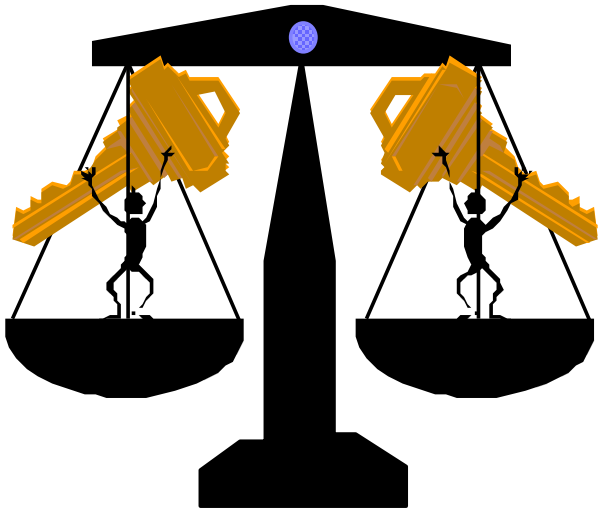
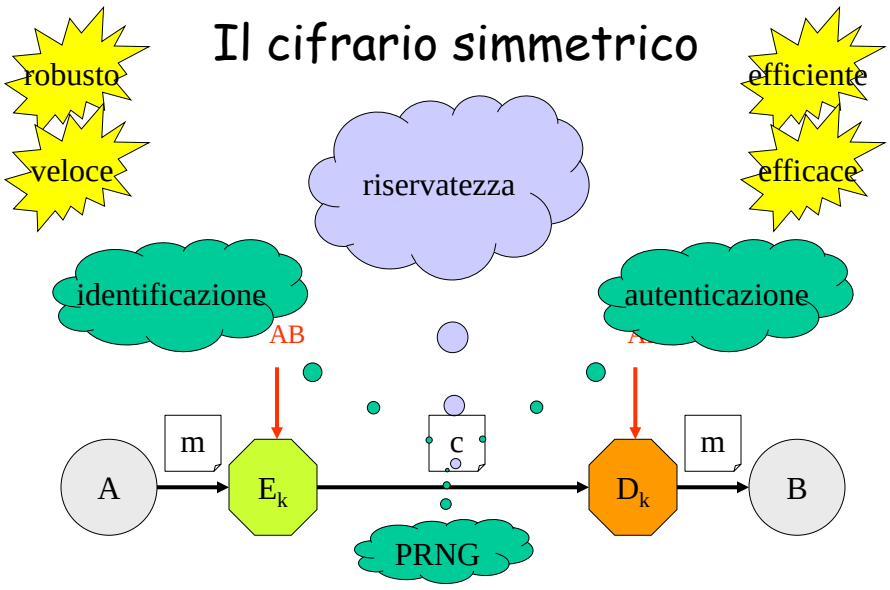
Meccanismi simmetrici



Il cifrario simmetrico



$m_1, m_2, .., m_N$
$c_1, c_2, .., c_N$

1. A: calcola $c = E_{AB}(m)$ e trasmette c
2. B: calcola $D_{AB}(c) = D_{AB}(E_{AB}(m)) = m$

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

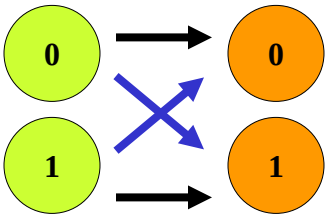
+ sicuro

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

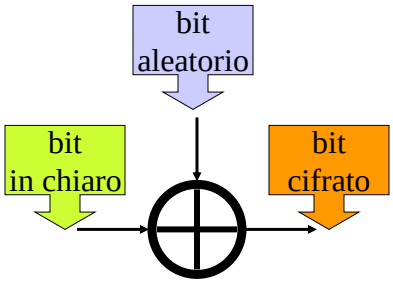
Il meccanismo per la sostituzione di un bit



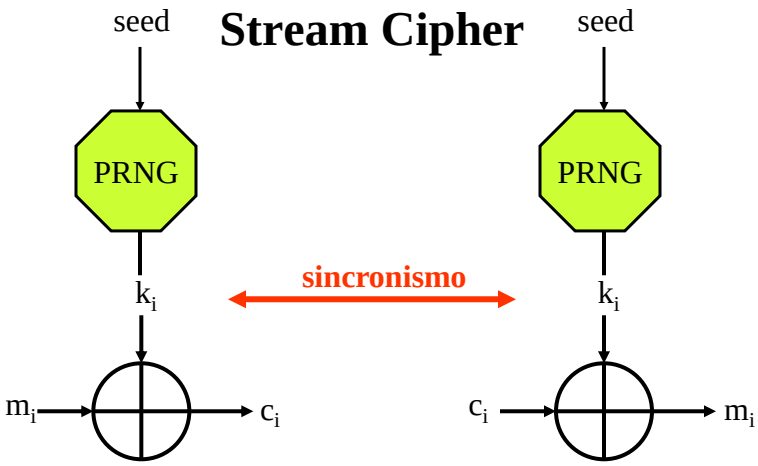
bit di chiave

bit di testo

⊕	0	1
0	0	1
1	1	0



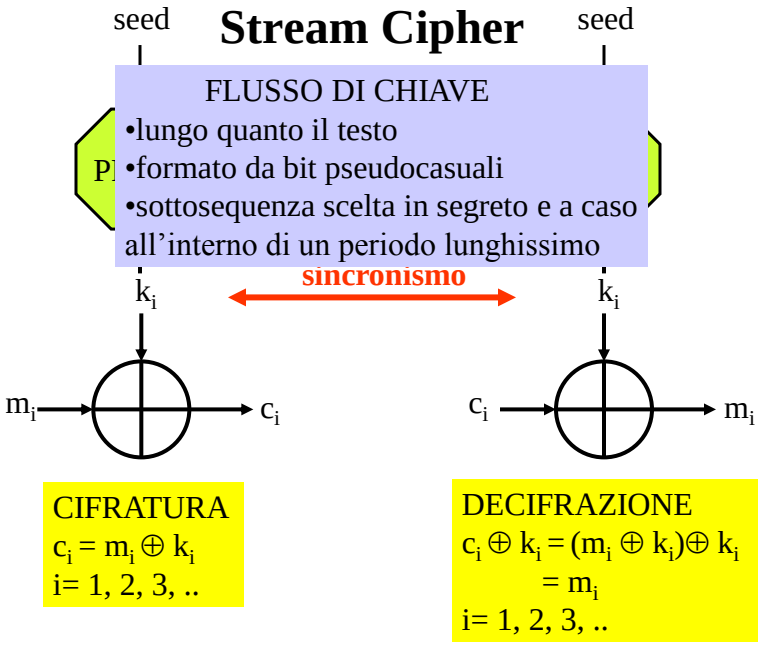
Stream Cipher



CIFRATURA
$c_i = m_i \oplus k_i$
i= 1, 2, 3, ..

DECIFRAZIONE
$c_i \oplus k_i = (m_i \oplus k_i) \oplus k_i$
$= m_i$
i= 1, 2, 3, ..

Stream Cipher

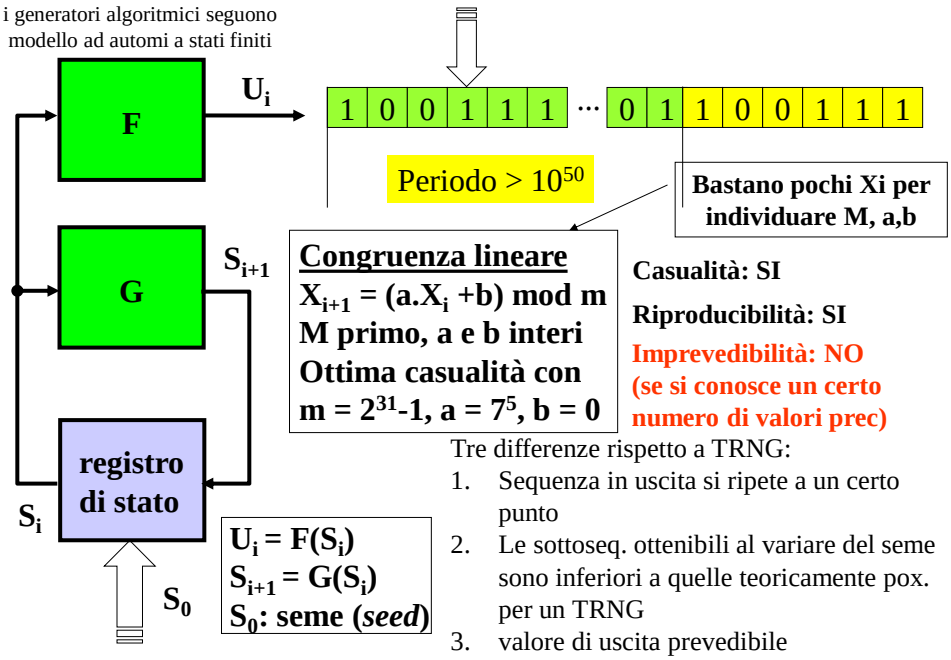


CIFRATURA
$c_i = m_i \oplus k_i$
i= 1, 2, 3, ..

DECIFRAZIONE
$c_i \oplus k_i = (m_i \oplus k_i) \oplus k_i$
$= m_i$
i= 1, 2, 3, ..

