Sicurezza

Argomenti trattati

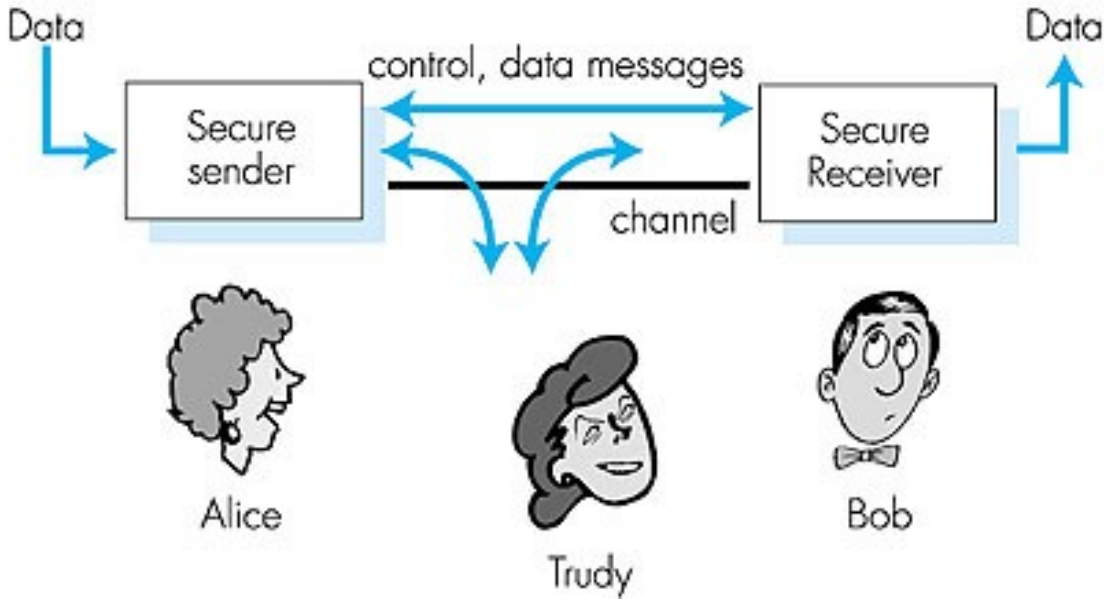
Metodologie

- Cosa significa sicurezza?
- Crittografia
- Integrità dei messaggi e firma digitale
- Autenticazione
- Distribuzione delle chiavi e certificazione

Sicurezza in pratica:

- Livello applicativo: e-mail sicura
- Livello di trasporto: commercio Internet, SSL….
- Strato di rete: sicurezza IP

Amici e nemici: Alice, Bob, Trudy



- Bob e Alice (amanti?) vogliono comunicare in modo "sicuro"
- Trudy ("intruso/intrusa") può intercettare, rimuovere o aggiungere messaggi
- Trudy vuole modificare, conoscere o impedire la comunicazione

Obiettivi

Il sistema deve soddisfare i seguenti requisiti di sicurezza

- Disponibilita'
- Riservatezza
- Integrità
- Autenticazione
- Non ripudiazione

Requisiti di disponibilitá

Rendere disponibili a ciascun utente abilitato le informazioni alle quali ha diritto di accedere, nei tempi e nei modi previsti.

Nei sistemi informatici, i requisiti di disponibilità includono prestazioni e robustezza.

Requisiti di integrità

Impedire la **alterazione** diretta o indiretta delle informazioni, sia da parte di utenti e processi non autorizzati, che a seguito di eventi accidentali.

Se i dati vengono alterati e' necessario fornire strumenti per poterlo verificare facilmente.

Requisiti di riservatezza

Nessun utente deve poter ottenere o dedurre dal sistema informazioni che non è autorizzato a conoscere.

Se una informazione e' protetta, o se esiste una comunicazione in atto fra due utenti o processi in un certo contesto, non deve permettere di dedurre altre informazioni riservate.

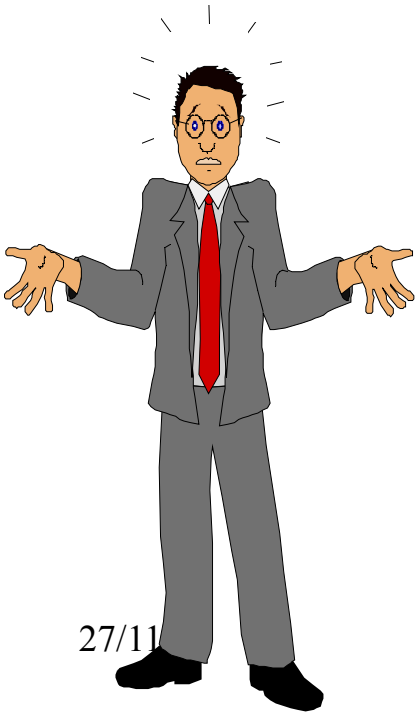
Requisiti di autenticazione

Ciascun utente deve poter verificare l'autenticita' delle informazioni.

Si richiede di poter verificare se una informazione - non necessariamente riservata - e' stata manipolata.

Requisiti di non ripudiazione

Nessun utente deve poter ripudiare o negare messaggi da lui spediti o firmati.



Evitare che le informazioni e i messaggi siano negati dal firmatario in tempi successivi (es. firma di un contratto)

Tipi di intruso

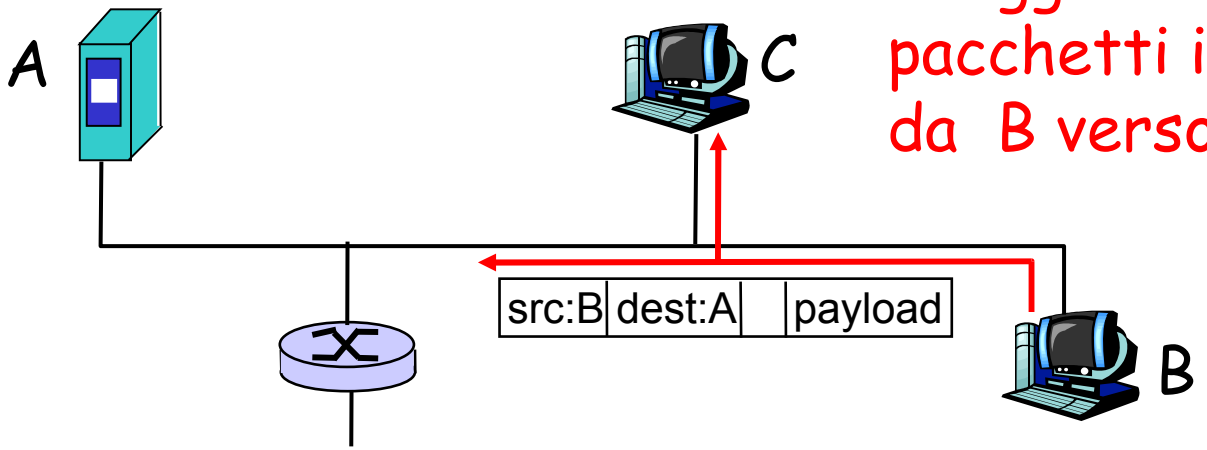
- Intruso passivo: legge il messaggio senza alterarlo (es. password)

- Intruso attivo:
  - Può alterare il messaggio
  - Può inviare messaggi falsi spacciandosi presso il ricevente per il mittente autentico

Minacce alla sicurezza in Internet

Packet sniffing (to sniff = "odorare"):

- Evidente in mezzi condivisi
- Un adattatore di rete programmato ad hoc (NIC) legge tutti i pacchetti in transito
- Tutti i dati non cifrati (es.: password) possono essere letti

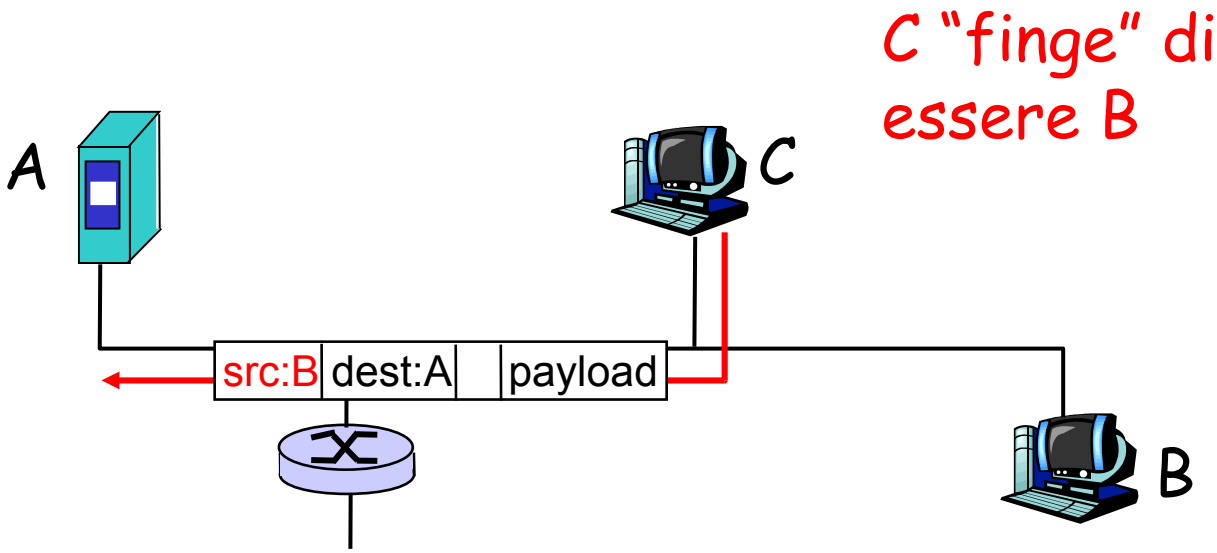


C legge tutti i pacchetti inviati da B verso A

Minacce alla sicurezza (cont.)

