Crittografia simmetrica

Alessandro Armando

Laboratorio di sicurezza informatica (CSec)
DIBRIS, Università di Genova

Sicurezza del computer

Contorno

Cifrari a blocchi e a flusso

- i cifrari a blocchi elaborano i messaggi in blocchi, ognuno dei quali viene quindi decrittografato
- come una sostituzione su un alfabeto molto grande
- 64 bit o più
- i cifrari a flusso elaborano i messaggi un bit o un byte alla volta quando la crittografia/
- decrittografia di molti codici correnti sono cifrari a blocchi
- gamma di applicazioni più ampia

Contorno

Cifrario a blocchi ideale

- I cifrari a blocchi sembrano una sostituzione estremamente ampia
- Avrebbe bisogno di un tavolo da 2^n voci per a n-bit blocco e quindi una dimensione "chiave" di $n \times 2^n$
- Un totale di $2^n!$ le trasformazioni sono possibili
- La cifratura ideale come informazione statistica del testo in chiaro è persa, ma irrealizzabile.

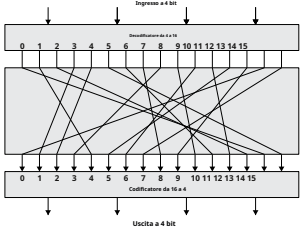


Figura 3.1 Generale *n*-po-*n*-bit Block Sostituzione (mostrato con $n = 4$)

Tabella 3.1 Tabelle di crittografia e decrittografia per la cifratura di sostituzione di Figura 3.4

Testo in chiaro	testo cifrato
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
0101	1111
0110	1011
0111	1000
1000	0011
1001	1010
1010	0110
1011	1100
1100	0101
1101	1001
1110	0000
1111	0111

testo cifrato	Testo in chiaro
0000	1110
0001	0011
0010	0100
0011	1000
0100	0001
0101	1100
0110	1010
0111	1111
1000	0111
1001	1101
1010	1001
1011	0110
1100	1011
1101	0010
1110	0000
1111	0101

Struttura cifrata di Feistel

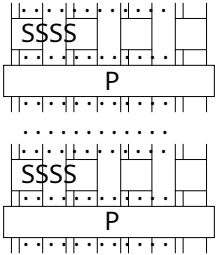
- **Idea:** approssimare il cifrario a blocchi ideale utilizzando il concetto di a *cifratura del prodotto* , cioè una combinazione di semplici cifrari in modo tale che il risultato finale o il prodotto sia crittograficamente più forte di qualsiasi cifrario componente.
- In pratica: sviluppare un cifrario a blocchi con una lunghezza chiave di K bit e una lunghezza di blocco di n bit, consentendo un totale di 2^K possibili trasformazioni, piuttosto che il $2^n!$ trasformazioni disponibili con il cifrario a blocchi ideale
- La maggior parte dei cifrari a blocchi simmetrici si basa su questa idea
- Necessario poiché deve essere in grado di decifrare il testo cifrato per recuperare i messaggi in modo efficiente

Claude Shannon e i cifrari a sostituzione-permutazione

- Claude Shannon ha introdotto l'idea delle reti di sostituzione-permutazione (SP) nel documento del 1949
- forma base dei moderni cifrari a blocchi
- Le reti SP si basano sulle due operazioni crittografiche primitive viste in precedenza:
  - sostituzione (S-box)

    confuso bit di ingresso.
  - permutazione (P-box)

    diffondere bit attraverso gli ingressi della S-box
- fornire confusione e diffusione del messaggio e della chiave



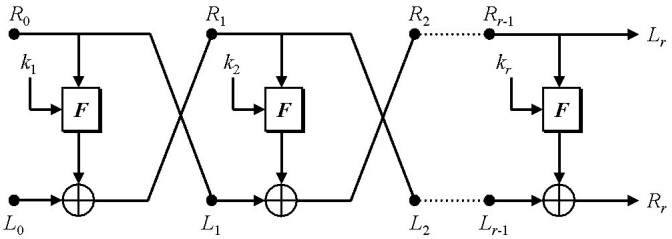
Confusione e diffusione

- il cifrario deve oscurare completamente le proprietà statistiche del messaggio originale
- un one-time pad fa questo
- più praticamente Shannon ha suggerito di combinare elementi S & P per ottenere:
  - *diffusione* dissipa la struttura statistica del testo in chiaro sulla maggior parte del testo cifrato
  - *confusione* rende la relazione tra testo cifrato e chiave il più complessa possibile

Struttura cifrata di Feistel

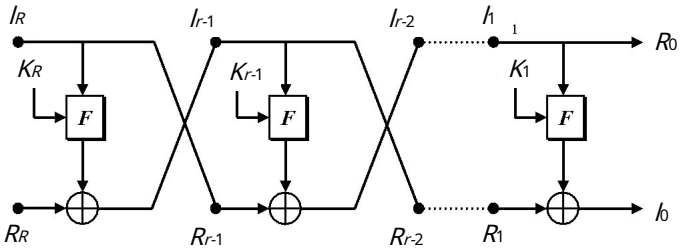
- Horst Feistel ha ideato il cifrario feistel
  - basato sul concetto di cifrario prodotto invertibile
- Partiziona il blocco di input in due metà
  - elaborare attraverso più round che eseguono una
  - sostituzione sulla metà dei dati a sinistra in base alla funzione
  - round della metà destra e della sottochiave quindi hanno
  - metà di scambio di permutazione
- implementa il concetto di rete SP di Shannon

Struttura cifrata di Feistel



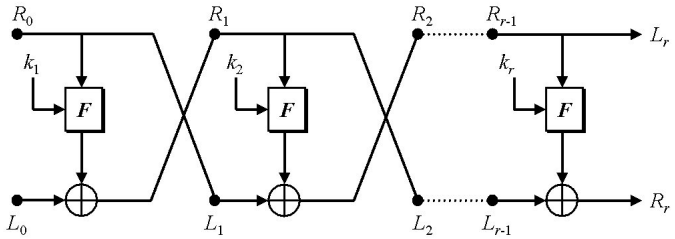
- La funzione rotonda *F* può essere una rete SP o qualsiasi cifrario (non necessariamente invertibile).
- *La crittografia e la decrittografia sono strutturalmente identiche* , sebbene le sottochiavi utilizzate durante la crittografia ad ogni round vengano prese in ordine inverso durante la decrittografia.
- Più precisamente, l'input nell'algoritmo di decrittazione è la coppia (R_R , L_R) al posto della coppia (l_0, R_0), e il *io* -la sottochiave è K_{r-i+1}, non K_{io} . Ciò significa che otteniamo(R_{r-i} , L_{r-i}) invece di (l_{io} , R_{io}) dopo il *io* -esimo giro.

