La crittografia a chiave simmetrica

Gregorio D'Agostino

7 Maggio 2021

La crittografia simmetrica

Le comunicazioni continuate: le sessioni

- Abbiamo visto che è possibile scambiarsi pacchetti tra indirizzi IP.

Le comunicazioni continuate: le sessioni

- Abbiamo visto che è possibile scambiarsi pacchetti tra indirizzi IP.
- Vedremo che una volta stabilito il primo contatto si stabilisce una serie di comunicazioni correlate tra loro che condividono le stesse modalità ed in particolare i soggetti comunicanti

Le comunicazioni continuate: le sessioni

- Abbiamo visto che è possibile scambiarsi pacchetti tra indirizzi IP.
- Vedremo che una volta stabilito il primo contatto si stabilisce una serie di comunicazioni correlate tra loro che condividono le stesse modalità ed in particolare i soggetti comunicanti
- Abbiamo visto che con una serie di comunizioni è possibile condividere una chiave segreta comunicandosela tramite un canale pubblico.

Le comunicazioni continuate: le sessioni

- Abbiamo visto che è possibile scambiarsi pacchetti tra indirizzi IP.
- Vedremo che una volta stabilito il primo contatto si stabilisce una serie di comunicazioni correlate tra loro che condividono le stesse modalità ed in particolare i soggetti comunicanti
- Abbiamo visto che con una serie di comunizioni è possibile condividere una chiave segreta comunicandosela tramite un canale pubblico.
- Una volta disponibile la chiave viene utilizzata per comunicazioni a chiave condivisa o chiave simmetrica.

Principi della criptografia simmetrica

- I principi della crittografia simmetrica sono sempre gli stessi: si tratta di trasformare un testo o altra informazione "leggibile" in un altra sequenza di bit inutilizzabile da chi non conosca una (unica) chiave segreta.

Principi della criptografia simmetrica

- I principi della crittografia simmetrica sono sempre gli stessi: si tratta di trasformare un testo o altra informazione "leggibile" in un altra sequenza di bit inutilizzabile da chi non conosca una (unica) chiave segreta.
- La cifratura $\mathcal{C}$ è una procedura che consente di trasformare i dati (m) in una forma equivalente (c), ma incomprensibile o indecifrabile in un assegnato lasso di tempo anche avvalendosi dei mezzi di calcolo esistenti più avanzati.

$$c = \mathcal{C}(m).$$

Principi della criptografia simmetrica

- I principi della crittografia simmetrica sono sempre gli stessi: si tratta di trasformare un testo o altra informazione "leggibile" in un altra sequenza di bit inutilizzabile da chi non conosca una (unica) chiave segreta.
- La cifratura $\mathcal{C}$ è una procedura che consente di trasformare i dati (m) in una forma equivalente (c), ma incomprensibile o indecifrabile in un assegnato lasso di tempo anche avvalendosi dei mezzi di calcolo esistenti più avanzati.

$$c = \mathcal{C}(m).$$

- Abbiamo visto alcuni metodi di trasformazione di un numero m di grandezza limitata (ad esempio 256bit) in un altro cifrato c. Basandoci sul tali trasformazioni è possibile cifrare una sequenza di bit di lunghezza arbitraria che può rappresentare un testo, un eseguibile, un file compresso, una musica o un immagine campionata o altro.

Cifratura sequenziale o a blocchi

- Vi sono due possibili approcci per trattare sequenze di bit di lunghezza arbitraria. Il primo detto Cifratura continua o sequenziale ("Stream Cipher") consiste nel trattare i bit in sequenza ed applicare trasformazioni che dipendono esclusivamente dal valore del bit corrente e (eventualmente) dai valori dei precedenti. Se s_i sono i bit della sequenza da cifrare:

$$b_i = \mathcal{C}(s_i; s_1, s_2 \ldots, s_{i-1}).$$

Cifratura sequenziale o a blocchi

- Vi sono due possibili approcci per trattare sequenze di bit di lunghezza arbitraria. Il primo detto Cifratura continua o sequenziale ("Stream Cipher") consiste nel trattare i bit in sequenza ed applicare trasformazioni che dipendono esclusivamente dal valore del bit corrente e (eventualmente) dai valori dei precedenti. Se s_i sono i bit della sequenza da cifrare:
$$b_i = \mathcal{C}(s_i; s_1, s_2 \ldots, s_{i-1}).$$
- Il secondo metodo, denominato Cifratura a blocchi ("Block Cipher") consiste nel suddividere la sequenza iniziale in unità di lunghezza predefinita e reiterare la procedura di cifratura su ciascuna di esse. Tipicamente si usano blocchi di lunghezza multipla di 64 bit essendo questa la lunghezza base su cui vengono eseguite le operazioni nei processori moderni. Abbiamo già visto esempi elementari di tale metodo.

Cifratura sequenziale o a blocchi

- Come già detto gli strumenti base per la cifratura sono l'alterazione della posizione (trasposizione) dei bit e la sostituzione di sequenza con altre.

Cifratura sequenziale o a blocchi

- Come già detto gli strumenti base per la cifratura sono l'alterazione della posizione (trasposizione) dei bit e la sostituzione di sequenza con altre.
- Un esempio già visto è la permutazione delle lettere dell'alfabeto. Una lettera è codificata con 8bit (1 byte) cioè un numero compreso tra 0 e 255. Una sostituzione non è altro che una mappa dei gruppi da 8 bit in altri.

Cifratura sequenziale o a blocchi

- Come già detto gli strumenti base per la cifratura sono l'alterazione della posizione (trasposizione) dei bit e la sostituzione di sequenza con altre.
- Un esempio già visto è la permutazione delle lettere dell'alfabeto. Una lettera è codificata con 8bit (1 byte) cioè un numero compreso tra 0 e 255. Una sostituzione non è altro che una mappa dei gruppi da 8 bit in altri.
- Una parola è (per convenzione) una sequenza di 8 byte; quindi se usiamo blocchi da 64 bit un blocco è una parola, se usiamo blocchi da 256 bit, un blocco sono 4 parole.

Cifratura a blocchi simmetrica

- L'applicazione $\mathcal{C}$ dipende parametricamente da una chiave k:

$$c = c_k = \mathcal{C}(m) = \mathcal{C}_k(m).$$

Cifratura a blocchi simmetrica

- L'applicazione $\mathcal{C}$ dipende parametricamente da una chiave k:

$$c = c_k = \mathcal{C}(m) = \mathcal{C}_k(m).$$

- Una cifratura si dice "simmetrica" quando la chiave utilizzata per cifrare e per decifrare è la stessa. la funzione $\mathcal{C}$ è nota a tutti, ma la chiave k è conosciuta solo al cifratore ed al decifratore; o più esattamente al circolo di persone o dispositivi che la condivide. La funzione di decifratura $\mathcal{D}$ è l'inversa della cifratura $\mathcal{C}$:

$$\mathcal{C} : \mathcal{D}(\mathcal{C}(m)) = m.$$

Cifratura a blocchi simmetrica

- L'applicazione $\mathcal{C}$ dipende parametricamente da una chiave k:

$$c = c_k = \mathcal{C}(m) = \mathcal{C}_k(m).$$

- Una cifratura si dice "simmetrica" quando la chiave utilizzata per cifrare e per decifrare è la stessa. la funzione $\mathcal{C}$ è nota a tutti, ma la chiave k è conosciuta solo al cifratore ed al decifratore; o più esattamente al circolo di persone o dispositivi che la condivide. La funzione di decifratura $\mathcal{D}$ è l'inversa della cifratura $\mathcal{C}$:

$$\mathcal{C} : \mathcal{D}(\mathcal{C}(m)) = m.$$

- Nel caso della crittografia simmetrica entrambe le funzioni di cifratura e decifratura sono tenute segrete. In molti casi sono note le procedure, ma non la chiave k. In questo caso la complessità della cifratura è data dalla complessità della chiave (la cardinalità dello spazio dei valori possibili).

Cifratura a blocchi simmetrica

- La crittanalisi studia i metodi per decrittare informazioni (sequenze di bit) senza disporre della chiave.

Cifratura a blocchi simmetrica

- La crittanalisi studia i metodi per decrittare informazioni (sequenze di bit) senza disporre della chiave.
- Una cifratura si dice sicura se il tempo necessario a decrittare è più lungo di un valore prefissato o della utilità dell'informazione protetta. Ad esempio il testo di un esame deve essere segreto fino al giorno dello stesso. Le informazioni classificate a 50 anni devono, invece resistere a attacchi di tale durata.

Cifratura a blocchi simmetrica

- La crittanalisi studia i metodi per decrittare informazioni (sequenze di bit) senza disporre della chiave.
- Una cifratura si dice sicura se il tempo necessario a decrittare è più lungo di un valore prefissato o della utilità dell'informazione protetta. Ad esempio il testo di un esame deve essere segreto fino al giorno dello stesso. Le informazioni classificate a 50 anni devono, invece resistere a attacchi di tale durata.
- Come sempre il concetto di sicurezza è legato agli strumenti ed al tempo a disposizione del potenziale attaccante.

La cifratura a blocchi

- Uno schema di cifratura (lo abbiamo già visto) si dice "a blocchi" quando la sequenza che si intende cifrare è suddivisa in sequenze binarie di uguale dimensione dette blocchi.

La cifratura a blocchi

- Uno schema di cifratura (lo abbiamo già visto) si dice "a blocchi" quando la sequenza che si intende cifrare è suddivisa in sequenze binarie di uguale dimensione dette blocchi.
- La dimensione del blocco è il numero dei suoi bit. Ogni blocco viene cifrato utilizzando lo stesso algoritmo. Abbiamo visto esempi di cifratura trasposizionale a blocchi basati su una permutazione di caratteri. Erano casi particolari a blocchi da 8 bit (la codifica ascii). Per avvalersi delle caratteristiche dei processori moderni si usano spesso blocchi da 64bit.

La cifratura a blocchi

- Uno schema di cifratura (lo abbiamo già visto) si dice "a blocchi" quando la sequenza che si intende cifrare è suddivisa in sequenze binarie di uguale dimensione dette blocchi.
- La dimensione del blocco è il numero dei suoi bit. Ogni blocco viene cifrato utilizzando lo stesso algoritmo. Abbiamo visto esempi di cifratura trasposizionale a blocchi basati su una permutazione di caratteri. Erano casi particolari a blocchi da 8 bit (la codifica ascii). Per avvalersi delle caratteristiche dei processori moderni si usano spesso blocchi da 64bit.
- La cifratura più semplice a blocchi è la ECB (Electronic Code Book) in cui la sequenza è decomposta in blocchi da 64 bit ognuno dei quali viene cifrato con la stessa chiave segreta. Vedremo che questo schema semplice si presta ad attacchi crittoanalitici.

La cifratura a blocchi

- Uno schema di cifratura (lo abbiamo già visto) si dice "a blocchi" quando la sequenza che si intende cifrare è suddivisa in sequenze binarie di uguale dimensione dette blocchi.
- La dimensione del blocco è il numero dei suoi bit. Ogni blocco viene cifrato utilizzando lo stesso algoritmo. Abbiamo visto esempi di cifratura trasposizionale a blocchi basati su una permutazione di caratteri. Erano casi particolari a blocchi da 8 bit (la codifica ascii). Per avvalersi delle caratteristiche dei processori moderni si usano spesso blocchi da 64bit.
- La cifratura più semplice a blocchi è la ECB (Electronic Code Book) in cui la sequenza è decomposta in blocchi da 64 bit ognuno dei quali viene cifrato con la stessa chiave segreta. Vedremo che questo schema semplice si presta ad attacchi crittoanalitici.
- Sicuramente la cifratura ECB non garantisce l'integrità perché è sufficiente per un attaccante mescolare i blocchi tra loro e il risultato sarebbe ancora leggibile (ma in ordine diverso).

