

Corso di Laurea in Ingegneria Informatica



Corso di Reti di Calcolatori I

Roberto Canonico ([roberto.canonico@unina.it](mailto:roberto.canonico@unina.it))

Giorgio Ventre ([giorgio.ventre@unina.it](mailto:giorgio.ventre@unina.it))

Sicurezza nella comunicazione in rete:  
tecniche crittografiche

I lucidi presentati al corso sono uno strumento didattico  
che NON sostituisce i testi indicati nel programma del corso  
I lucidi sono adattati dagli originali di J. Kurose e K. Ross e fanno riferimento al testo  
*Reti di calcolatori e Internet - Un approccio top-down (4a ed.)*

Nota di copyright per le slide COMICS



## Nota di Copyright

Questo insieme di trasparenze è stato ideato e realizzato dai ricercatori del Gruppo di Ricerca COMICS del Dipartimento di Informatica e Sistemistica dell'Università di Napoli Federico II. Esse possono essere impiegate liberamente per fini didattici esclusivamente senza fini di lucro, a meno di un esplicito consenso scritto degli Autori. Nell'uso dovranno essere esplicitamente riportati la fonte e gli Autori. Gli Autori non sono responsabili per eventuali imprecisioni contenute in tali trasparenze né per eventuali problemi, danni o malfunzionamenti derivanti dal loro uso o applicazione.

Autori:

Simon Pietro Romano, Antonio Pescapè, Stefano Avallone,  
Marcello Esposito, Roberto Canonico, Giorgio Ventre

## Sicurezza della comunicazione in rete

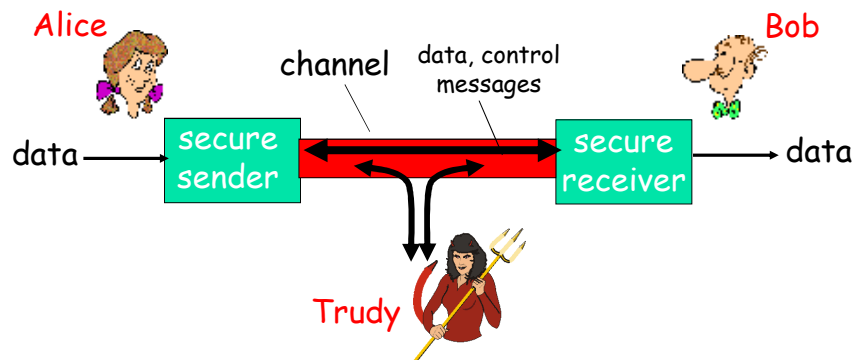


- La sicurezza della comunicazione in rete coinvolge diversi aspetti
- **Riservatezza:** solo il mittente ed il destinatario “legittimo” dovrebbero essere in grado di comprendere il contenuto del messaggio
  - Le tecniche crittografiche servono innanzitutto a proteggere la confidenzialità della comunicazione
  - Il mittente effettua un'operazione di cifratura del messaggio (*encryption*) ed il ricevente un'operazione duale di decifratura (*decryption*)
- **Integrità dei messaggi:** mittente e destinatario di un messaggio desiderano essere certi che i messaggi scambiati non siano alterati da una terza parte senza che se ne possano accorgere
- **Autenticazione:** mittente e destinatario di un messaggio desiderano essere reciprocamente sicuri dell'identità della controparte
- **Accessibilità e disponibilità dei servizi:** i servizi offerti in rete devono essere protetti da eventuali attacchi (es. attacchi *Denial of Service*, DoS)

## Amici e nemici: Alice, Bob, Trudy



- Alice e Bob sono le due parti che intendono comunicare in maniera sicura attraverso la rete
  - Potrebbero essere programmi client e server, o dispositivi (es. router) ...
- Trudy è una terza parte che può ascoltare i messaggi scambiati da Alice e Bob ed eventualmente alterarli, cancellarli o crearne di falsi



## Esempi di comportamenti malevoli



La terza parte Trudy può:

- intercettare i messaggi inviati da Alice a Bob (*eavesdropping*)
- inserire messaggi fasulli nel flusso della comunicazione da Alice a Bob
- inviare pacchetti con il campo source address fasullo (*spoofing*) in modo da fingere di essere Alice
- dirottare la comunicazione tra Alice e Bob, piazzandosi “in mezzo”, ad es. fingendo con Alice di essere Bob e con Bob di essere Alice (*hijacking*)
- impedire al servizio offerto da Bob di essere utilizzabile (es. sovraccaricando le risorse di Bob) (*denial of service*)