Pseudo Random Number Generator

i generatori algoritmici seguono modello ad automi a stati finiti



Casualità: SI

Riproducibilità: SI

Imprevedibilità: NO
(se si conosce un certo numero di valori prec)

Tre differenze rispetto a TRNG:

1. Sequenza in uscita si ripete a un certo punto
2. Le sottoseq. ottenibili al variare del seme sono inferiori a quelle teoricamente pox. per un TRNG
3. valore di uscita prevedibile

Cryptographically Secure PseudoRandom Bit Generator

casualità dell'uscita (periodo grandissimo)

imprevedibilità dell'uscita

indeducibilità di stati precedenti (imprevedibile e segreto)

Test statistici

Next-bit test: dati L bit della stringa d'uscita non deve esistere un algoritmo polinomiale in grado di predire il bit L+1-esimo con probabilità > 0,5

One-way function

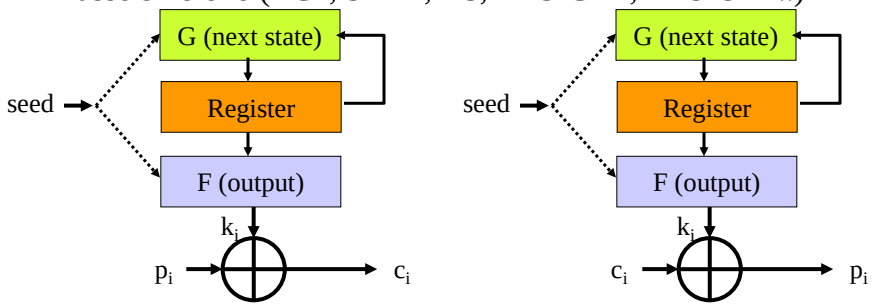
Funzione di stato futuro o di uscita

Crittografia simmetrica:
- verifica sperimentale
- alta velocità

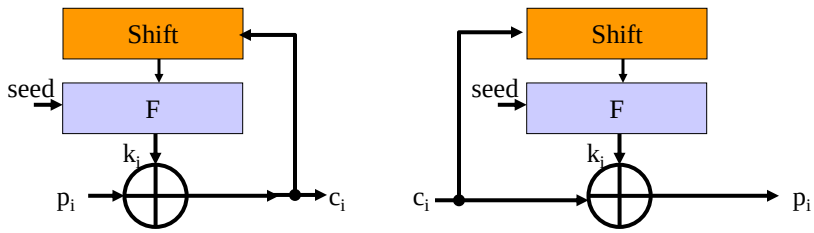
Crittografia asimmetrica:
- dimostrazione teorica
- bassa velocità

Stream ciphers

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



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



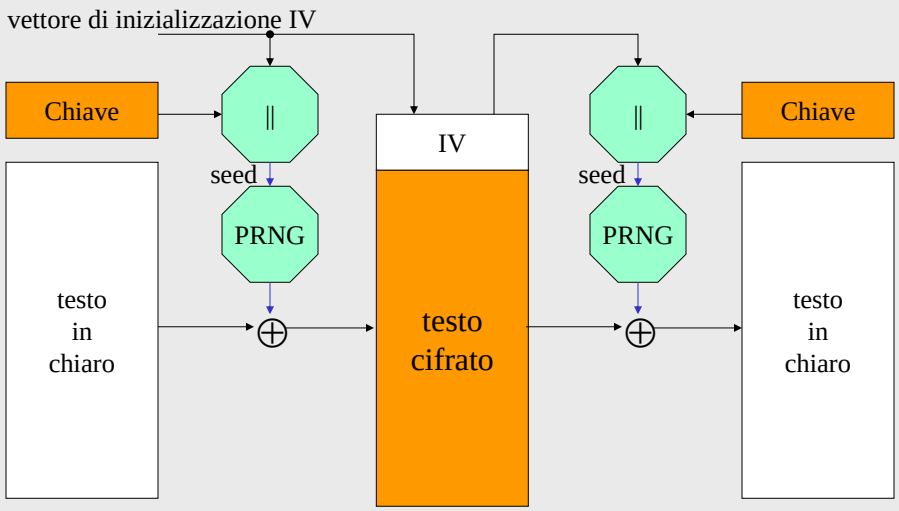
Problemi dei Cifrari a flusso

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

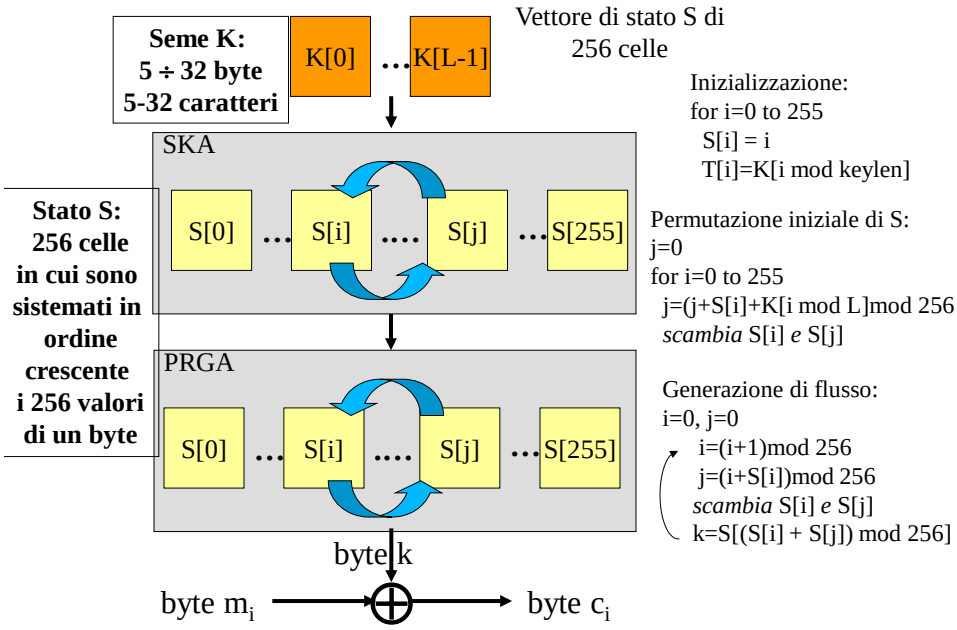
più usati

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

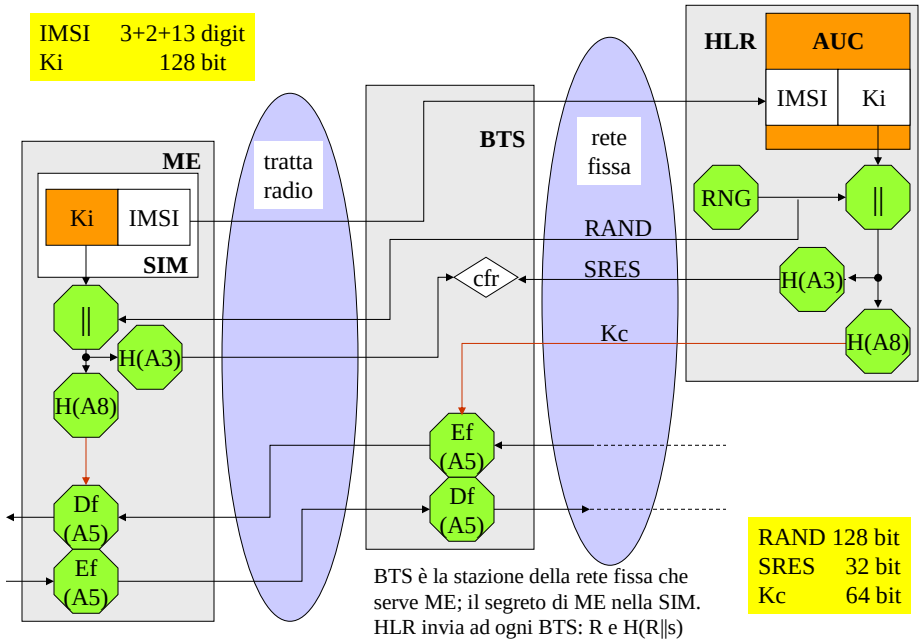
Segretezza e variabilità del seme (WEP)



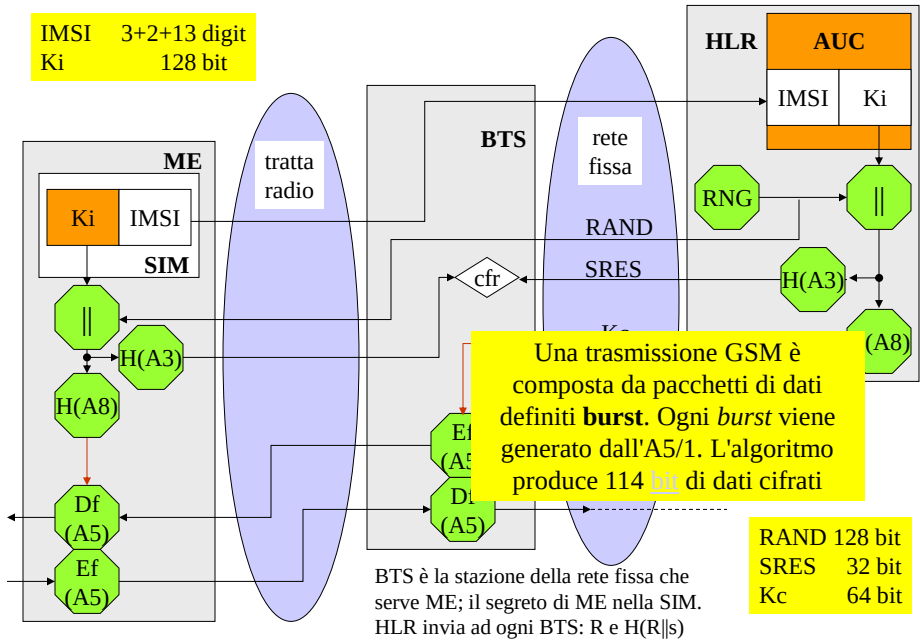
Il Cifrario a flusso RC4 (Rivest, 1984)



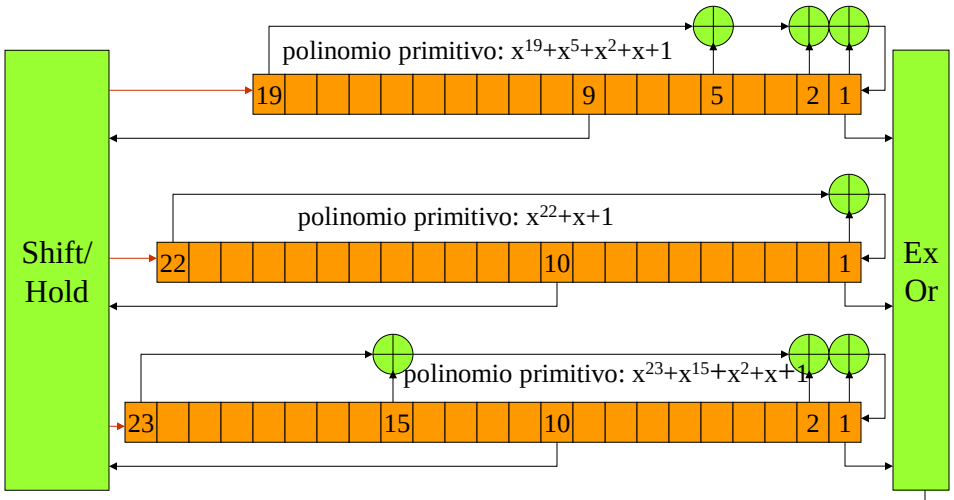
GSM: identificazione e riservatezza



GSM: identificazione e riservatezza

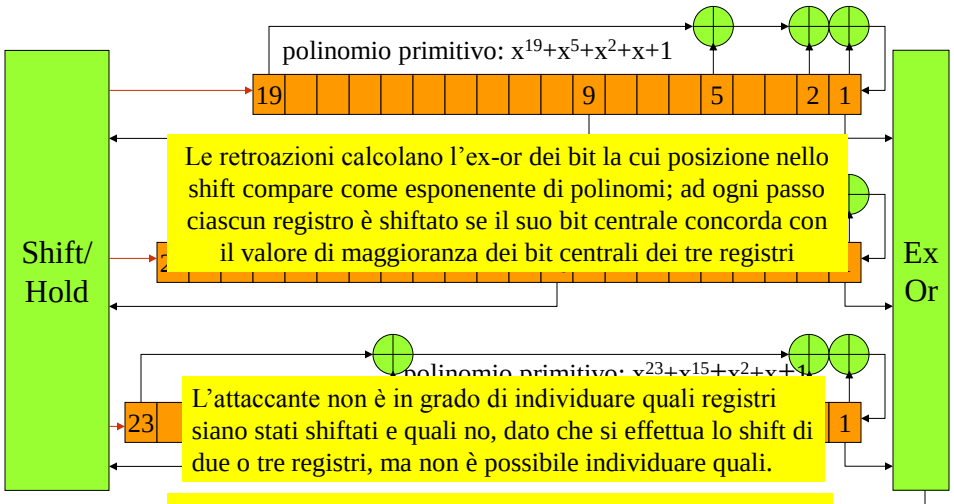


GSM: il generatore di flusso di chiave



3 registri a scorrimento a retroazione lineare inizializzati con una combinazione non lineare della chiave