Cifratura a blocchi in cascata

- Per rendere più difficile la decrittazione trasposizionale a blocchi si aggiungono degli espedienti.

Cifratura a blocchi in cascata

- Per rendere più difficile la decrittazione trasposizionale a blocchi si aggiungono degli espedienti.
- Una versione appena più sofisticata della precedente è la crifratura a blocchi in cascata CBC (Cipher Block Chain), in cui non si cifra il singolo blocco ma lo XOR con il risultato della cifratura del blocco precedente.

Cifratura a blocchi in cascata

- Per rendere più difficile la decrittazione trasposizionale a blocchi si aggiungono degli espedienti.
- Una versione appena più sofisticata della precedente è la crifratura a blocchi in cascata CBC (Cipher Block Chain), in cui non si cifra il singolo blocco ma lo XOR con il risultato della cifratura del blocco precedente.
- Purtroppo anche questo piccolo artificio non rafforza molto la cifratura.

O esclusivi e permutazioni

- Due degli elementi chiave che possono essere combinati per rafforzare la cifratura sono la permutazione (trasposizione) dei bit e la composizione tramite un or esclusivo con una chiave della stessa lunghezza della sequenza.

O esclusivi e permutazioni

- Due degli elementi chiave che possono essere combinati per rafforzare la cifratura sono la permutazione (trasposizione) dei bit e la composizione tramite un or esclusivo con una chiave della stessa lunghezza della sequenza.
- Abbiamo visto che se usassimo sempre nuove chiavi per ogni blocco di dati la cifratura sarebbe perfetta. Se invece riutilizziamo la stessa chiave non aggiungiamo nuova aleatorietà ad ogni blocco.

O esclusivi e permutazioni

- Due degli elementi chiave che possono essere combinati per rafforzare la cifratura sono la permutazione (trasposizione) dei bit e la composizione tramite un or esclusivo con una chiave della stessa lunghezza della sequenza.
- Abbiamo visto che se usassimo sempre nuove chiavi per ogni blocco di dati la cifratura sarebbe perfetta. Se invece riutilizziamo la stessa chiave non aggiungiamo nuova aleatorietà ad ogni blocco.
- Vedremo cosa succede se si compongono pemutazioni ed o esclusivi con chiavi diverse sulla stesso messaggio (o blocco).

Permutazioni

- Una permutazione di una sequenza $a = \{a_1 a_2 \dots a_n\}$ è una operazione che cambia l'ordine degli elementi senza scartarne nessuno o duplicarne alcuno:
  $a \to b = \{b_1 b_2 \dots b_n\} = \{a_{i_1} a_{i_2} \dots a_{i_n}\}$. In cui l'insieme degli elementi $\{i_1, i_2, \dots, i_n\}$ coincide a meno dell'ordine con l'insieme dei primi numeri $\{1, 2, \dots, n\}$:

$$b_k = a_{i_k}.$$

Permutazioni

- Una permutazione di una sequenza $a = \{a_1 a_2 \dots a_n\}$ è una operazione che cambia l'ordine degli elementi senza scartarne nessuno o duplicarne alcuno:
  $a \to b = \{b_1 b_2 \dots b_n\} = \{a_{i_1} a_{i_2} \dots a_{i_n}\}$. In cui l'insieme degli elementi $\{i_1, i_2, \dots, i_n\}$ coincide a meno dell'ordine con l'insieme dei primi numeri $\{1, 2, \dots, n\}$:

$$b_k = a_{i_k}.$$

- Per questa ragione per indicare una permutazione è sufficiente indicare la sequenza ordinata dei numeri $i_1, i_2, \dots, i_n$. In forma operatoriale:

$$\pi_{i_1, i_2, \dots, i_n}(a) = b.$$

Composizione di due permutazioni

- Se si eseguono due pemutazioni successive sulla stesso blocco il risultato è equivalente ad effettuare una sola permutazione:

$$\pi^1 \circ \pi^2(a) \stackrel{def}{=} \pi^1(\pi^2(a));$$

$$\pi_{j_1,j_2,\ldots,j_n}\left(\pi_{i_1,i_2,\ldots,i_n}(a)\right) = \pi_{j_1,j_2,\ldots,j_n}(b) = c = \pi_{k_1,k_2,\ldots,k_n}(a).$$

in cui $k_m = j_{i_m}$.

Composizione di due permutazioni

- Se si eseguono due pemutazioni successive sulla stesso blocco il risultato è equivalente ad effettuare una sola permutazione:

$$\pi^1 \circ \pi^2(a) \stackrel{def}{=} \pi^1(\pi^2(a));$$

$$\pi_{j_1,j_2,\ldots,j_n}\left(\pi_{i_1,i_2,\ldots,i_n}(a)\right) = \pi_{j_1,j_2,\ldots,j_n}(b) = c = \pi_{k_1,k_2,\ldots,k_n}(a).$$

  in cui $k_m = j_{i_m}$.
- Esempio $\pi^1 \circ \pi^2 = \pi_{3142} \circ \pi_{2341} = \pi_{4213}$ agendo sulla sequenza $ABCD$ fornisce:

$$\pi^2(ABCD) = BCDA$$
$$\pi^1(BCDA) = DBAC$$
$$\pi^1 \circ \pi^2(ABCD) = DBAC$$

Composizione di due chiavi

- Comporre un messagio m (o un blocco) tramite uno xor con una chiave due volte di seguito equivale ad applicare una sola volta la composizione della chiavi:

$$(K_1 \oplus m) \oplus K_2 = (m \oplus K_1) \oplus K_2 = m \oplus (K_1 \oplus K_2) = m \oplus K;$$

a causa delle proprietà associativa e commutativa.

Composizione di due chiavi

- Comporre un messagio m (o un blocco) tramite uno xor con una chiave due volte di seguito equivale ad applicare una sola volta la composizione della chiavi:

$$(K_1 \oplus m) \oplus K_2 = (m \oplus K_1) \oplus K_2 = m \oplus (K_1 \oplus K_2) = m \oplus K;$$

  a causa delle proprietà associativa e commutativa.
- Quindi cifrare due volte con due chiavi diverse non aggiunge entropia (variabilità) al messaggio. La difficoltà per l'attaccante non aumenta.

Composizione di o esclusivi e permutazioni

- L'operazioni di permutazione gode della proprietà distribuitiva rispetto all' "o aut" (XOR ovvero $\oplus$). Ovvero se si esegue prima la permutazione e poi l'aut per una diversa sequenza permutata con la stessa regola il risultato non cambia:

$$\pi(K \oplus m) = \pi(m) \oplus \pi(K) = \pi(m) \oplus K'.$$

Composizione di o esclusivi e permutazioni

- L'operazioni di permutazione gode della proprietà distribuitiva rispetto all' "o aut" (XOR ovvero $\oplus$). Ovvero se si esegue prima la permutazione e poi l'aut per una diversa sequenza permutata con la stessa regola il risultato non cambia:

$$\pi(K \oplus m) = \pi(m) \oplus \pi(K) = \pi(m) \oplus K'.$$

- Quindi eseguire una sequenza qualsiasi di muduli permutazionali e composizioni di aut con una chiave equivale ad un'unica permutazione ed un solo aut per una unica chiave:

$$\pi_n(K_n \oplus \pi_{n-1}(K_{n-1} \oplus \ldots)) = \pi(m) \oplus K'.$$

Composizione di o esclusivi e permutazioni

- L'operazioni di permutazione gode della proprietà distribuitiva rispetto all' "o aut" (XOR ovvero $\oplus$). Ovvero se si esegue prima la permutazione e poi l'aut per una diversa sequenza permutata con la stessa regola il risultato non cambia:

$$\pi(K \oplus m) = \pi(m) \oplus \pi(K) = \pi(m) \oplus K'.$$

- Quindi eseguire una sequenza qualsiasi di muduli permutazionali e composizioni di aut con una chiave equivale ad un'unica permutazione ed un solo aut per una unica chiave:

$$\pi_n(K_n \oplus \pi_{n-1}(K_{n-1} \oplus \ldots)) = \pi(m) \oplus K'.$$

- Basandosi su questa osservazione è stata sviluppata una tecnica crittografica denominata crittografia differenziale che ricava una chiave equivalente.

Composizione di o esclusivi e permutazioni

- L'operazioni di permutazione gode della proprietà distribuitiva rispetto all' "o aut" (XOR ovvero $\oplus$). Ovvero se si esegue prima la permutazione e poi l'aut per una diversa sequenza permutata con la stessa regola il risultato non cambia:

$$\pi(K \oplus m) = \pi(m) \oplus \pi(K) = \pi(m) \oplus K'.$$

- Quindi eseguire una sequenza qualsiasi di muduli permutazionali e composizioni di aut con una chiave equivale ad un'unica permutazione ed un solo aut per una unica chiave:

$$\pi_n(K_n \oplus \pi_{n-1}(K_{n-1} \oplus \ldots)) = \pi(m) \oplus K'.$$

- Basandosi su questa osservazione è stata sviluppata una tecnica crittografica denominata crittografia differenziale che ricava una chiave equivalente.
- Per questa ragione furono definiti i moduli s nell'algoritmo di Feistel che non commutano con lo XOR.

Cifrature a blocchi alla Horst Feistel

- Anche il CBC è vulnerabile rispetto ad attacchi crittoanalitici quando la mole di dati è sufficientemente ampia; inoltre parte della chiave è un blocco fittizio necessario per cifrare il primo blocco vero.

Cifrature a blocchi alla Horst Feistel

- Anche il CBC è vulnerabile rispetto ad attacchi crittoanalitici quando la mole di dati è sufficientemente ampia; inoltre parte della chiave è un blocco fittizio necessario per cifrare il primo blocco vero.
- Uno degli schemi più diffusi per la cifratura a blocchi è lo schema di cifratura alla Horst Feistel (IBM in 1973). Gli schemi fondamentali di uso corrente (cioè gli standard): Data Encryption Standard (DES) e Triple DES (3DES) rientrano in questo schema di cifratura.
  Anche la "Advanced Encryption Standard" (AES), divenuto lo standard, dal 2011 possiede uno schema di cifratura molto simile e comunque uno schema di cifratura a blocchi a chiave simmetrica.

Schema di cifratura alla Feistel

- Si tratta di cifratura a blocchi (di dimensione pari) divisi in due semiblocchi, in cui si itera una procedura di base un numero di volte, denominate round). Ogni iterazione è basata su un algoritmo di cifratura elementare che impiega una chiave di cifratura. Le chiavi possono essere indipendenti o derivate da un'unica chiave generale.

Schema di cifratura alla Feistel

- Si tratta di cifratura a blocchi (di dimensione pari) divisi in due semiblocchi, in cui si itera una procedura di base un numero di volte, denominate round). Ogni iterazione è basata su un algoritmo di cifratura elementare che impiega una chiave di cifratura. Le chiavi possono essere indipendenti o derivate da un'unica chiave generale.
- I parametri caratterisitici sono: la dimensione del semiblocco w, il numero di iterazioni (round) e la complessità delle chiavi (la cardinalità dello spazio delle chiavi). L'algoritmo iterativo di base, detto round function, e l'algoritmo di generazione delle sottochiavi completano la caratterizzazione della cifratura.

