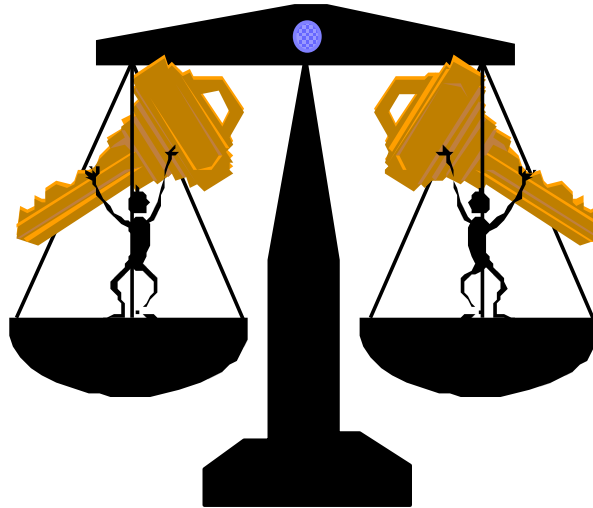


# Meccanismi simmetrici



## Definizioni di sicurezza per un Cifrario

### SICUREZZA COMPUTAZIONALE

Un Cifrario è detto **computazionalmente sicuro** se il calcolare  $M$  da un  $C$  è possibile, ma richiede una potenza di elaborazione superiore a quella a disposizione dell'attaccante.

**PRINCIPIO DI KERCKOFF** la sicurezza deve dipendere dalla chiave e non dall'algoritmo, perchè quest'ultimo non può essere mantenuto segreto

**PRINCIPIO DI SHANNON** detto anche principio di CONFUSIONE e DIFFUSIONE

CONFUSIONE il messaggio criptato non deve fornire informazioni sulla chiave

DIFFUSIONE la modifica di una solo carattere del messaggio in chiaro deve provocare una modifica sostanziale del messaggio criptato

## Confusione & Diffusione (C. Shannon)

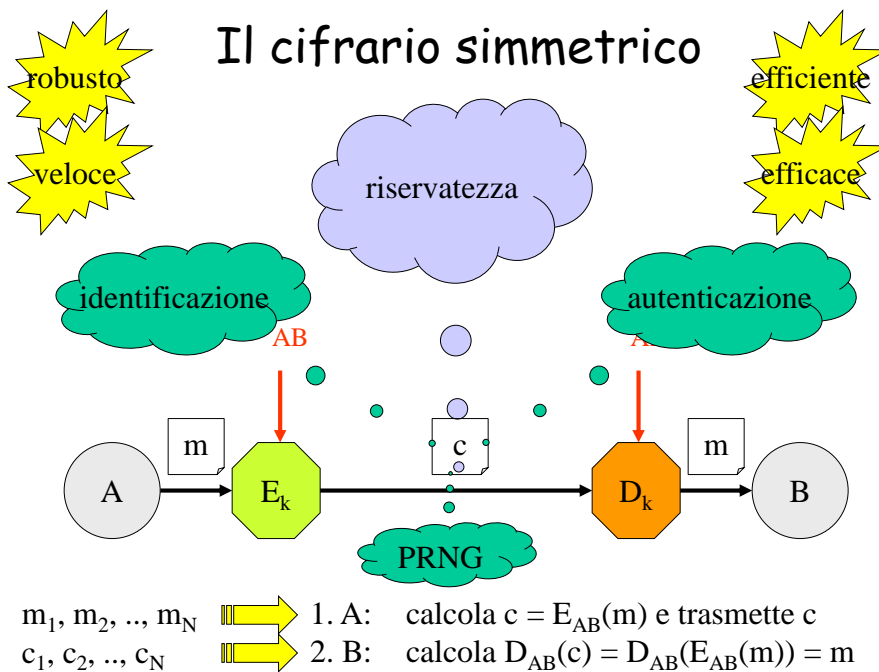
La **confusione** nasconde la relazione esistente tra testo in chiaro e testo criptato e rende poco efficace lo studio del secondo basato su statistiche e ridondanze del primo. Rende difficile prevedere che cosa accadrà al criptato anche modificando un solo simbolo del testo in chiaro

La **sostituzione** è il mezzo più semplice ed efficace per creare confusione.

La **diffusione** nasconde la ridondanza del testo in chiaro spargendola all'interno del testo criptato. Si impone ad ogni simbolo del testo in chiaro di influire su molti se non tutti i simboli del testo criptato. Difficile prevedere quali e quanti si modificano se si modifica anche un solo simbolo del testo in chiaro

La **trasposizione** è il mezzo più semplice ed efficace per ottenere diffusione





## Cifrari a flusso ed a blocchi

One time pad

**+** veloce

**Cifrario a flusso** (stream cipher): trasforma, **con una regola variabile al progredire del testo**, uno o pochi bit alla volta del testo da cifrare e da decifrare.

Protezione dei singoli bit di una trasmissione seriale

WEP, GSM

Cifrario poligrafico  
Cifrario composto

**+** sicuro

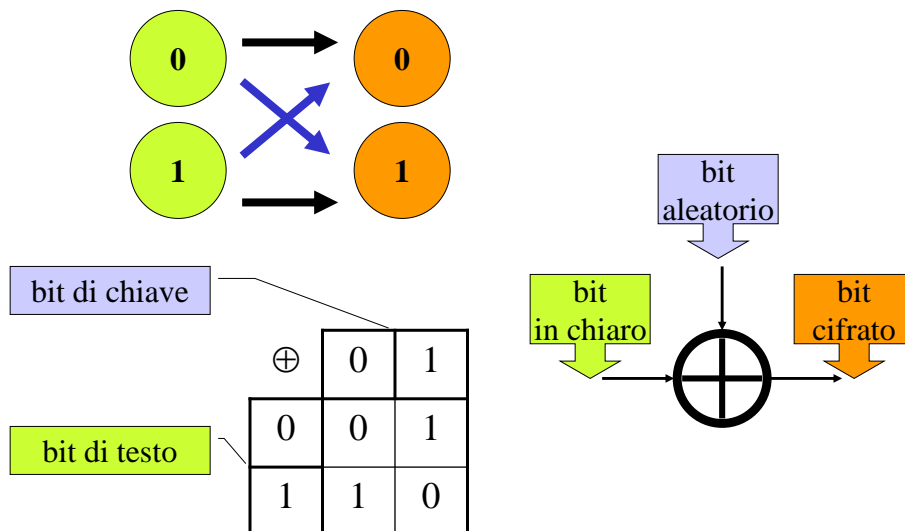
**Cifrario a blocchi** (block cipher): trasforma, **con una regola fissa** ed uno alla volta, blocchi di messaggio formati da molti bit.

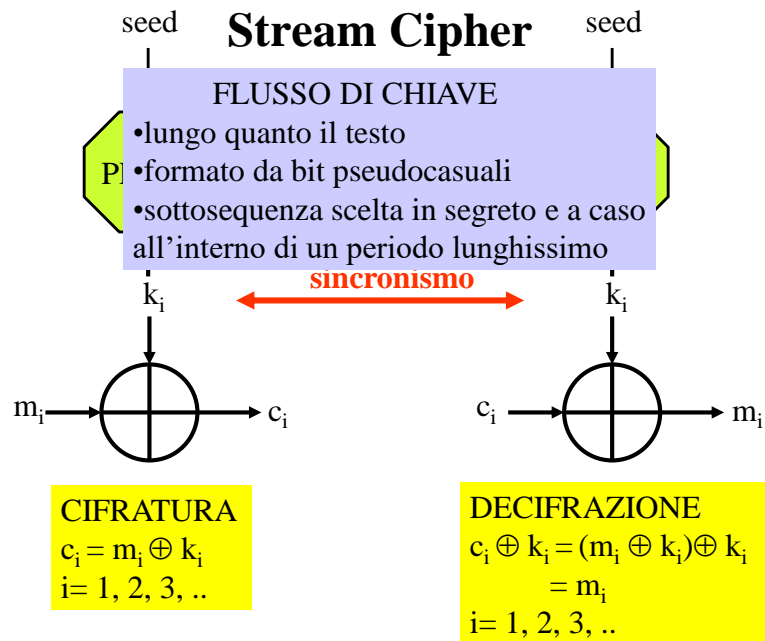
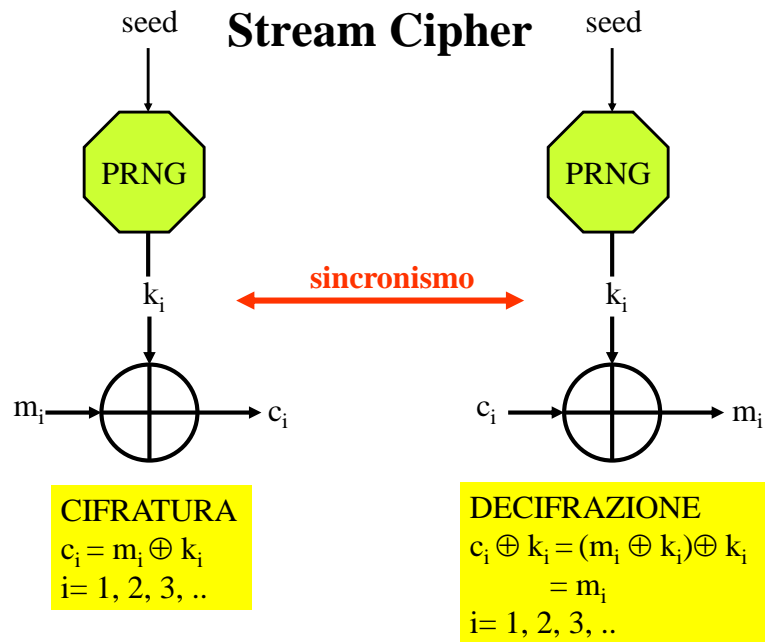
Protezione di pacchetti, di file e di strutture di dati