Struttura cifrata di Feistel



- La funzione rotonda F può essere una rete SP o qualsiasi cifrario (non necessariamente invertibile).
- *La crittografia e la decrittografia sono strutturalmente identiche*, sebbene le sottochiavi utilizzate durante la crittografia ad ogni round vengano prese in ordine inverso durante la decrittografia.
- Più precisamente, l'input nell'algoritmo di decrittazione è la coppia (R_R, L_R) al posto della coppia (I_0, R_0), e il io-la sottochiave è K_{r-i+1}, non K_{io}. Ciò significa che otteniamo(R_{r-i}, L_{r-i}) invece di (I_{io}, R_{io}) dopo il io-esimo giro.

Struttura del cifrario di Feistel: decrittazione



$$l_R = R_{r-1} \quad (1)$$

$$R_R = l_{r-1} \mathbin{?} F(K_R, R_{r-1}) \quad (2)$$

Fatto: $R_R \mathbin{?} l_{r-1} = F(K_R, R_{r-1})$

Prova: Mettendo entrambi i lati di (2) in $?$ insieme a l_{r-1} noi abbiamo:

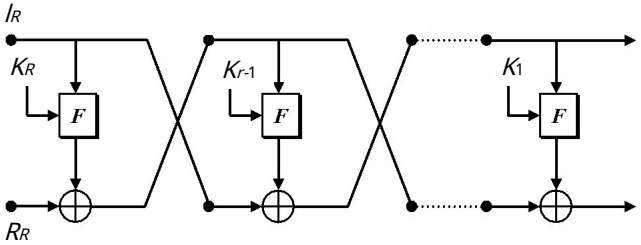
$R_R \mathbin{?} l_{r-1} = l_{r-1} \mathbin{?} F(K_R, R_{r-1}) \mathbin{?} l_{r-1}$

Questa equazione semplifica il fatto di cui sopra sfruttando le proprietà di $?$.

Proprietà di $?$

$$X \mathbin{?} X = 0$$
$$X \mathbin{?} 0 = X$$
$$X \mathbin{?} sì = sì \mathbin{?} X$$
$$X \mathbin{?} (sì \mathbin{?} z) = (X \mathbin{?} sì) \mathbin{?} z$$

Struttura del cifrario di Feistel: decrittazione



$$I_R = R_{r-1} \quad (1)$$

$$R_R = I_{r-1} \oplus F(K_R, R_{r-1}) \quad (2)$$

Fatto: $R_R \oplus I_{r-1} = F(K_R, R_{r-1})$

Prova: Mettendo entrambi i lati di (2) in $\oplus$ insieme a I_{r-1} noi abbiamo:

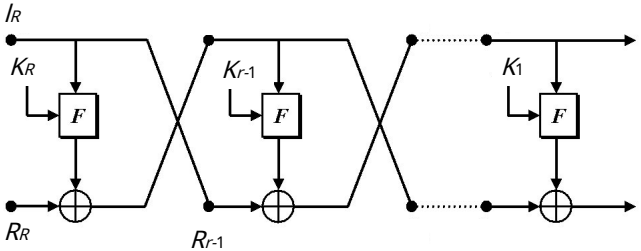
$R_R \oplus I_{r-1} = I_{r-1} \oplus F(K_R, R_{r-1}) \oplus I_{r-1}$

Questa equazione semplifica il fatto di cui sopra sfruttando le proprietà di $\oplus$.

Proprietà di $\oplus$

$$X \oplus X = 0$$
$$X \oplus 0 = X$$
$$X \oplus sì = sì \oplus X$$
$$X \oplus (sì \oplus z) = (X \oplus sì) \oplus z$$

Struttura del cifrario di Feistel: decrittazione



$$l_R = R_{r-1} \quad (1)$$

$$R_R = l_{r-1} \oplus F(K_R, R_{r-1}) \quad (2)$$

Fatto: $R_R \oplus l_{r-1} = F(K_R, R_{r-1})$

Prova: Mettendo entrambi i lati di (2) in $\oplus$ insieme a l_{r-1} noi abbiamo:

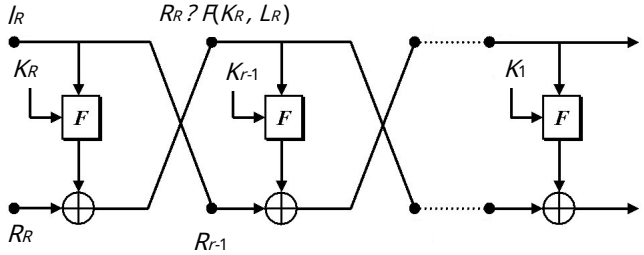
$R_R \oplus l_{r-1} = l_{r-1} \oplus F(K_R, R_{r-1}) \oplus l_{r-1}$

Questa equazione semplifica il fatto di cui sopra sfruttando le proprietà di $\oplus$.

Proprietà di $\oplus$

$$X \oplus X = 0$$
$$X \oplus 0 = X$$
$$X \oplus sì = sì \oplus X$$
$$X \oplus (sì \oplus z) = (X \oplus sì) \oplus z$$

Struttura del cifrario di Feistel: decrittazione



$$I_R = R_{r-1} \quad (1)$$

$$R_R = I_{r-1} ? F(K_R, R_{r-1}) \quad (2)$$

Fatto: $R_R ? I_{r-1} = F(K_R, R_{r-1})$

Prova: Mettendo entrambi i lati di (2) in $?$ insieme a I_{r-1} noi abbiamo:

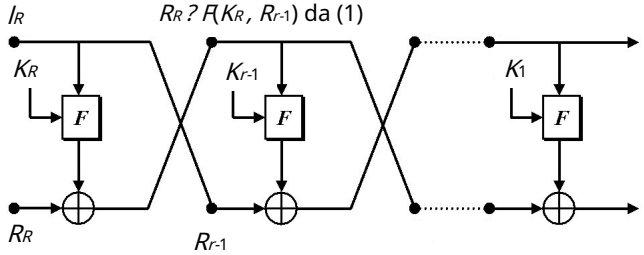
$R_R ? I_{r-1} = I_{r-1} ? F(K_R, R_{r-1}) ? I_{r-1}$

Questa equazione semplifica il fatto di cui sopra sfruttando le proprietà di $?$.