Schema di cifratura alla Feistel

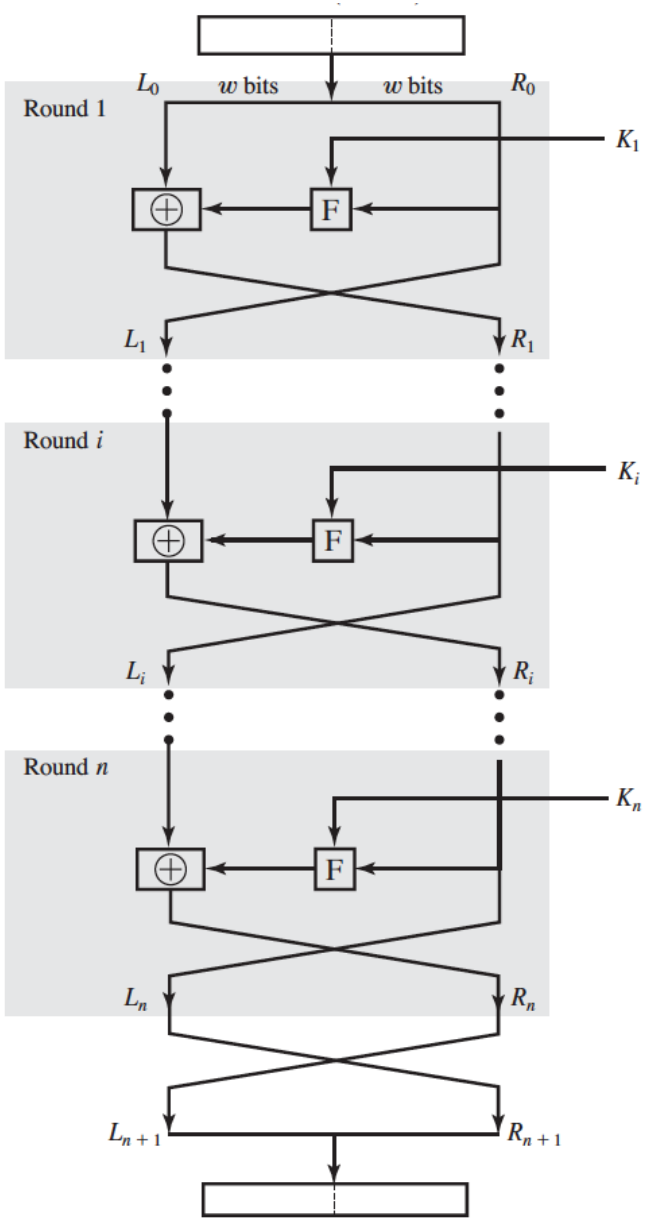
- Si tratta di cifratura a blocchi (di dimensione pari) divisi in due semiblocchi, in cui si itera una procedura di base un numero di volte, denominate round). Ogni iterazione è basata su un algoritmo di cifratura elementare che impiega una chiave di cifratura. Le chiavi possono essere indipendenti o derivate da un'unica chiave generale.
- I parametri caratterisitici sono: la dimensione del semiblocco w, il numero di iterazioni (round) e la complessità delle chiavi (la cardinalità dello spazio delle chiavi). L'algoritmo iterativo di base, detto round function, e l'algoritmo di generazione delle sottochiavi completano la caratterizzazione della cifratura.
- Uno degli aspetti utili della cifratura alla Feistel è la semplicità della decifratura che consiste nell'applicare in ordine inverso la funzione inversa della round function. Nota la chiave questa funzione deve essere di facile computazione.

Horst Feistel

- Crittografo tedesco 30 Gennaio 1915 - 14 Novembre 1990

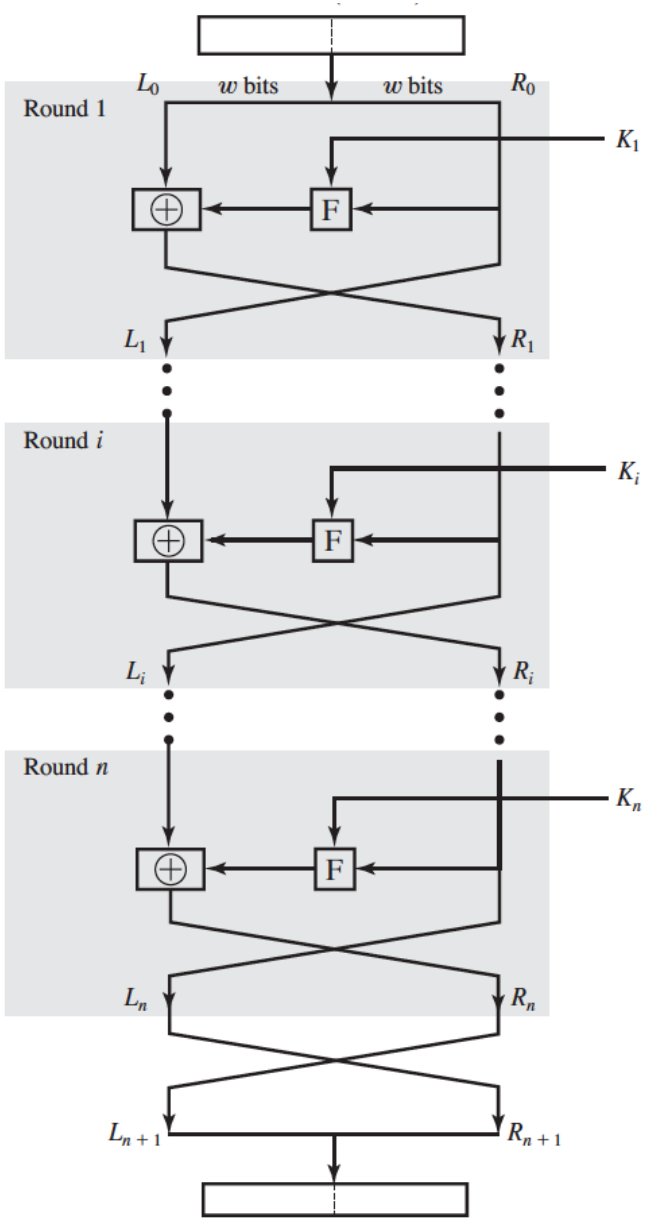
Figura: Horst Feistel, uno dei padri della crittografia a chiave simmetrica, in particolare DES.

Schema di cifratura alla Feistel



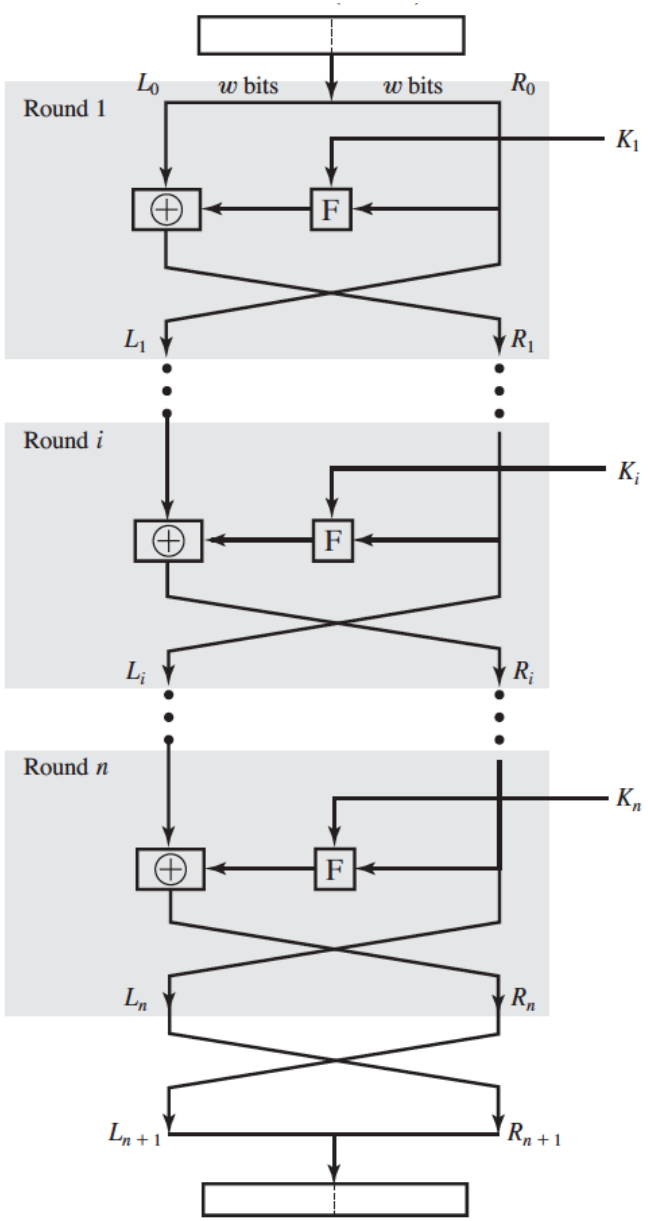
- Il simbolo F rappresenta l'algoritmo della "round function" applicata ad ogni iterazione.

Schema di cifratura alla Feistel



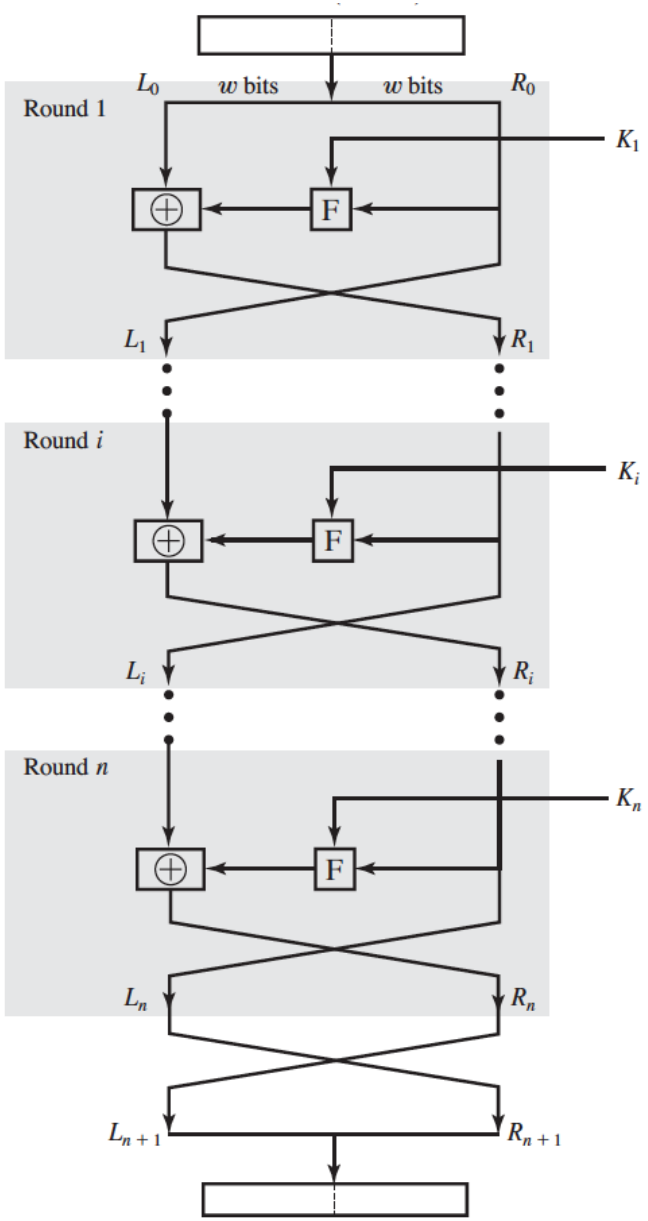
- Il simbolo F rappresenta l'algoritmo della "round function" applicata ad ogni iterazione.
- w è la lunghezza del semiblocco.