GSM: il generatore di flusso di chiave



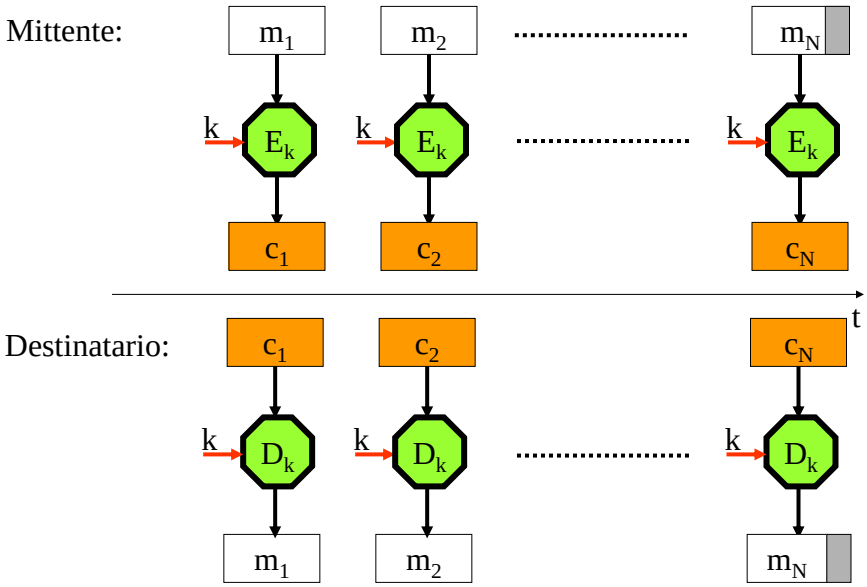
Le retroazioni calcolano l'ex-or dei bit la cui posizione nello shift compare come esponenente di polinomi; ad ogni passo ciascun registro è shiftato se il suo bit centrale concorda con il valore di maggioranza dei bit centrali dei tre registri

L'attaccante non è in grado di individuare quali registri siano stati shiftati e quali no, dato che si effettua lo shift di due o tre registri, ma non è possibile individuare quali.

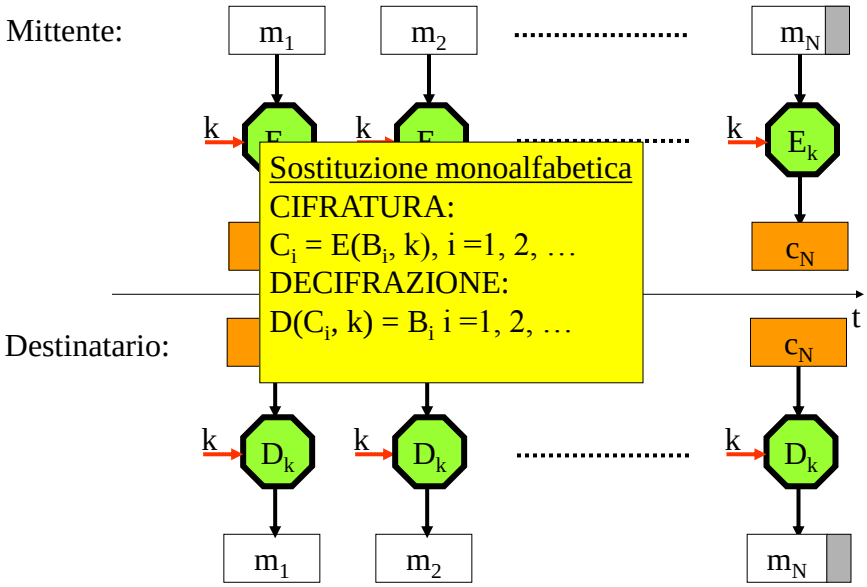
L'uso dei polinomi attribuisce al registro un periodo di massima lunghezza. Il periodo massimo di un registro a k bit è 2k– 1, che è la cardinalità dell'insieme delle sequenze di k bit, tranne quella composta da tutti 0.

Cifrari a blocchi

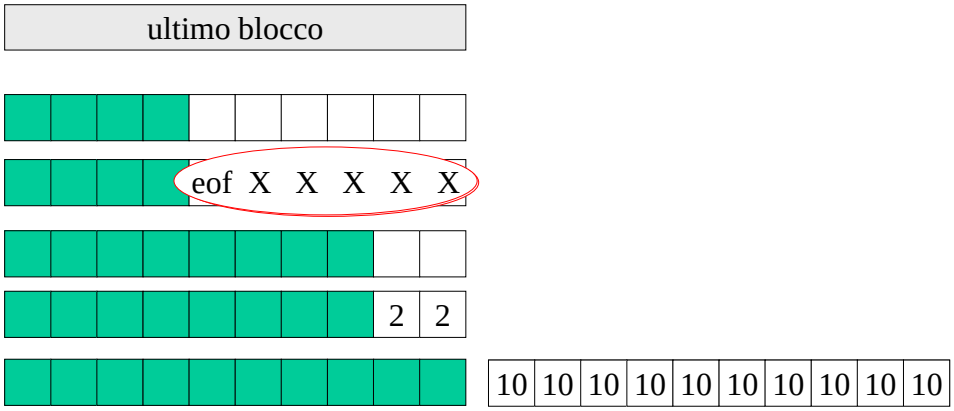
Block cipher (modalità ECB)



Block cipher (modalità ECB)



Padding: standard PKCS#5 e #7



Time to break a code

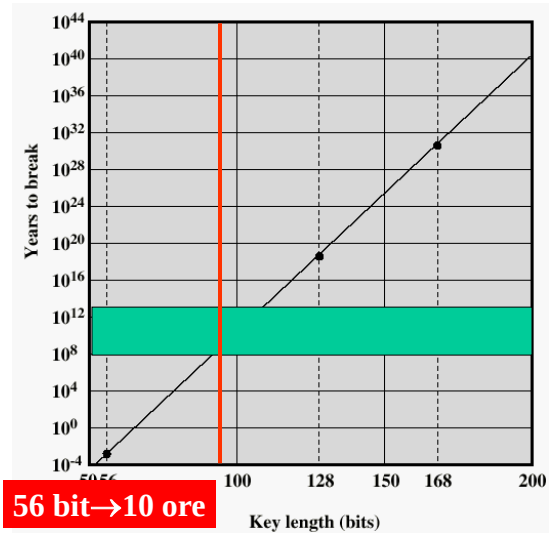
Spazio delle chiavi

N bit	2^N chiavi
32	$2^{32} = 4{,}3 \times 10^9$
56	$2^{56} = 7{,}2 \times 10^{16}$
128	$2^{128} = 3{,}4 \times 10^{38}$
168	$2^{168} = 3{,}7 \times 10^{50}$
192	$2^{192} = 6{,}3 \times 10^{57}$

$p = 2^{-N}$

Valutazione sicurezza a breve termine (1996)
R28: "75 bit
(6×10^{11} anni MIPS)
+14 bit ogni vent'anni"

Forza bruta: $T = 2^{N-1}/10^{12}$ s



56 bit→10 ore

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)".

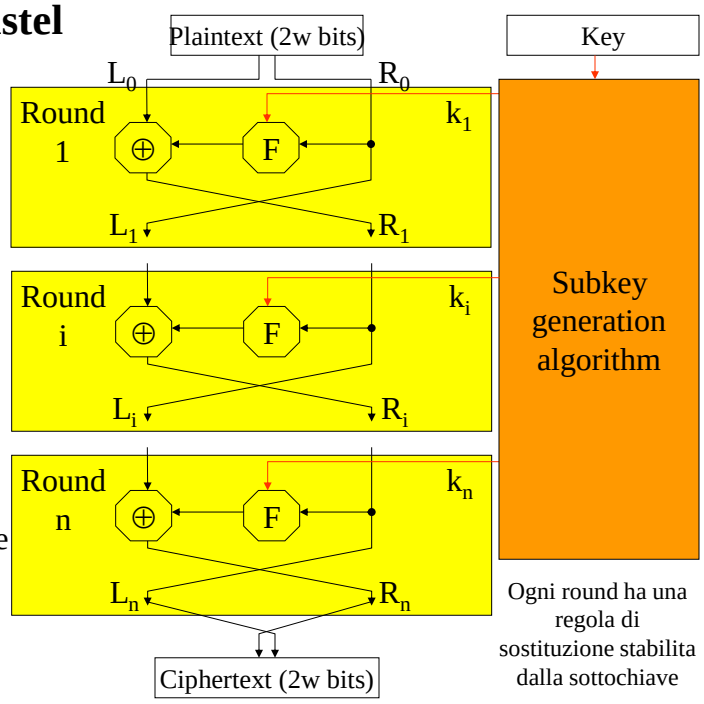
La rete di Feistel

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

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

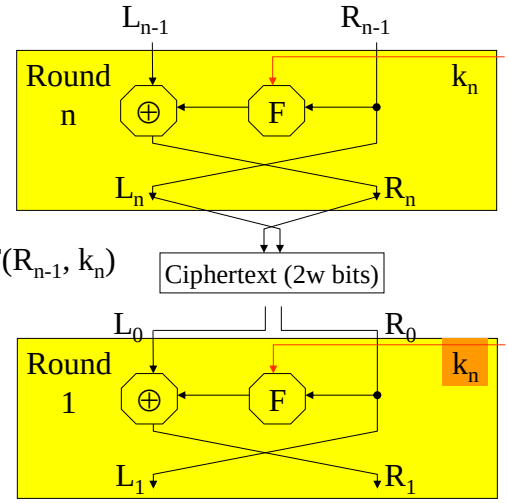
$L_i=R_{i-1}$
$R_i=L_{i-1}\oplus F(R_{i-1}, k_i)$