Proprietà di $?$

$$X ? X = 0$$
$$X ? 0 = X$$
$$X ? sì = sì ? X$$
$$X ? (sì ? z) = (X ? sì) ? z$$

Struttura del cifrario di Feistel: decrittazione



$$I_R = R_{r-1} \quad (1)$$

$$R_R = I_{r-1} \,?\, F(K_R, R_{r-1}) \quad (2)$$

Fatto: $R_R \,?\, I_{r-1} = F(K_R, R_{r-1})$

Prova: Mettendo entrambi i lati di (2) in $?$ insieme a I_{r-1} noi abbiamo:

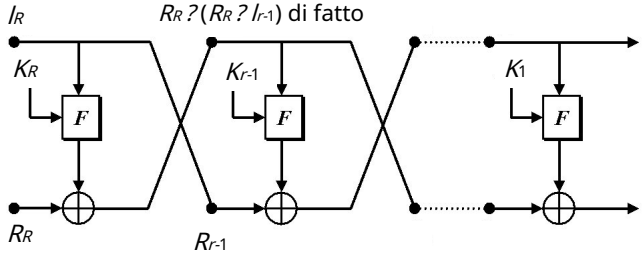
$R_R \,?\, I_{r-1} = I_{r-1} \,?\, F(K_R, R_{r-1}) \,?\, I_{r-1}$

Questa equazione semplifica il fatto di cui sopra sfruttando le proprietà di $?$.

Proprietà di $?$

$$X \,?\, X = 0$$
$$X \,?\, 0 = X$$
$$X \,?\, sì = sì \,?\, X$$
$$X \,?\, (sì \,?\, z) = (X \,?\, sì) \,?\, z$$

Struttura del cifrario di Feistel: decrittazione



$$I_R = R_{r-1} \quad (1)$$

$$R_R = I_{r-1} ? F(K_R, R_{r-1}) \quad (2)$$

Fatto: $R_R ? I_{r-1} = F(K_R, R_{r-1})$

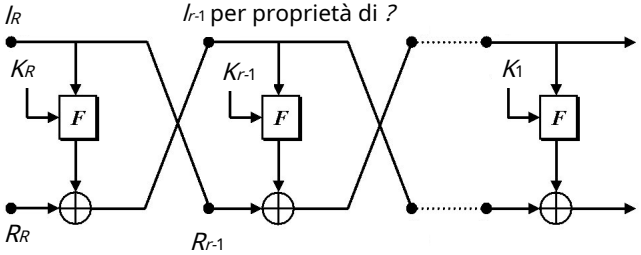
Prova: Mettendo entrambi i lati di (2) in $?$ insieme a I_{r-1} noi abbiamo:

$R_R ? I_{r-1} = I_{r-1} ? F(K_R, R_{r-1}) ? I_{r-1}$

Questa equazione semplifica il fatto di cui sopra sfruttando le proprietà di $?$.

Proprietà di $?$

$$X ? X = 0$$
$$X ? 0 = X$$
$$X ? sì = sì ? X$$
$$X ? (sì ? z) = (X ? sì) ? z$$



$$l_R = R_{r-1} \quad (1)$$

$$R_R = l_{r-1} \,?\, F(K_R, R_{r-1}) \quad (2)$$

Fatto: $R_R \,?\, l_{r-1} = F(K_R, R_{r-1})$

Prova: Mettendo entrambi i lati di (2) in $?$ insieme a l_{r-1} noi abbiamo:

$R_R \,?\, l_{r-1} = l_{r-1} \,?\, F(K_R, R_{r-1}) \,?\, l_{r-1}$

Questa equazione semplifica il fatto di cui sopra sfruttando le proprietà di $?$.

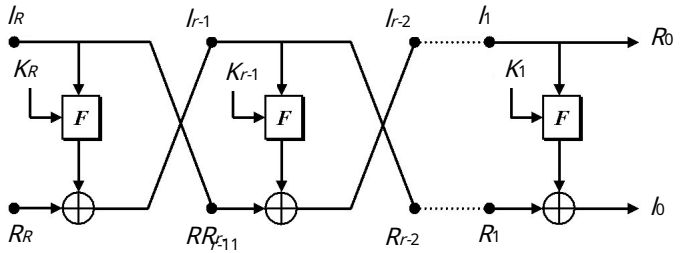
Proprietà di $?$

$$X \,?\, X = 0$$

$$X \,?\, 0 = X$$

$$X \,?\, sì = sì \,?\, X$$

$$X \,?\, (sì \,?\, z) = (X \,?\, sì) \,?\, z$$



$$I_R = R_{r-1} \quad (1)$$

$$R_R = I_{r-1} \mathbin{?} F(K_R, R_{r-1}) \quad (2)$$

Fatto: $R_R \mathbin{?} I_{r-1} = F(K_R, R_{r-1})$

Prova: Mettendo entrambi i lati di (2) in $?$ insieme a I_{r-1} noi abbiamo:

$R_R \mathbin{?} I_{r-1} = I_{r-1} \mathbin{?} F(K_R, R_{r-1}) \mathbin{?} I_{r-1}$

Questa equazione semplifica il fatto di cui sopra sfruttando le proprietà di $?$.

Proprietà di $?$

$$X \mathbin{?} X = 0$$

$$X \mathbin{?} 0 = X$$

$$X \mathbin{?} sì = sì \mathbin{?} X$$

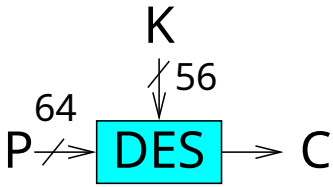
$$X \mathbin{?} (sì \mathbin{?} z) = (X \mathbin{?} sì) \mathbin{?} z$$

Elementi di design di cifratura Feistel

- misura del blocco
- dimensione della chiave
- numero di round
- funzione rotonda dell'algoritmo di
- generazione della sottochiave
- software veloce en/decrittazione
- facilità di analisi

DES

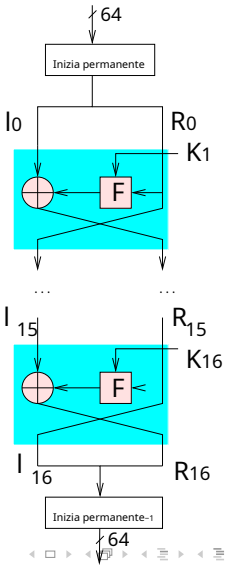
- **Standard di crittografia dei dati**, 1993
  Basato su invenzione IBM, LUCIFER