Schema di cifratura alla Feistel



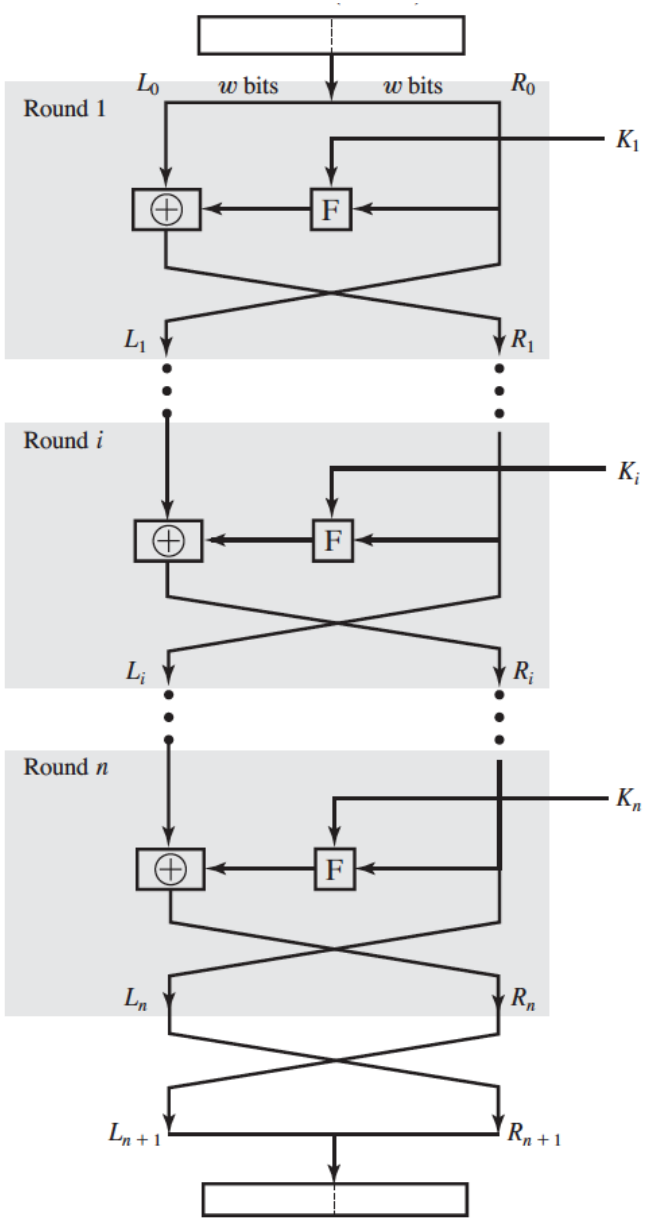
- Il simbolo F rappresenta l'algoritmo della "round function" applicata ad ogni iterazione.
- w è la lunghezza del semiblocco.
- k_i è la i-esima chiave derivata dalla chiave globale.

Schema di cifratura alla Feistel



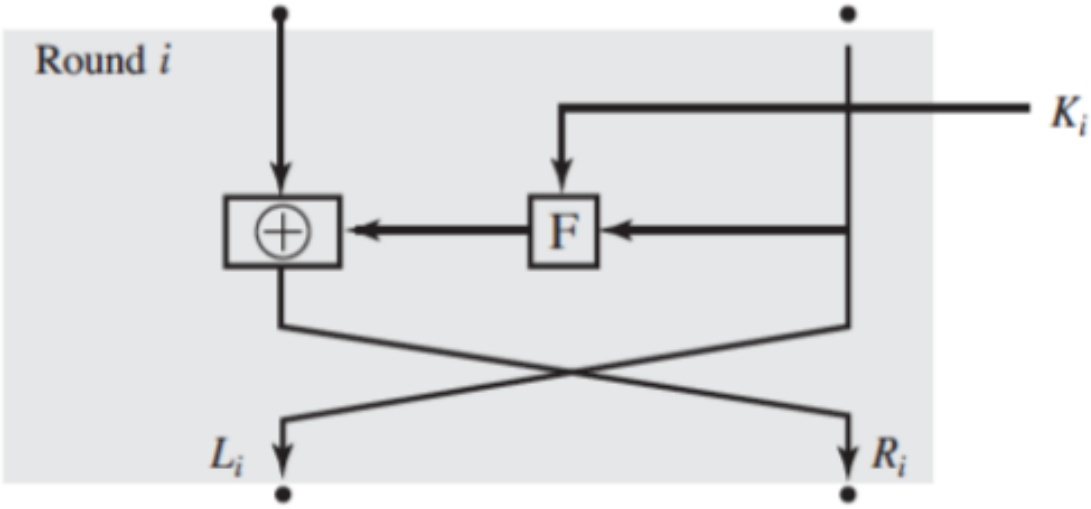
- Il simbolo F rappresenta l'algoritmo della "round function" applicata ad ogni iterazione.
- w è la lunghezza del semiblocco.
- k_i è la i-esima chiave derivata dalla chiave globale.
- L_i ed R_i sono le semisequenze sinistre e destre ad ogni iterazione.

Schema di cifratura alla Feistel



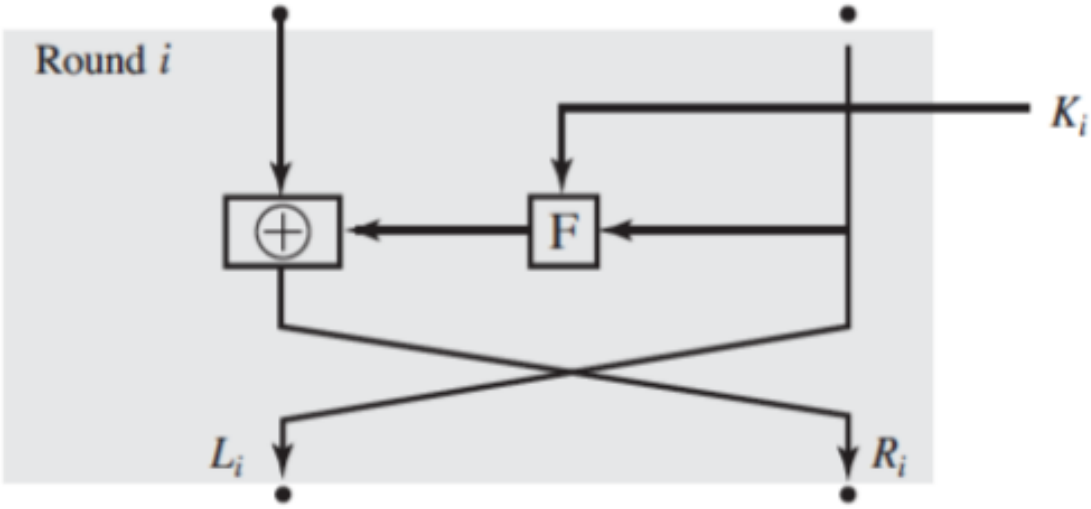
- Il simbolo F rappresenta l'algoritmo della "round function" applicata ad ogni iterazione.
- w è la lunghezza del semiblocco.
- k_i è la i-esima chiave derivata dalla chiave globale.
- L_i ed R_i sono le semisequenze sinistre e destre ad ogni iterazione.
- Il simbolo $\oplus$ indica l'or esclusivo (XOR) tra le sequenze in ingresso.

Funzione di Round



Riferendosi allo schema precedente, in generale una funzione di Round alla Feistel esegue le seguenti azioni:

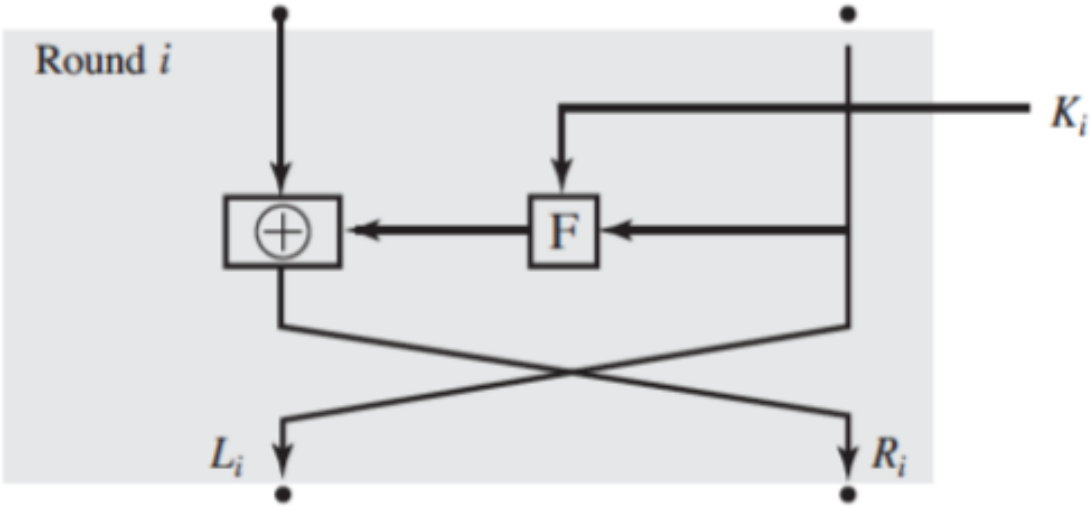
Funzione di Round



Riferendosi allo schema precedente, in generale una funzione di Round alla Feistel esegue le seguenti azioni:

- Manda la semisequenza destra nella sequenza sinistra della iterazione successiva: $L_{i+1} = R_i$.

Funzione di Round

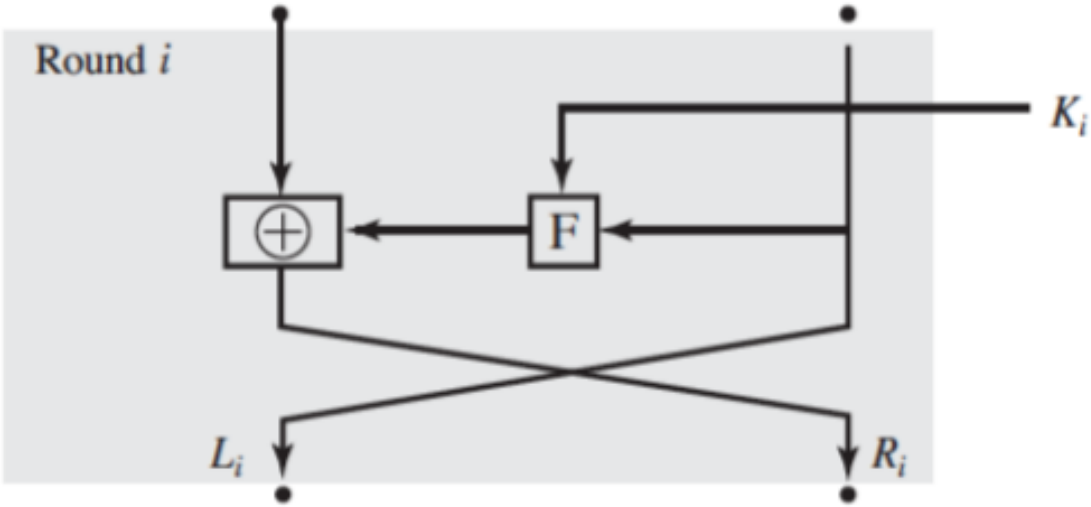


Riferendosi allo schema precedente, in generale una funzione di Round alla Feistel esegue le seguenti azioni:

- Manda la semisequenza destra nella sequenza sinistra della iterazione successiva: $L_{i+1} = R_i$.
- Cifra la semisequenza destra R_i con l'algoritmo $\mathcal{F}$ e chiave i-esima k_i.

$$R_i \to R_i' = \mathcal{F}_{k_i}(R_i).$$

Funzione di Round



Riferendosi allo schema precedente, in generale una funzione di Round alla Feistel esegue le seguenti azioni:

- Manda la semisequenza destra nella sequenza sinistra della iterazione successiva: $L_{i+1} = R_i$.
- Cifra la semisequenza destra R_i con l'algoritmo $\mathcal{F}$ e chiave i-esima k_i.

$$R_i \to R_i' = \mathcal{F}_{k_i}(R_i).$$

- Manda la sequenza ottenuta con l'o esclusivo (o aut, "xor" o ($\oplus$) della semisequenza Sinistra con la destra cifrata nella nuova sequenza destra: $R_{i+1} = R_i' \oplus L_i$.