IPSec, SFS

# Cifrari a flusso

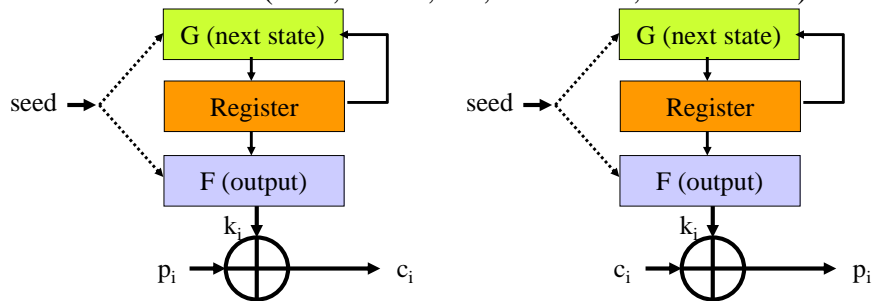
## Il meccanismo per la sostituzione di un bit



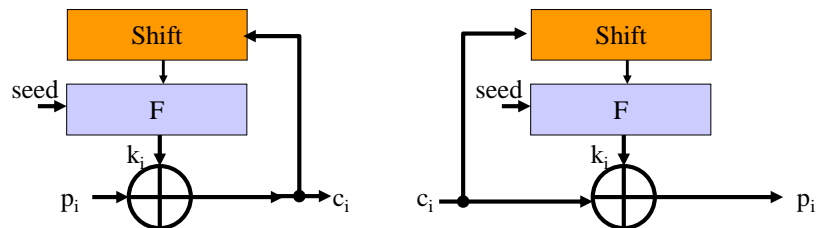


## Stream ciphers

- A flusso sincrono (RC4, SEAL, A5, DES-CTR, DES-OFB..)



- Con auto-sincronizzazione (DES-CFB, ..)




## Problemi dei Cifrari a flusso

ATTACCHI	FLUSSO SINCRONO	AUTOSINCR.
Cancellazione di bit	perdita di sincronismo	transitorio
Inserzione di bit	perdita di sincronismo	transitorio
Modifica di bit	non propagazione	transitorio



↑  
più usati

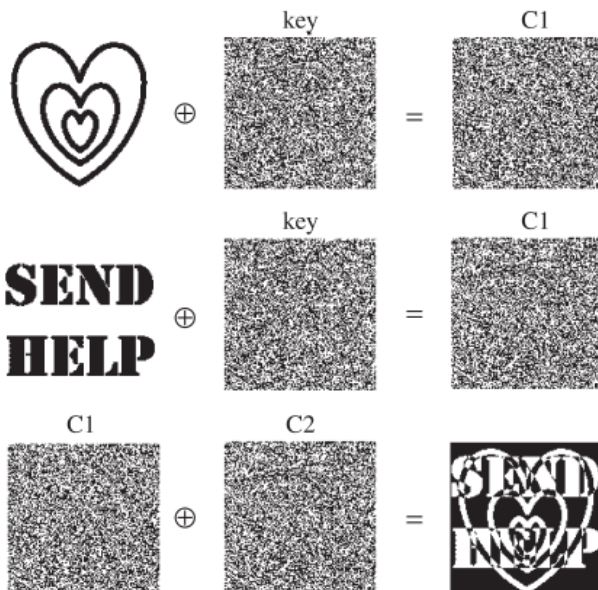
Diffusione (ogni bit di testo in chiaro influisce su molti bit di testo cifrato (ridotta la probabilità di successo di attacchi basati su ridondanza)

# Two-time keys

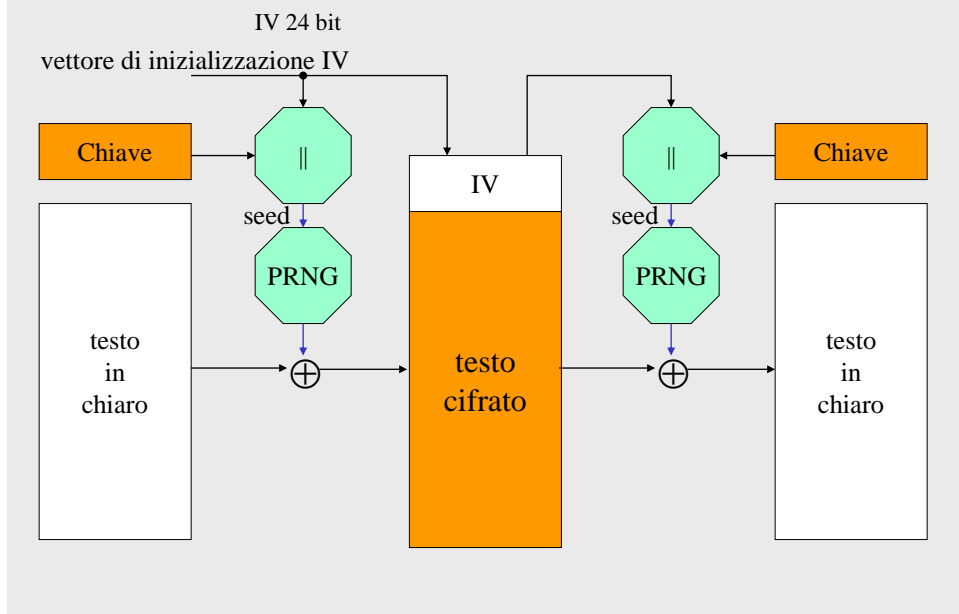
m1  k= c1

m2  k=c2

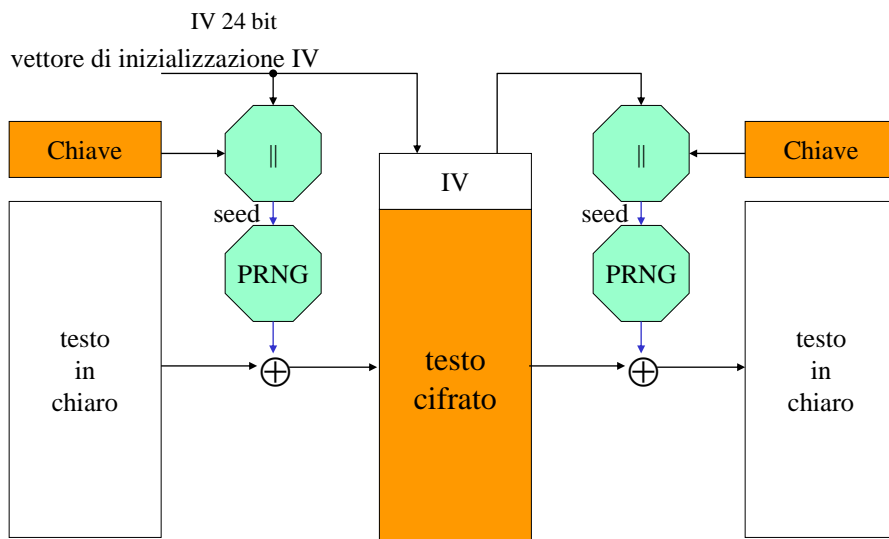
c1  c2 = m1  m2



## Segretezza e variabilità del seme (WEP)



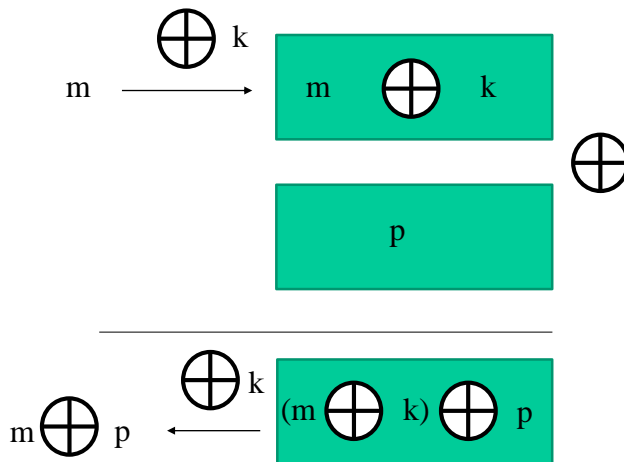
## Segretezza e variabilità del seme (WEP)



