^{-a} c_2) \bmod p = ((g^k)^{-a} M z^k) \bmod p = (g^{-ak} M g^{ak}) \bmod p = M$
- L'inversione necessita di  $a$

# Crittosistema di El Gamal: sicurezza

- per risalire al messaggio in chiaro  $M$  a partire dal cifrato  $C$  e dalla chiave pubblica  $(p, g, z)$  un attaccante deve calcolare  $a$  e per fare questo deve risolvere il problema del logaritmo discreto

# Firma digitale 1/2

- Sistema a chiave pubblica:
  - procedura per cifrare E
  - procedura per decifrare D
- Proprietà:
  - $D(E(M))=M$
  - $E(D(M))=M$
- Il procedimento di cifratura-decifratura che caratterizza un sistema a chiave pubblica può anche essere invertito => usare la chiave segreta per cifrare e la chiave pubblica per decifrare
- Questo non produce sicurezza: se A usa la sua chiave segreta per cifrare un messaggio e lo invia a B, chiunque altro oltre a B può decifrare usando la chiave pubblica di A
- Questa procedura assicura autenticità: se B riesce a decifrare il messaggio con la chiave pubblica di A significa che il messaggio è stato cifrato con la chiave privata di A, che solo A possiede

# Firma digitale 2/2

- A vuole inviare un messaggio M a B
  - Per garantire riservatezza => cifra M con la chiave pubblica di B
  - Per garantire la autenticità => cifra M con la propria chiave privata
  - Combinando i due procedimenti => si generano le firme digitali
- Firme digitali: sono utilizzate quando si necessita del riconoscimento del mittente di un messaggio cifrato e per garantire che il messaggio non possa essere modificato dal destinatario
- ESEMPIO: A vuole inviare un messaggio firmato a B.
  - A calcola la sua firma F corrispondente al messaggio M, utilizzando la sua chiave segreta:  $F=D(M,K_A)$
  - A cifra la firma F tramite la chiave pubblica di B:  $C=E(F,K_B)$  e aggiunge in chiaro il nome del mittente A ed invia a B
  - B utilizza la propria chiave segreta per decifrare C.  $D(C,K_B)=F$
  - B utilizza la chiave pubblica di A per ottenere il messaggi originale  $E(F,K_A)=M$
- In questo modo M è autenticato e non modificabile: A non può negare di esser stato il mittente e B non può cambiare il contenuto di M non essendo in grado di riprodurre la firma posta da A
- La firma digitale dipende sia dall'autore sia dal messaggio per cui viene calcolata

# Protocollo di Massey-Omura 1/2

- Realizza una comunicazione segreta tra due entità A e B senza la necessità di scambiarsi nessuna chiave.
- Può essere utilizzato come crittosistema a chiave privata o come metodo sicuro per scambiarsi una chiave tra due entità
- Principio di funzionamento:
  - A mette un messaggio M in una scatola, chiude la scatola con una chiave KA e la invia a B
  - B aggiunge anche una propria chiave KB e invia ad A la scatola chiusa con KA e KB
  - A apre la propria chiave e rispedisce a B la scatola, ora chiusa con KB
  - B può aprire con la propria chiave KB e reperire il contenuto della scatola
- Si osserva che in nessuna fase la scatola viaggia senza almeno una chiave e dunque senza protezione

# Protocollo di Massey-Omura 2/2

- Protocollo: A e B scelgono un primo elevato  $p$  in comune, che non deve essere necessariamente mantenuto segreto
- Ciascuno calcola le coppie di chiavi  $eA, dA$  e  $eB, dB$  in questo modo
  - $eA, eB$  relativamente primi con  $F(p)=p-1$ ,  $\gcd(eA, F(p))=1$ ,  $\gcd(eB, F(p))=1$
  - $dA, dB$  inverso moltiplicativo modulo  $F(p)$  di  $eA, eB$  rispettivamente, cioè si ha che  $eAdA \bmod F(p)=1$ ,  $eBdB \bmod F(p)=1$
- A vuole inviare  $M$ ,  $1 < M < p$  a B
  - A invia ad B  $J1 = M^{eA} \bmod p$
  - B invia a A  $J2 = J1^{eB} \bmod p = M^{(eB eA)} \bmod p$
  - A invia ad B  $J3 = J2^{dA} \bmod p = M^{(dA eB eA)} \bmod p = M^{eB} \bmod p$
  - B calcola  $J4 = J3^{dB} \bmod p = M^{(eB dB)} \bmod p = M$