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



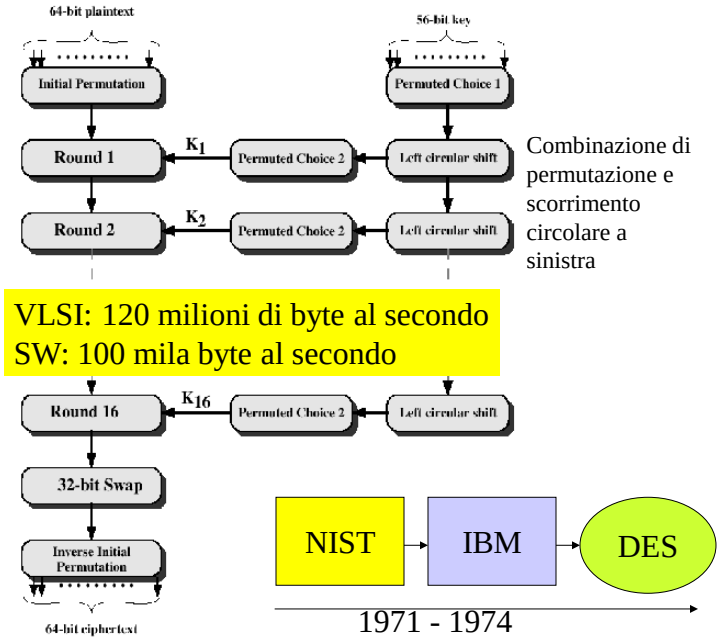
Ogni round ha una regola di sostituzione stabilita dalla sottochiave

Reti di Feistel: Cifratura/Decifrazione



$L_0 = R_n = L_{n-1} \oplus F(R_{n-1}, k_n)$
$R_0 = L_n = R_{n-1}$

$L_1 = R_0 = R_{n-1}$
$R_1 = L_0 \oplus F(R_0, k_n) = [L_{n-1} \oplus F(R_{n-1}, k_n)] \oplus F(R_{n-1}, k_n) = L_{n-1}$



Combinazione di permutazione e scorrimento circolare a sinistra

VLSI: 120 milioni di byte al secondo
SW: 100 mila byte al secondo

Figure 2.3 General Depiction of DES Encryption Algorithm

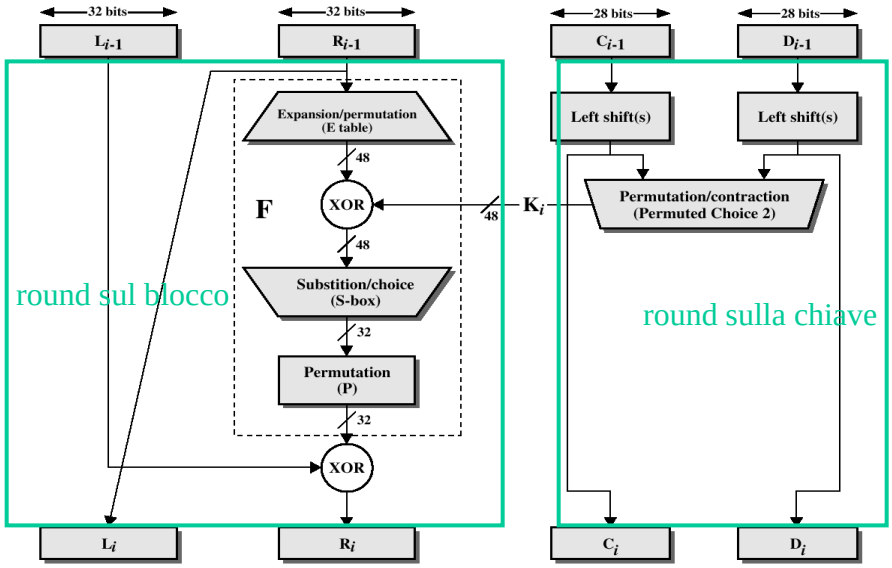


Figure 2.4 Single Round of DES Algorithm

Ogni S-box produce una sostituzione reversibile

I successori del DES

Hw → Sw

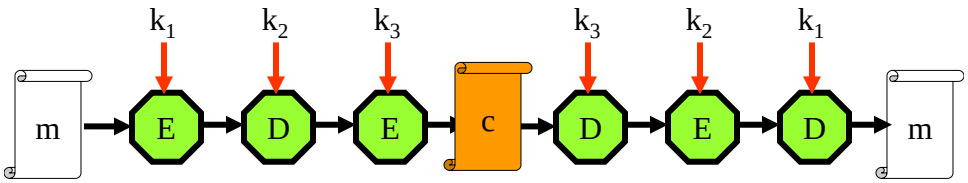
K: 64 → 128+

B: 64 → 128+

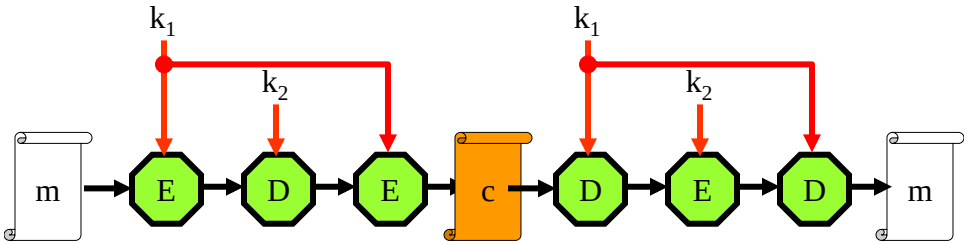
IDEA
TDES
BLOWFISH
CAST-128
ecc.

opera su 3 gruppi:
- Somma modulo 2 su vettori di 16 bit
- Addizione tra interi modulo 2^{16}
- Moltiplicazione tra interi modulo $2^{16}+1$

Il Triplo DES (TDEA, EDE)



La versione con 168 bit di chiave



La versione con 112 bit di chiave

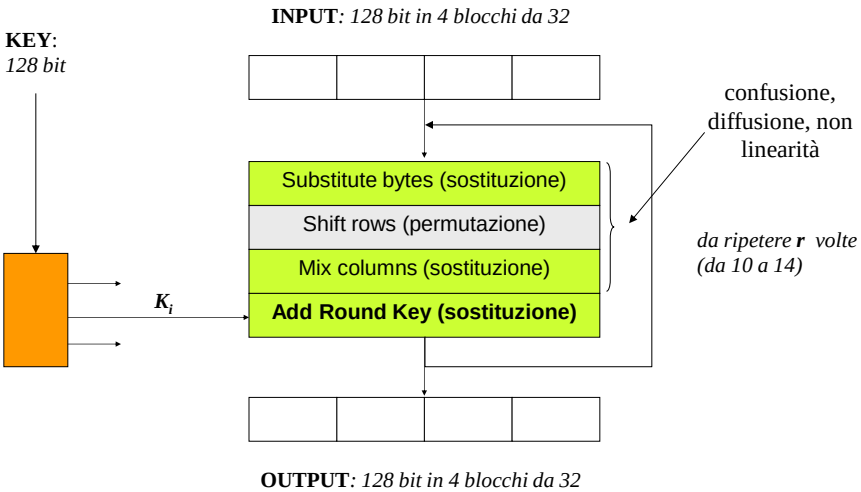
Advanced Encryption Standard

Nel 1997 il NIST emise una richiesta di proposte per un nuovo algritmo
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.

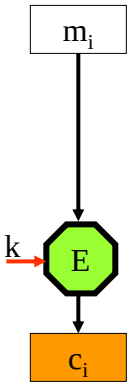
Un round di Rijndael



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



☹ **ECB**: Electronic Code Book — padding

☺ **CBC**: Cipher block Chaining — padding

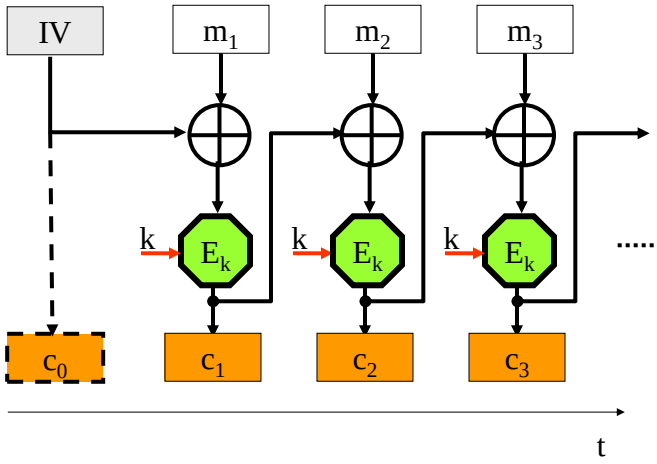
☺ **CFB**: Cipher Feedback, — no padding

☺ **OFB**: Output Feedback — no padding

☺ **CTR**: Counter — no padding

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



Per impedire a un intruso di sapere se sono stati inviati due messaggi uguali si usa IV

DECIFRAZIONE

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

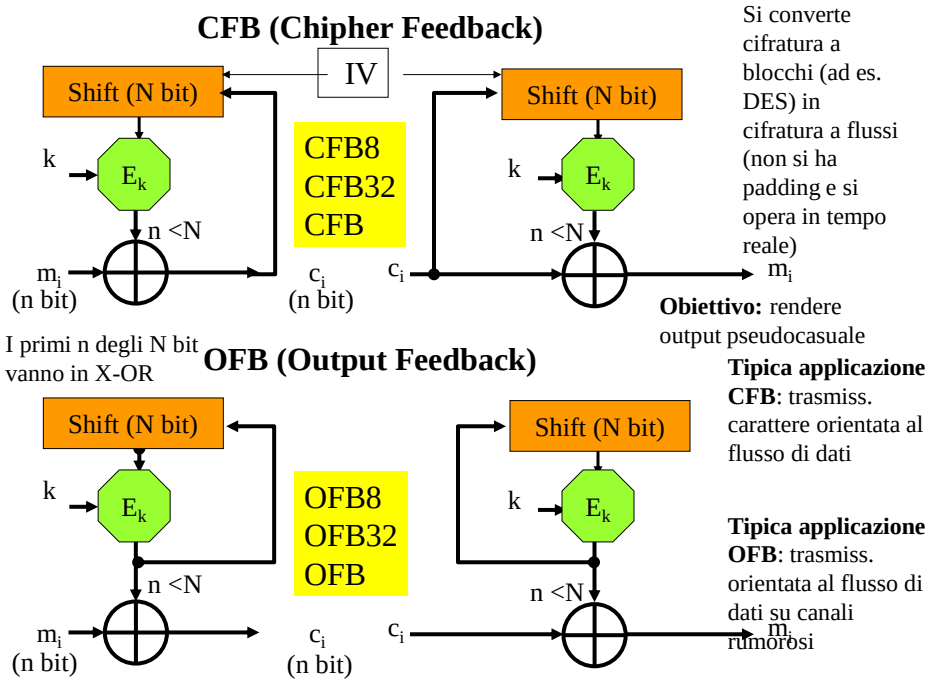
$D(c_i,k) \oplus c_{i-1} = m_i \oplus c_{i-1} \oplus c_{i-1} = m_i$