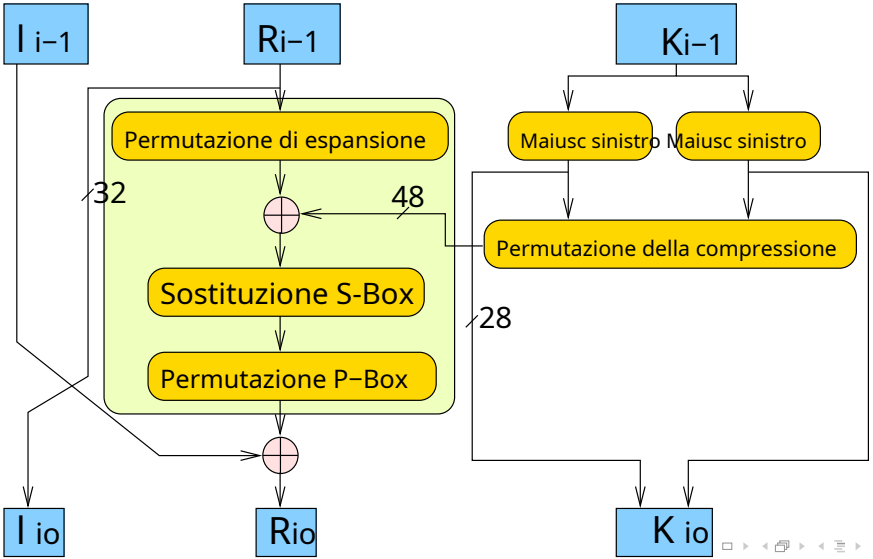
- Cifratura a blocchi, crittografia di blocchi a 64 bit
  Utilizza chiavi a 56 bit
  Espresso come numeri a 64 bit (controllo di parità a 8 bit)
- Primo standard crittografico.
  - 1977 Standard federale degli Stati Uniti (US Bureau of Standards)
  - 1981 ANSI standard del settore privato
- Molto utilizzato nelle applicazioni bancarie.
  Estensioni come triple-DES utilizzate per superare brevi lunghezze di chiave.

DES – forma generale

- Cifrario Feistel a 16 round + schedulatore chiavi.
- L'algoritmo di pianificazione delle chiavi deriva le sottochiavi K_i dalla chiave originale K.
- Dentro al permutazione all'inizio e permutazione inversa alla fine.
- F consiste di due permutazioni e una sostituzione s-box.



DES – 1 round


I i−1
Ri−1
Ki−1
Permutazione di espansione
Maiusc sinistro
Maiusc sinistro
32
48
Permutazione della compressione
Sostituzione S-Box
28
Permutazione P−Box
I io
Rio
K io

Effetto valanga

- proprietà chiave desiderabile dell'algoritmo di crittografia
- dove la modifica di un bit di input o chiave determina la modifica di circa la metà dei bit di output
- DES mostra un forte effetto valanga

Forza di DES – Dimensione chiave

- Le chiavi a 56 bit hanno $2^{56} = 7.2 \times 10^{16}$ valori
- la ricerca della forza bruta sembra difficile che i recenti
- progressi abbiano dimostrato che è possibile
  - nel 1997 su Internet in pochi mesi
  - nel 1998 su h/w dedicato (EFF) in pochi giorni nel
  - 1999 sopra combinato in 22 ore!
- deve ancora essere in grado di riconoscere il testo in
- chiaro deve ora considerare alternative a DES

Forza del DES – Attacchi analitici

- ora hanno diversi attacchi analitici su DES questi
- utilizzano una struttura profonda del cifrario
  - raccogliendo informazioni sulla crittografia può eventualmente
  - recuperare alcuni/tutti i bit della sottochiave se necessario,
  - quindi cercare in modo esauriente il resto
- generalmente questi sono attacchi statistici
- includono
  - crittoanalisi differenziale
  - crittoanalisi lineare
  - attacchi chiave correlati

Forza del DES – Attacchi a tempo

- attacca l'effettiva implementazione della cifratura
- utilizzare la conoscenza delle conseguenze dell'implementazione per ricavare
- informazioni su alcuni/tutti i bit della sottochiave
- utilizzare specificamente il fatto che i calcoli possono richiedere tempi variabili a seconda del valore degli input ad esso
- particolarmente problematico sulle smartcard

Sicurezza di DES

Le persone hanno a lungo messo in dubbio la sicurezza di DES. Ci sono state molte speculazioni sulla lunghezza della chiave, sul numero di iterazioni e sul design delle S-box. Le S-box erano particolarmente misteriose: tutte quelle costanti, senza alcuna ragione apparente sul perché oa cosa servano. Sebbene IBM sostenesse che i meccanismi interni fossero il risultato di 17 anni-uomo di crittoanalisi intensiva, alcune persone temevano che la NSA avesse incorporato una botola nell'algoritmo in modo da avere un mezzo facile per decifrare i messaggi.

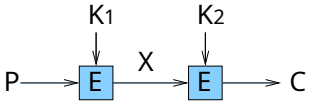
– Bruce Schneier, *Crittografia applicata* p278.

La National Security Agency ha anche fornito consulenza tecnica a IBM. E Konheim è stato citato come dicendo "abbiamo spedito le S-box a Washington. Sono tornati ed erano tutti diversi. Abbiamo fatto i nostri test e li hanno superati". La gente ha indicato questo come prova che la NSA ha messo una botola nel DES.

– Bruce Schneier, *Crittografia applicata* p279.

Aumento della sicurezza DES – Doppio DES

- **Idea**: esegue due crittografie



Equivalente a 112 chiavi bit?

- **attacco**: Incontro nel mezzo
  - Per $C = E_K(E_K(P))$ permettere $X = E_K(P) = D_K(C)$.
  - Dato noto P e C crittografare P per tutti 2^{56} possibile K_1. Memorizza
  - nella tabella, ordinato per X.
  - decifrare C con tutto 2^{56} possibile K_2 e cercare una corrispondenza.
  - Ogni hit deve essere convalidato rispetto a una coppia aggiuntiva di testo normale/cifrato. (Per un dato testo in chiaro P il numero medio di diverse chiavi a 112 bit che produrranno un dato testo cifrato C è $2^{112}/2^{64} = 2^{48}$.)
  - Un noto attacco di testo in chiaro contro il doppio DES (chiavi a 112 bit) riesce con uno sforzo dell'ordine di 2^{56} operazioni.

Aumento della sicurezza DES – Triple DES

- Utilizza tre fasi di crittografia invece di due.
- Notare che K_1 è usato due volte *?* Chiave a 112 bit. La compatibilità
- viene mantenuta con il DES standard se $K_2 = K_1$.
- Nessun attacco pratico noto
  ? ricerca a forza bruta con 2^{112} operazioni.
- Per ulteriore sicurezza viene utilizzato 3DES a tre chiavi (ad es. in PGP e S/MIME):

$$C = E_K(D_{K_2}(E_{K_1}(P)))$$

- Compatibilità con le versioni precedenti di DES con $K_3 = K_2$ e $K_1 = K_2$.

Standard di crittografia avanzata (AES)

- Selezionato come standard NIST nel 2001 (su 15 cifrari concorrenti)
- Basato sul cifrario Rijndael sviluppato dai crittografi belgi, Joan Daemen e Vincent Rijmen
- Rijndael è una famiglia di cifrari con chiavi e blocchi di diverse dimensioni.
- Per AES, il NIST ha selezionato tre membri della famiglia Rijndael:
  - dimensione del blocco di 128 bit,
  - tre diverse lunghezze di chiave: 128, 192 e 256 bit.
- Veloce sia nel software che nell'hardware.
- AES è ora utilizzato in tutto il mondo e sostituisce DES

Standard di crittografia avanzata (continua)

- AES si basa su una rete di sostituzione-permutazione A
- differenza di DES, AES non utilizza una rete Feistel.
- Mentre AES ha una dimensione di blocco fissa di 128 bit e una dimensione di chiave di 128, 192 o 256 bit, Rijndael funziona con dimensioni di blocco e chiave che possono essere multipli di 32 bit, sia con un minimo di 128 che un massimo di 256 bit.
- La dimensione della chiave utilizzata per un cifrario AES specifica il numero di ripetizioni dei cicli di trasformazione che convertono il testo in chiaro in testo cifrato:
  - 10 cicli di ripetizione per chiavi a 128 bit.
  - 12 cicli di ripetizione per chiavi a 192 bit.
  - 14 cicli di ripetizione per chiavi a 256 bit.