Data Encryption Standard (DES)

Le principali caratteristiche sono:

- dimensione del blocco 64 bit $w = 32$;

Data Encryption Standard (DES)

Le principali caratteristiche sono:

- dimensione del blocco 64 bit $w = 32$;
- Si eseguono 16 round.

Data Encryption Standard (DES)

Le principali caratteristiche sono:

- dimensione del blocco 64 bit $w = 32$;
- Si eseguono 16 round.
- dimensione della chiave 56 bit. L'input è di 64 bit suddivisi i 8 gruppi da 8, ma in ciascuno si ha una ridondanza per il controllo di parità.

Data Encryption Standard (DES)

Le principali caratteristiche sono:

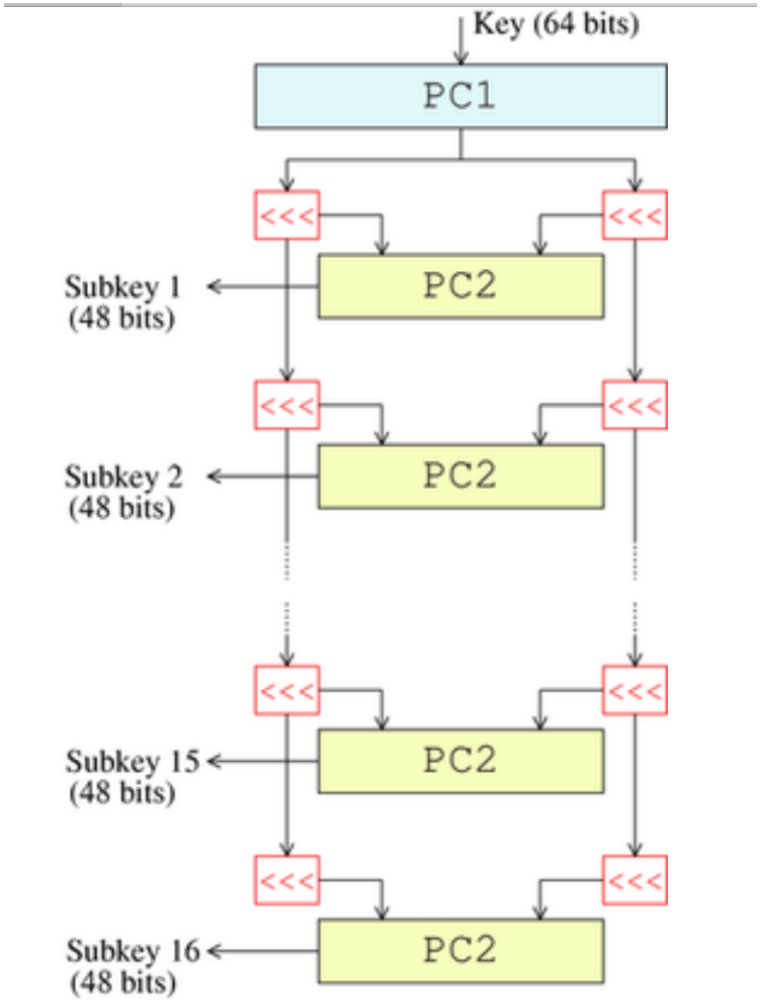
- dimensione del blocco 64 bit $w = 32$;
- Si eseguono 16 round.
- dimensione della chiave 56 bit. L'input è di 64 bit suddivisi i 8 gruppi da 8, ma in ciascuno si ha una ridondanza per il controllo di parità.
- Le chiavi intermedie vengono generate dalla chiave iniziale deterministicamente.

Data Encryption Standard (DES)

Le principali caratteristiche sono:

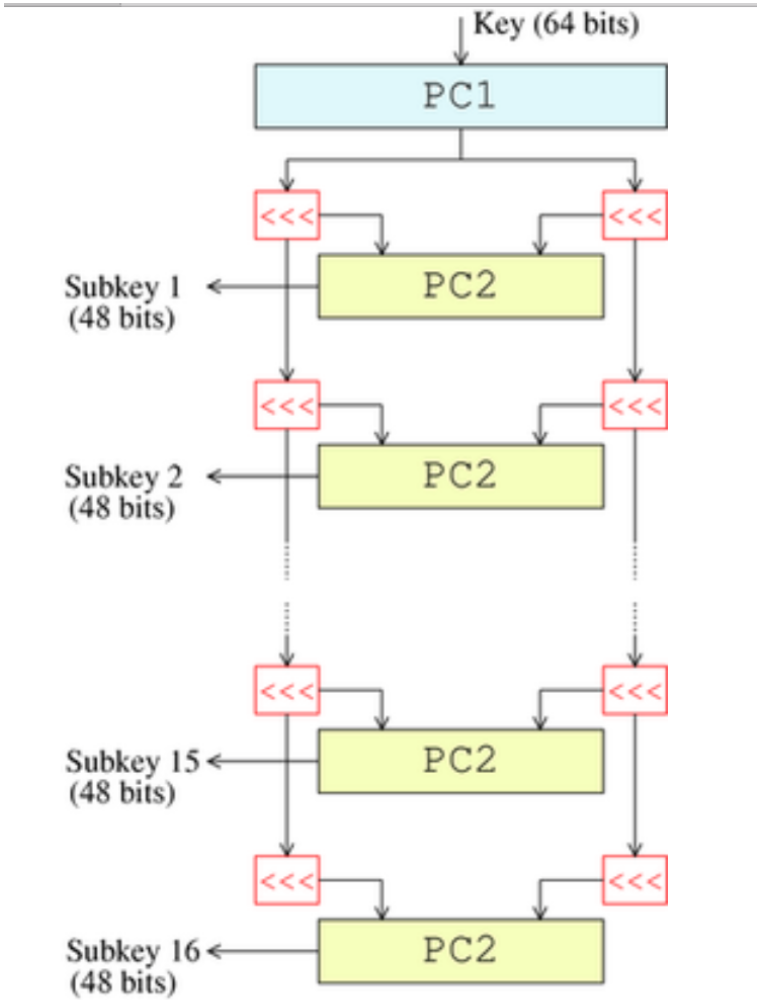
- dimensione del blocco 64 bit $w = 32$;
- Si eseguono 16 round.
- dimensione della chiave 56 bit. L'input è di 64 bit suddivisi i 8 gruppi da 8, ma in ciascuno si ha una ridondanza per il controllo di parità.
- Le chiavi intermedie vengono generate dalla chiave iniziale deterministicamente.
- La funzione di Feistel è non lineare nell'o aut ($\oplus$).

Generazione delle chiavi intermedie



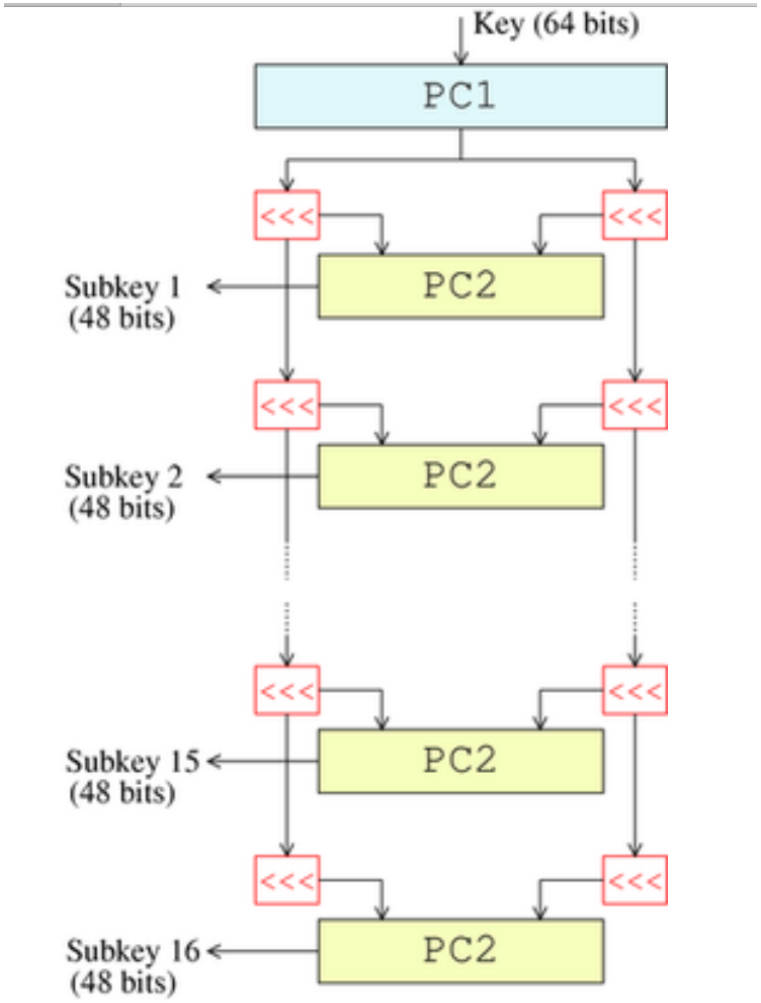
- Riferendosi allo schema generale di Feistel la parte relativa alla generazione delle chiavi utilizza un'algoritmo deterministico iterativo.

Generazione delle chiavi intermedie



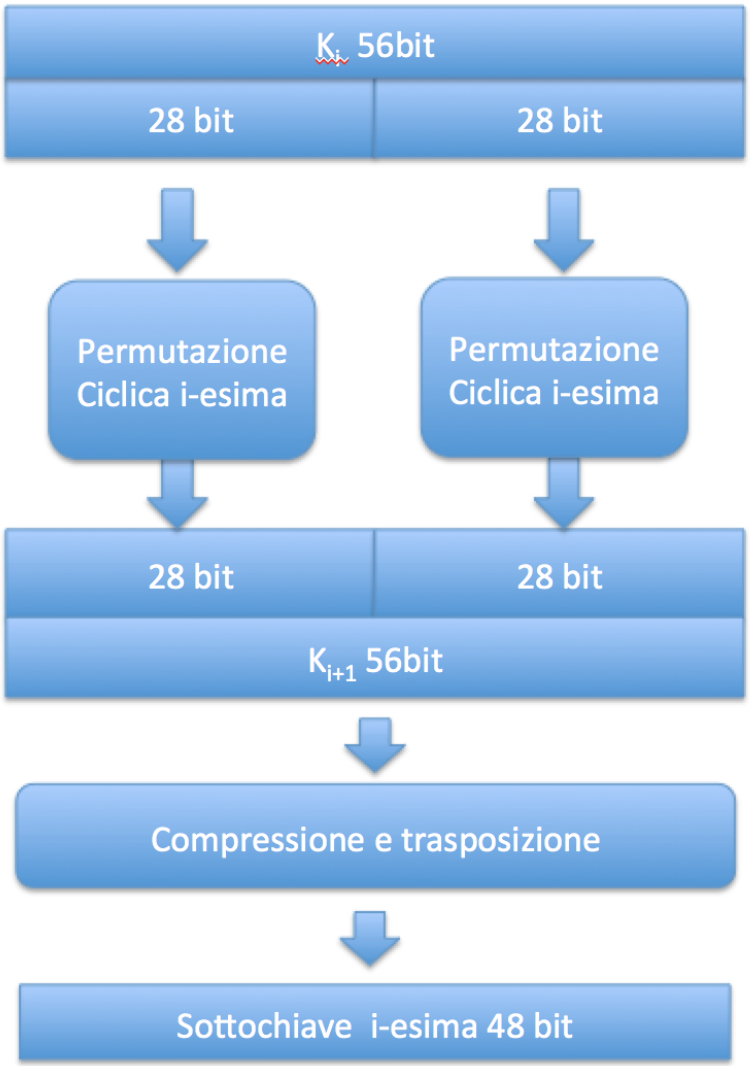
- Riferendosi allo schema generale di Feistel la parte relativa alla generazione delle chiavi utilizza un'algoritmo deterministico iterativo.
- La chiave globale di input è composta da 64 bit (quindi è una parola) , ma questa è suddivisa in 8 byte. Ogni byte ha 7 bit codificanti ed un bit di parità. Quindi in totale la chiave è da 56 bit.