**Prominent example of two-time keys: WEP**

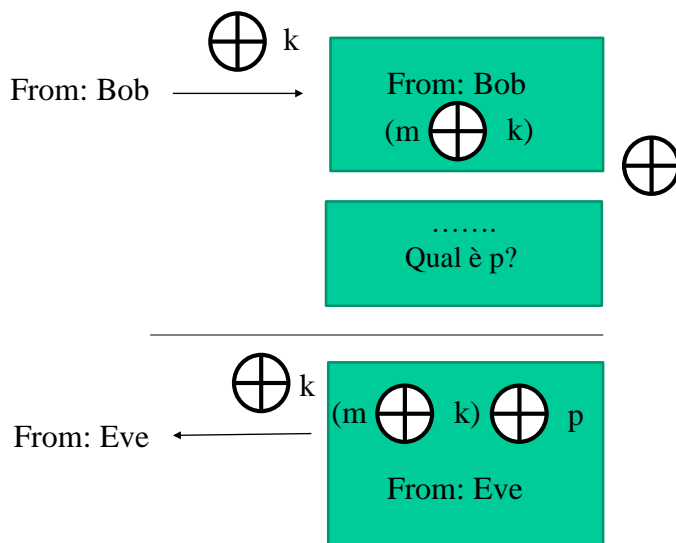


## Malleabilità

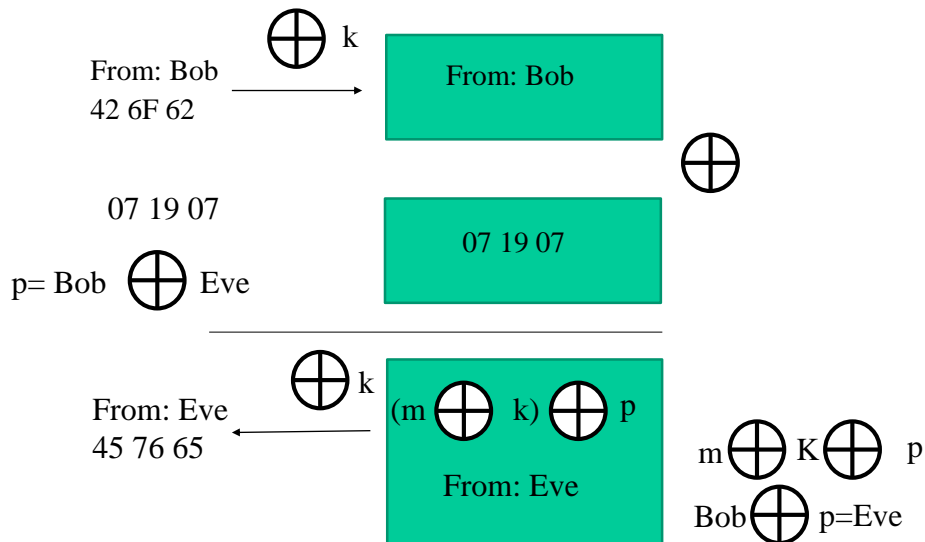


Se l'attaccante conosce  $M$ , può scegliere un opportuno  $p$  tale che, in fase di decifratura, il testo in chiaro originario  $M$  sia sostituito da un testo arbitrario  $M'$ , scelto dall'attaccante

## Malleabilità



## Malleabilità



## Quali Stream Ciphers?

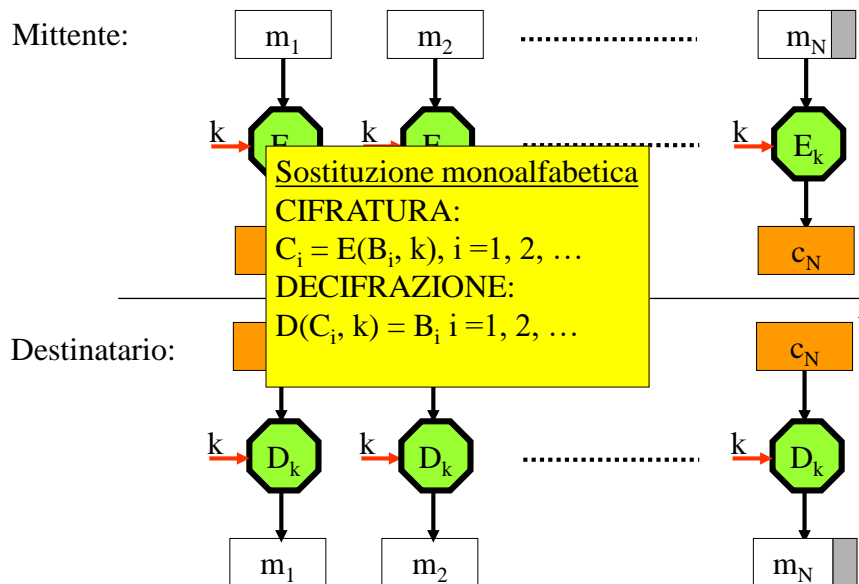
No RC4.....ormai è stato disabilitato nei browser

Nuovi Stream ciphers:

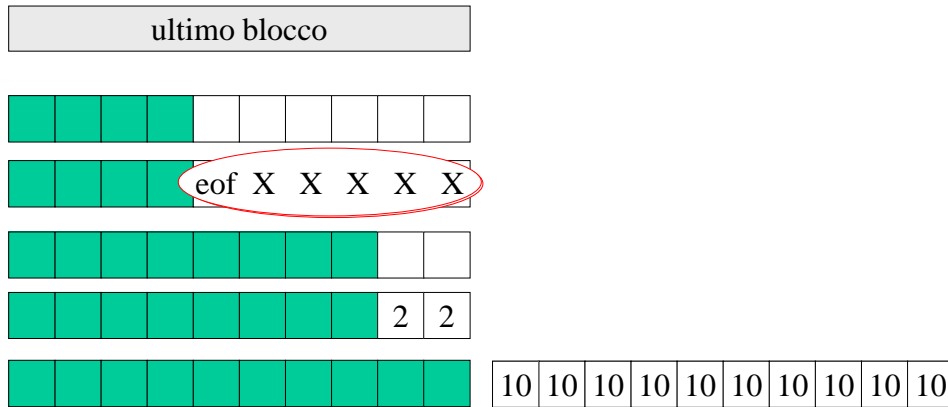
- Salsa
- Sosemanuk

# Cifrari a blocchi

## Block cipher (modalità ECB)



## Padding: standard PKCS#5 e #7



## Time to break a code

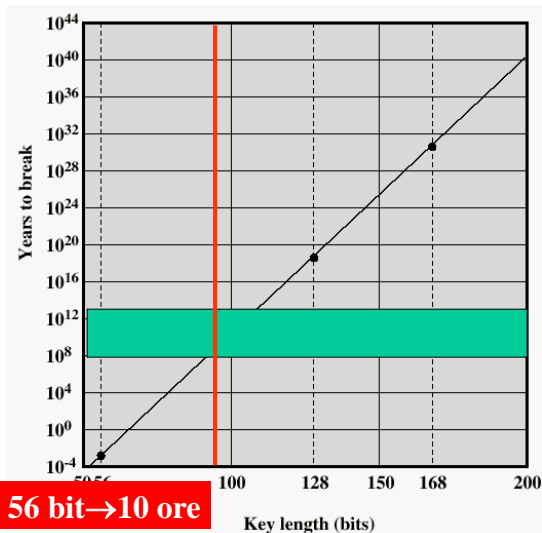
**Spazio delle chiavi    Forza bruta:  $T = 2^{N-1}/10^{12} \text{ s}$**

N bit	2 <sup>N</sup> chiavi
32	2 <sup>32</sup> = 4,3 × 10 <sup>9</sup>
56	2 <sup>56</sup> = 7,2 × 10 <sup>16</sup>
128	2 <sup>128</sup> = 3,4 × 10 <sup>38</sup>
168	2 <sup>168</sup> = 3,7 × 10 <sup>50</sup>
192	2 <sup>192</sup> = 6,3 × 10 <sup>57</sup>

$$p = 2^{-N}$$

## Valutazione sicurezza a breve termine (1996)

R28: “75 bit  
( $6 \times 10^{11}$  anni MIPS)  
+14 bit ogni vent’anni”



## Dimensioni della chiave e del blocco

**DES Cracker** (1998): macchina parallela costata 250.000 \$ ha individuato in meno di 3 giorni una chiave di 56 bit.

Con una chiave di 168 bit impiegherebbe  $10^{31}$  anni!

**FBI, CIA:** esportazione solo di crittografia "debole" (40 bit)

**Attacchi con testo noto e scelto:** **dimensione del blocco**

**DES** (56 bit di chiave e 64 bit di blocco): anni '80 e '90;

**TDES** (112 o 168 bit di chiave e 64 bit di blocco): anni '90;

**AES** (da 128 a 256 bit di chiave con blocchi da 128 a 256 bit):

Rijndael, prossimi 30 anni

"la chiave segreta deve essere scelta caso (R12) e frequentemente modificata (R24)".

## La rete di Feistel

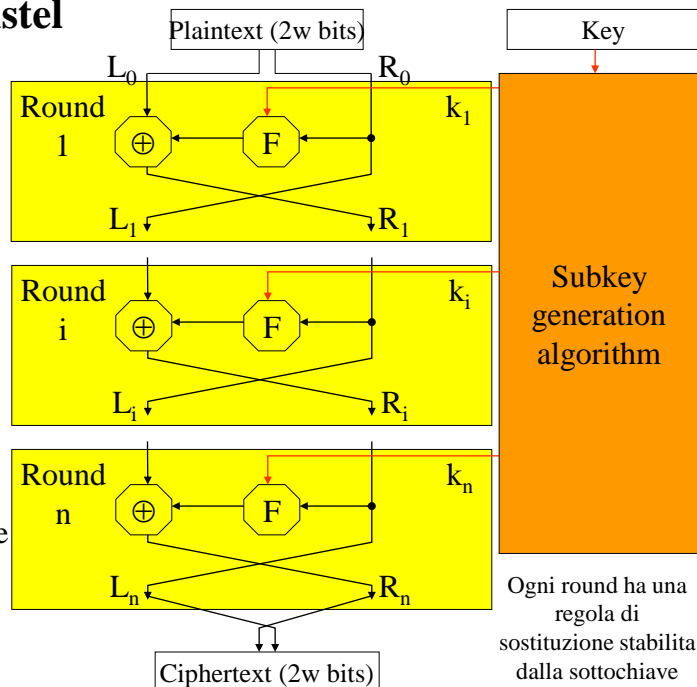
Obiettivo: ogni blocco di testo in chiaro deve produrre un blocco di testo cifrato univoco