# Poker elettronico

- Applicazione del protocollo di Massey-Omura
- Due giocatori A e B devono avere 5 carte per uno, da un mazzo di 52, senza vedere le carte dell'altro
  - A codifica tutte le 52 carte con  $e_A$  e le invia a B
  - B rimescola, le codifica con  $e_B$  e invia il mazzo di carte ad A
  - A sceglie 10 carte, ne decifra 5 con  $d_A$  e le invia a B
  - B applica alle 10 carte la propria chiave di decifratura  $d_B$  (e quindi ne riesce a decifrare solo 5, che costituiscono le carte di B). Invia le 10 carte ad A
  - A applica la propria chiave e decifra le restanti 5 carte

# Sicurezza dei protocolli 1/2

- ARP:
  - A manda un broadcast con la richiesta del MAC address di B
  - Problema se un'altra macchina, C, risponde fingendosi di essere B (C aspetta che B risponda ma risponde anche C, sfruttando il fatto che A aggiorna la propria tabella in ogni caso)
- IP:
  - Source routing: è possibile forzare un source routing e far instradare i pacchetti attraverso l'attaccante
  - TTL: vario il TTL così il pacchetto muore prima di arrivare a destinazione
- ICMP:
  - Visto che è quello che controlla IP è il più soggetto ad attacchi.
  - Redirect il traffico viene forzato a passare presso l'attaccante
  - Echo request viene innondato un determinato host
- RIP: DV (distance vector) con distanze errate forzano dei percorsi falsati
- OSPF: LS (link state) errati indicano un link funzionante come non buono e quindi deviano il traffico

# Sicurezza dei protocolli 2/2

- TCP: inserimento in una comunicazione tramite sincronizzazione di: porta, ordine dei pacchetti, sequence number, etc
- DNS: forzatura del caricamento di una tabella tra primario e secondario, meccanismi di aggiornamento dinamico delle entry, meccanismi di forward
- SMTP: non c'è un'autenticazione, potenziale sostituzione; il demone può funzionare da root con relativi problemi
- TELNET: trasferimento in chiaro coppia login/password
- NTP: sincronizzare clock di computer in rete; forzatura re-schedulare dei processi, crash legato errori parametri temporali a livello kernel
- FINGER: informazioni sugli utenti del sistema; molto attaccato quindi solitamente la sua porta viene bloccata
- NFS: gestione di un file system distribuito; possibilità modifica file, problematiche gestione dei diritti

# Firewall

- Sistema di sicurezza posto tra una rete interna e internet che controlla il collegamento tra le due reti, cerca di proteggere la rete interna da attacchi provenienti da internet e fornisce un unico punto di accesso tra rete interna e internet in cui concentrare e imporre la sicurezza dell'intero sistema
  - tutto il traffico proveniente da interno verso esterno e viceversa deve essere analizzato attraverso il firewall
  - solo il traffico autorizzato dalla politica di sicurezza della rete locale può attraversare il firewall
- Per controllare l'accesso il firewall può utilizzare diverse tecniche:
  - controllo dei servizi: decide quali servizi permettere e quali bloccare; filtra traffico sulla base di indirizzo IP e porta TCP
  - controllo di direzione: determina la direzione in cui possono essere attivate alcune richieste di servizio
  - controllo degli utenti: determina l'accesso ad un servizio sulla base dell'utente che lo richiede
  - controllo del comportamento: controlla il modo in cui vengono utilizzati certi servizi (esempio eliminare messaggi spam dalla posta)

# Firewall: architetture

- Router a filtraggio di pacchetti:
  - un router che implementa una serie di regole ad ogni pacchetto IP in ingresso e quindi inoltra o elimina il pacchetto; tali regole si basano su indirizzi IP sorg e dest, porte sorg e dest a livello di trasporto, interfaccia del router da cui proviene il pacchetto
- Gateway a livello di applicazioni
  - chiamato anche proxy server
  - l'utente contatta il gateway tramite telnet o ftp e il gateway chiede all'utente il nome dell'host remoto a cui accedere; utente invia le informazioni di autenticazione e il gateway contatta l'host remoto e invia i segmenti TCP della comunicazione tra i due host; il gateway può rendere disponibili solo alcune delle funzionalità di una certa applicazione e impedirne altre per ragioni di sicurezza