IP Spoofing:

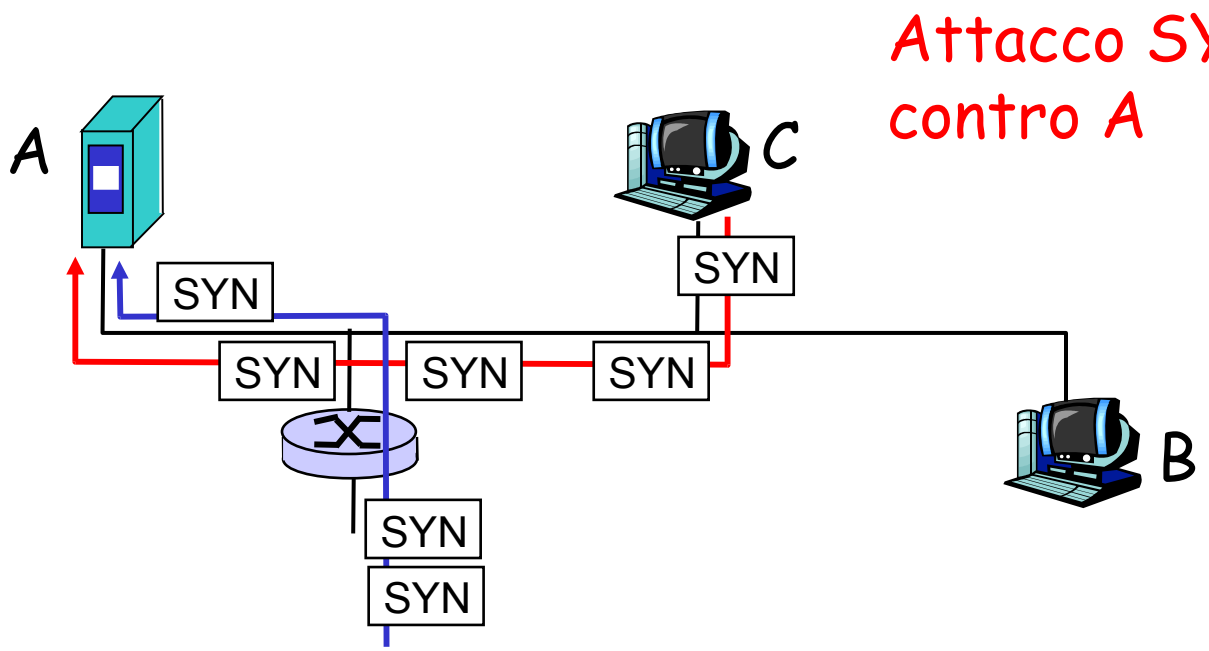
- Un host genera pacchetti IP con indirizzi di sorgente falsi
- Il ricevente non è in grado di stabilire se l'origine dei pacchetti sia quella autentica



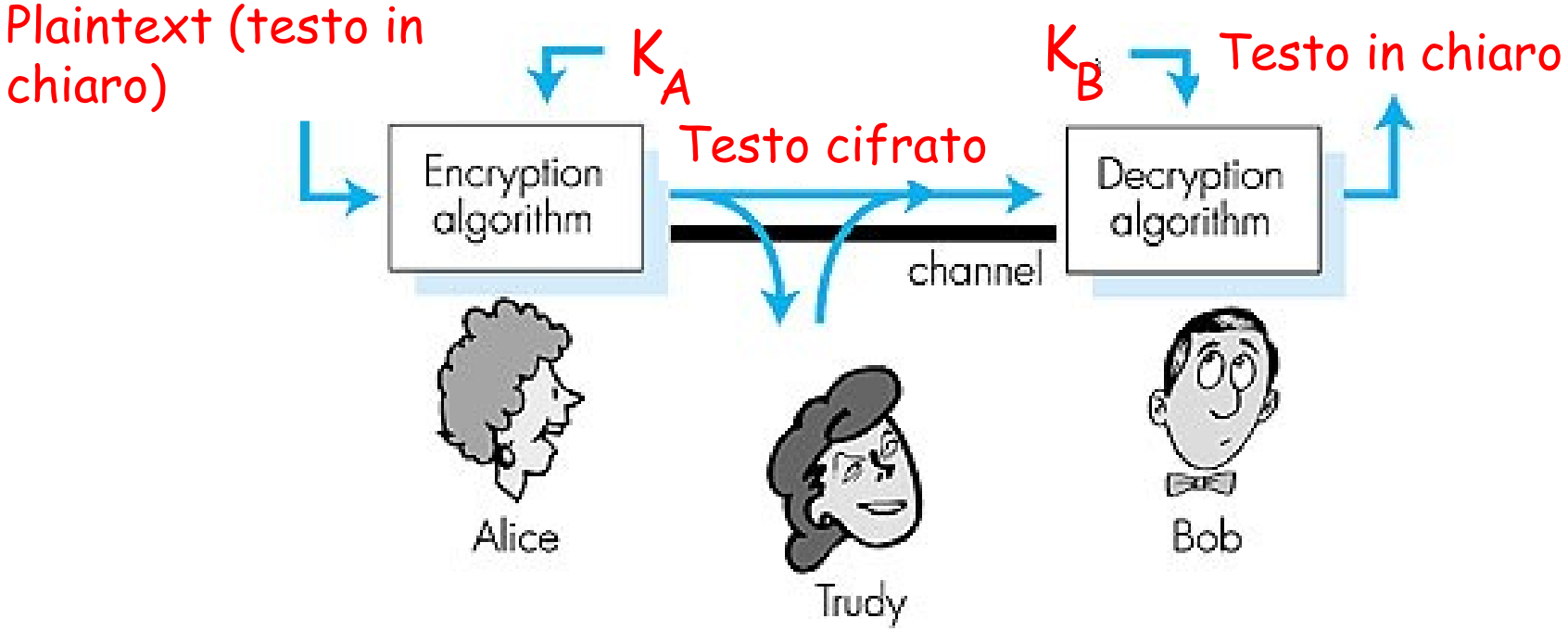
Minacce alla sicurezza (cont.)

Denial of service (DoS):

- Flusso di pacchetti "maligni" che sommerge il ricevente
- Distributed DoS (DDoS): attacco coordinato multiplo



Crittografia: nomenclatura



Chiave segreta (o simmetrica): chiavi del mittente e del ricevente identiche

Chiave pubblica (o asimmetrica): chiave di cifratura e , di decifratura diverse

Crittografia simmetrica

Algoritmi in grado di cifrare (trasformare) il testo in modo da renderlo incomprensibile

- la codifica è funzione di una **chiave (segreta)**.
- l' operazione di **decifratura** e' relativamente semplice nel caso in cui si conosca la chiave.

Se non si conosce la chiave

- risulta molto laborioso ottenere informazioni - anche limitate - sul messaggio
- dedurre la chiave con cui è stato cifrato un documento anche conoscendo il testo in chiaro

Sicurezza di un protocollo

La sicurezza in crittografia è valutata in base alle risorse di tempo e di calcolo necessarie per dedurre informazioni

- *ogni protocollo crittografico puo' essere "rotto" con sufficienti risorse di tempo e calcolo*
- *se un algoritmo puo' essere "rotto" usando per 30 anni un sistema di calcolo del valore di 10 miliardi di Euro allora puo' essere sicuro.*

La sicurezza di un algoritmo dipende dal campo di applicazione.