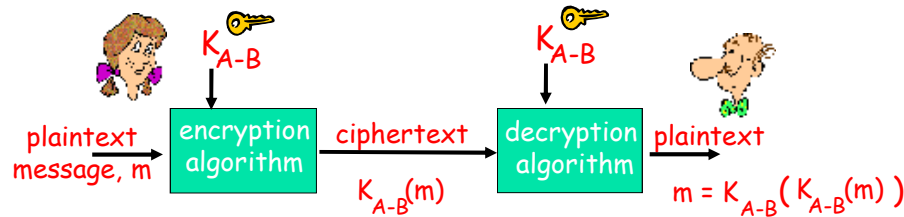
## Crittografia: concetti generali



- Un *sistema crittografico* è un sistema in grado di cifrare e decifrare un messaggio attraverso l'uso di un *algoritmo* e di una *chiave* (una stringa alfanumerica)
- Il messaggio da cifrare è detto “testo in chiaro” (*plaintext*) mentre il risultato dell'algoritmo crittografico è detto “testo cifrato” (*ciphertext*)



## Crittografia a chiave simmetrica



**Crittografia a chiave simmetrica:** Bob ed Alice entrambi conoscono la chiave crittografica (simmetrica)  $K_{A-B}$

- Come fanno Bob ed Alice a mettersi d'accordo sul valore della chiave? Lo scambio deve avvenire in maniera sicura (es. in un incontro di persona...)

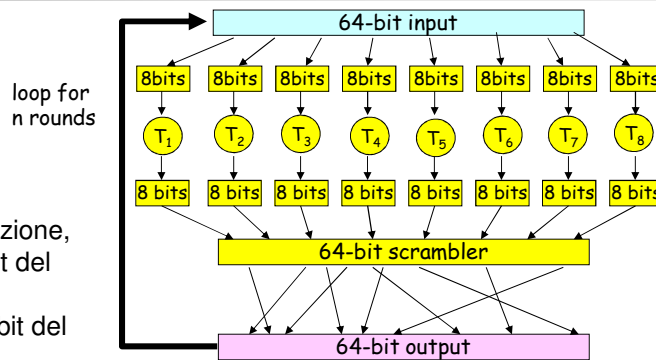
## Cifrari a blocchi



- Un blocco di  $k$  bit del testo in chiaro è codificato con altri  $k$  bit nel testo in codice secondo uno schema fisso
- Es.:  $k=3$   

000 → 110	001 → 111	010 → 101	011 → 100
100 → 011	101 → 010	110 → 000	111 → 001
- Il messaggio (010)(110)(001)(111) è codificato in (101)(000)(111)(001)
- Quante corrispondenze si possono creare con  $k=3$  bit ?  
N. permutazioni di  $2^k=8$  oggetti (triple di bit) =  $2^k! = 8! = 40320$
- Per una maggiore sicurezza occorre aumentare  $k$  (es.  $k=64$ )
- Per valori di  $k$  grandi, difficoltà di implementazione: cifratura e decifratura richiedono una tabella di  $2^k$  elementi in memoria
- Elevati valori di  $k$  ottenuti per composizione di valori più piccoli

## Cifrari a blocchi



- dopo una iterazione, ogni singolo bit del testo in chiaro influenza otto bit del testo cifrato
- Se la funzione di rimescolamento è fissa, la chiave è costituita dalle 8 tabelle di permutazione
- Esempi di cifrari a blocchi: DES, 3DES, AES

## Crittografia a chiave simmetrica: DES



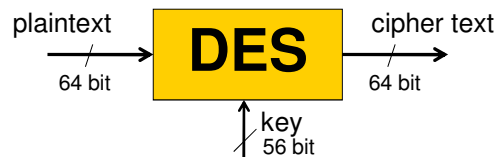
### DES: Data Encryption Standard

- US encryption standard [NIST 1993]
- 56-bit symmetric key, 64-bit plaintext input
- How secure is DES?
  - DES Challenge: 56-bit-key-encrypted phrase ("Strong cryptography makes the world a safer place") decrypted (brute force) in 4 months
  - no known "backdoor" decryption approach
- making DES more secure:
  - use three keys sequentially (3-DES) on each datum
  - use cipher-block chaining