Contorno

Modalità di funzionamento – Codebook elettronico

- Come viene utilizzato un cifrario a blocchi quando i messaggi superano la larghezza di blocco? Diverso possibilemodalità di funzionamento. Consideriamo (solo) due!
- Il più semplice è Modalità registro elettronico

  Messaggio suddiviso in m blocchi. Ciascuno crittografato individualmente.
- Limitazioni:

  Fuga di informazioni: blocchi di testo cifrato identici si associano a blocchi di testo in chiaro identici

  Integrità limitata: la decrittazione non indica se i blocchi di testo cifrato sono stati modificato, cancellato o duplicato.

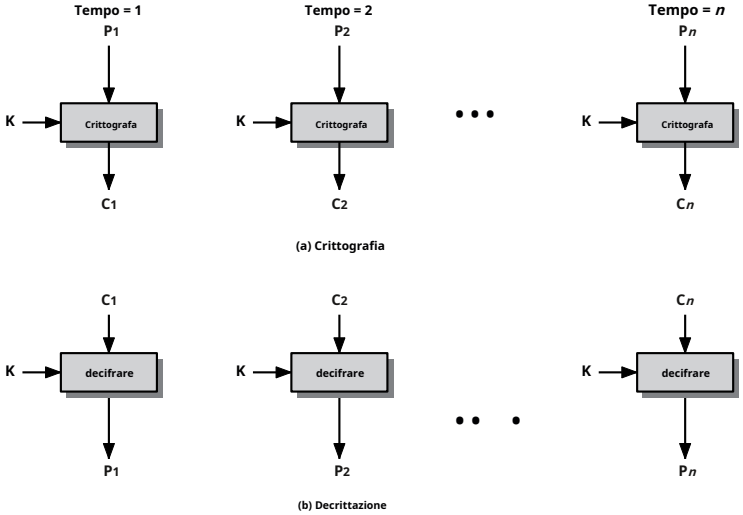


Figura 6.3 Modalità Electronic Codebook (ECB)

- L'input di cifratura è XOR del blocco di testo in chiaro con testo cifrato precedente.
- Per $C_0 = IV$ (un vettore di inizializzazione), $io = 1..m$:

  Crittografia: $C_{io} = E_K(P_{io} \mathbin{?} C_{io-1}) P$

  Decrittografia: $_{io} = C_{io-1} \mathbin{?} D_K(C_{io})$

- Correttezza?

$$\begin{aligned} P_{io} &= C_{io-1} \mathbin{?} D_K(C_{io}) \\ &= C_{io-1} \mathbin{?} D_K(E_K(P_{io} \mathbin{?} C_{io-1})) \\ &= C_{io-1} \mathbin{?} (P_{io} \mathbin{?} C_{io-1}) \\ &= P_{io} \end{aligned}$$

- Proprietà
  - Blocchi di testo in chiaro identici mappati su testo cifrato diverso *Dipendenze di concatenamento:* C_j
  - dipende da tutto il testo in chiaro precedente. *Auto-sincronizzazione:* se si verifica un errore (bit
  - modificati, blocchi persi) in C_j ma no C_{j+1}, poi C_{j+2} è correttamente decifrato.

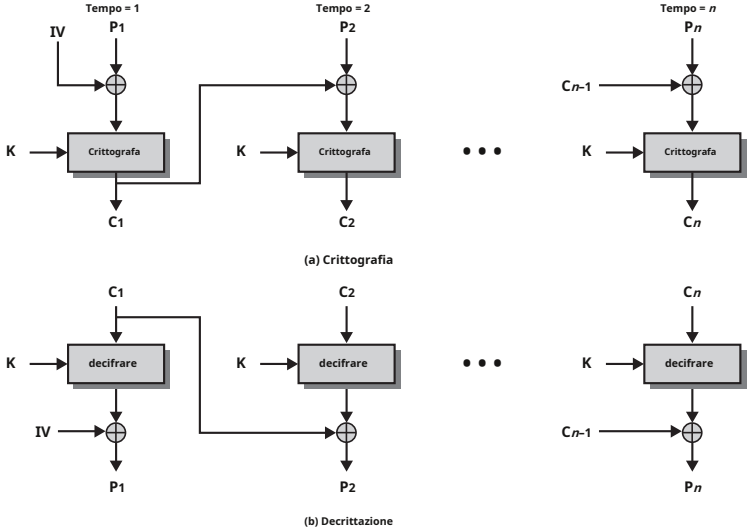


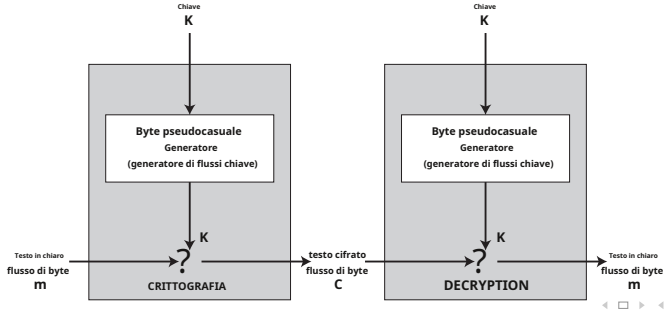
Figura 6.4 Modalità Cipher Block Chaining (CBC)

Cifrari a flusso

- Stessa idea del cifrario di Vernam ma usa un generatore pseudocasuale (al posto di un generatore veramente casuale), cioè

$$E_{K_1 \cdots K_n}(m_1 \cdots m_n) = (m_1 \,?\, K_1) \cdots (m_n \,?\, K_n)\,(C$$
$$D_{K_1 \cdots K_n}(C_1 \cdots C_n) = {}_1 \,?\, K_1) \cdots (C_n \,?\, K_n)$$

- Usa il seme come chiave



Cifrari a flusso e cifrari a blocchi

Cifra	Lunghezza chiave	Velocità (Mbps)
DES	56	9
3DES	168	3
RC4	variabile	45

- I codici di flusso sono

  solitamente più veloce e più facile da implementare
- Con un generatore di numeri pseudocasuali adeguatamente progettato, un cifrario a flusso può essere sicuro quanto un cifrario a blocchi di lunghezza di chiave comparabile
- Con i cifrari a blocchi le chiavi possono essere riutilizzate
- Con cifrari a flusso se due testi in chiaro, P_{io} e $P_{io\ io}$ insieme a $io = 1, 2, \ldots$, sono crittografati con la stessa chiave usando un cifrario a flusso, cioè $C_{io} = P_{io} \oplus K_{io}$ e $C_{io\ io} = P_{io io} \oplus K_{io}$, poi $C_{io} \oplus C_{io io} = P_{io} \oplus P_{io\ io}$.