Sicurezza di un protocollo

La sicurezza di un protocollo dipende anche dal numero di possibili chiavi:

se ci sono molte possibili chiavi allora ci vuole molto tempo (o molta fortuna) per trovare la chiave segreta:

- *20 bit (circa 1 milione di diverse chiavi) allora non e' affatto sicuro*
- *56 bit (circa 66 milioni di miliardi diverse chiavi) andava bene dieci anni fa ma oggi e' "poco" sicuro*
- *512 bit (piu' di 40000000....0000000000 - 4 seguito da 153 zeri - diverse chiavi) oggi e' sicuro; domani?*

Sicurezza di un protocollo

Grandi Numeri

- Colonne Enalotto 622.614.630=1.15 2^{29}
- Secondi dalla creazione del sistema solare 1.38 2^{57}
- Cicli in un secolo di una macchina a 2 GHz 2.7 2^{61}
- Cicli in un secolo di 1000000 di macchine a 2 GHz 2.7 2^{81}
- Numeri primi di 249 bit 1.8 2^{244}
- Elettroni nell'universo 1.8 2^{258}

Sicurezza di un protocollo

La Crittoanalisi studia le modalità di attacco dei protocolli crittografici

Diversi tipi di attacco basati su

- ***Conoscenza*** *di testo cifrato*
- ***Conoscenza*** *testo in chiaro e corrispondente testo cifrato*
- ***Scelta*** *di testo in chiaro e conoscenza del corrispondente testo cifrato*
- ***Scelta*** *di testo crittato e conoscenza del corrispondente testo in chiaro*

Storia della crittografia

La crittografia è un scienza molto antica:

- Mesopotamia: Assiri e Babilonesi (scritture cuneiformi): usavano sostituire parti terminali delle parole
- Bibbia: Atbash (alfabeto rovesciato), cifratura di Babilonia nel libro di Geremia
- India, Kama Sutra: tra le 64 arti la 45-esima Mlecchita-vikalpa, che le donne devono conoscere
- Plutarco, Vite parallele: gli spartani usavano la Scytala (cilindro su cui si arrotolava la pergamena)

Storia della crittografia

Nel passato algoritmi a chiave segreta:

Codice di Cesare (a sostituzione di lettere)

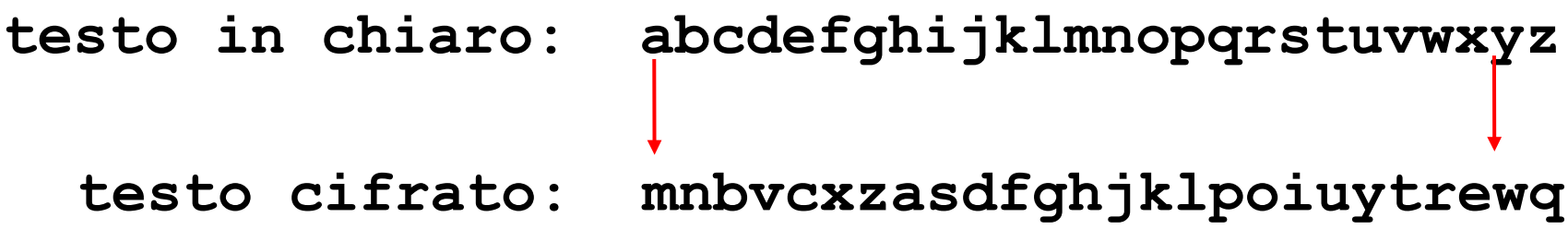
- Esempio con chiave 5
  ABCDEFGHILMNOPQRSTUVZ

  FGHILMNOPQRSTUVZABCDE

Storia della crittografia

Sostituzione di cifra

Cifratura monoalfabetica



Esempio: testo in ch: bob. i love you. alice

testo cif.: nkn. s gktc wky. mgsbc

Storia della crittografia

Sostituzione di cifra

Permutazione del testo:

- Scrivi il messaggio in blocchi di lunghezza fissa
- Riordina le lettere usando la chiave

Esempio: **Nel mezzo del cammin di nostra vita..**

Chiave: **domani**

264153	123456
nelmez	mnzlee
zodelc	ezcdlo
ammind	iadmnm

Storia della crittografia

Proprietà sulla frequenza dei caratteri e le vicinanze rende facile rompere i codici precedenti

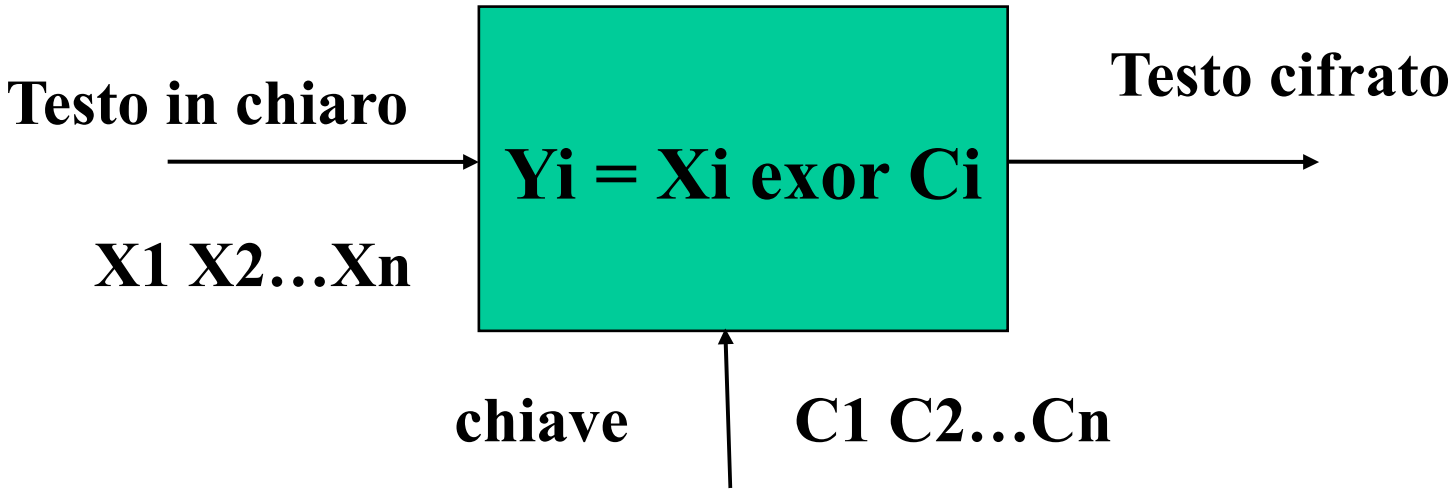
(es: la sequenza "zq" non esiste, "tt" è molto probabile, conoscenza parziale testo in chiaro)

Molte altre proposte:

- Codici a sostituzione di gruppi di lettere (Porta 1563, Vigenere 1586, ...): resistono meglio ad attacchi basati sulla analisi della frequenza delle lettere nel testo, Vigenere usa sostituzioni variabili.
- Codici basati su rotori: la macchina ENIGMA (usata dalle forze dell'asse durante la seconda guerra mondiale)

Storia della crittografia

- Un cifrario perfetto: One Time Pad
  - G.Vernam,impiegato ATT, 1917
  - La chiave è una sequenza casuale lunga come il testo



La decodifica è uguale alla codifica

Storia della crittografia

Cifrario perfetto: poichè C e X sono indipendenti
conoscere Y non fornisce informazioni su X

Prob(Yi=0)=

Prob(Xi=0| Ci=0) + Prob(Xi=1| Ci=1) =

Prob(Xi=0,Ci=0)/2 + Prob(Xi=1,Ci=1)/2 =

Prob(Xi=0)/4 + Prob(Xi=1)/4 = . . .

Prob(Yi=1)

Problemi con one time pad:

- la chiave deve essere lunga come il testo
- la chiave può essere usata solo una volta

Storia della crittografia

Principio di Kerckhoffs':

"La sicurezza di un sistema crittografico deve dipendere solo dalla segretezza della chiave e non dalla segretezza del metodo" Jean Guillaume Hubert Victor Francois Alexandre Auguste Kerckhoffs von Nieuwenhof (1835-1903), "*La Cryptographie militaire*", 1883.

Quanto è complesso "rompere" un codice?:

- forza bruta: se il numero di chiavi e' piccolo: il codice si "rompe" facilmente con un computer
- altri sistemi? (attacchi basati sulla conoscenza dell'algoritmo di cifratura)

Algoritmi di crittografia

- Gli algoritmi di crittografia possono essere classificati come
  - **simmetrici**, anche detti ***a chiave segreta*** *(o* ***privata***): usano la stessa chaive per codificare e decodificare
  - **asimmetrici**, anche detti ***a chiave pubblica:*** usano due chiavi distinte: una per codificare e una per decodificare.

- Tutti codificano il testo a blocchi (es. 64, 128 byte)

Data Encryption Standard: DES

DES codifica blocchi di 64 bit e usa chiavi di 56 bit; versioni successive (DES triplo) usano 2 o 3 chiavi da 56 bit (112,168 bit).