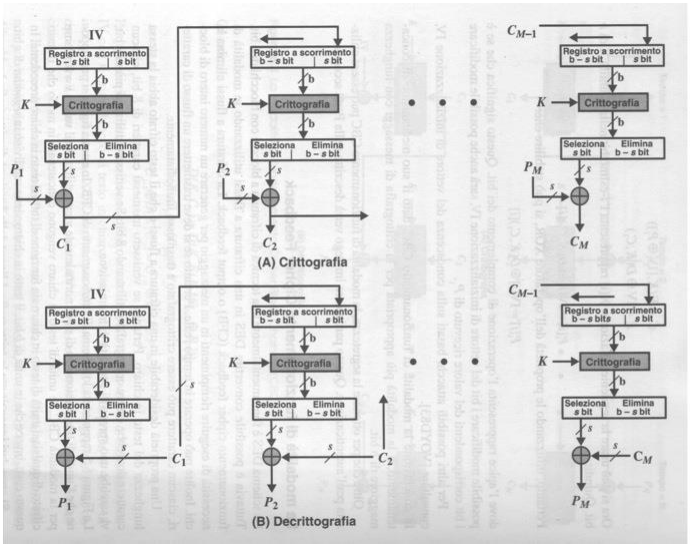
DECIFRAZIONE

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

Modalità CBC: vantaggi/svantaggi

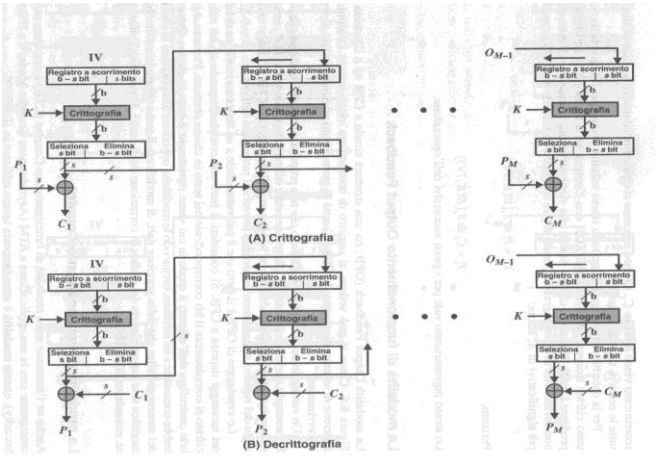
- Ciascun blocco di ciphertext dipende da **tutti i precedenti** blocchi di plaintext
- Un cambiamento in un singolo blocco ha effetto su tutti i blocchi cifrati seguenti
- C'è bisogno di un vettore di inizializzazione (IV) noto al trasmettitore e al ricevitore, non dovrebbe essere riutilizzato





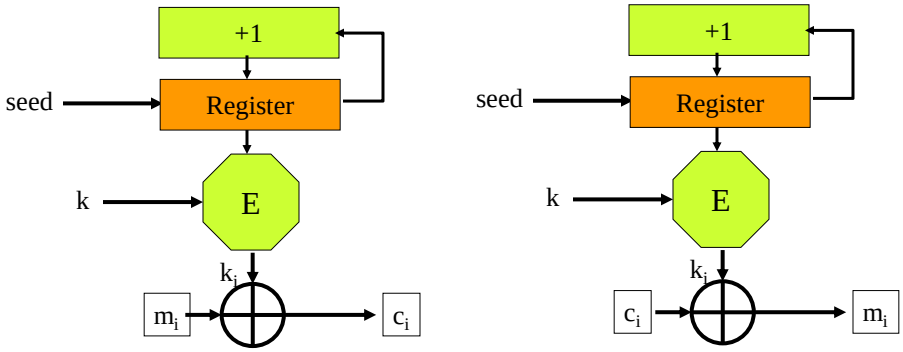
$C_1 = P_1 \oplus S_s[E(k,IV)]$

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

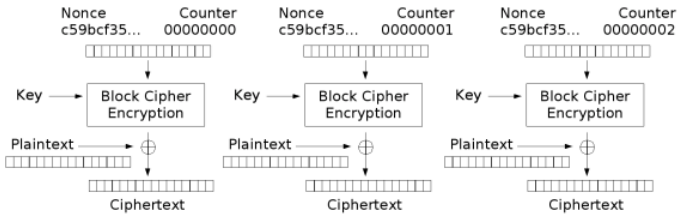


CTR (Counter)

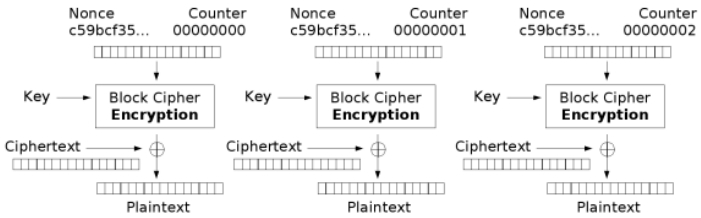
contatore di dimensione pari a quella del blocco; il valore del contatore differente per ogni blocco



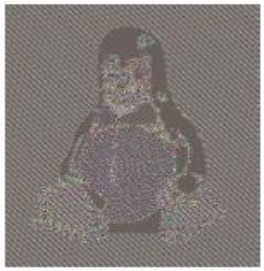
Tipica applicazione: trasmissione di carattere orientata al blocco, utile per requisiti di alta velocità (pox. di esecuzione parallela su più blocchi di testo in chiaro)



Counter (CTR) mode encryption



Counter (CTR) mode decryption



Plain text ECB mode Other modes

- Se Alice volesse velocemente decifrare un grande file usando molti processori quale modo preferirebbe tra CBC e CTR?

- CBC e CFB richiedono che E sia invertibile?
- Richiedono che IV sia segreto?
- Quale puo' usare piccoli blocchi di messaggio?
- Sono resistenti alle modifiche? ossia se un intrusore modifica o scambia i blocchi la decifrazione puo' ancora avere senso?
- Il blocco finale di cifrato dipende da tutti i blocchi di messaggio?
- Alice cifra con CBC, si dimentica IV ma dispone di c e K. Puo risalire ad m:
- Caso 1. no
- Caso 2 quasi tutto tranne m0 (X)
- Caso 3 quasi tutto tranne mo e m1
- Caso 4 solo il blocco mn-1

Ricapitolando:

ATTACCO	CONOSCENZE DELL'INTRUSO
con solo testo cifrato	linguaggio e probabilità d'occorrenza
con testo in chiaro noto	coppie di testo in chiaro e cifrato
con testo in chiaro scelto	testi cifrati di testi in chiaro scelti
con testo cifrato scelto	testi in chiaro di testi cifrati scelti

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)

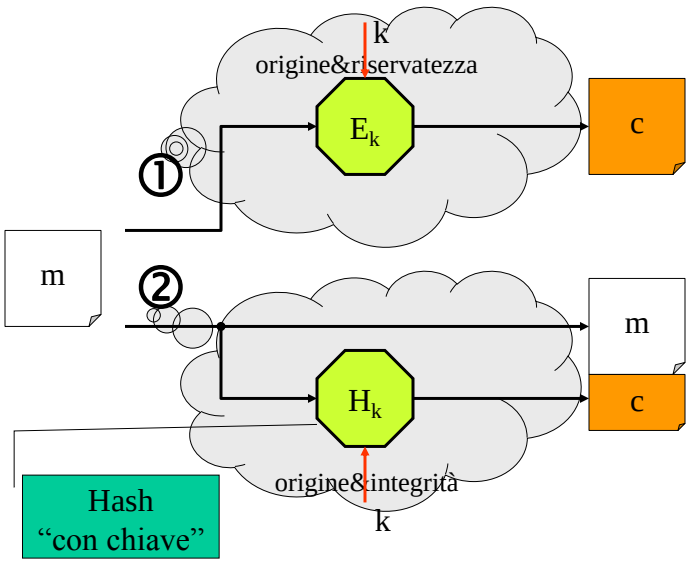
  Vulnerabile sia ad attacchi ciphertext only and chosen plaintext => ok se ECB cifra un solo blocco!
- Cifrario a blocco con CBC: si garantisce protezione contro ciphertext only attacks purche' ben utilizzato!!! (IV casuale e imprevedibile)

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

Integrità ed origine di un messaggio

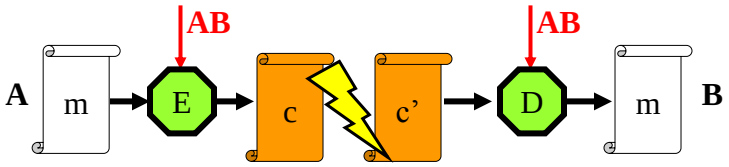
Message Authentication

Ipotesi:Uso della crittografia simmetrica



Se si riesce ad ottenere un testo dotato di significato decifrando con una chiave un messaggio cifrato si puo' pensare che tale testo sia stato inviato dal corrispondente con cui si condivide il segreto su quella chiave

Autenticazione di m con E(m)



SCENARIO FAVOREVOLE

PUNTI CRITICI

Efficienza

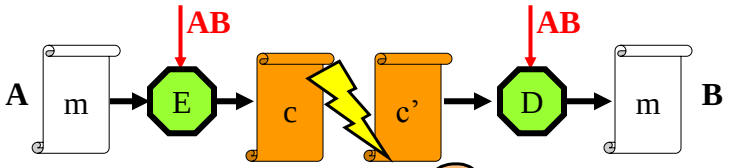
- documento riservato
- un destinatario
- documento non riservato
- più destinatari

Sicurezza

- fiducia reciproca
- controllo del significato
- attacco attivo impossibile
- ripudio e falsificazione
- significato incontrollabile
- attacco attivo possibile

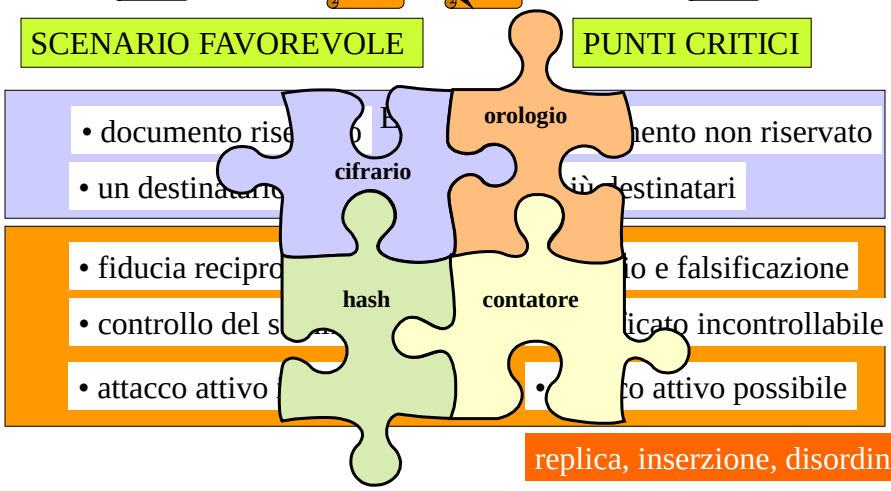
replica, inserzione, disordine

Autenticazione di m con E(m)