## DES: caratteristiche generali



- DES codifica blocchi di 64 bit e usa una chiave di 56 bit
- Chiave da 56 bit + 8 bit di parità



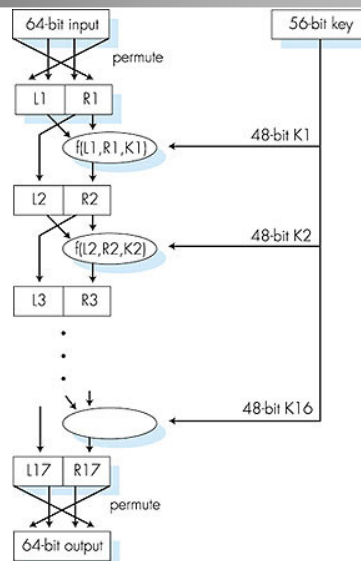
- La chiave è memorizzata su 64 bit, con l'ottavo, il 16-esimo, ... , il 64-esimo bit calcolati come bit di parità per i 7 bit precedenti

## Symmetric key crypto: DES



### Funzionamento del DES

- Permutazione iniziale
- 16 iterazioni in cui si applica una funzione  $f$ , usando in ciascuna iterazione 48 bit della chiave
- permutazione finale



## AES: Advanced Encryption Standard

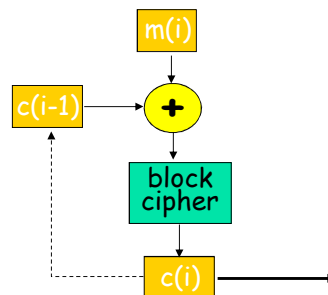
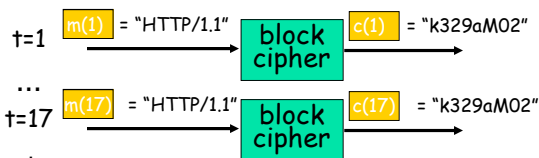


- Standard NIST standard, ha sostituito DES
- Elabora i dati a blocchi da 128 bit
- Chiavi da 128, 192, o 256 bit
- Un elaboratore in grado di decifrare DES con un attacco a forza bruta in 1 secondo, impiegherebbe 149 trilioni di anni a decifrare AES

## Cipher Block Chaining



- Debolezza dei cifrari a blocchi: blocchi in input uguali producono lo stesso testo cifrato
  - Si prestano a crittoanalisi
- Per ovviare a questo inconveniente si usa la tecnica detta **cipher block chaining**:  
 $c(i)$  si calcola con l'algoritmo di cifratura applicato al blocco ottenuto tramite XOR del testo in chiaro  $m(i)$  con il blocco di testo cifrato  $c(i-1)$  calcolato sull'input precedente
  - $c(0)$  trasmesso in chiaro (vettore di inizializzazione)



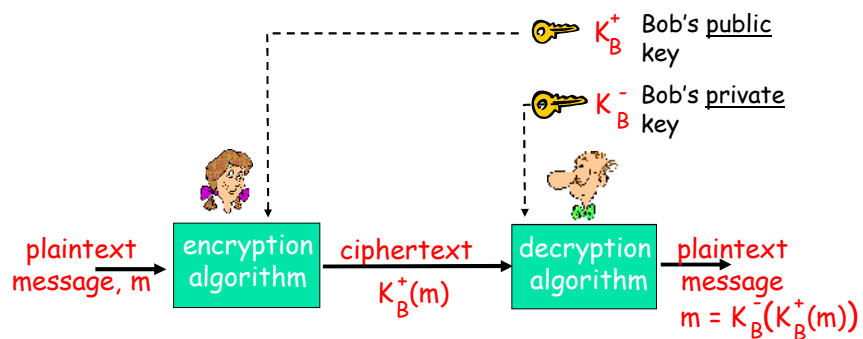


## Crittografia a chiave pubblica



- Approccio radicalmente differente
  - Diffie-Hellman 1976, RSA78
- Mittente e destinatario non condividono una chiave segreta
- *Chiave pubblica* usata per la cifratura, nota a tutti
- *Chiave privata* usata per la decifratura, nota solo al destinatario

## Public key cryptography



## Algoritmi di cifratura a chiave pubblica



Requisiti:

- 1 Occorre trovare una coppia di chiavi  $K_B^+(\cdot)$  and  $K_B^-(\cdot)$  tali che

$$K_B^-(K_B^+(m)) = m$$

- 2 Nota la chiave pubblica  $K_B^+$ , dovrebbe essere impossibile calcolare la chiave privata  $K_B^-$

*Algoritmo RSA: Rivest, Shamir, Adleman*

## RSA: scelta delle chiavi



1. Si scelgono due numeri primi grandi  $p$  e  $q$
2. Si calcolano  $n = p \cdot q$ ,  $z = (p-1)(q-1)$
3. Si sceglie un numero  $e$  ( $e < n$ ) che non abbia fattori comuni con  $z$  ( $e, z$  "relativamente primi")
4. Si sceglie un numero  $d$  tale che  $e \cdot d - 1$  sia esattamente divisibile per  $z$  (in altre parole:  $e \cdot d \bmod z = 1$ )
5. La chiave pubblica è  $(n, e)$ , la chiave privata è  $(n, d)$   
 $\underbrace{(n, e)}_{K_B^+}$        $\underbrace{(n, d)}_{K_B^-}$

## RSA: cifratura e decifratura



0. Dati  $(n,e)$  ed  $(n,d)$  come calcolati precedentemente ...

1. La cifratura del testo in chiaro  $m$  (*stringa di bit*)  
si effettua calcolando

$$c = m^e \bmod n \text{ (resto della divisione di } m^e \text{ per } n)$$

2. La decifratura del testo cifrato  $c$  (*stringa di bit*)  
si effettua calcolando

$$m = c^d \bmod n \text{ (resto della divisione di } c^d \text{ per } n)$$

$$m = \underbrace{(m^e \bmod n)}_c \bmod n$$

## RSA: esempio



Bob sceglie  $p=5, q=7 \rightarrow n=35, z=24$

$e=5$  (in modo che  $e, z$  siano relativamente primi)

$d=29$  (in modo che  $e \cdot d - 1$  sia divisibile per  $z$ )

Bob rende pubblica la coppia  $(n,e)=(35,5)$

e mantiene segreto il valore  $d=29$

	<u>lettera</u>	<u>m</u>	<u><math>m^e</math></u>	<u><math>c = m^e \bmod n</math></u>
cifratura:	I	12	248832	17
	<u>c</u>	<u><math>c^d</math></u>	<u><math>m = c^d \bmod n</math></u>	<u>lettera</u>
decifratura:	17	481968572106750915091411825223071697	12	I

## RSA: perchè $m = (m^e \bmod n)^d \bmod n$ ?



Un utile risultato di teoria dei numeri  
Se  $p$  e  $q$  sono primi ed  $n = p \cdot q$ , allora:

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

$$\begin{aligned} (m^e \bmod n)^d \bmod n &= m^{e \cdot d} \bmod n \\ &= m^{e \cdot d \bmod (p-1)(q-1)} \bmod n \\ &\quad \text{(grazie al risultato di sopra)} \\ &= m^1 \bmod n \end{aligned}$$

(dal momento che  $e \cdot d$  è stato scelto in modo che  
la sua divisione per  $(p-1)(q-1)$  dia resto 1)

$$= m$$

## RSA: un'altra proprietà notevole



$$K_B^-(K_B^+(m)) = m = K_B^+(K_B^-(m))$$

Si usa prima la  
chiave pubblica  
e poi la privata

Si usa prima la  
chiave privata e  
poi la pubblica

*Il risultato è lo stesso →  
le due chiavi possono  
intertirsi i ruoli*

## RSA: discussione



- La robustezza di RSA dipende dal fatto che non sono noti algoritmi veloci per la fattorizzazione di numeri interi grandi
- Anche se  $n = p \cdot q$  è noto, i due fattori  $p$  e  $q$  non sono velocemente determinabili, di conseguenza non è velocemente determinabile  $z$ , da cui si potrebbe facilmente risalire alla componente  $d$  della chiave privata, nota la parte  $e$  della chiave pubblica
- L'elevamento a potenza richiesto da RSA è computazionalmente oneroso
  - Algoritmi a chiave simmetrica come DES sono 100-1000 volte più veloci di RSA
  - Nelle comunicazioni su rete, RSA è spesso usato in combinazione con DES o 3DES