Codifica e decodifica con DES sono veloci; esistono implementazioni hardware

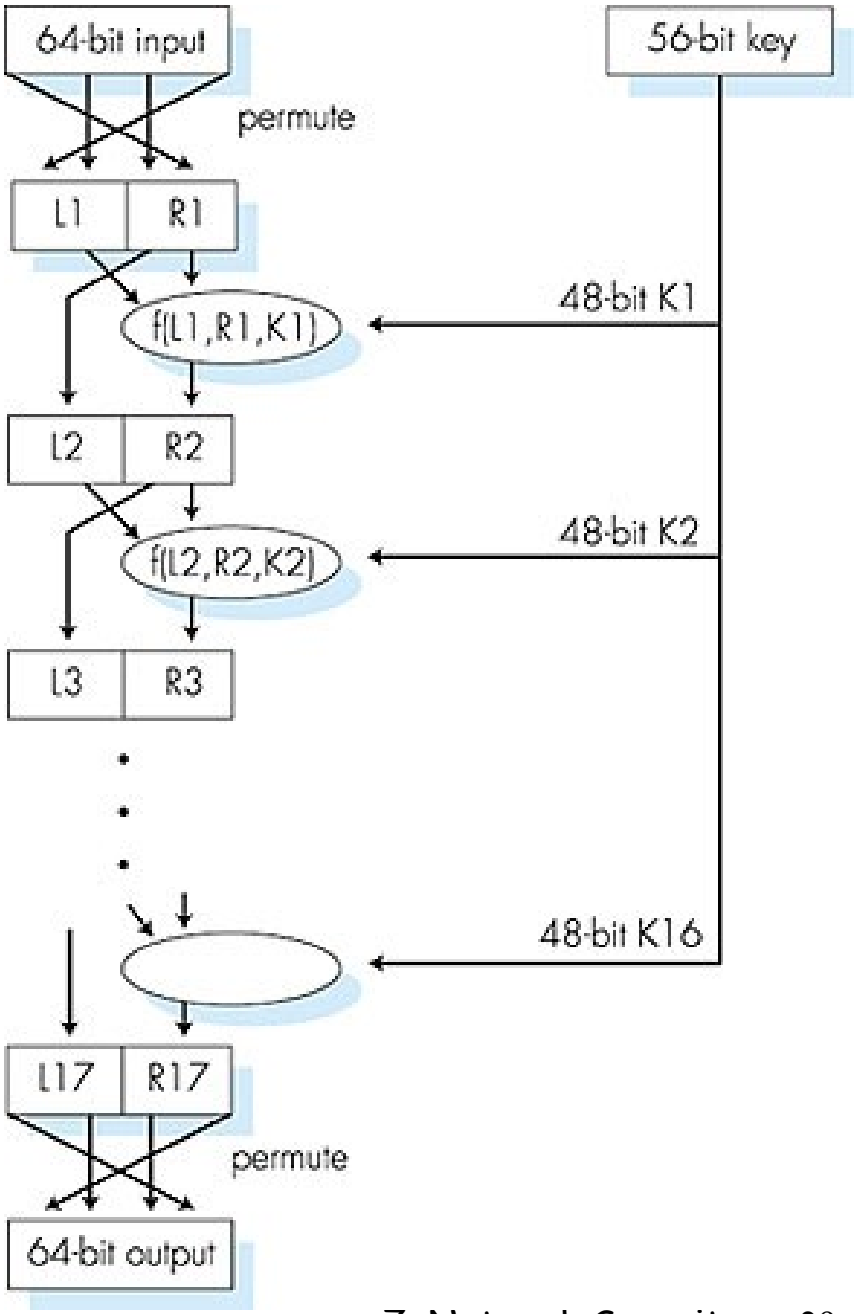
Storia

- Maggio 1973: richiesta pubblica per standard
- 1976: Modifica di Lucifer (IBM)
- 1977: viene pubblicato lo standard
- 2001: Advanced Encryption Standard (AES)