# IPsec 1/4

- Specifica funzioni relative alla sicurezza per essere applicate su IPv4 e IPv6
- Consente alle applicazioni di utilizzare comunicazioni cifrate ed autenticate
- Provvede a tre funzioni:
  - AUTENTICAZIONE: garantisce che un pacchetto ricevuto sia in effetti stato trasmesso dall'origine indicata nell'intestazione; garantisce che il pacchetto non sia stato alterato
  - SEGRETEZZA: consente ai nodi di crittografare messaggi per impedire intercettazioni
  - SCAMBIO SICURO DI CHIAVI
- Protocolli
  - Authentication Header (AH): sola autenticazione
  - Encapsulating Security Payload (ESP): Autenticazione con cifratura delle comunicazioni

# IPsec 2/4

- IPsec assicura le seguenti APPLICAZIONI:
  - Login remoti, posta elettronica, trasferimento files
  - Aziende con reti locali disperse geograficamente: all'interno di ogni rete locale il traffico IP è non sicuro; il traffico esterno viene protetto da funzionalità IPsec che eseguono la crittografia del traffico proveniente da interno verso esterno e la decrittografia del traffico da esterno verso interno
  - Accesso remoto sicuro via internet: l'utente, con un sistema dotato di protocolli IPsec, può eseguire una chiamata locale a un provider internet e poi eseguire un accesso sicuro ad una rete aziendale
  - Connettività con partner commerciali: la garanzia delle tre funzionalità prima esposte garantisce comunicazioni sicure tra aziende diverse
  - Miglioramento della sicurezza del commercio elettronico: anche se alcune applicazioni web sono dotate di protocolli interni per la sicurezza, IPsec aiuta a migliorarne la sicurezza
  - A livello di routing può garantire che:
    - Il messaggio di pubblicizzazione di un router provenga da un router autorizzato
    - il messaggio di redirection provenga dal router al quale è stato inviato il pacchetto iniziale
    - L'aggiornamento dei routing non venga falsificato da un estraneo

# IPsec 3/4

- ASSOCIAZIONI DI SICUREZZA
- Un'associazione di sicurezza è una relazione monodirezionale tra un mittente e un destinatario che riguarda i servizi di sicurezza rispetto al traffico trasportato. E' caratterizzata dai seguenti parametri:
  - Security Parameter Index: identificativo
  - IP destinazione
  - Tipo di protocollo : AH o ESP
  - Sequence Number (32 bit): contatore di pacchetti
  - Lifetime: tempo di vita dell'associazione
  - Informazioni specifiche dell'AH: algoritmo di autenticazione, chiavi, tempo di vita delle chiavi
  - Informazioni specifiche dell'ESP: algoritmo di autenticazione e crittografia, chiavi, tempo di vita delle chiavi
  - Antireply window: indica se il pacchetto AH o ESP in ingresso è un reply (in un attacco a reply un estraneo ottiene una copia di un pacchetto autenticato e successivamente lo trasmette alla destinazione prevista; la ricezione di pacchetti duplicati può disturbare il funzionamento del servizio)



# IPsec 4/4

- MODALITA' FUNZIONAMENTO: sia AH che ESP possono essere trasportati direttamente o tramite tunnel
- trasporto diretto:
  - per proteggere protocolli superiori (TCP, UDP)
  - IPv4: AH e ESP messi come payload
  - IPv6: inseriti come header successivi a quello base
  - ESP: protegge solo payload
  - AH: protegge anche parte dell'header
- trasporto in modalità tunnel:
  - si ha la protezione dell'intero pacchetto, payload e header; il pacchetto e il campo sicurezza diventano il payload di un nuovo pacchetto IP esterno, con un nuovo header esterno; il pacchetto originale viaggia in un tunnel e nessun router presente nel percorso di instradamento è in grado di esaminare o modificare il contenuto del pacchetto incapsulato

# Sistemi di sicurezza biometrici

- L'utilizzo di metodi quali la scansione dell'iride o il riconoscimento delle impronte digitali
- Firma digitale (con attivazione biometrica)
  - nella pratica, i certificati digitali necessari per la firma sono rilasciati, dalle società certificatrici, all'interno di Smart Card; l'apposizione della firma digitale richiede il possesso di una smart card e la conoscenza di una password per l'abilitazione del certificato contenuto all'interno di essa. Questi due elementi non sono in grado di verificare che colui che firma digitalmente a distanza un documento sia veramente il proprietario del certificato di firma, in quanto la carta e la password potrebbero essere state rubate o più semplicemente prestate. L'utilizzo di tecniche biometriche (ad esempio la verifica dell'impronta digitale) è un modo di accertare con sicurezza la presenza dell'individuo.