Ogni iterazione genera due vettori di  $w$  bit ( $L_i$ ,  $R_i$ ) a partire dai risultati del round  $i-1$ -esimo

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus F(R_{i-1}, k_i)$$

$F$  funzione non lineare per produrre confusione  
La diffusione discende dallo scambio tra  $L$  e  $R$



Ogni round ha una regola di sostituzione stabilita dalla sottochiave

## Reti di Feistel: Cifratura/Decifrazione

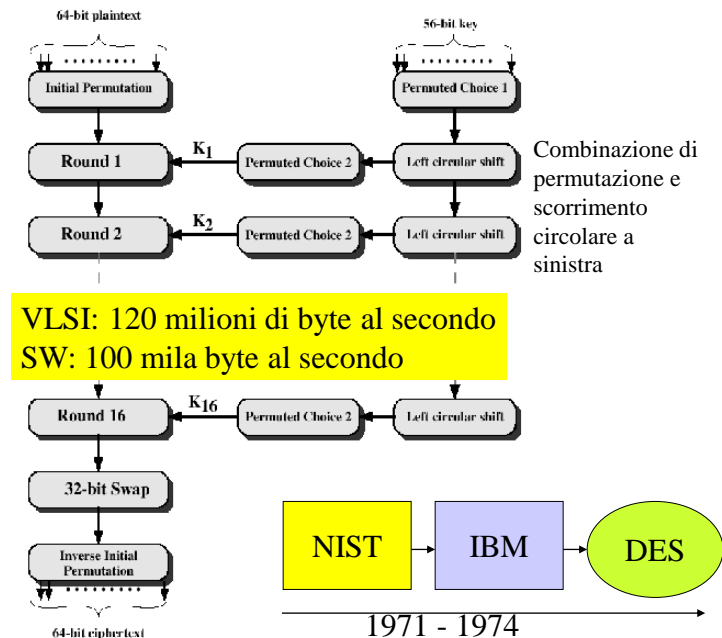
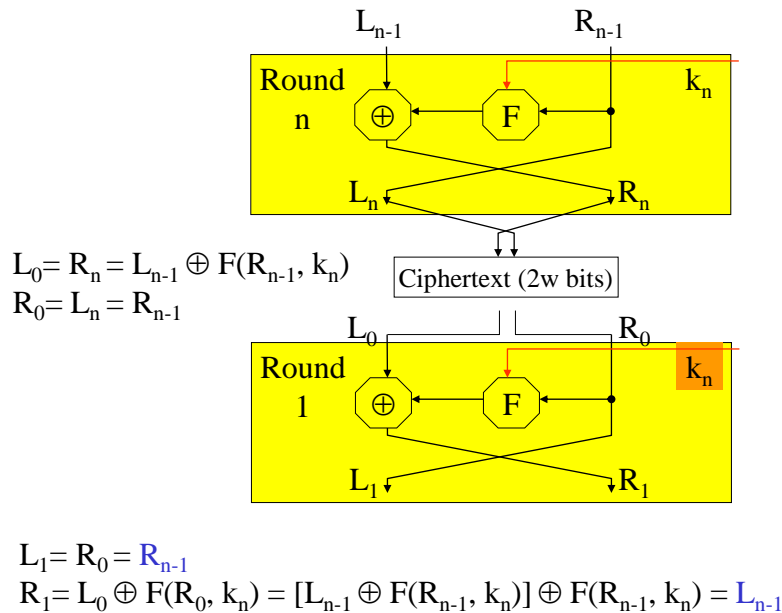
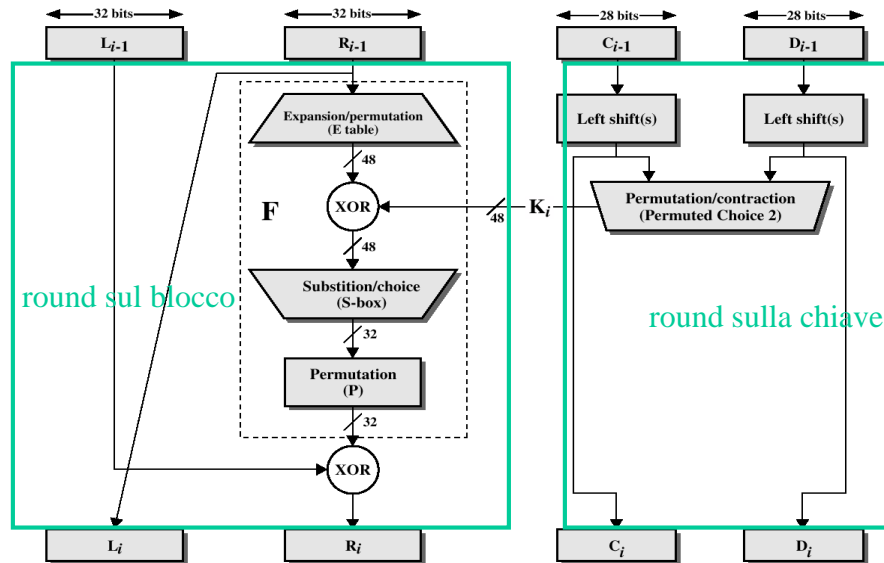


Figure 2.3 General Depiction of DES Encryption Algorithm



Ogni S-box produce una sostituzione reversibile

Figure 2.4 Single Round of DES Algorithm

## I successori del DES

Hw  $\rightarrow$  Sw

K: 64  $\rightarrow$  128+

B: 64  $\rightarrow$  128+

IDEA  
TDES  
BLOWFISH  
CAST-128  
ecc.

# Advanced Encryption Standard

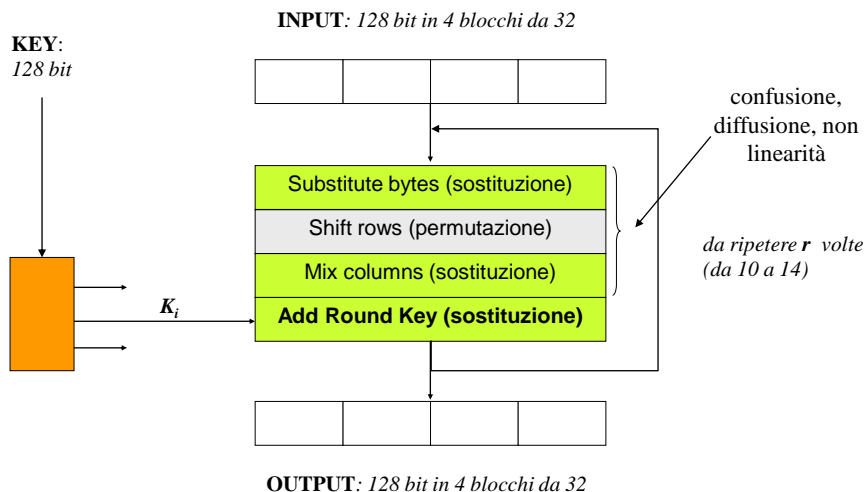
Nel 1997 il NIST emise una richiesta di proposte per un nuovo algoritmo  
5 finalisti su 16 candidati:

MARS, RC6, Rijndael, Serpent, Twofish

## Valutazione di Rijndael

- eccellenti prestazioni su tutte le piattaforme (dai main frame alle smart card),
- buon margine di sicurezza a fronte di ogni attacco conosciuto,
- bassa richiesta di memoria, sia ROM che RAM,
- veloce procedura di key setup,
- buone caratteristiche per l'esecuzione parallela delle istruzioni,
- chiavi e blocchi di principio di lunghezza variabile per multipli di 32 bit.

## Un round di Rijndael

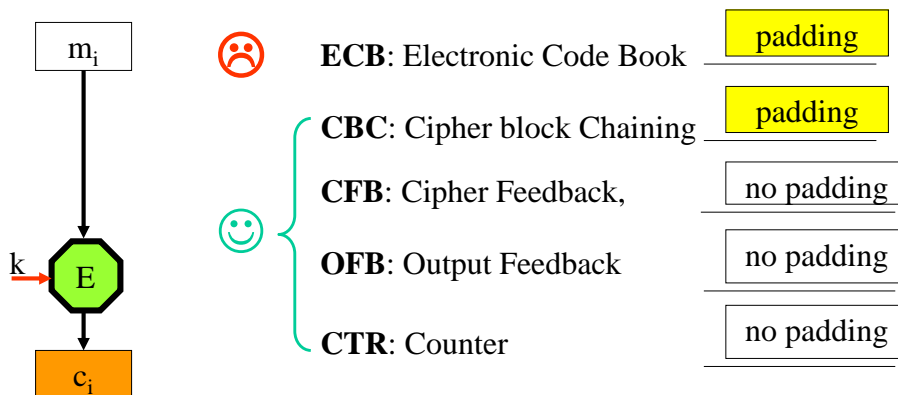


No struttura di Feistel: no a metà del blocco che viene utilizzata per modificare l'altra e poi le metà scambiate