SCENARIO FAVOREVOLE

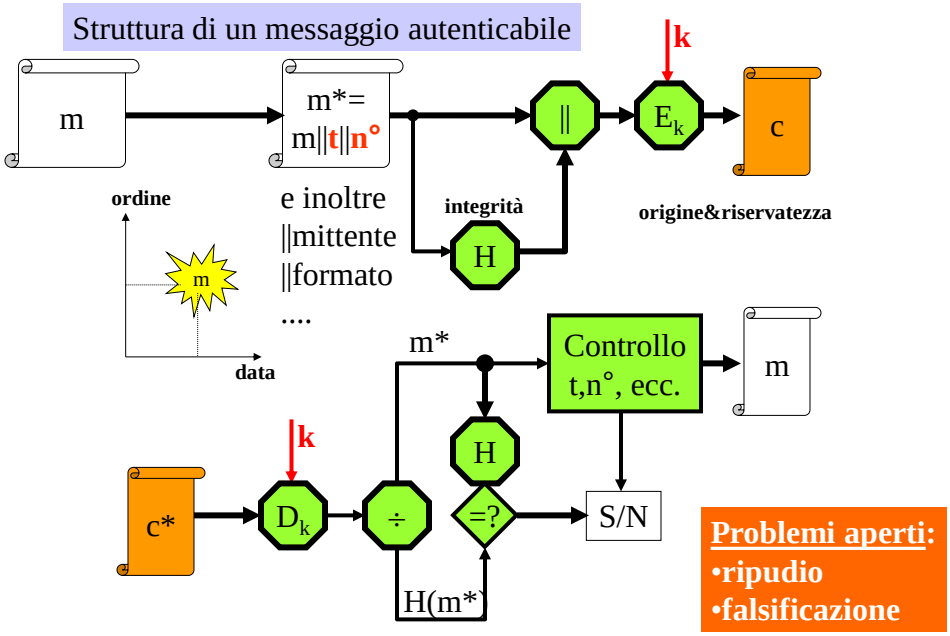
PUNTI CRITICI



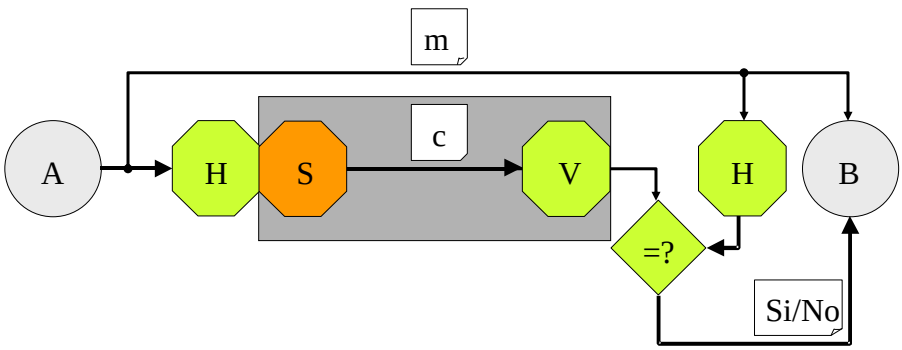
- documento rise… …nento non riservato
- un destinatario… …destinatari
- fiducia recipro… …o e falsificazione
- controllo del s… …icato incontrollabile
- attacco attivo … …co attivo possibile

replica, inserzione, disordine

Autenticazione di m con E(m*||H(m*))



Autenticazione di un messaggio in chiaro 1: firma digitale

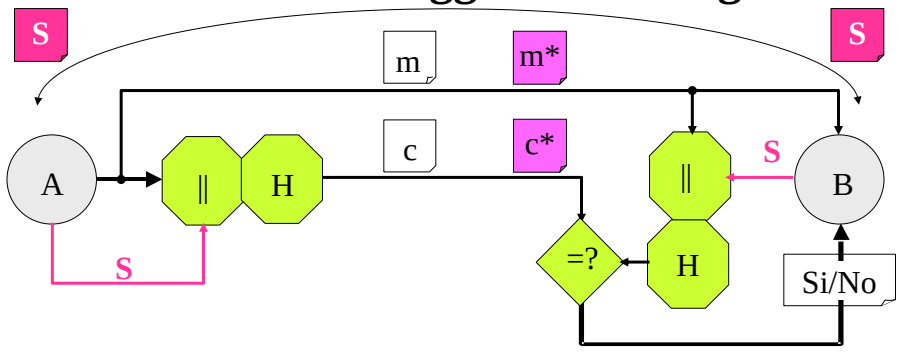


NON ripudiabile

a) canale sicuro

b) canale reso sicuro

Autenticazione di un messaggio in chiaro 2: hash del messaggio e di un segreto



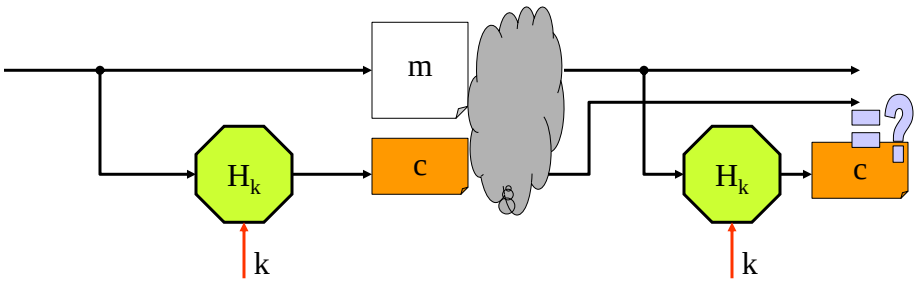
1. **calcola H(m||s)**
2. **invia m e H(m||s)**

ripudiabile

H^{-1} difficile

3. **riceve m*e H*(m||s)**
4. **calcola H(m*||s)**
5. **H*(m||s) =? H(m*||s)**

Integrità ed origine di un testo in chiaro

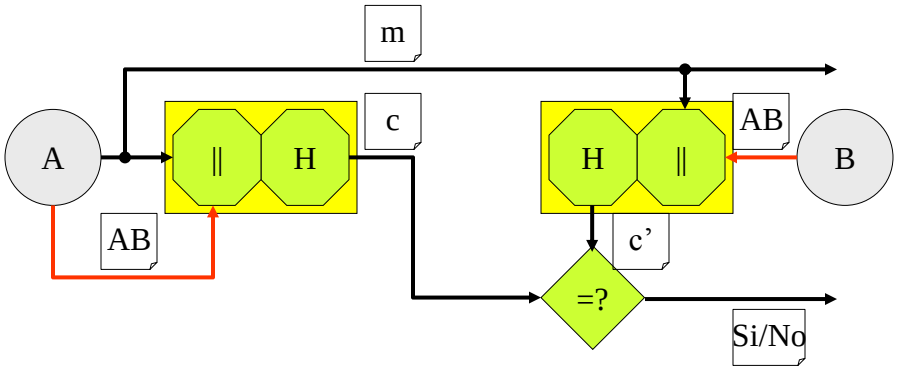


IPOTESI sulla H:
1. impossibilità di inversione
2. impossibilità di individuare collisioni

Problemi aperti:
- **ripudio**
- **falsificazione**

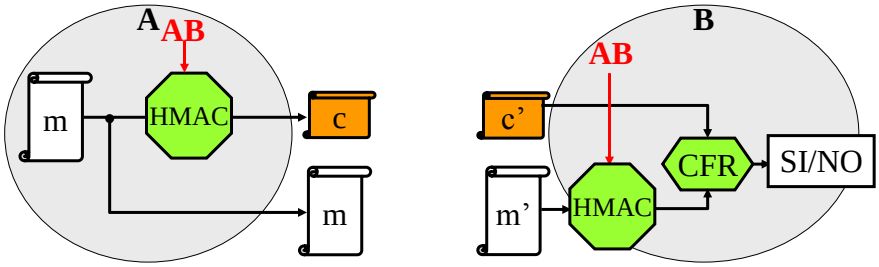
- **MAC** (*hash with CBC encryption*)
- **HMAC** (*hash with key*)

Hash a 2 ingressi o con chiave



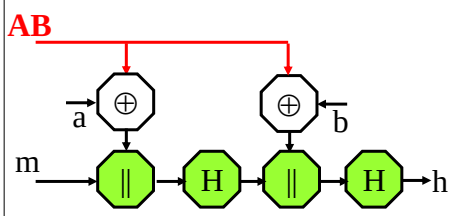
Usando il segreto **AB**, **A** dichiara a **B** di essere l'autore della prova di integrità **c** del messaggio **m**

RFC 2104: HMAC (hash "con chiave")



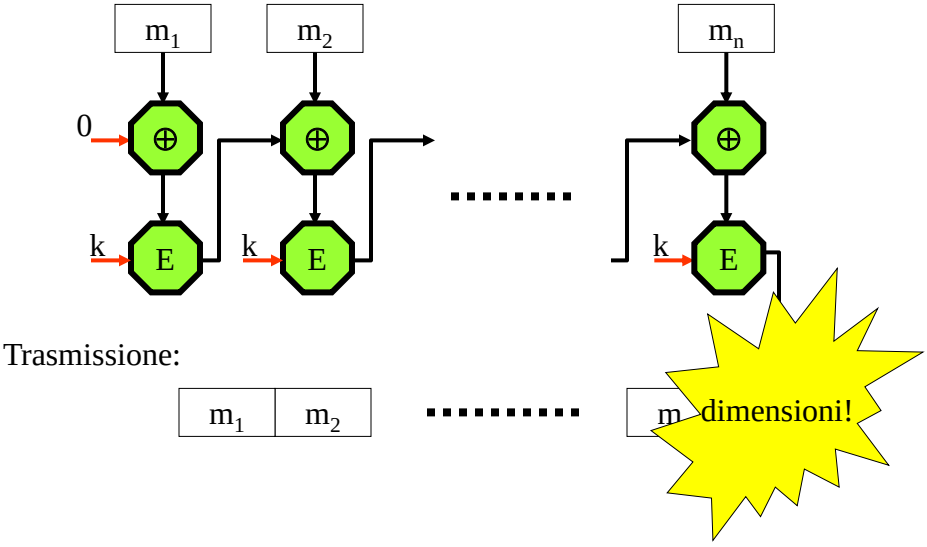
Standard Internet per dare sicurezza al livello **IP**