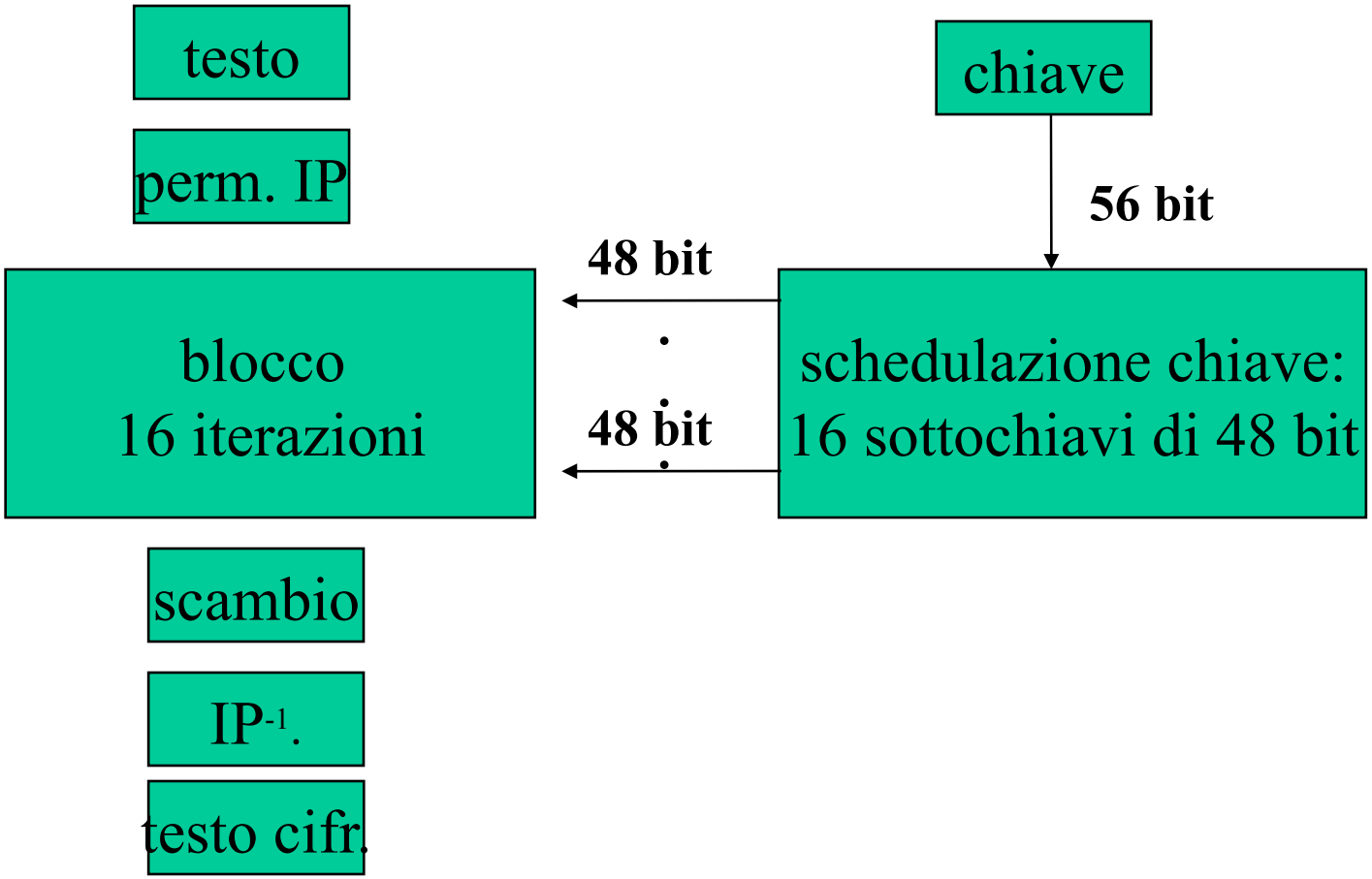
DES

Struttura DES

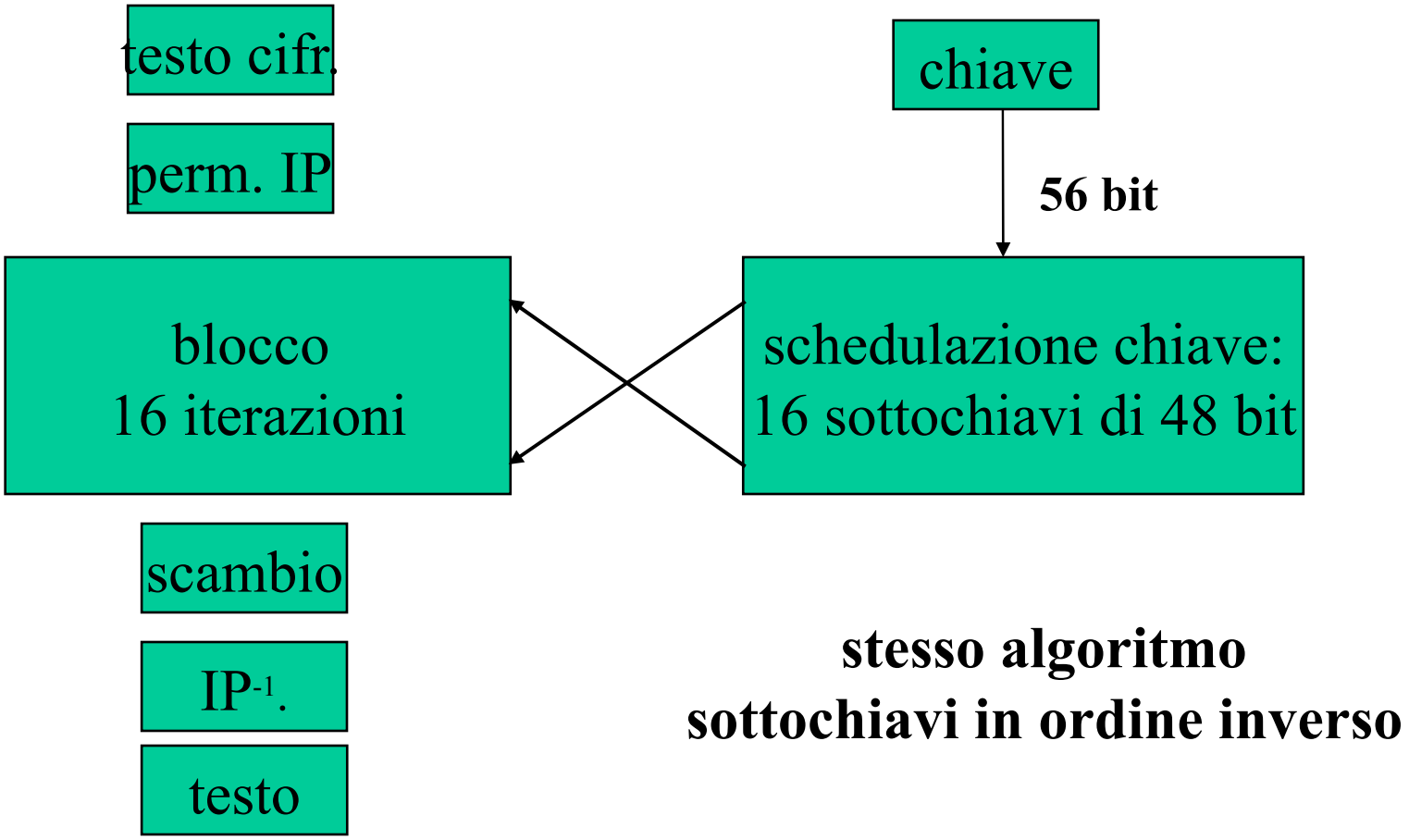
- Permutazione iniziale
- 16 "rounds" identici in ciascuno si applica una funzione basata su 48 bit della chiave (ogni volta diversi)
- Permutazione finale



Struttura DES



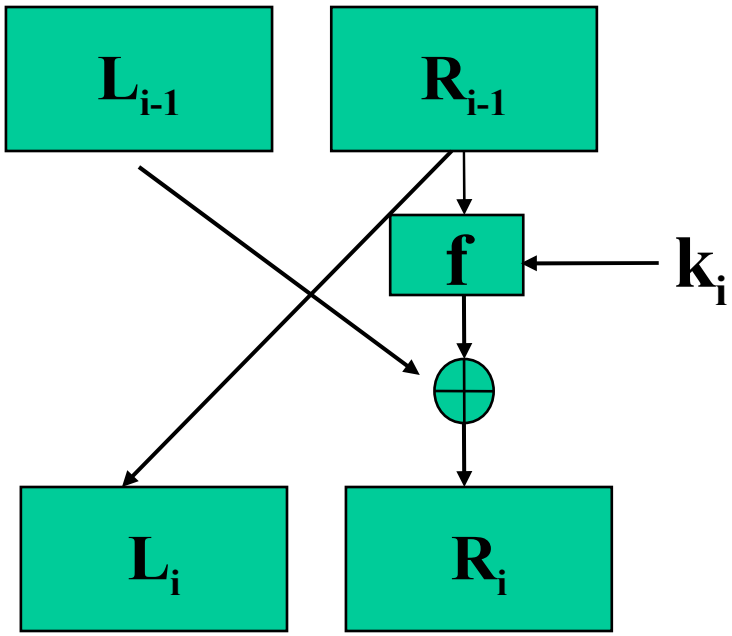
Struttura DES/ decodifica



Struttura DES

-

Singola iterazione

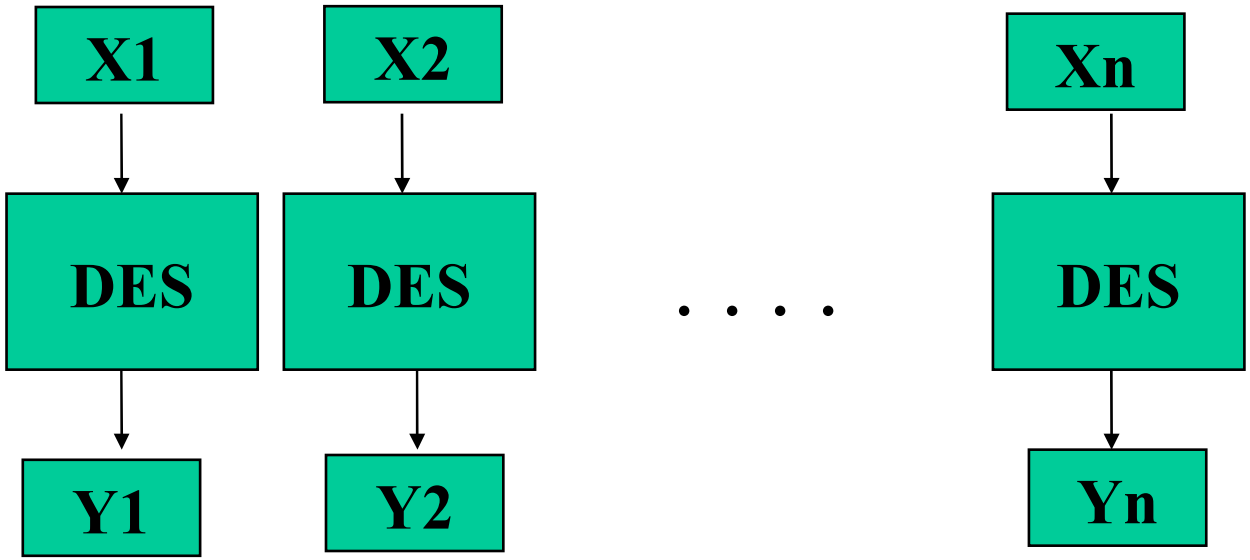


k_i è sottochiave i.
L_{i-1} e R_{i-1}(32 bit) sono parte sin. e destra di un blocco di 64 bit prima iterazione i.
L_i e R_i dopo iterazione i

Codifica testi

Electronic codebook chaining (ECB):

Dividi il messaggio in blocchi di 64 bit e codifica ciascun blocco con DES (padding se il testo non è multiplo di 64)