## Modalità di cifratura

### Modalità di elaborazione a blocchi

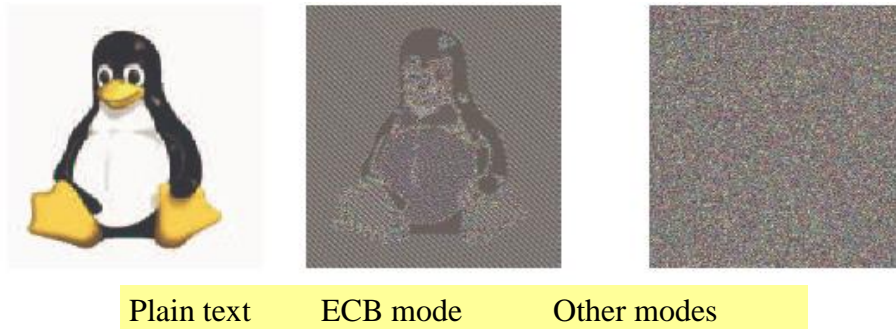


blocchi identici di testo in chiaro

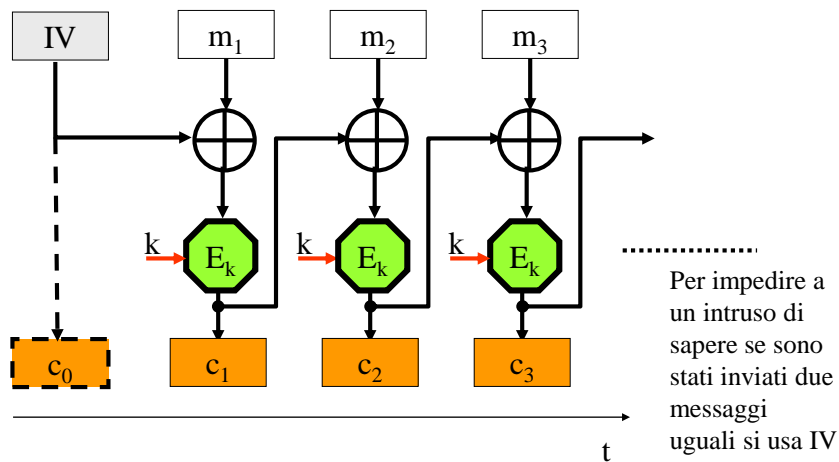
producono

blocchi identici di testo cifrato

Se il messaggio è strutturato l'analisi crittografica può sfruttarne le regolarità



## Cipher Block Chaining



### DECIFRAZIONE

$$D(c_i, k) = m_i \oplus c_{i-1}$$

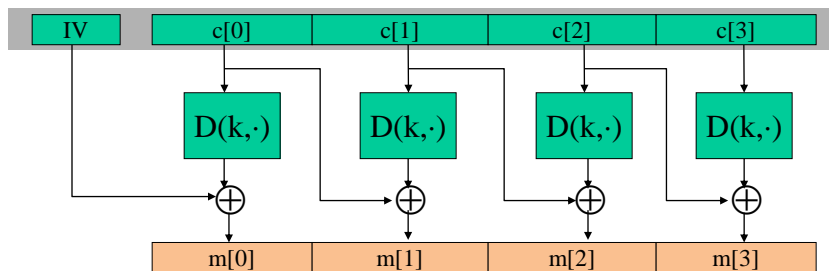
$$D(c_i, k) \oplus c_{i-1} = m_i \oplus c_{i-1} \oplus c_{i-1} = m_i$$

### DECIFRAZIONE

$$m_1 = D(c_1, k) \oplus IV$$

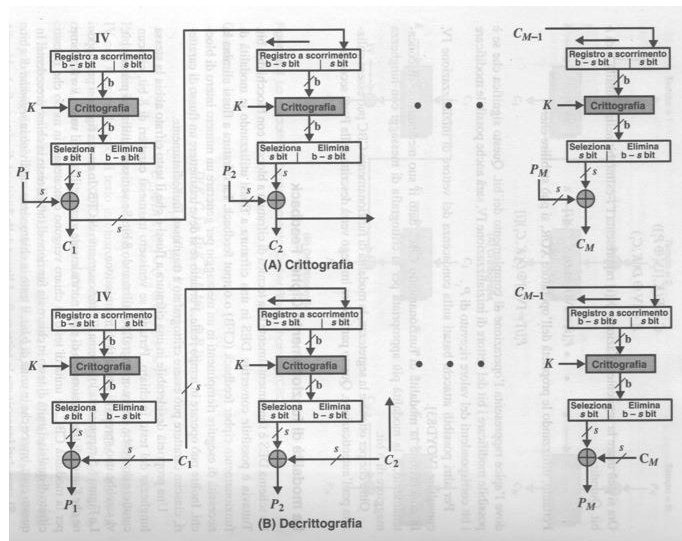
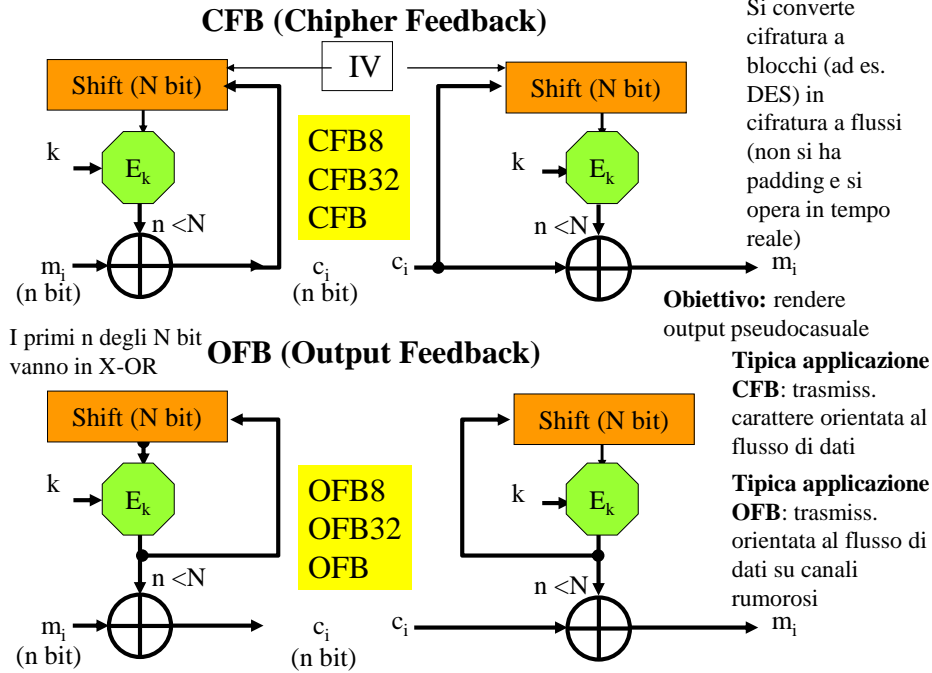
## Decryption circuit

In symbols:  $c[0] = E(k, IV \oplus m[0]) \Rightarrow m[0] = D(k, c[0]) \oplus IV$



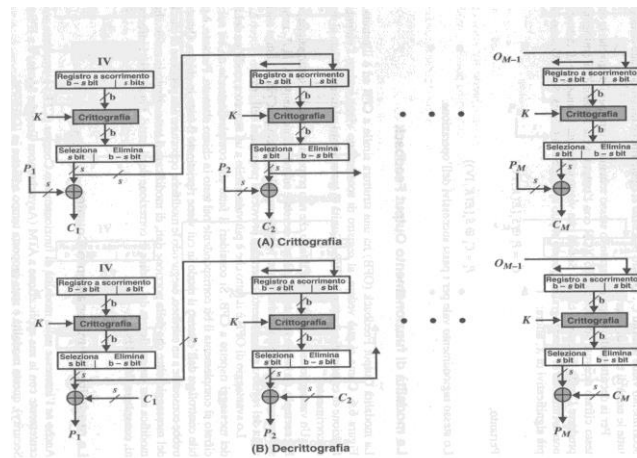
## Modalità CBC: vantaggi/svantaggi

- Ciascun blocco di ciphertext dipende da **tutti i precedenti** blocchi di plaintext
- Un cambiamento in un singolo blocco ha effetto su tutti i blocchi cifrati seguenti
- C'è bisogno di un vettore di inizializzazione (IV) noto al trasmettitore e al ricevitore, non dovrebbe essere riutilizzato



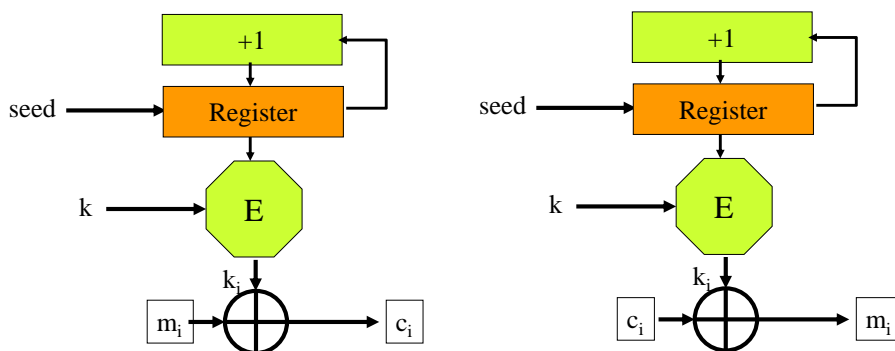
$$C_1 = P_1 \oplus S_s[E(k, IV)]$$

$$P_1 = C_1 \oplus S_s[E(k, IV)]$$

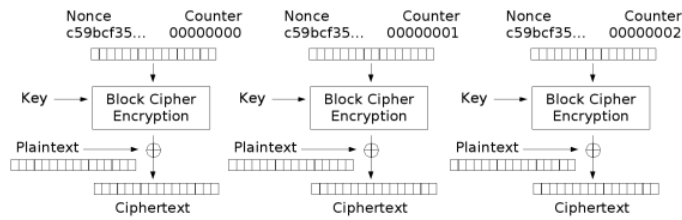


## CTR (Counter)

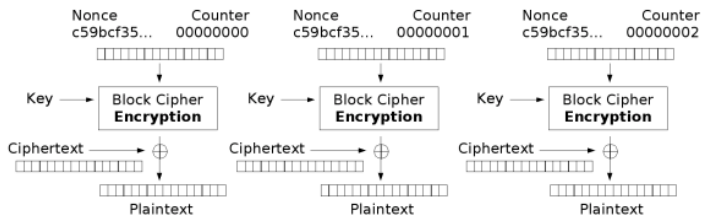
contatore di dimensione pari a quella del blocco; il valore del contatore differente per ogni blocco