Generazione delle chiavi intermedie



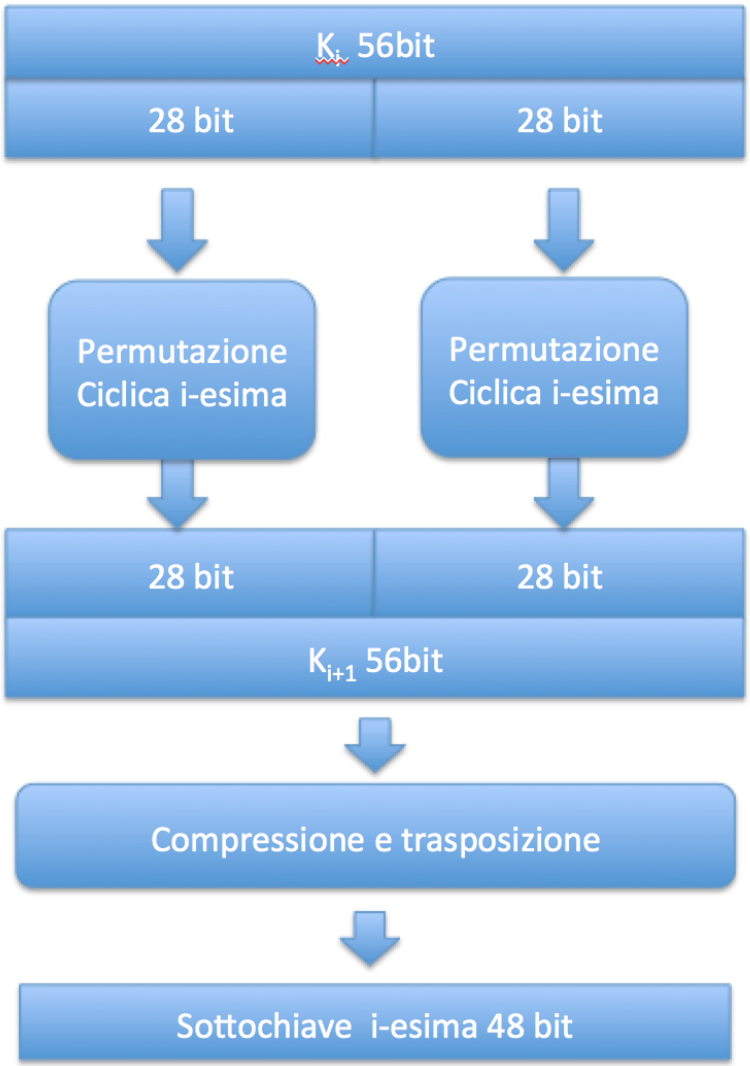
- Riferendosi allo schema generale di Feistel la parte relativa alla generazione delle chiavi utilizza un'algoritmo deterministico iterativo.
- La chiave globale di input è composta da 64 bit (quindi è una parola) , ma questa è suddivisa in 8 byte. Ogni byte ha 7 bit codificanti ed un bit di parità. Quindi in totale la chiave è da 56 bit.
- Il meccanismo di generazione delle sottochiavi è di facile esecuzione. In ognuno dei 16 passi genera una sottochiave 48 bit.

Passo iterativo per la generazione della sottochiave



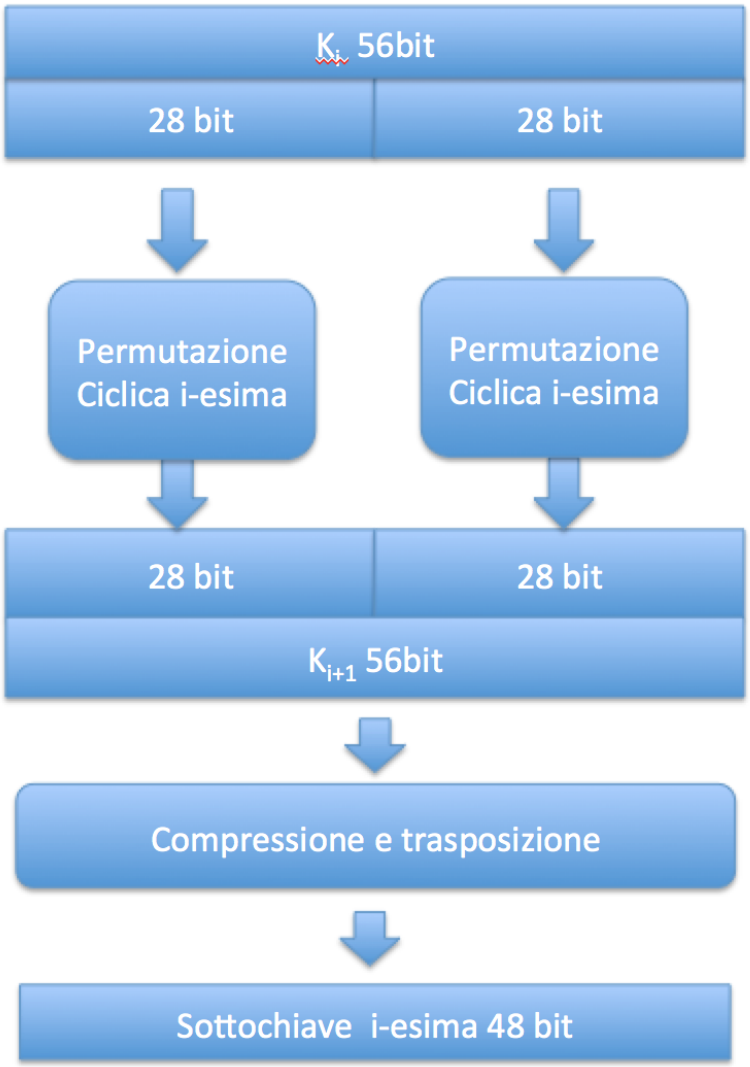
- Ad ogni iterazione la chiave di 56bit viene modificata e genera una sottochiave da 48 bit.

Passo iterativo per la generazione della sottochiave



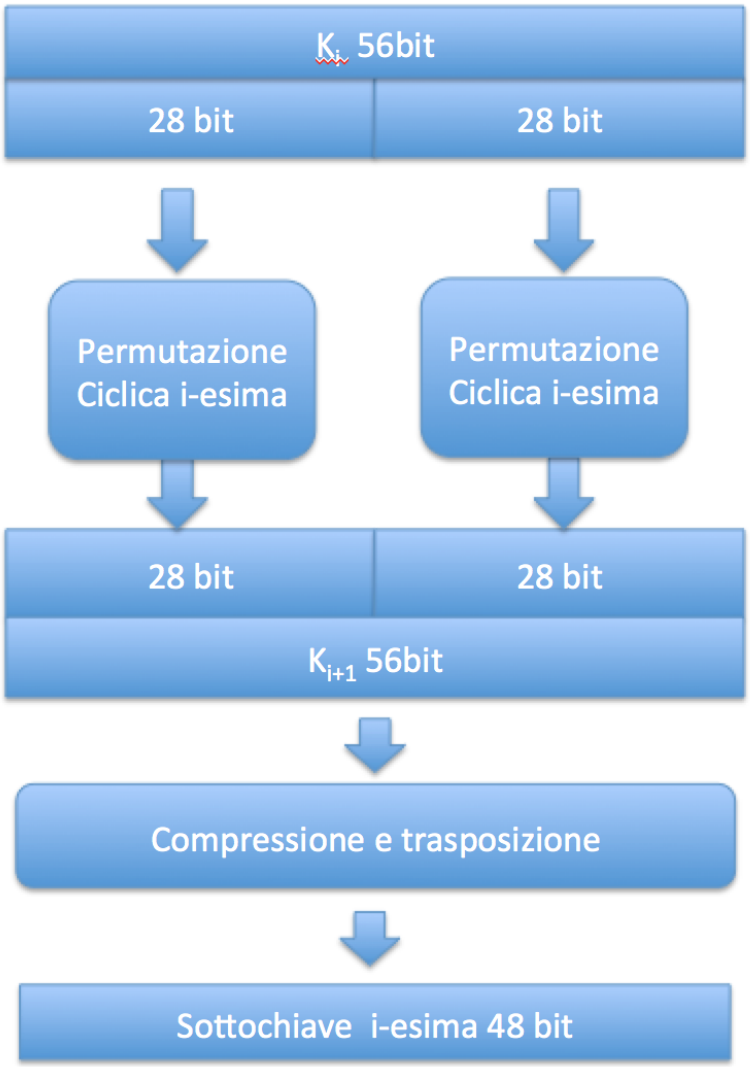
- Ad ogni iterazione la chiave di 56bit viene modificata e genera una sottochiave da 48 bit.
- Il modulo di ciclicità applica una permutazione ciclica di un passo (scorrimento ciclico) in tutti i round tranne in quelli 1,2,9 e 16 in cui applica due volte.

Passo iterativo per la generazione della sottochiave



- Ad ogni iterazione la chiave di 56bit viene modificata e genera una sottochiave da 48 bit.
- Il modulo di ciclicità applica una permutazione ciclica di un passo (scorrimento ciclico) in tutti i round tranne in quelli 1,2,9 e 16 in cui applica due volte.
- La nuova chiave da 56 diverrà input per l'iterazione successiva.

Passo iterativo per la generazione della sottochiave



- Ad ogni iterazione la chiave di 56bit viene modificata e genera una sottochiave da 48 bit.
- Il modulo di ciclicità applica una permutazione ciclica di un passo (scorrimento ciclico) in tutti i round tranne in quelli 1,2,9 e 16 in cui applica due volte.
- La nuova chiave da 56 diverrà input per l'iterazione successiva.
- Il modulo di compressione e trasposizione è descritto tramite una tabella.