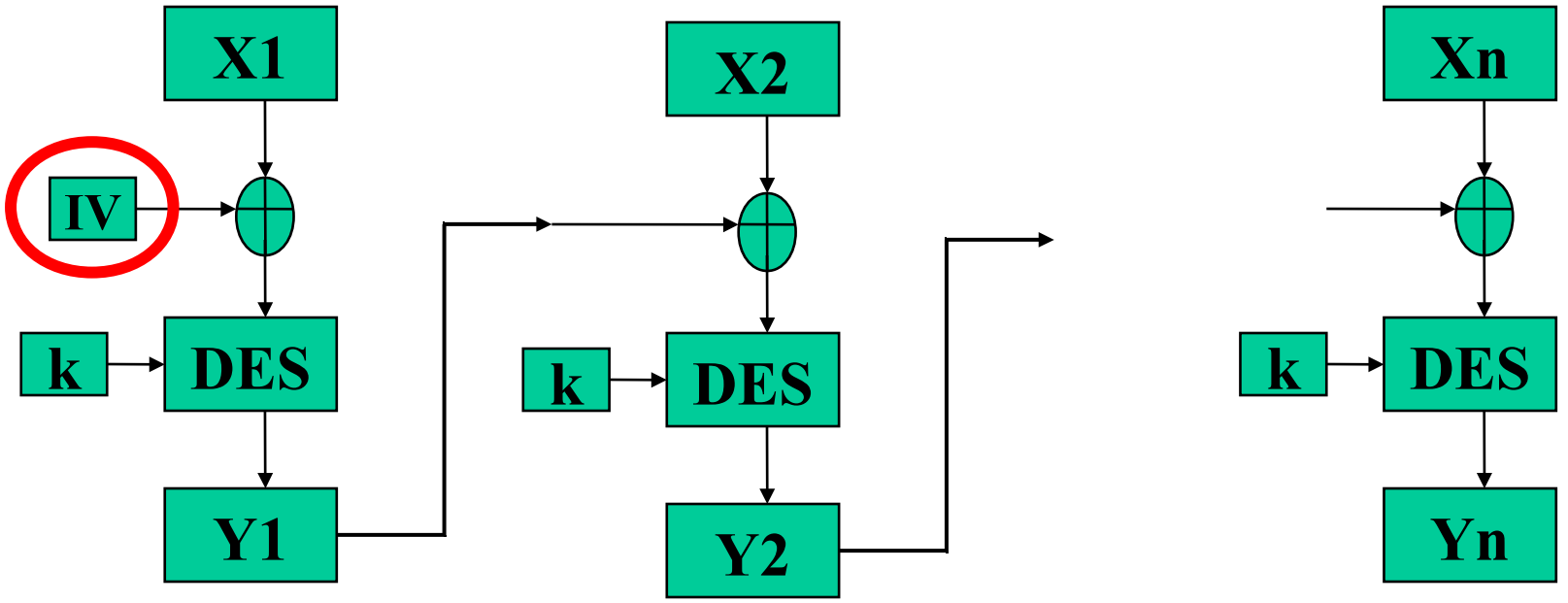


Codifica testi

Cipher block chaining (CBC):

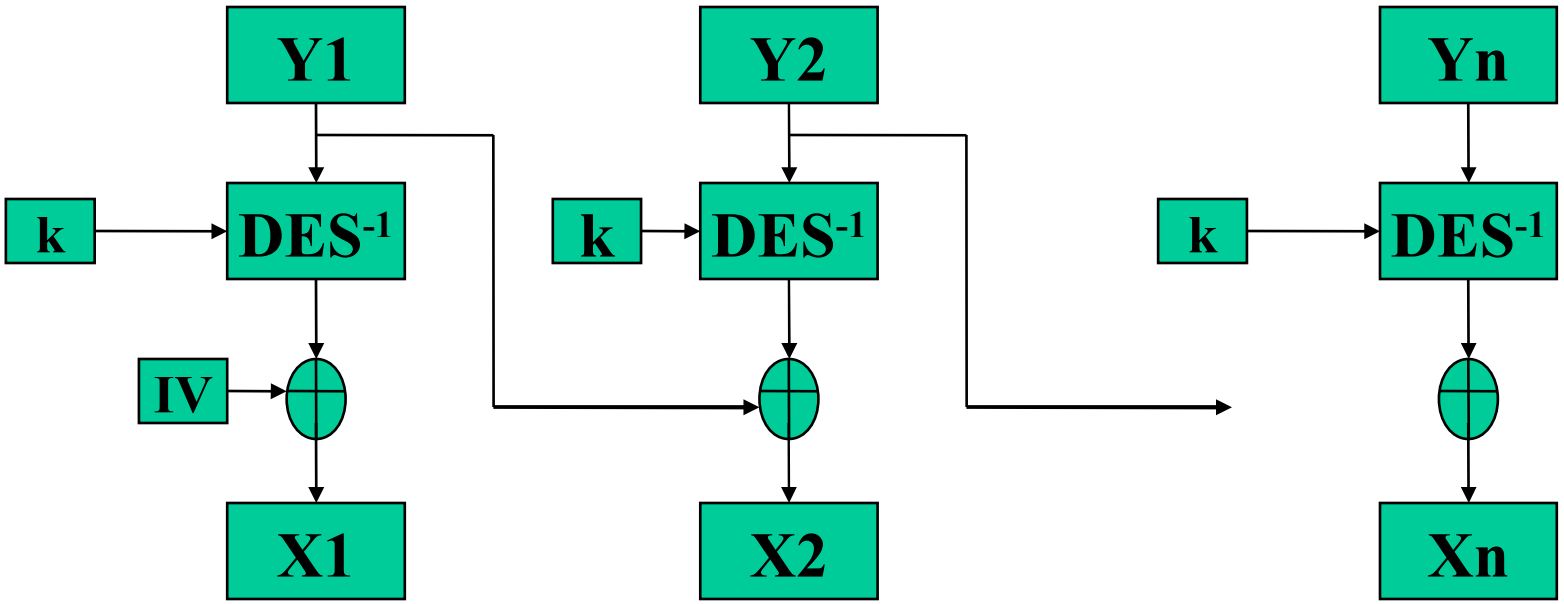
Dividi testo in blocchi X1,X2,…,Xn di 64 bit

IV è valore iniziale



Decodifica testi

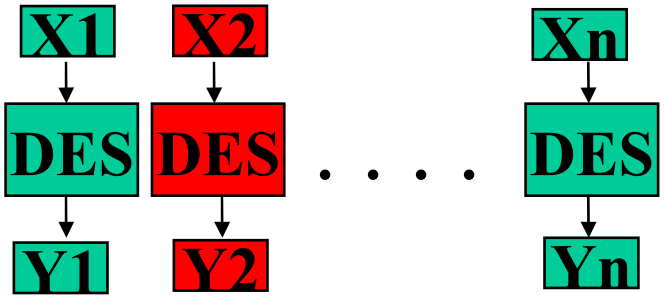
Cipher block chaining (CBC):



Codifica blocchi

Cipher Block Chaining vs. Electronic book Chaining

- CBC più lento di ECB
- in ECB non c'è dipendenza fra i blocchi e quindi sono possibili attacchi di sostituzione (stesso in,k → stesso out)
- in CBC ho dipendenza fra blocchi e quindi non sono possibili attachi per sostituzione ma ho anche propagazione di errori; in ECB no



Altri algoritmi a chiave segreta

Esistono molte altre proposte. Di solito:

- La codifica si basa su permutazione e sostituzione di parti del testo
- Le operazioni di permutazione e sostituzione sono ripetute diverse volte con modalità diverse che dipendono dalla chiave o da parti di essa
- La lunghezza della chiave è variabile

Advanced Encryption Standard (AES)

1997 : NIST (National Inst. of Standard and Tech., USA)
Concorso pubblico per AES : nuovo standard a blocchi per uso commerciale.

Perché nuovo standard?

- Chiave DES corta, problemi di sicurezza

Advanced Encryption Standard

Requisiti: Cifrario a blocchi con chiavi fra 128 e 256 bit

- Resistente ai tipi di attacco noti
- Semplice, veloce e con codice compatto (implementabile su smart card)
- Senza royalties

Selezione attraverso pubblico esame ed è basata su sicurezza, efficenza implementazione (velocità e memoria richiesta)

Advanced Encryption Standard

SELEZIONE

- Giugno 1998: 15 candidati: descrizione algoritmo con analisi efficienza e robustezza rispetto ad attacchi piu' comuni
- Agosto 1999: dopo pubblico scrutinio scelta 5 finalisti: Mars(IBM), RC6 (RSA Lab.), Rijndael (Daemen e Rjinen), Serpent (Anderson et al.), Twofish (Schneider et al.)
- Ottobre 2000: RIJNDAEL è stato scelto: pubblico esame e eventuale revisione
- 2001 : RIJNDAEL è proposto come standard AES

Advanced Encryption Standard

AES usa chiavi di 128, 192 o 256 bit

128 bit: 10 (round) iterazioni

- La chiave è espansa in 10 chiavi da 128 bit ciascuna
- Ogni round cambia lo stato e poi si esegue EXOR con la chiave
- Stato: 128 bit organizzati in una matrice 4 x 4 di byte

E' facile rompere AES con 1 o 2 round; ma non si sa come fare con 5 round; con 10 round è considerato impossibile!

Algoritmi asimmetrici

Problemi con crittografia a chiave simmetrica

- Richiede che mittente e ricevente abbiano la stessa chiave segreta
  - D.: come ci si accorda sulla chiave (specialmente tra computer)? Ci si incontra?
- Nella comunicazione fra n soggetti, si usano complessivamente $O(n^2)$ chiavi
  - Bisogna fare attenzione a non usare la chiave sbagliata

Esiste un sistema crittografico in cui non c'è bisogno di una chiave segreta?