Tipica applicazione: trasmissione di carattere orientata al blocco, utile per requisiti di alta velocità (pox. di esecuzione parallela su più blocchi di testo in chiaro)



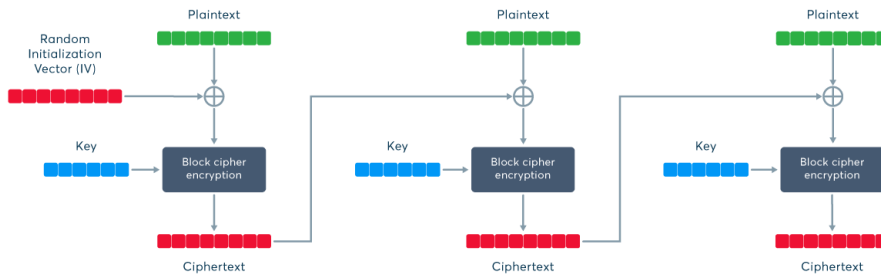
Counter (CTR) mode encryption



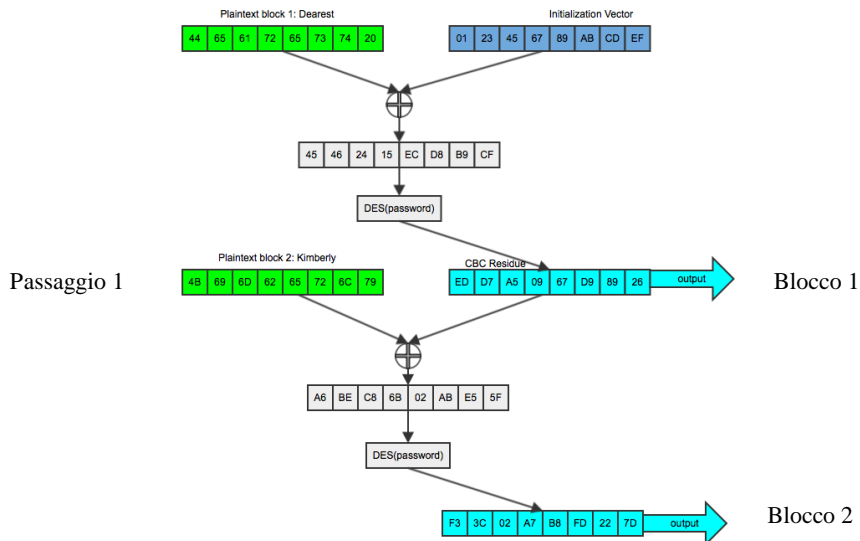
Counter (CTR) mode decryption

## Beast Attack

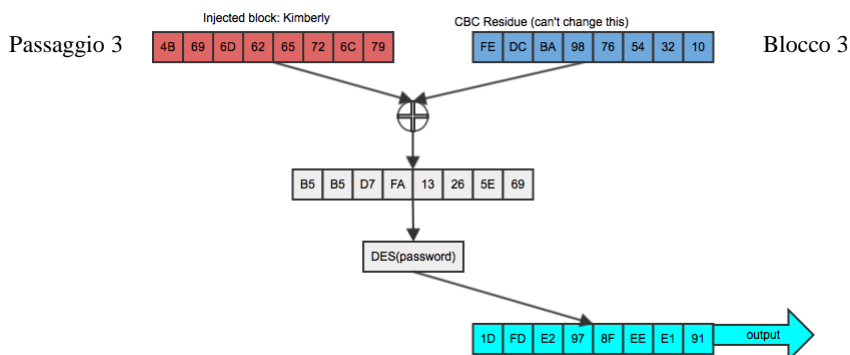
### Browser Exploit Against SSL/TLS



# Beast Attack



# Beast Attack



Supponiamo per semplicità che il CBC residuo al passo precedente prima di fare injection di Kimberly sia quello indicato in figura. Ovviamente dato lo stesso input l'output è diverso da quello al blocco 2 perché diversi sono i vettori di inizializzazione e l'attaccante non può modificare il CBC residuo ma solo iniettare input. **L'obiettivo è cercare di capire se dando un input opportuno al passaggio 3 si ha lo stesso output del blocco 2 così da poter confrontare e dedurre che Kimberly è l'input al blocco 2**

# Beast Attack....

basta conoscere come si comporta l'XOR, cioè se si fa l'xor di uno stesso valore due volte, il secondo xor annulla il primo

$$m1 \oplus K = m1 \oplus K1 \oplus K \oplus K1$$

$m1 = \text{Kimberly}$

$K = \text{CBC residuo al blocco 1}$

$K1 = \text{CBC residuo al blocco 3}$

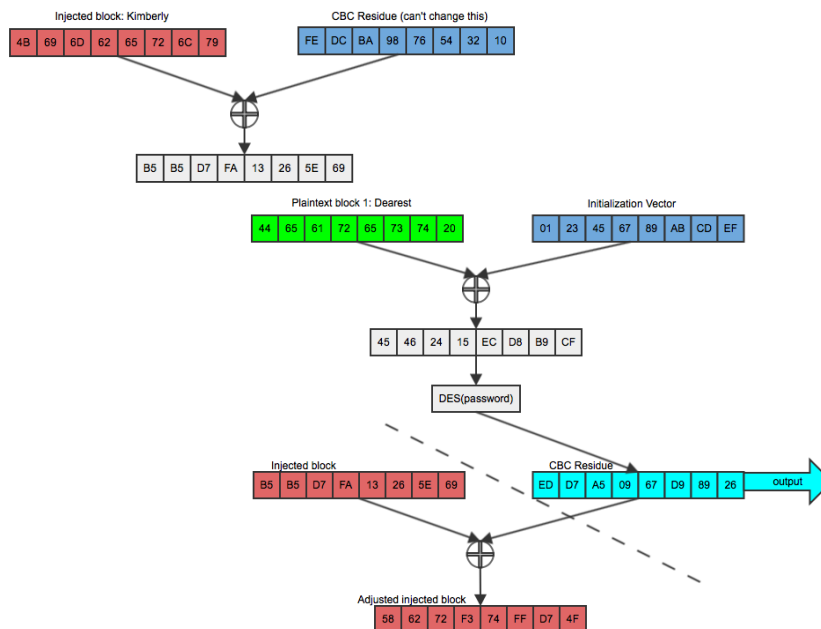
## TRUCCO

Al blocco 3 non do come input Kimberly ma:

$$\text{Kimberly} \oplus K1 \oplus K \text{ così } K1 \text{ si annulla}$$

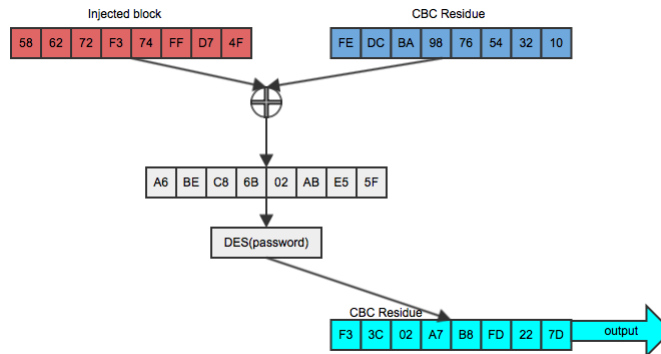
La cifratura diventa quindi esattamente quella al passaggio 1

# Beast Attack





# Beast Attack



## Birthday Attacks 64-bit Cipher Blocks

La dimensione della chiave è importante ma anche la dimensione del blocco che definisce quanti dati possono essere cifrati con la stessa chiave. La cifratura deve essere robusta fino a  $2^n$  input diversi.

Purtroppo molte modalità di cifratura diventano insicure dopo  $2^{n/2}$  cifrature a causa dell'aumento di probabilità di collisioni tra due blocchi di cifrato.

La collisioni tra due blocchi permette di rivelare l'X-OR tra i testi in chiaro dei corrispondenti blocchi. Se l'attaccante riesce a fare ipotesi su un testo in chiaro può recuperare l'altro testo in chiaro

Ipotesi: l'attaccante può osservare l'intero traffico e ha informazioni su alcune porzioni di testo in chiaro

## Il paradosso del giorno del compleanno

### **Birthday paradox**

Nell'ipotesi che le date di nascita siano equiprobabili, è sufficiente scegliere a caso **23** persone per avere una probabilità  $> 0,5$  che una di queste compia gli anni in un dato giorno.

Sono invece sufficienti **23** persone scelte a caso per avere una probabilità  $> 0,5$  che due o più compiano gli anni nello stesso giorno.

## Birthday Attacks 64-bit Cipher Blocks

Standard bodies recommend to change the key just before  $2^{n/2}$  blocks, and many implementations don't enforce any limit on the use of a key.

There are many uses of block ciphers with 64-bit blocks where large amount of data are potentially encrypted under the same key, such as:

- 3G telephony (UMTS), encrypted with KASUMI;
- OpenVPN, which uses Blowfish as the default cipher;
- many Internet protocols, such as TLS, IPsec and SSH, support Triple-DES as a legacy cipher.