AB≤64 **byte**, *completato con* 00_H
fino a 512 bit (segreto)
1. k1 = AB⊕a, a=36_H *per 64 volte*
   k2 = AB⊕b, b=$5C_H$ *per 64 volte*
2. h1 = H(k1||m)
3. **h1': h1** *completato con* 00_H
4. h = H(k2||h1') = HMAC(AB, m)



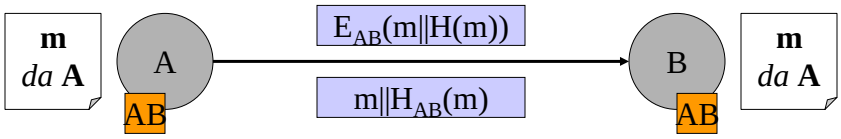
H con compressione iterata

MAC (Message Authentication Code)



Integrità & Origine & Non ripudio

Ripudio e Falsificazione



La condivisione del segreto: problemi di sicurezza
1: A **ripudia m**, affermando che B l'ha alterato o forgiato
2: B **altera** o **forgia m**, affermando che l'ha fatto A

Firma digitale nel contesto della Crittografia simmetrica

Firma digitale

La firma digitale di un documento informatico deve:

1- consentire a **chiunque** di identificare **univocamente** il firmatario,

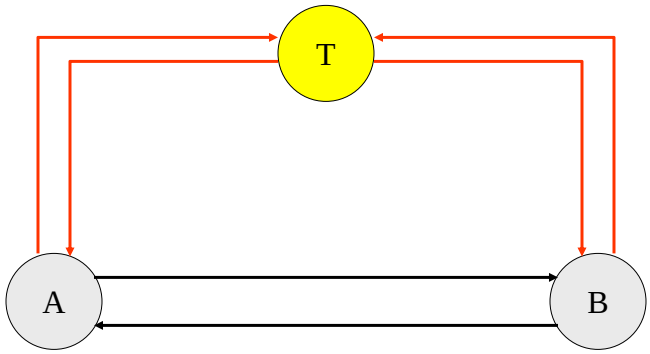
2- non poter essere **imitata** da un impostore,

3- non poter essere **trasportata** da un documento ad un altro,

4- non poter essere **ripudiata** dall'autore,

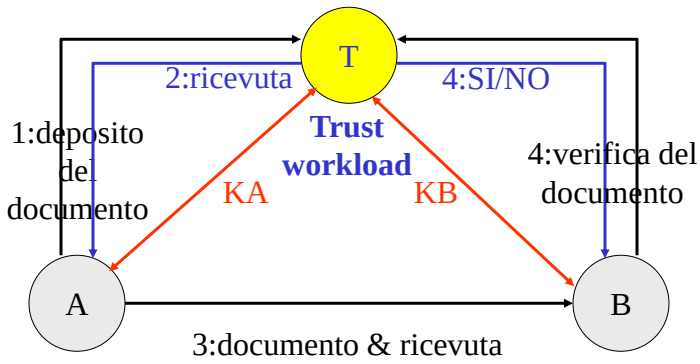
5- rendere **inalterabile** il documento in cui è stata apposta.

Il principio della terza parte fidata

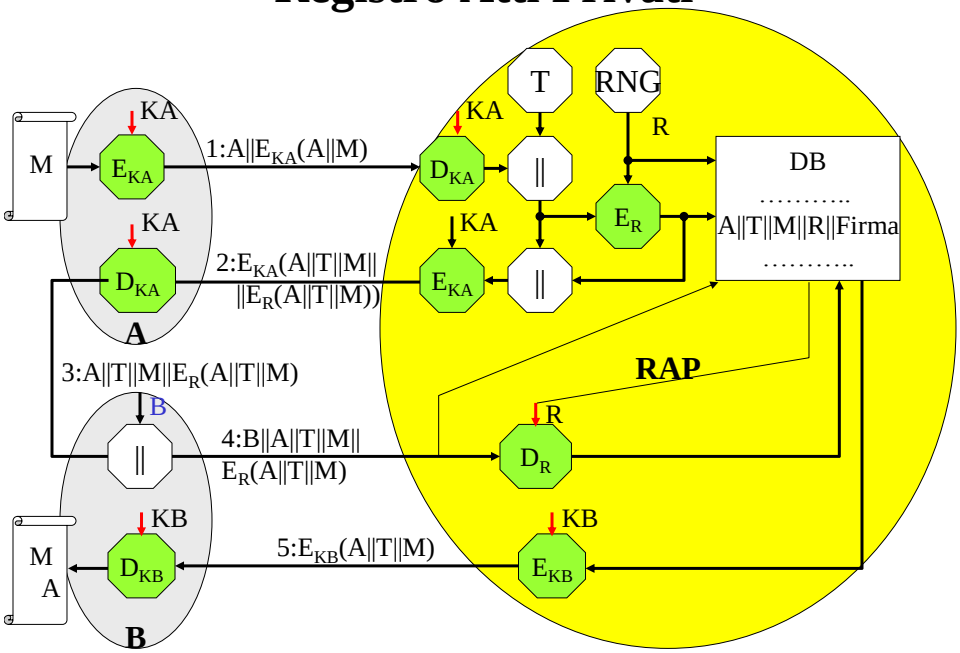


Protocolli resi sicuri dalla partecipazione di una terza parte:
il "notaio" interviene durante lo svolgimento per impedire scorrettezze
il "giudice" interviene al termine per dirimere dispute

Firma digitale
con un Cifrario simmetrico



Registro Atti Privati



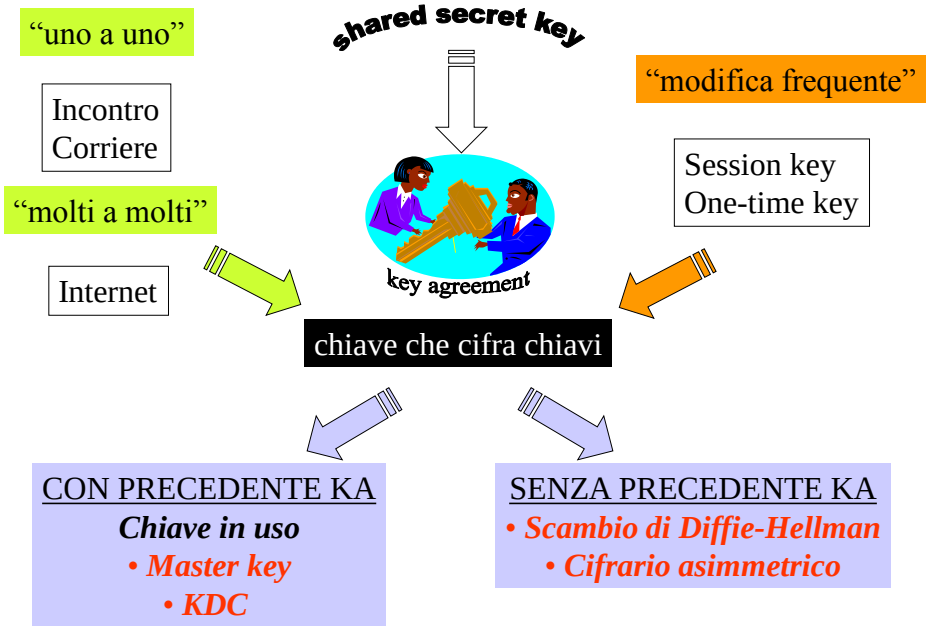
Problemi risolti e nuovi problemi

- Ripudio
- Falsificazione

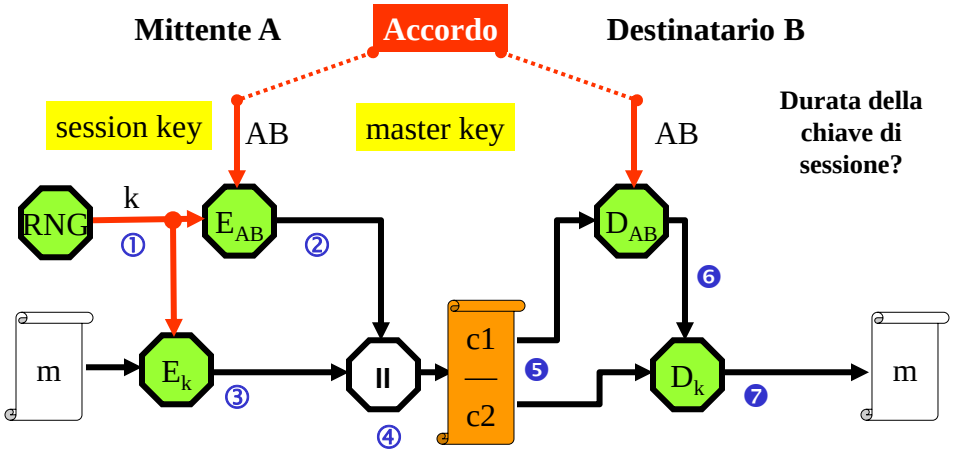
- L'Autorità deve essere sempre **on-line**.
- L'Autorità non deve costituire un **collo di bottiglia**.
- L'Autorità non deve creare **documenti falsi**.
- L'Autorità deve tenere le chiavi in una **memoria sicura**.

Key Management

Accordo sulla chiave segreta



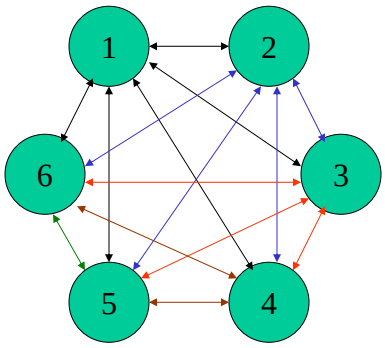
La master key



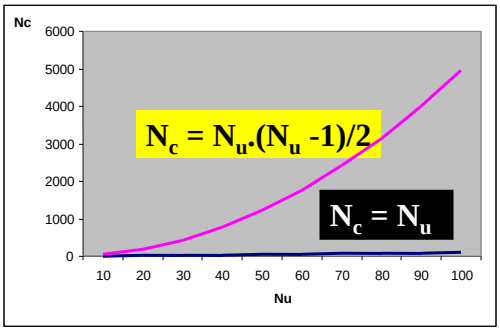
La chiave AB cifra solo le chiavi k e può avere una vita "lunga"
La chiave k cifra messaggi anche "lunghi" ed è usata una volta sola

Numero di chiavi in circolazione

Comunità di utenti



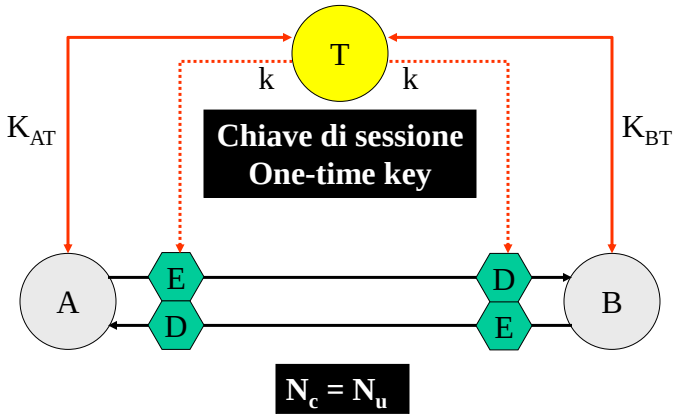
Non è scalabile!