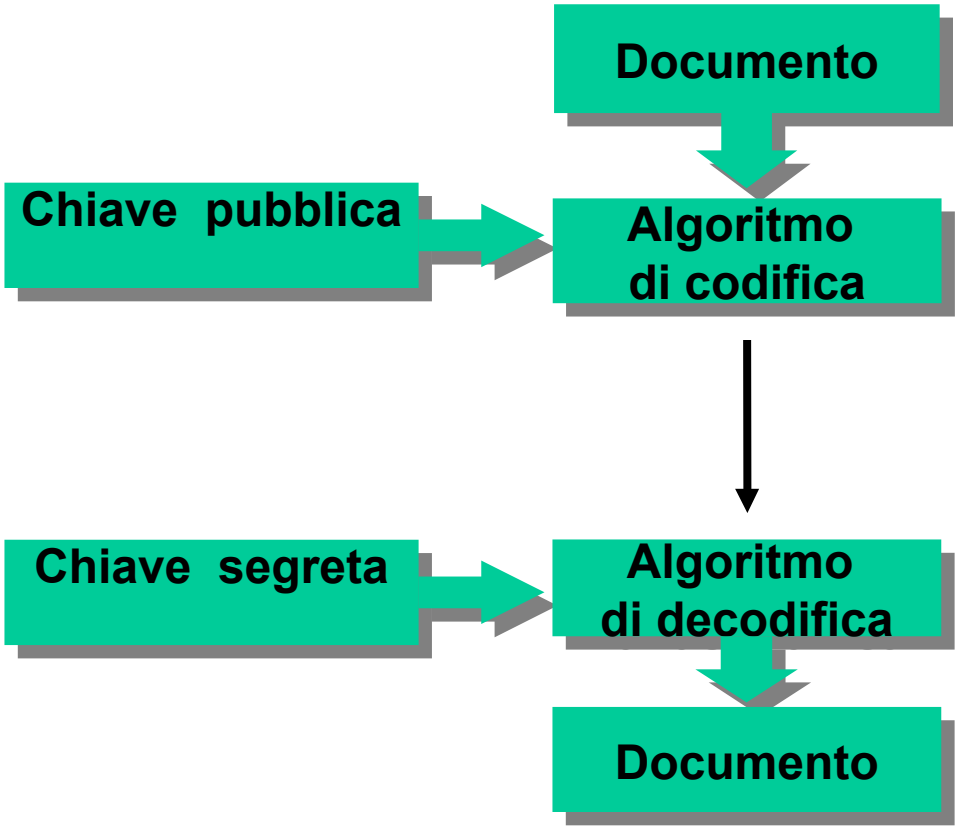
Algoritmi asimmetrici

Gli algoritmi **asimmetrici** utilizzano due chiavi distinte **generate insieme**:

- La ***chiave pubblica*** usata per codificare e puo' essere distribuita
- La ***chiave privata*** usata per decodificare (deve essere segreta)

- La stessa chiave pubblica è usata da tutti gli utenti
- Nella comunicazione fra n soggetti, un algoritmo asimmetrico usa 2n chiavi.

Algoritmi asimmetrici: codifica e decodifica



Conoscere la chiave pubblica non deve permettere di conoscere la chiave segreta

Algoritmi asimmetrici: storia

Idea originaria crittografia chiave pubblica circa 1950 (non pubblicata)

Diffie- Hellman proposero un metodo (non sicuro)

1978: Rivest Shamir e Adleman proposero il metodo **RSA**, che è ancora oggii il più usato inpratica.

RSA: scelta delle chiavi

1. Scegli due numeri **primi grandi** ***p, q.***
   (es., 1024 bit ciascuno)

2. Calcola $n = pq$, $z = (p-1)(q-1)$

3. Scegli e (con $e<n$) tale che MCD(e,z) =1; cioè e privo di fattori comuni con z. (es. $e=99$ e $z=100$ non sono primi ma non hanno fattori comuni)
   e = encryption
4. scegli d tale che $ed-1$ sia esattamente divisibile per z.
   (cioè: ed mod $z = 1$)
   d = decryption
5. La chiave *Pubblica* è (n,e), quella *Privata* è (n,d).

RSA: cifratura, decifrazione

Dati (n,e) e (n,d) calcolati come al punto precedente, supponiamo di voler cifrare un messaggio m < n

1. Per cifrare la stringa di bit corrispondente a m, calcola

$$c = m^e \bmod n$$

(resto di m^e diviso n)

2. Per decifrare c, calcola

$$m = c^d \bmod n$$

(resto di c^d diviso n)

Questa è vera !

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

RSA: Esempio

Bob sceglie *p=5, q=7.* Allora *n=35, z=24.*

e=5

d=29 (*ed-1* esattamente divisibile per z)

Cifra:

lettera	m	m^e	$c = m^e \bmod n$
l	12	1524832	17

Decifra:

c	c^d	$m = c^d \bmod n$	letter
17	481968572106750915091411825223072000	12	l

RSA: Perché funziona

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

Teorema: se p,q primi, $n = pq$, allora

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

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

$$m^e = x + cn$$

$$(x+cn)^d = \sum bin(d,k)\, x^{(d-k)} (cn)^k$$

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

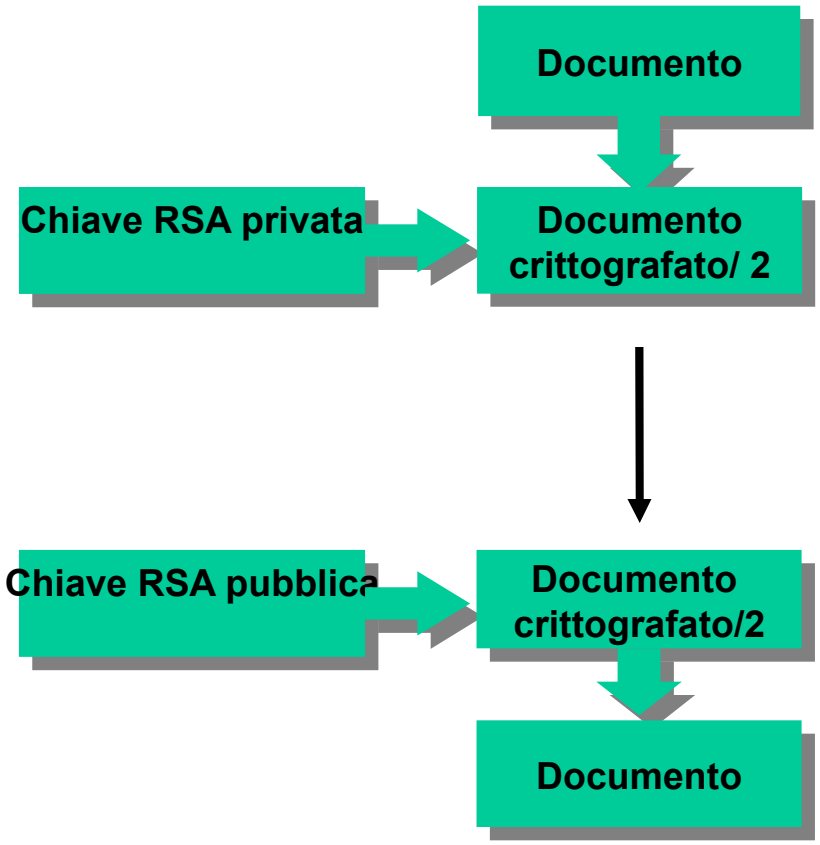
(perché (ed) è divisibile per $(p-1)(q-1)$ con resto 1)

$$= m^1 \bmod n$$

$(m<n)$

$$= m$$

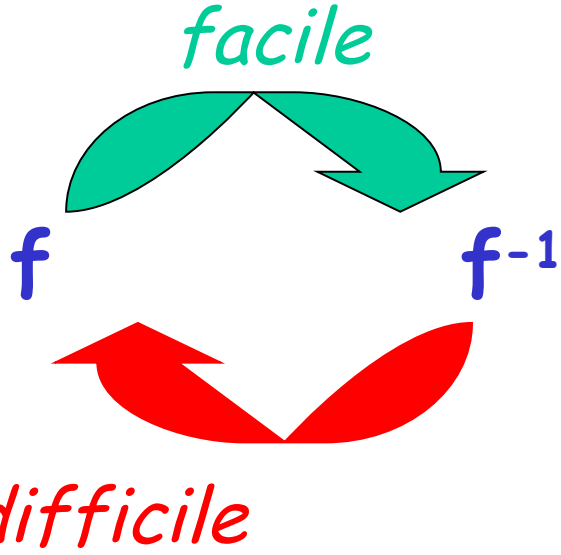
RSA: scambio delle chiavi



RSA: Sicurezza

Funzione one way:

- facile da calcolare
- difficile da invertire



Facile: Esiste Algoritmo veloce (tempo polin.)

Difficile: Non esiste (o si crede che non esista) alg. polinomiale

Recap

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

(n,e) chiave pubblica

(n,d) chiave privata

1. (n,e,) chiave pubblica →(n,e) NOTI
2. se da n fosse FACILE calcolare p e q , cioè FATTORIZZARE →
3. sarebbe facile calocalare d e dunque la chiave privata ...

RSA: Sicurezza (cont.)