In all these scenarios, 32 GB of data can be transferred in less than one hour with a fast connection.

# Birthday Attacks su CBC

Una collisione su due cifrati CBC significa avere due input identici:

$$m_i \oplus c_{i-1} = m_j \oplus c_{j-1}$$

$$m_i \oplus m_{i-1} = c_{i-1} \oplus c_{j-1}$$

Per il paradosso del compleanno la probabilità di avere una collisione tra due blocchi è proporzionale a  $2^{n/2}$

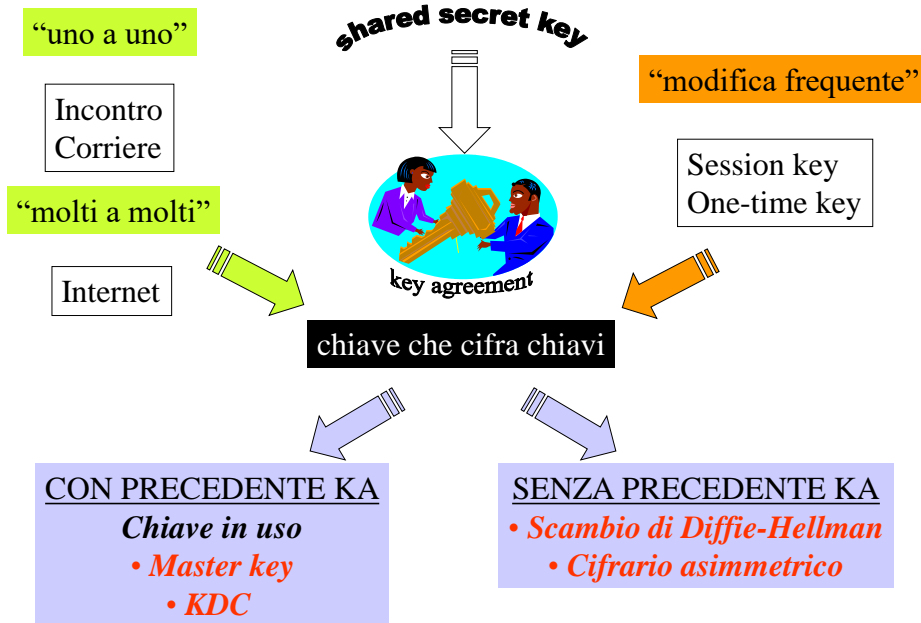
## Ricapitolando:

- Cifrario a blocco con ECB con almeno due blocchi. Caso di uso di una chiave una sola volta:  
=> anche nel caso di un solo campione di testo cifrato non ho sicurezza (dal cifrato si apprendono info sul testo in chiaro)  
Vulnerabile sia ad attacchi ciphertext only and chosen plaintext => ok se ECB cifra un solo blocco!
- Cifrario a blocco con CBC: si garantisce protezione contro ciphertext only attacks purché ben utilizzato!!! (IV casuale e imprevedibile)

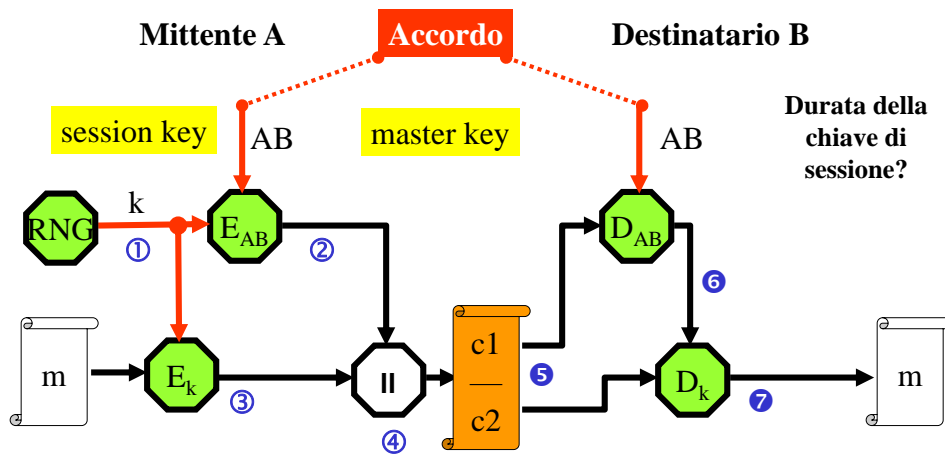
- Se CBC non usa IV imprevedibili (ossia l'intruso può predire quale IV verrà usato per un messaggio successivo), anche CBC vulnerabile ad attacchi chosen-plaintext.



## Accordo sulla chiave segreta



## La master key

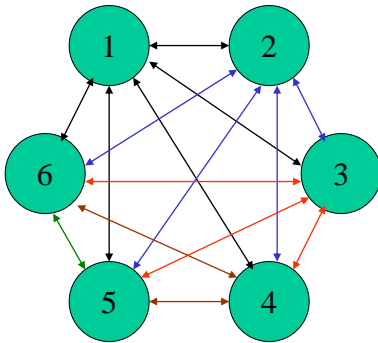


La chiave  $AB$  cifra solo le chiavi  $k$  e può avere una vita “lunga”  
 La chiave  $k$  cifra messaggi anche “lunghi” ed è usata una volta sola

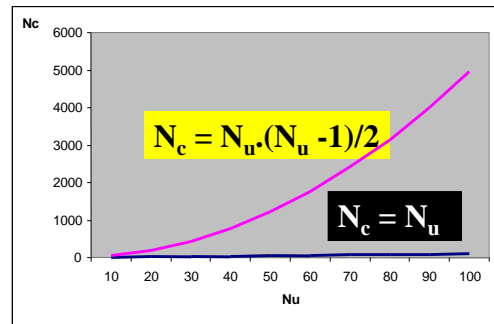
**Key Distribution Center**

## Numero di chiavi in circolazione

Comunità di utenti



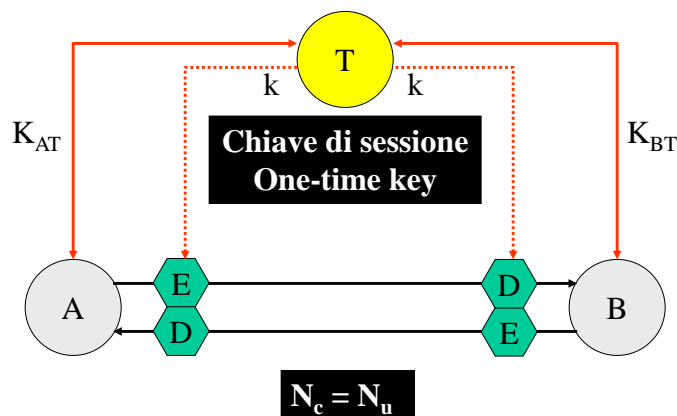
**Non è scalabile!**

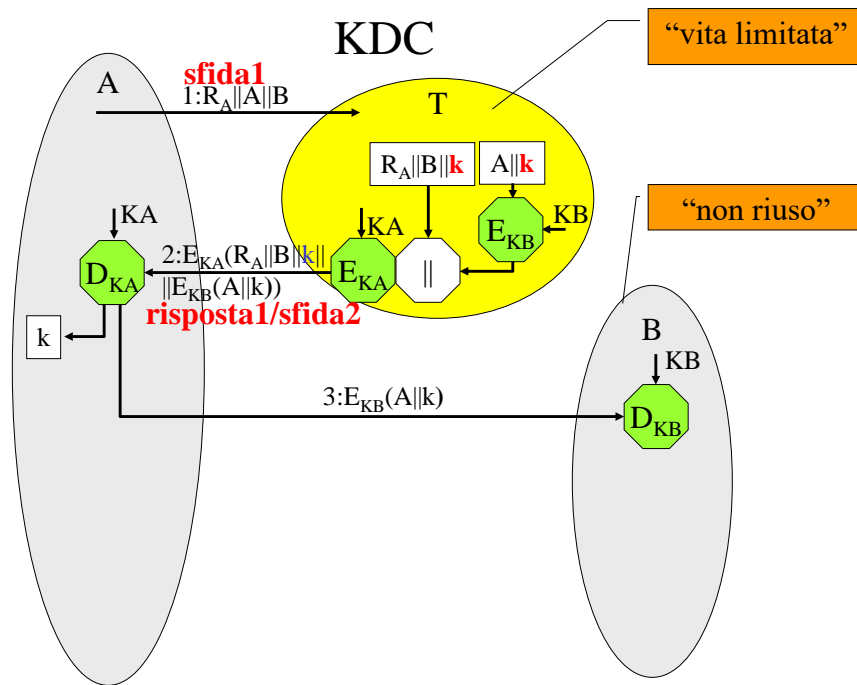


**Obiettivo da perseguire:**  
“una chiave per utente”

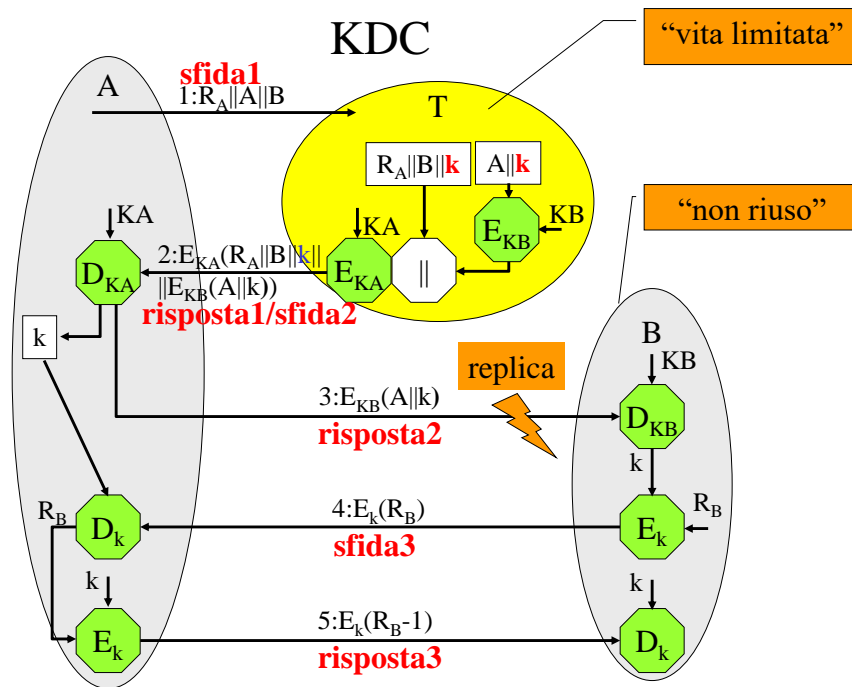
**Soluzione: ogni utente concorda la sua chiave con una terza parte**