RC4

- Progettato nel 1987 da Rivest, segreto commerciale pubblicato anonimamente su Internet nel 1994
- Utilizzato in SSL/TLS, WEP e WiFi Protected Access (WPA)

```
# definire SWAP(a,b) (((a) ^ (b)) && ((b) ^= (a) ^= (b), (a) ^= (b)))int principale() {

    carattere non firmato S[256],
    chiave[]="Chiave";corto senza segno i, j,
    lunghezza chiave=3;per(i=0;i<256;i++)
        S[i]=i;
    j=0;
    per(i=0;i<256;i++) {
        j=(j+S[i]+tasto[i%lunghezza chiave])%256;
        SCAMBIO(S[i],S[j]);
    }
    int K;
    io=j=0;
    mentre(1) {
        i=(i+1)%256; j=(j+S[i])%256; SCAMBIO(S[i],S[j]); K=S[(S[i]+S[j])
        %256];
        printf("%X",K);
    }
```

Contorno

Posizionamento della crittografia

- La crittografia può essere posizionata a vari livelli in il modello di riferimento OSI
- crittografia dei collegamenti ai livelli 1 o 2
- crittografia end-to-end ai livelli 3, 4, 6, 7 man mano
- che ci spostiamo più in alto nello stack OSI
  - meno informazioni sono
  - crittografate più sicure, ma
  - più complesso e con più entità e chiavi

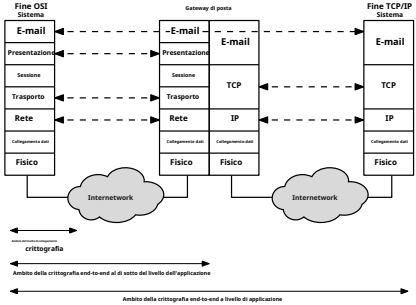


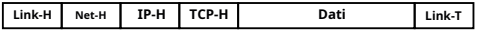
Figura 7.4 Implicazioni sulla copertura della crittografia delle comunicazioni Store-and-Forward

Posizionamento della crittografia (continua)

- Quando si utilizza la crittografia end-to-end, le intestazioni devono essere lasciate in chiaro
  - in modo che la rete possa instradare correttamente le informazioni
- Quindi, sebbene i contenuti siano protetti, i flussi del modello di traffico non lo sono. Idealmente
- vogliamo entrambi contemporaneamente
  - la crittografia end-to-end protegge i contenuti dei dati su tutto il percorso e fornisce l'autenticazione
  - collegamento protegge i flussi di traffico dal monitoraggio

(a) Crittografia a livello di applicazione (su collegamenti e su router e gateway)

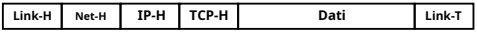
Sui link e sui router



nei gateway

(b) Crittografia a livello TCP

Sui link



Nei router e gateway

(c) Crittografia a livello di collegamento

L'ombreggiatura indica la crittografia.

TCP-H	=	Intestazione TCP
IP-H	=	Intestazione IP
Net-H	=	Intestazione a livello di rete (ad esempio, intestazione del pacchetto X.25, intestazione LLC) Intestazione
Link-H	=	del protocollo di controllo del collegamento dati
Link-T	=	Trailer del protocollo di controllo del collegamento dati

Link vs crittografia end-to-end: riepilogo

Crittografia dei collegamenti	Crittografia end-to-end
Sicurezza all'interno dei sistemi finali e dei sistemi intermedi	
Messaggio esposto nell'host di invio/ricezione Messaggio esposto nei nodi intermedi	Messaggio crittografato nell'host mittente/ricevente Messaggio crittografato nei nodi intermedi
Ruolo dell'utente	
Applicato inviando/ricevendo host Trasparente per l'utente L'host mantiene la struttura di crittografia Una struttura per tutti gli utenti Può essere eseguito dall'hardware Tutti o nessun messaggio crittografato	Applicato dal processo di invio/ricezione L'utente applica la crittografia L'utente deve determinare l'algoritmo Gli utenti selezionano lo schema di crittografia Implementazione del software L'utente sceglie di crittografare o meno per ogni messaggio
Problemi di implementazione	
Richiede una chiave per coppia (host-nodo intermedio) e coppia (nodo intermedio-nodo intermedio) Fornisce l'autenticazione dell'host	Richiede una chiave per coppia di utenti Fornisce l'autenticazione dell'utente

Link vs crittografia end-to-end: riepilogo

Crittografia dei collegamenti	Crittografia end-to-end
Sicurezza all'interno dei sistemi finali e dei sistemi intermedi	
Messaggio esposto nell'host di invio/ricezione Messaggio esposto nei nodi intermedi	Messaggio crittografato nell'host di invio/ricezione Messaggio crittografato nei nodi intermedi
Ruolo dell'utente	
Applicato inviando/ricevendo host Trasparente per l'utente L'host mantiene la struttura di crittografia Una struttura per tutti gli utenti Può essere eseguito dall'hardware Tutti o nessun messaggio crittografato	Applicato dal processo di invio/ricezione L'utente applica la crittografia L'utente deve determinare l'algoritmo Gli utenti selezionano lo schema di crittografia Implementazione del software L'utente sceglie di crittografare o meno per ogni messaggio
Problemi di implementazione	
Richiede una chiave per coppia (host-nodo intermedio) e coppia (nodo intermedio-nodo intermedio) Fornisce l'autenticazione dell'host	Richiede una chiave per coppia di utenti Fornisce l'autenticazione dell'utente

Link vs crittografia end-to-end: riepilogo

Crittografia dei collegamenti	Crittografia end-to-end
Sicurezza all'interno dei sistemi finali e dei sistemi intermedi	
Messaggio esposto nell'host di invio/ricezione Messaggio esposto nei nodi intermedi	Messaggio crittografato nell'host di invio/ricezione Messaggio crittografato nei nodi intermedi
Ruolo dell'utente	
Applicato inviando/ricevendo host Trasparente per l'utente L'host mantiene la struttura di crittografia Una struttura per tutti gli utenti Può essere eseguito dall'hardware Tutti o nessun messaggio crittografato	Applicato dal processo di invio/ricezione L'utente applica la crittografia L'utente deve determinare l'algoritmo Gli utenti selezionano lo schema di crittografia Implementazione del software L'utente sceglie di crittografare o meno per ogni messaggio
Problemi di implementazione	
Richiede una chiave per coppia (host-nodo intermedio) e coppia (nodo intermedio-nodo intermedio) Fornisce l'autenticazione dell'host	Richiede una chiave per coppia di utenti Fornisce l'autenticazione dell'utente