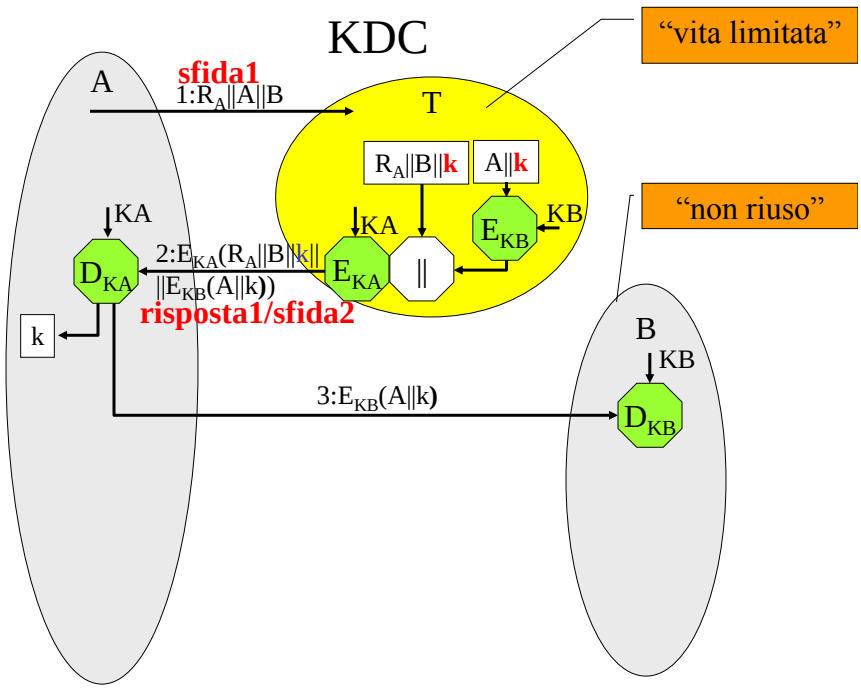


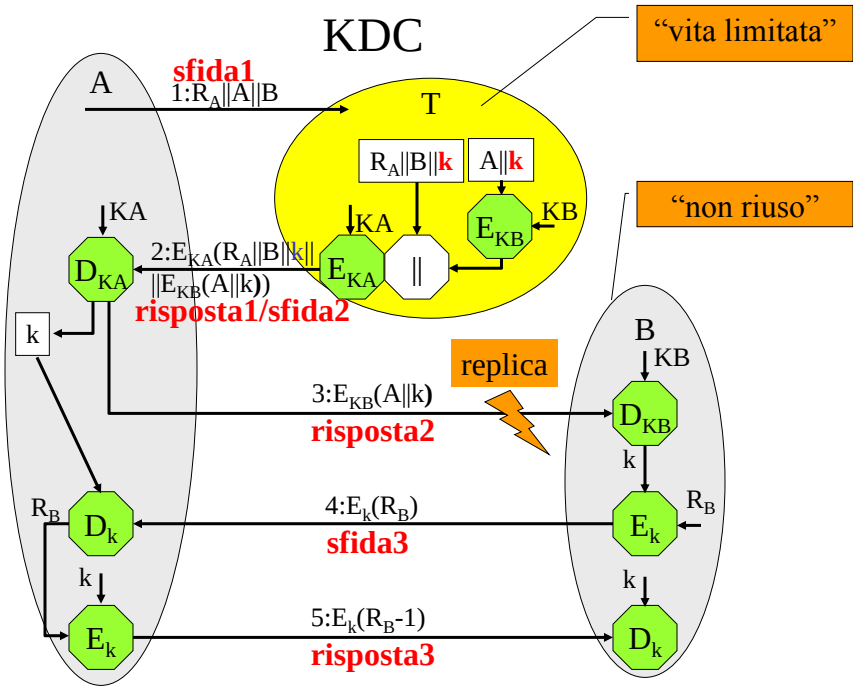
Obiettivo da perseguire: "una chiave per utente"

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° max di utenti)
- Memoria sicura
- Ente degno di fiducia

KryptoKnight,
Kerberos,
Distributed Computing Environment,
Windows 2000

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

$g \bmod p$, $g^2 \bmod p$,.., $g^p \bmod p$

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

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

$b = g^i \pmod p$ dove $0 <= i <= (p-1)$

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

$Y_A = g^{XA} \bmod p$ e $Y_B = g^{XB} \bmod p$

3. Calcolo della chiave del Cifrario simmetrico

$K_A = Y_B^{XA} \bmod p = (g^{XB})^{XA} \bmod p$

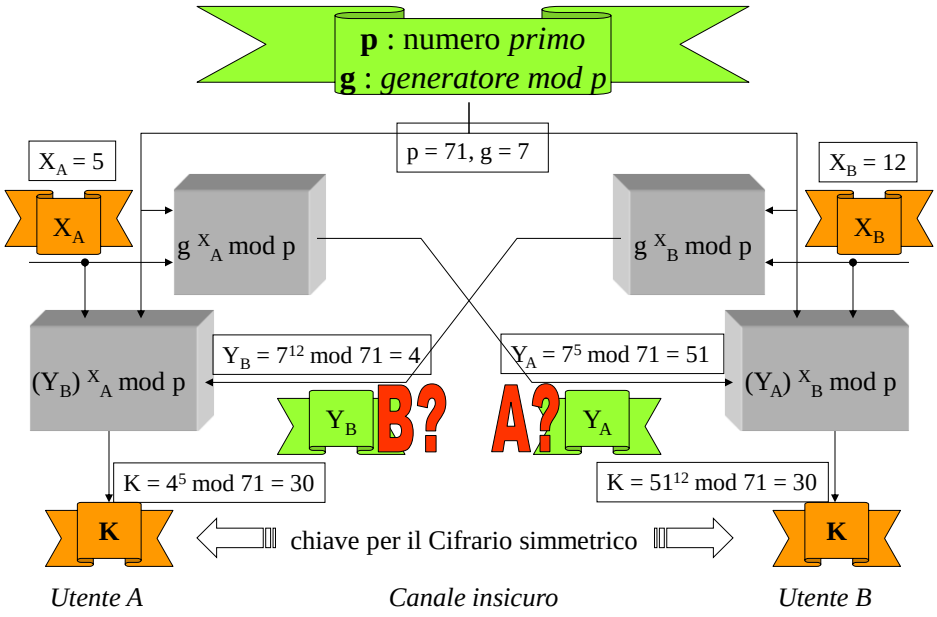
$K_B = Y_A^{XB} \bmod p = (g^{XA})^{XB} \bmod p$

$K_A = K_B$

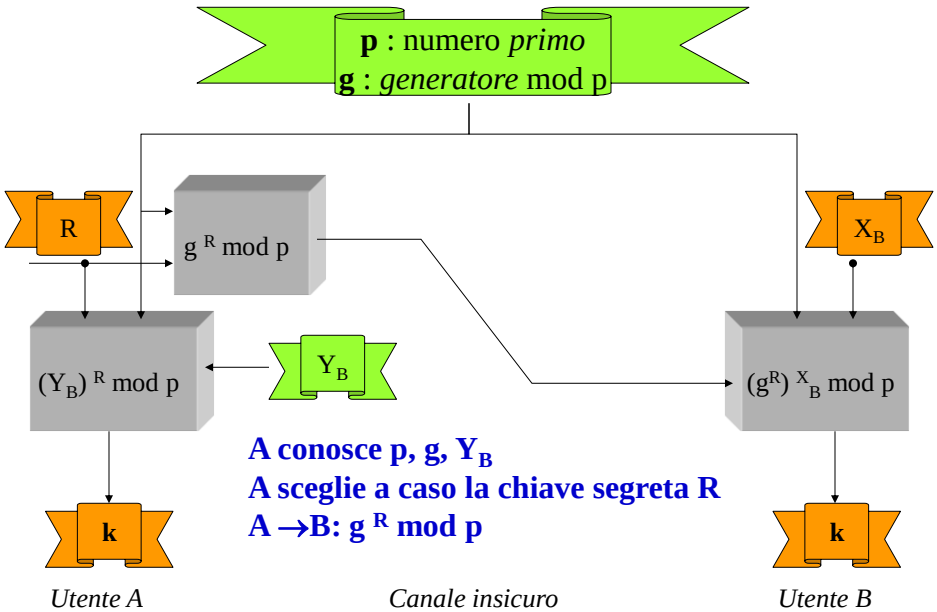
La dimensione è grande (quella di p): occorre scegliere k

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

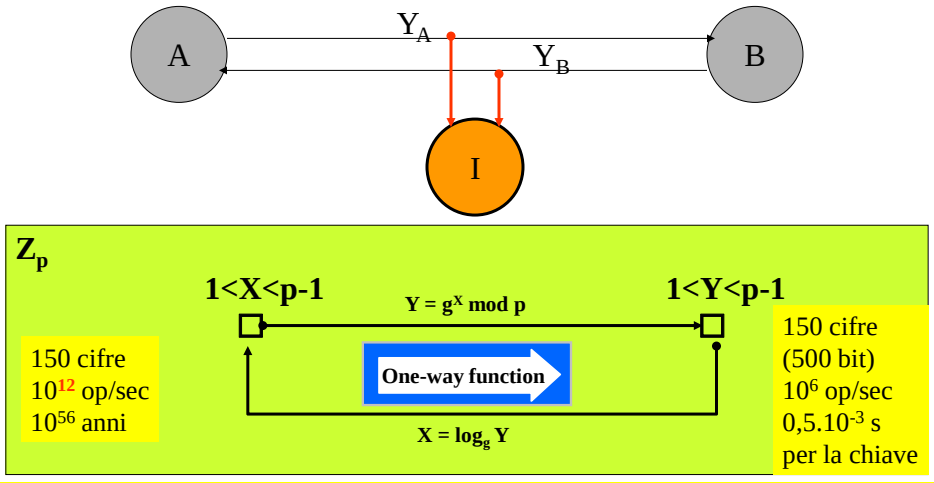
Lo scambio di Diffie-Hellman



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** $\in$ **Z^*_p**, trovare l'intero **x** **$1 \le x \le p-1$**, tale che **$g^x \bmod p = c$**"

Z^*_p insieme di interi non negativi non nulli minori di p