## L'Autorità per la distribuzione chiavi









## Problemi di KDC

- On-line
- Collo di bottiglia ( $n^\circ$  max di utenti)
- Memoria sicura
- Ente degno di fiducia

*KryptoKnight,  
Kerberos,  
Distributed Computing Environment,  
Windows 2000*



## Diffie-Hellman key agreement

### Il contesto “tutti con tutti”

	Accordi precedenti	Numero di chiavi	Valutazione della realizzabilità
Incontri & Corrieri	SI	Enorme	difficile
Rete mondiale di N KDC	SI	$1+N(N-1)/2$	difficile
Scambio D-H	NO	1 e one-time!	facile

## Scambio di chiavi Diffie-Hellman

Si basa sulla difficoltà del calcolo dei logaritmi discreti

$p$  numero primo grande

Generatore di  $p$  numero le cui potenze modulo  $p$  generano tutti gli interi compresi tra 1 e  $p-1$

Se  $g$  generatore allora

$g \bmod p, g^2 \bmod p, \dots, g^{p-1} \bmod p$

sono distinti e costituiti dai numeri compresi tra 1 e  $p-1$

Per un qualsiasi intero  $b$  e un generatore  $g$  di  $p$ , si può trovare un esponente univoco  $i$  tale che:

$$b = g^i \bmod p \text{ dove } 0 \leq i \leq (p-1)$$

$i$  è chiamato logaritmo discreto di  $b$  per la base  $g$ , modulo  $p$

## Algoritmo DH *anonimo* per l'accordo di una chiave di sessione tra gli utenti A e B

*Numero primo  $p$  e generatore  $g$*  prefissati e noti

### 1. Generazione delle chiavi segrete

$X_A$  e  $X_B$  scelti a caso  $> 1$  e  $< p-1$

### 2. Generazione e comunicazione delle chiavi pubbliche

$$Y_A = g^{X_A} \bmod p \text{ e } Y_B = g^{X_B} \bmod p$$

### 3. Calcolo della chiave del Cifrario simmetrico

$$K_A = Y_B^{X_A} \bmod p = (g^{X_B})^{X_A} \bmod p$$

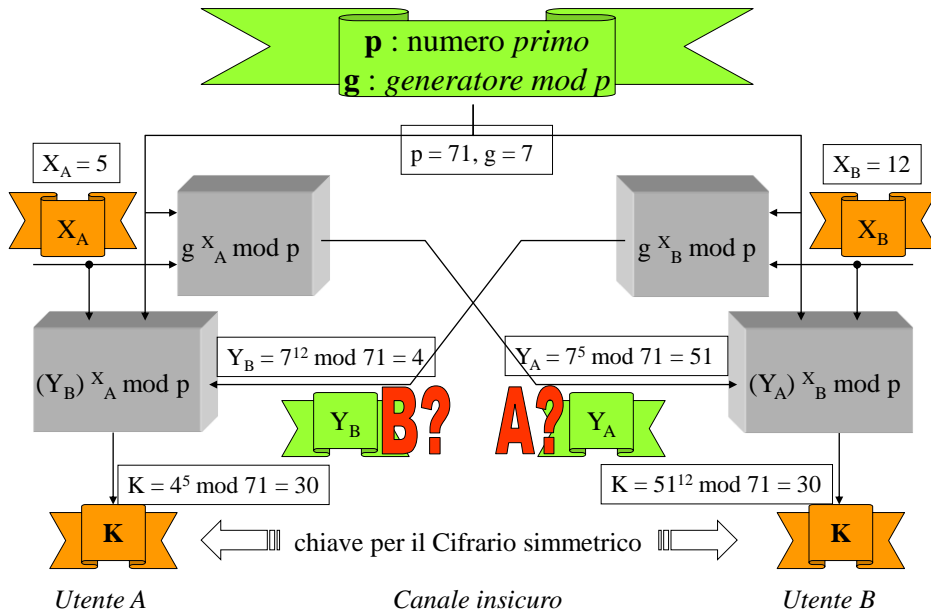
$$K_B = Y_A^{X_B} \bmod p = (g^{X_A})^{X_B} \bmod p$$

$$K_A = K_B$$

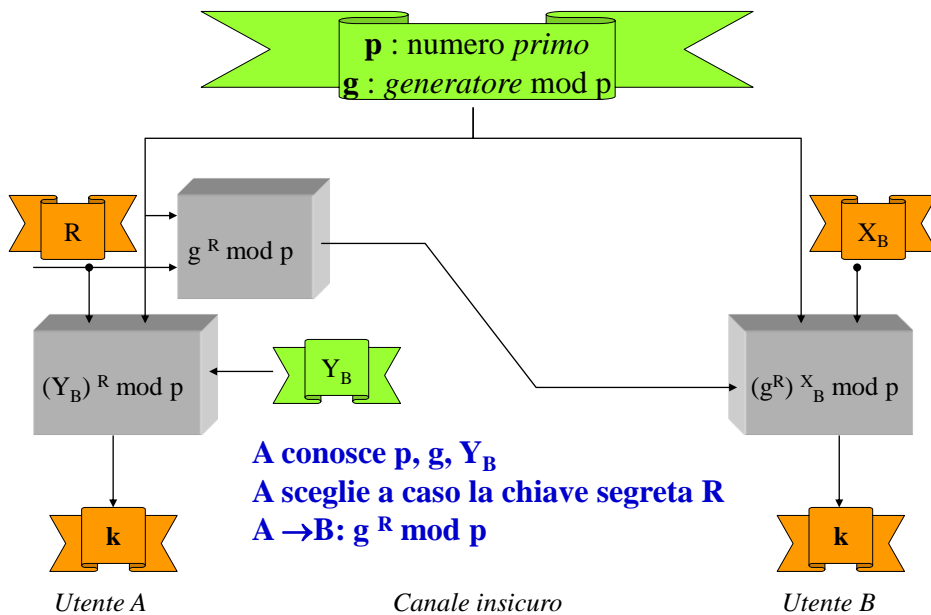
La dimensione è grande (quella di  $p$ ): occorre scegliere  $k$

**DH *anonimo*: l'origine di  $Y$  non è attestata in modo sicuro**

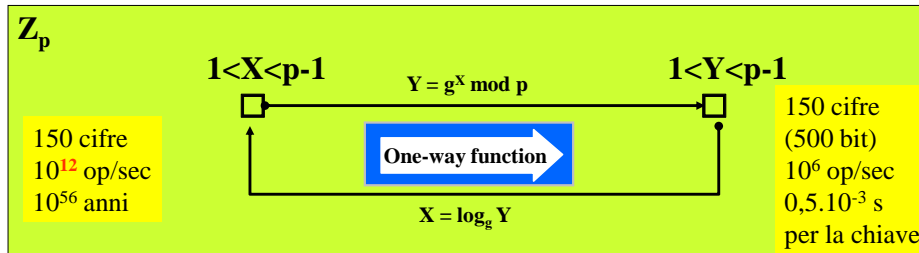
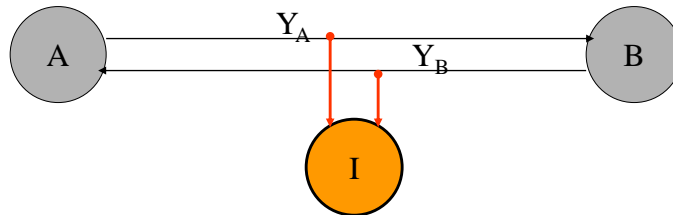
## Lo scambio di Diffie-Hellman



## Variante DH/ElGamal



## Sicurezza dello scambio DH



**P1: problema (difficile) del logaritmo discreto su un campo di Galois**

“ Dato un primo  $p$ , un generatore  $g$  ed un intero  $c \in \mathbb{Z}_p^*$ , trovare l'intero  $x$   $1 \leq x \leq p-1$ , tale che  $g^x \bmod p = c$ ”

$\mathbb{Z}_p^*$  insieme di interi non negativi non nulli minori di  $p$

## Problemi risolti e nuovi problemi

- Ripudio
- Falsificazione

- L'Autorità deve essere sempre **on-line**.
- L'Autorità non deve costituire un **collo di bottiglia**.
- L'Autorità non deve creare **documenti falsi**.
- L'Autorità deve tenere le chiavi in una **memoria sicura**.