Modulo di compressione e trasposizione

14	17	11	24	01	05
03	28	15	06	21	10
23	19	12	04	26	19
16	07	27	20	13	02
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

b_{14}	b_{17}	b_{11}	b_{24}	b_{01}	b_{05}
b_{03}	b_{28}	b_{15}	b_{06}	b_{21}	b_{10}
b_{23}	b_{19}	b_{12}	b_{04}	b_{26}	b_{19}
b_{16}	b_{07}	b_{27}	b_{20}	b_{13}	b_{02}
b_{41}	b_{52}	b_{31}	b_{37}	b_{47}	b_{55}
b_{30}	b_{40}	b_{51}	b_{45}	b_{33}	b_{48}
b_{44}	b_{49}	b_{39}	b_{56}	b_{34}	b_{53}
b_{46}	b_{42}	b_{50}	b_{36}	b_{29}	b_{32}

- La tabella a sinistra va letta come segue:

Modulo di compressione e trasposizione

14	17	11	24	01	05
03	28	15	06	21	10
23	19	12	04	26	19
16	07	27	20	13	02
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

b_{14}	b_{17}	b_{11}	b_{24}	b_{01}	b_{05}
b_{03}	b_{28}	b_{15}	b_{06}	b_{21}	b_{10}
b_{23}	b_{19}	b_{12}	b_{04}	b_{26}	b_{19}
b_{16}	b_{07}	b_{27}	b_{20}	b_{13}	b_{02}
b_{41}	b_{52}	b_{31}	b_{37}	b_{47}	b_{55}
b_{30}	b_{40}	b_{51}	b_{45}	b_{33}	b_{48}
b_{44}	b_{49}	b_{39}	b_{56}	b_{34}	b_{53}
b_{46}	b_{42}	b_{50}	b_{36}	b_{29}	b_{32}

- La tabella a sinistra va letta come segue:
- I 56 bit di input $b_1, b_2, \ldots, b_{56}$ danno luogo ad un output di 48 bit composto dalla sequenza dei bit b_i con indice l'elemento della tabella.

Modulo di compressione e trasposizione

14	17	11	24	01	05
03	28	15	06	21	10
23	19	12	04	26	19
16	07	27	20	13	02
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

b_{14}	b_{17}	b_{11}	b_{24}	b_{01}	b_{05}
b_{03}	b_{28}	b_{15}	b_{06}	b_{21}	b_{10}
b_{23}	b_{19}	b_{12}	b_{04}	b_{26}	b_{19}
b_{16}	b_{07}	b_{27}	b_{20}	b_{13}	b_{02}
b_{41}	b_{52}	b_{31}	b_{37}	b_{47}	b_{55}
b_{30}	b_{40}	b_{51}	b_{45}	b_{33}	b_{48}
b_{44}	b_{49}	b_{39}	b_{56}	b_{34}	b_{53}
b_{46}	b_{42}	b_{50}	b_{36}	b_{29}	b_{32}

- La tabella a sinistra va letta come segue:
- I 56 bit di input $b_1, b_2, \ldots, b_{56}$ danno luogo ad un output di 48 bit composto dalla sequenza dei bit b_i con indice l'elemento della tabella.
- Ovviamente 12 bit non saranno presenti in uscita.

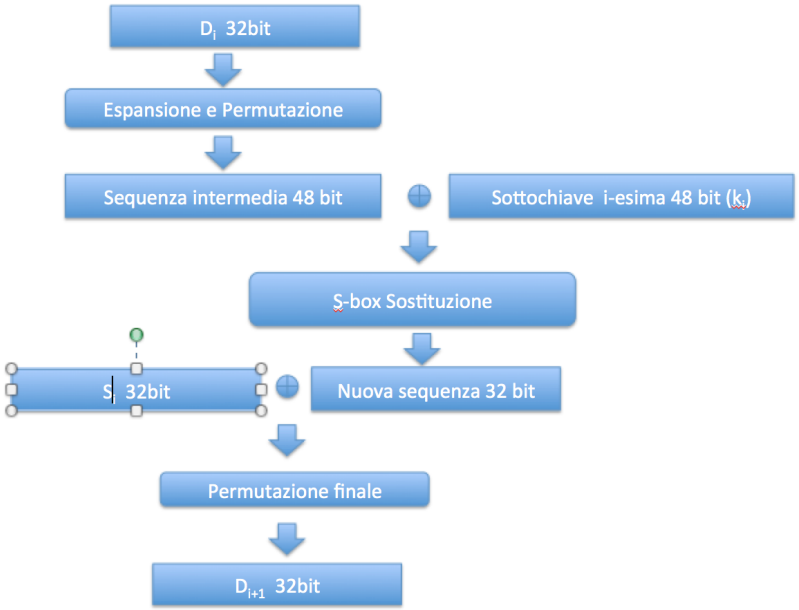
Modulo di compressione e trasposizione

14	17	11	24	01	05
03	28	15	06	21	10
23	19	12	04	26	19
16	07	27	20	13	02
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

b_{14}	b_{17}	b_{11}	b_{24}	b_{01}	b_{05}
b_{03}	b_{28}	b_{15}	b_{06}	b_{21}	b_{10}
b_{23}	b_{19}	b_{12}	b_{04}	b_{26}	b_{19}
b_{16}	b_{07}	b_{27}	b_{20}	b_{13}	b_{02}
b_{41}	b_{52}	b_{31}	b_{37}	b_{47}	b_{55}
b_{30}	b_{40}	b_{51}	b_{45}	b_{33}	b_{48}
b_{44}	b_{49}	b_{39}	b_{56}	b_{34}	b_{53}
b_{46}	b_{42}	b_{50}	b_{36}	b_{29}	b_{32}

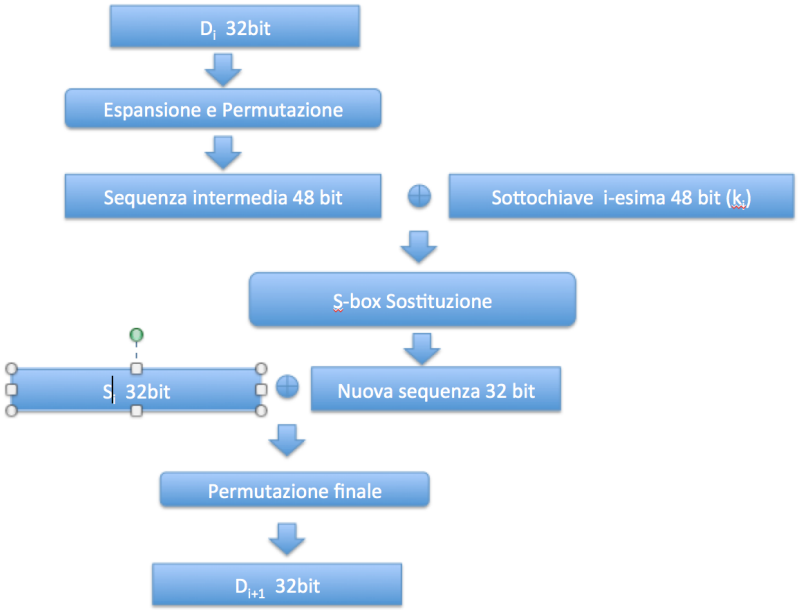
- La tabella a sinistra va letta come segue:
- I 56 bit di input $b_1, b_2, \ldots, b_{56}$ danno luogo ad un output di 48 bit composto dalla sequenza dei bit b_i con indice l'elemento della tabella.
- Ovviamente 12 bit non saranno presenti in uscita.
- La tabella utilizzata è sempre la stessa per ogni round.

Vediamo adesso la funzione centrale di Feistel



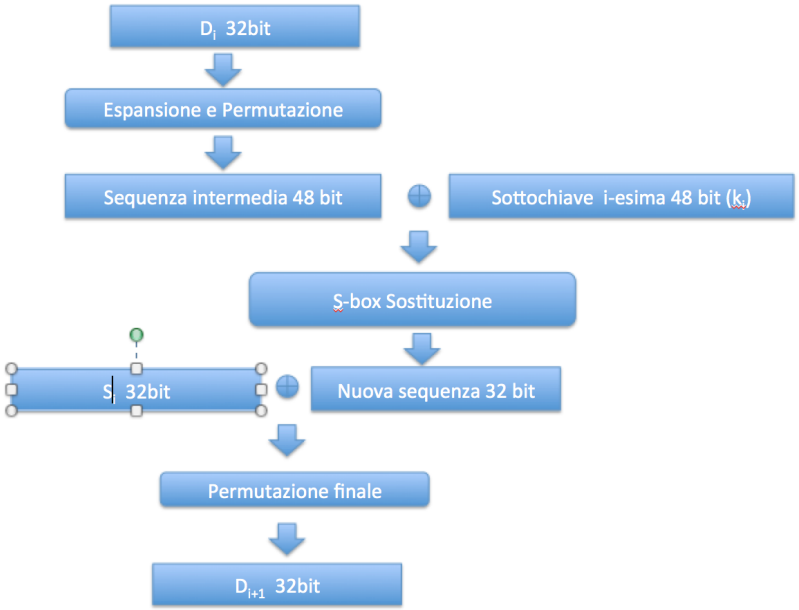
- I dati in ingresso sono: i due semi blocchi S_i e D_i da 32 bit dell'iterazione precedente e la sottochiave i-esima k_i da 48 bit.

Vediamo adesso la funzione centrale di Feistel



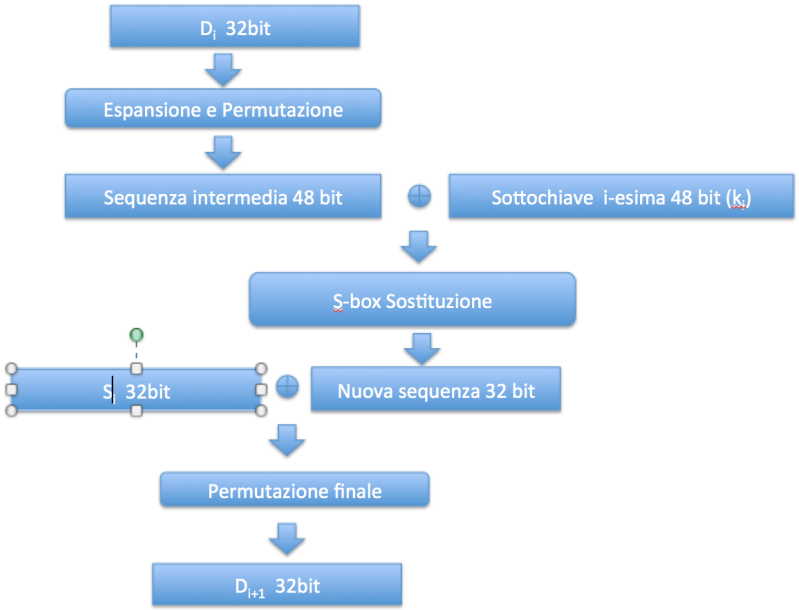
- I dati in ingresso sono: i due semi blocchi S_i e D_i da 32 bit dell'iterazione precedente e la sottochiave i-esima k_i da 48 bit.
- Il dato in uscita è il blocco destro per l'terazione successiva.

Vediamo adesso la funzione centrale di Feistel



- I dati in ingresso sono: i due semi blocchi S_i e D_i da 32 bit dell'iterazione precedente e la sottochiave i-esima k_i da 48 bit.
- Il dato in uscita è il blocco destro per l'terazione successiva.
- I moduli di espansione e permutazione sono definiti tramite tabelle.

Vediamo adesso la funzione centrale di Feistel



- I dati in ingresso sono: i due semi blocchi S_i e D_i da 32 bit dell'iterazione precedente e la sottochiave i-esima k_i da 48 bit.
- Il dato in uscita è il blocco destro per l'terazione successiva.
- I moduli di espansione e permutazione sono definiti tramite tabelle.
- Il modulo di sostituzione è il cuore dell'algorimo

Modulo di espansione e permutazione

32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	01

- Riceve in ingresso una sequenza di 32 bit

Modulo di espansione e permutazione

32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	01

- Riceve in ingresso una sequenza di 32 bit
- Fornisce in uscita una sequenza di 48 bit

Modulo di espansione e permutazione

32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	01

- Riceve in ingresso una sequenza di 32 bit
- Fornisce in uscita una sequenza di 48 bit
- Alcuni bit sono ripetuti.