Link vs crittografia end-to-end: riepilogo

Crittografia dei collegamenti	Crittografia end-to-end
Sicurezza all'interno dei sistemi finali e dei sistemi intermedi	
Messaggio esposto nell'host di invio/ricezione Messaggio esposto nei nodi intermedi	Messaggio crittografato nell'host mittente/ricevente Messaggio crittografato nei nodi intermedi
Ruolo dell'utente	
Applicato inviando/ricevendo l'host Trasparente per l'utente L'host mantiene la struttura di crittografia Una struttura per tutti gli utenti Può essere eseguito dall'hardware Tutti o nessun messaggio crittografato	Applicato tramite processo di invio/ricezione L'utente applica la crittografia L'utente deve determinare l'algoritmo Gli utenti selezionano lo schema di crittografia Implementazione del software L'utente sceglie di crittografare o meno per ogni messaggio
Problemi di implementazione	
Richiede una chiave per coppia (host-nodo intermedio) e coppia (nodo intermedio-nodo intermedio) Fornisce l'autenticazione dell'host	Richiede una chiave per coppia di utenti Fornisce l'autenticazione dell'utente

Link vs crittografia end-to-end: riepilogo

Crittografia dei collegamenti	Crittografia end-to-end
Sicurezza all'interno dei sistemi finali e dei sistemi intermedi	
Messaggio esposto nell'host di invio/ricezione Messaggio esposto nei nodi intermedi	Messaggio crittografato nell'host mittente/ricevente Messaggio crittografato nei nodi intermedi
Ruolo dell'utente	
Applicato inviando/ricevendo l'host Trasparente per l'utente L'host mantiene la struttura di crittografia Una struttura per tutti gli utenti Può essere eseguito dall'hardware Tutti o nessun messaggio crittografato	Applicato tramite processo di invio/ricezione L'utente applica la crittografia L'utente deve determinare l'algoritmo Gli utenti selezionano lo schema di crittografia Implementazione del software L'utente sceglie di crittografare o meno per ogni messaggio
Problemi di implementazione	
Richiede una chiave per coppia (host-nodo intermedio) e coppia (nodo intermedio-nodo intermedio) Fornisce l'autenticazione dell'host	Richiede una chiave per coppia di utenti Fornisce l'autenticazione dell'utente

Link vs crittografia end-to-end: riepilogo

Crittografia dei collegamenti	Crittografia end-to-end
Sicurezza all'interno dei sistemi finali e dei sistemi intermedi	
Messaggio esposto nell'host di invio/ricezione Messaggio esposto nei nodi intermedi	Messaggio crittografato nell'host mittente/ricevente Messaggio crittografato nei nodi intermedi
Ruolo dell'utente	
Applicato inviando/ricevendo host Trasparente per l'utente L'host mantiene la funzione di crittografia Una struttura per tutti gli utenti Può essere eseguita dall'hardware Tutti o nessun messaggio crittografato	Applicato dal processo di invio/ricezione L'utente applica la crittografia L'utente deve determinare l'algoritmo Gli utenti selezionano lo schema di crittografia Implementazione del software L'utente sceglie di crittografare o meno per ogni messaggio
Problemi di implementazione	
Richiede una chiave per coppia (host-nodo intermedio) e coppia (nodo intermedio-nodo intermedio) Fornisce l'autenticazione dell'host	Richiede una chiave per coppia di utenti Fornisce l'autenticazione dell'utente

Link vs crittografia end-to-end: riepilogo

Crittografia dei collegamenti	Crittografia end-to-end
Sicurezza all'interno dei sistemi finali e dei sistemi intermedi	
Messaggio esposto nell'host di invio/ricezione Messaggio esposto nei nodi intermedi	Messaggio crittografato nell'host mittente/ricevente Messaggio crittografato nei nodi intermedi
Ruolo dell'utente	
Applicato inviando/ricevendo host Trasparente per l'utente L'host mantiene la funzione di crittografia Una struttura per tutti gli utenti Può essere eseguito dall'hardware Tutti o nessun messaggio crittografato	Applicato dal processo di invio/ricezione L'utente applica la crittografia L'utente deve determinare l'algoritmo Gli utenti selezionano lo schema di crittografia Implementazione del software L'utente sceglie di crittografare o meno per ogni messaggio
Problemi di implementazione	
Richiede una chiave per coppia (host-nodo intermedio) e coppia (nodo intermedio-nodo intermedio) Fornisce l'autenticazione dell'host	Richiede una chiave per coppia di utenti Fornisce l'autenticazione dell'utente

Link vs crittografia end-to-end: riepilogo

Crittografia dei collegamenti	Crittografia end-to-end
Sicurezza all'interno dei sistemi finali e dei sistemi intermedi	
Messaggio esposto nell'host di invio/ricezione Messaggio esposto nei nodi intermedi	Messaggio crittografato nell'host mittente/ricevente Messaggio crittografato nei nodi intermedi
Ruolo dell'utente	
Applicato inviando/ricevendo host Trasparente per l'utente L'host mantiene la struttura di crittografia Una struttura per tutti gli utenti Può essere fatto dall'hardware Tutti o nessun messaggio crittografato	Applicato dal processo di invio/ricezione L'utente applica la crittografia L'utente deve determinare l'algoritmo Gli utenti selezionano lo schema di crittografia Implementazione del software L'utente sceglie di crittografare o meno per ogni messaggio
Problemi di implementazione	
Richiede una chiave per coppia (host-nodo intermedio) e coppia (nodo intermedio-nodo intermedio) Fornisce l'autenticazione dell'host	Richiede una chiave per coppia di utenti Fornisce l'autenticazione dell'utente

Link vs crittografia end-to-end: riepilogo

Crittografia dei collegamenti	Crittografia end-to-end
Sicurezza all'interno dei sistemi finali e dei sistemi intermedi	
Messaggio esposto nell'host di invio/ricezione Messaggio esposto nei nodi intermedi	Messaggio crittografato nell'host mittente/ricevente Messaggio crittografato nei nodi intermedi
Ruolo dell'utente	
Applicato inviando/ricevendo host Trasparente per l'utente L'host mantiene la struttura di crittografia Una struttura per tutti gli utenti Può essere fatto dall'hardwareTutti o nessun messaggio crittografato	Applicato dal processo di invio/ricezione L'utente applica la crittografia L'utente deve determinare l'algoritmo Gli utenti selezionano lo schema di crittografia Implementazione del software L'utente sceglie di crittografare o meno per ogni messaggio
Problemi di implementazione	
Richiede una chiave per coppia (host-nodo intermedio) e coppia (nodo intermedio-nodo intermedio) Fornisce l'autenticazione dell'host	Richiede una chiave per coppia di utenti Fornisce l'autenticazione dell'utente