- Se l'avversario sa fattorizzare $N = pq$ allora può conoscere la chiave segreta:
  Calcola $(p-1)(q-1)$ e poi $d = e^{-1} \bmod (p-1)(q-1)$

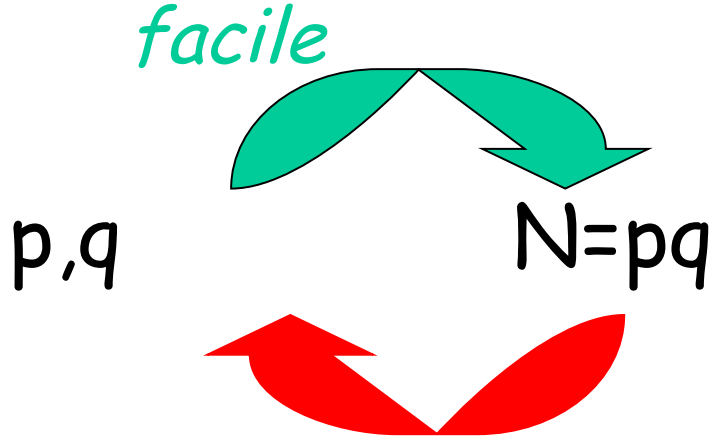
- Inoltre la definizione di RSA implica che se si conosce la chiave segreta allora è facile conoscere come si fattorizza N

Fattorizzare N e decodificare sono problemi computazionalmente equivalenti

Non sono noti attacchi migliori che fattorizzare N:
Es. Conoscere $(p-1)(q-1)$ equivale a conoscere la fattorizzazione di N

RSA: Sicurezza (cont.)

Moltiplicare e fattorizzare:



Funzione one way:

Facile da calcolare

Difficile da invertire

Moltiplicazione facile → codifica veloce

Fattorizzazione difficile → decodifica molto lunga (se non si conosce la chiave)

RSA: Sicurezza (cont.)

Test primalità : facile (p,q)

Fattorizzazione: difficile (N) (algoritmo semplice $O(N^{1/2})$; algoritmi migliori sono noti ma non sono polinomiali)

Nota bene: algoritmo polinomiale nella lunghezza dell'input. La rappresentazione di N richiede (log N) bit.

Es.: N da 1024 bit (155 cifre decim.)

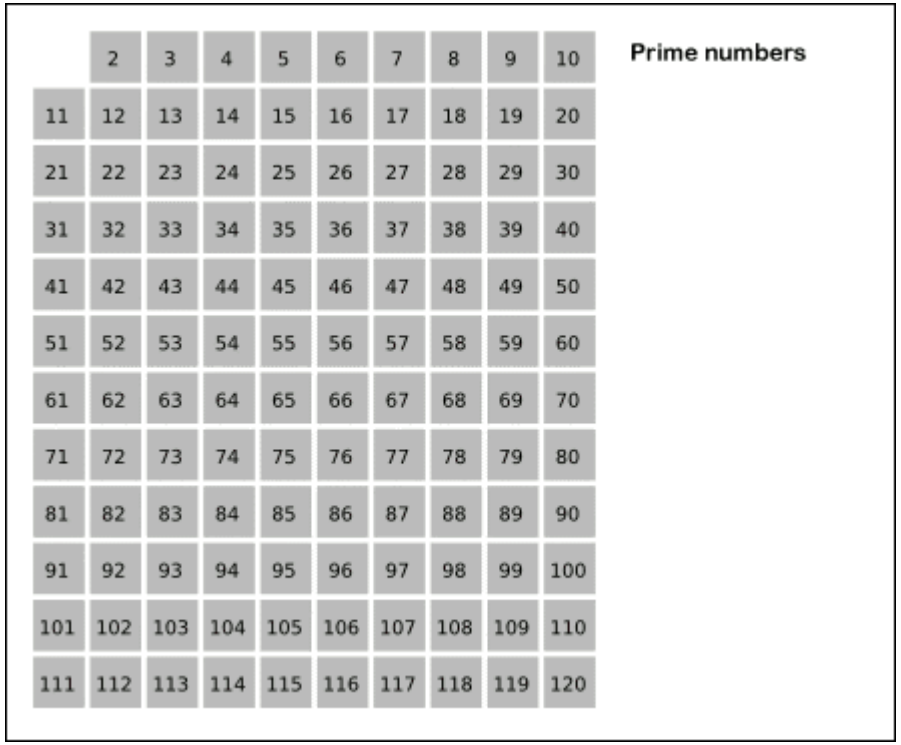
$O(N^{1/2}) = O(2^{512})$

non è log N

Metodo forza bruta: si divide n per tutti i numeri che gli sono minori - O(n). Metodo forza bruta migliorato: si considerano solo i numeri primi minori o uguali alla radice quadrata del numero

Test Primalità

- p è il più piccolo divisore primo di N quindi: N=pr con r>p
- N=pr>=pp=p^2 → p<=sqrt(N)



RSA: Sicurezza (cont.)

Esempio: Ago. 1999 RSA-155 è stato fattorizzato usando 300 elaboratori (tipo Pentium 2,ws Sun-400 MHz) in 7,4 mesi

- I numeri più difficili da fattorizzare sono quelli n= pq, dove p e q hanno la stessa lunghezza
- Per essere tranquilli
  - 768 bit (230 digit) uso personale
  - 1024 bit per aziende
  - 2048 per chiavi importanti

RSA: utilizzo in pratica

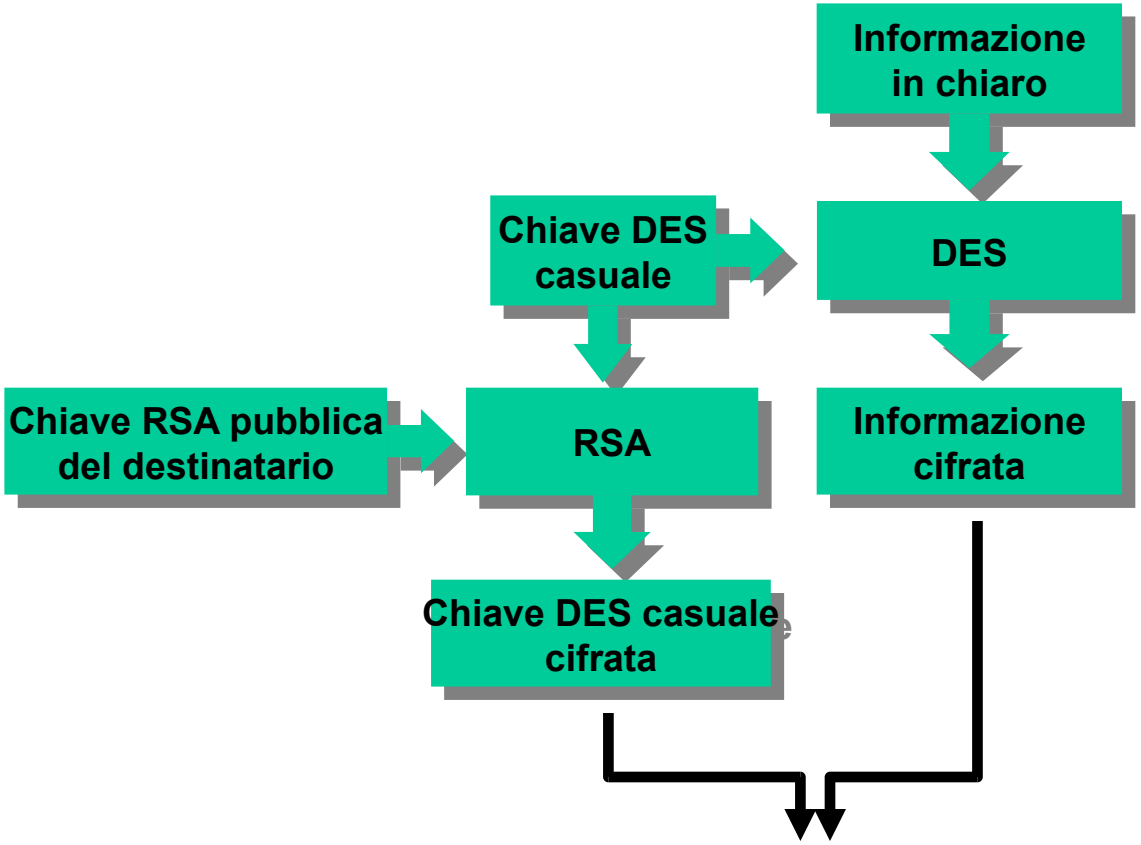
Per molte applicazioni RSA e' lento:

si può usare RSA con DES (o con altro metodo a chiave segreta)

Esempio: A invia un messaggio M a B

- A genera una chiave C, cifra C con RSA e M con DES (con chiave C)
- A invia il documento cifrato con DES e la chiave cifrata con RSA a B
- B usa la sua chiave privata RSA per conoscere la chiave C e poi usa DES per ottenere il messaggio.

RSA: utilizzo in pratica (cont.)



RSA: utilizzo in pratica (cont.)