Modulo di espansione e permutazione

32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	01

- Riceve in ingresso una sequenza di 32 bit
- Fornisce in uscita una sequenza di 48 bit
- Alcuni bit sono ripetuti.
- Nessun bit conserva la posizione.

Modulo di permutazione finale

- Si definisce tramite la tabella:

16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
30	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

Modulo di permutazione finale

- Si definisce tramite la tabella:

16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
30	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

- Riceve in ingresso una sequenza di 32 bit

Modulo di permutazione finale

- Si definisce tramite la tabella:

16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
30	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

- Riceve in ingresso una sequenza di 32 bit
- Fornisce in uscita una sequenza di 32 bit

Modulo di permutazione finale

- Si definisce tramite la tabella:

16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
30	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

- Riceve in ingresso una sequenza di 32 bit
- Fornisce in uscita una sequenza di 32 bit
- Si tratta di una permutazione quindi non ci sono bit ripetuti.

Modulo di permutazione finale

- Si definisce tramite la tabella:

16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
30	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

- Riceve in ingresso una sequenza di 32 bit
- Fornisce in uscita una sequenza di 32 bit
- Si tratta di una permutazione quindi non ci sono bit ripetuti.
- Nessun bit conserva la posizione.

Modulo di sostituzione

- Riceve in ingresso una sequenza di 48 bit divisi in 8 blocchi da 6 bit $s_1, s_2, \ldots s_8$.

x \y	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
01	00	15	07	04	14	02	13	01	10	06	12	11	09	05	03	08
02	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
03	15	12	09	02	04	09	01	07	05	11	03	14	10	00	06	13

Modulo di sostituzione

- Riceve in ingresso una sequenza di 48 bit divisi in 8 blocchi da 6 bit $s_1, s_2, \ldots s_8$.

x \y	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
01	00	15	07	04	14	02	13	01	10	06	12	11	09	05	03	08
02	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
03	15	12	09	02	04	09	01	07	05	11	03	14	10	00	06	13

Modulo di sostituzione

- Riceve in ingresso una sequenza di 48 bit divisi in 8 blocchi da 6 bit $s_1, s_2, \ldots s_8$.
- Fornisce in uscita una sequenza di 32 bit

x \y	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
01	00	15	07	04	14	02	13	01	10	06	12	11	09	05	03	08
02	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
03	15	12	09	02	04	09	01	07	05	11	03	14	10	00	06	13

Modulo di sostituzione

- Riceve in ingresso una sequenza di 48 bit divisi in 8 blocchi da 6 bit $s_1, s_2, \ldots s_8$.
- Fornisce in uscita una sequenza di 32 bit
- Ogni blocco da 6 bit è trattato separatamente. Si estraggono i primi due bit $b_1 b_2$ (oppure primo ed ultimo) ed i successivi 4 $b_3 b_4 b_5 b_6$. Si costruiscono $x = b_1 + 2b_2$ e $y = b_3 + 2b_4 + 4b_5 + 8b_6$. $0 \leq x \leq 3$ e $0 \leq y \leq 15$

x \y	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
01	00	15	07	04	14	02	13	01	10	06	12	11	09	05	03	08
02	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
03	15	12	09	02	04	09	01	07	05	11	03	14	10	00	06	13

Modulo di sostituzione

- Riceve in ingresso una sequenza di 48 bit divisi in 8 blocchi da 6 bit $s_1, s_2, \ldots s_8$.
- Fornisce in uscita una sequenza di 32 bit
- Ogni blocco da 6 bit è trattato separatamente. Si estraggono i primi due bit $b_1 b_2$ (oppure primo ed ultimo) ed i successivi 4 $b_3 b_4 b_5 b_6$. Si costruiscono $x = b_1 + 2b_2$ e $y = b_3 + 2b_4 + 4b_5 + 8b_6$. $0 \leq x \leq 3$ e $0 \leq y \leq 15$
- Per ogni blocco s_i si utilizza una tablella diversa. Ad esempio, per il primo blocco (s_1) ad ogni coppia (x,y) si associa l'elemento di posizione x, y nella seguente tablella:

x \y	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
01	00	15	07	04	14	02	13	01	10	06	12	11	09	05	03	08
02	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
03	15	12	09	02	04	09	01	07	05	11	03	14	10	00	06	13

Tabelle sostituizionali 1-4

S1	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
01	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
02	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
03	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S2	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
01	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
02	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
03	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S3	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
01	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
02	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
03	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S4	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
01	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
02	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
03	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

Tabelle sostituizionali 5-8

S5	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
01	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
02	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
03	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S6	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
01	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
02	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
03	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S7	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
01	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
02	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
03	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S8	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
01	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
02	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
03	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

3DES Triple Data Encryption Standard

- Nel 1998 la Electronic Frontier Foundation (EFF) creò il DES cracker , una macchina che riuscì con grandi allocazioni computazionali (per quei tempi) a violare il cifrario DES con il metodo della forza bruta (ispezione casuale).

3DES Triple Data Encryption Standard

- Nel 1998 la Electronic Frontier Foundation (EFF) creò il DES cracker , una macchina che riuscì con grandi allocazioni computazionali (per quei tempi) a violare il cifrario DES con il metodo della forza bruta (ispezione casuale).
- Per questo motivo, la chiave a 56 bit venne ritenuta poco complessa e quindi fu introdotto un nuovo standard 3DES che usa tre chiavi da 56 bit successive ($k = \{k_1, k_2, k_3\}$).

3DES Triple Data Encryption Standard

- Nel 1998 la Electronic Frontier Foundation (EFF) creò il DES cracker , una macchina che riuscì con grandi allocazioni computazionali (per quei tempi) a violare il cifrario DES con il metodo della forza bruta (ispezione casuale).
- Per questo motivo, la chiave a 56 bit venne ritenuta poco complessa e quindi fu introdotto un nuovo standard 3DES che usa tre chiavi da 56 bit successive ($k = \{k_1, k_2, k_3\}$).
- La sequenza crittata si ottiene reiterando il processo alla base di DES con 3 chiavi diverse. In realtà si usa due volte l'algoritmo diretto intramezzato con l'algoritmo inverso $\mathcal{D}$.

$$c = \mathcal{C}_{k_3}(\mathcal{D}_{k_2}(\mathcal{C}_{k_1}(t))).$$

3DES Triple Data Encryption Standard

- Nel 1998 la Electronic Frontier Foundation (EFF) creò il DES cracker , una macchina che riuscì con grandi allocazioni computazionali (per quei tempi) a violare il cifrario DES con il metodo della forza bruta (ispezione casuale).
- Per questo motivo, la chiave a 56 bit venne ritenuta poco complessa e quindi fu introdotto un nuovo standard 3DES che usa tre chiavi da 56 bit successive ($k = \{k_1, k_2, k_3\}$).
- La sequenza crittata si ottiene reiterando il processo alla base di DES con 3 chiavi diverse. In realtà si usa due volte l'algoritmo diretto intramezzato con l'algoritmo inverso $\mathcal{D}$.

$$c = \mathcal{C}_{k_3}(\mathcal{D}_{k_2}(\mathcal{C}_{k_1}(t))).$$

- Questo garantisce la continuità con la crittazione DES perché basta usare tre volte la stessa chiave: $(k = \{k_1, k_1, k_1\})$

$$c = \mathcal{C}_{k_1}(\mathcal{D}_{k_1}(\mathcal{C}_{k_1}(t))) \equiv \mathcal{C}_{k_1}(t).$$

AES Triple Data Encryption Standard

- Attualmente si è imposto l'Advanced Encryption Standard.

AES Triple Data Encryption Standard

- Attualmente si è imposto l'Advanced Encryption Standard.
- A partire dal 2001 il NIST dopo una gara vinta dal belga Vincent Rijndael, ha definito 3 standard con chiavi di diversa misura: 128, 192 e 256 bit, chiamati DES128, DES192 e DES256. Il 3DES ha una chiave equivalente da 3x56=168 bit.

AES Triple Data Encryption Standard

- Attualmente si è imposto l'Advanced Encryption Standard.
- A partire dal 2001 il NIST dopo una gara vinta dal belga Vincent Rijndael, ha definito 3 standard con chiavi di diversa misura: 128, 192 e 256 bit, chiamati DES128, DES192 e DES256. Il 3DES ha una chiave equivalente da 3x56=168 bit.
- Il numero di round dipende dalla dimensione della chiave ed è pari a 10, 12 e 14 (DES256).

AES Triple Data Encryption Standard

- Attualmente si è imposto l'Advanced Encryption Standard.
- A partire dal 2001 il NIST dopo una gara vinta dal belga Vincent Rijndael, ha definito 3 standard con chiavi di diversa misura: 128, 192 e 256 bit, chiamati DES128, DES192 e DES256. Il 3DES ha una chiave equivalente da 3x56=168 bit.
- Il numero di round dipende dalla dimensione della chiave ed è pari a 10, 12 e 14 (DES256).
- Le informazioni su DES ed AES sono disponibili sul Stillings-Brown cap 20.3.

AES Triple Data Encryption Standard

- Attualmente si è imposto l'Advanced Encryption Standard.
- A partire dal 2001 il NIST dopo una gara vinta dal belga Vincent Rijndael, ha definito 3 standard con chiavi di diversa misura: 128, 192 e 256 bit, chiamati DES128, DES192 e DES256. Il 3DES ha una chiave equivalente da 3x56=168 bit.
- Il numero di round dipende dalla dimensione della chiave ed è pari a 10, 12 e 14 (DES256).
- Le informazioni su DES ed AES sono disponibili sul Stillings-Brown cap 20.3.
- La definizione dello standard EAS si trova al NIST.FIPS.197

Messaggio

- Gran parte della cifratura utilizzata nella comunicazione dei dati è basata sulla cifratura a blocchi.

Messaggio

- Gran parte della cifratura utilizzata nella comunicazione dei dati è basata sulla cifratura a blocchi.
- Uno schema fondamentale della cifratura a blocchi è quello di Feistel. Le cifrature DES Data Encryption Standard e 3DES Triple Data Encryption Standard esempi molto diffusi di cifratura alla Feistel.

Messaggio

- Gran parte della cifratura utilizzata nella comunicazione dei dati è basata sulla cifratura a blocchi.
- Uno schema fondamentale della cifratura a blocchi è quello di Feistel. Le cifrature DES Data Encryption Standard e 3DES Triple Data Encryption Standard esempi molto diffusi di cifratura alla Feistel.
- Un miglioramento delle cifrature DES e 3DES è dato dalla AES (Advanced Encryption Standard) che dal 2011 ha sostituito in gran parte il DES.

Messaggio

- Gran parte della cifratura utilizzata nella comunicazione dei dati è basata sulla cifratura a blocchi.
- Uno schema fondamentale della cifratura a blocchi è quello di Feistel. Le cifrature DES Data Encryption Standard e 3DES Triple Data Encryption Standard esempi molto diffusi di cifratura alla Feistel.
- Un miglioramento delle cifrature DES e 3DES è dato dalla AES (Advanced Encryption Standard) che dal 2011 ha sostituito in gran parte il DES.
- La robustezza delle cifrature a blocchi è data dalla complessità della chiave e dalla presenza di moduli di sostituzione non lineari rispetto al connettivo "o esclusivo" ($\oplus$) come la funzione di Feistel.