Link vs crittografia end-to-end: riepilogo

Crittografia dei collegamenti	Crittografia end-to-end
Sicurezza all'interno dei sistemi finali e dei sistemi intermedi	
Messaggio esposto nell'host di invio/ricezione Messaggio esposto nei nodi intermedi	Messaggio crittografato nell'host mittente/ricevente Messaggio crittografato nei nodi intermedi
Ruolo dell'utente	
Applicato inviando/ricevendo host Trasparente per l'utente L'host mantiene la struttura di crittografia Una struttura per tutti gli utenti Può essere eseguito dall'hardware Tutti o nessun messaggio crittografato	Applicato dal processo di invio/ricezione L'utente applica la crittografia L'utente deve determinare l'algoritmo Gli utenti selezionano lo schema di crittografia Implementazione del software L'utente sceglie di crittografare o meno per ogni messaggio
Problemi di implementazione	
Richiede una chiave per coppia (host-nodo intermedio) e coppia (nodo intermedio-nodo intermedio) Fornisce l'autenticazione dell'host	Richiede una chiave per coppia di utenti Fornisce l'autenticazione dell'utente

Link vs crittografia end-to-end: riepilogo

Crittografia dei collegamenti	Crittografia end-to-end
Sicurezza all'interno dei sistemi finali e dei sistemi intermedi	
Messaggio esposto nell'host di invio/ricezione Messaggio esposto nei nodi intermedi	Messaggio crittografato nell'host mittente/ricevente Messaggio crittografato nei nodi intermedi
Ruolo dell'utente	
Applicato inviando/ricevendo host Trasparente per l'utente L'host mantiene la struttura di crittografia Una struttura per tutti gli utenti Può essere eseguito dall'hardware Tutti o nessun messaggio crittografato	Applicato dal processo di invio/ricezione L'utente applica la crittografia L'utente deve determinare l'algoritmo Gli utenti selezionano lo schema di crittografia Implementazione del software L'utente sceglie di crittografare o meno per ogni messaggio
Problemi di implementazione	
Richiede una chiave per coppia (host-nodo intermedio) e coppia (nodo intermedio-nodo intermedio) Fornisce l'autenticazione dell'host	Richiede una chiave per coppia di utenti Fornisce l'autenticazione dell'utente

Analisi del traffico

- Monitoraggio dei flussi di comunicazione tra le parti
  - utile sia in ambito militare che commerciale può essere
  - utilizzato anche per creare un canale nascosto
- La crittografia del collegamento oscura i dettagli dell'intestazione
  - ma i volumi di traffico complessivi nelle reti e agli end-point sono ancora visibili
- Il riempimento del traffico può oscurare ulteriormente i flussi
  - ma a scapito del traffico continuo

Contorno

- Gli schemi simmetrici richiedono che entrambe le parti condividano una chiave segreta comune. Il
- problema è come distribuire in modo sicuro questa chiave.
- Spesso il guasto del sistema sicuro è dovuto a un'interruzione dello schema di distribuzione delle chiavi.

Approcci alla distribuzione delle chiavi

Date le parti A e B hanno varie alternative di distribuzione delle chiavi:

- A può selezionare la chiave e consegnarla fisicamente a B La terza
- parte può selezionare e consegnare la chiave ad A e B
- Se A e B hanno comunicato in precedenza, possono utilizzare la chiave precedente per crittografare una nuova chiave
- Se A e B hanno comunicazioni sicure con una terza parte C, C può inoltrare la chiave tra A e B

Gerarchia dei tasti

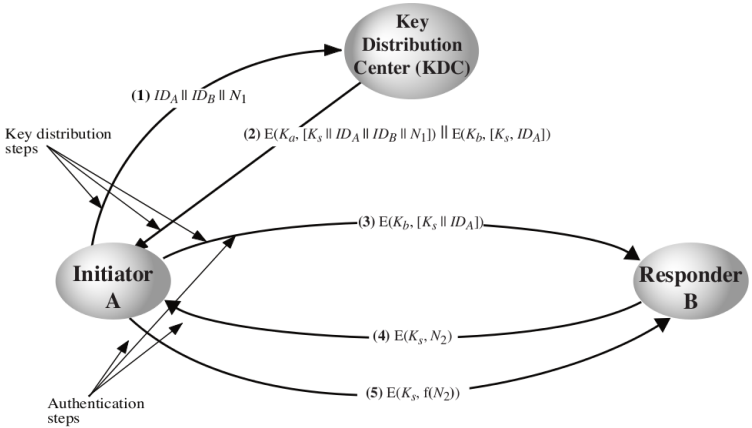
In genere viene utilizzata una gerarchia di chiavi:

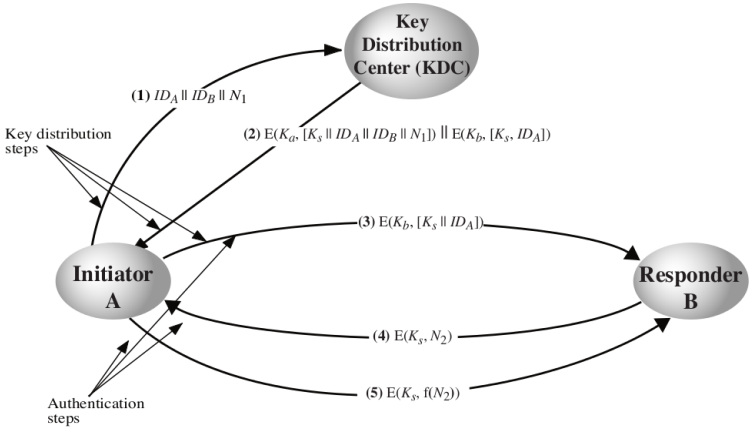
Chiave di sessione: utilizzato per la crittografia dei dati tra utenti per una sessione logica quindi scartato

Chiave principale: utilizzato per crittografare le chiavi di sessione

condiviso dall'utente e dal centro di distribuzione delle chiavi

Scenario di distribuzione delle chiavi





Debolezza: *B* non è possibile controllare la freschezza di *Ks*. Se *Ks* è compromesso, quindi può essere riprodotto dall'attaccante.

- Gerarchie di KDC necessarie per reti di grandi dimensioni, ma devono fidarsi l'una dell'altra Le durate delle
- chiavi di sessione dovrebbero essere limitate per una maggiore sicurezza
- Uso della distribuzione automatica delle chiavi per conto degli utenti, ma deve fidarsi del sistema Uso
- della distribuzione delle chiavi decentralizzata
- Controllo dell'utilizzo delle chiavi