Algoritmi di criptografia a chiave simmetrica

Cifrari a blocchi

Criptografia Simmetrica

- Gli algoritmi di criptografia a chiave simmetrica, o chiave segreta, sono preferiti perché, più è corta la chiave, maggiore è la velocità di elaborazione.
  - Gli algoritmi a chiave simmetrica si basano su semplici operazioni matematiche, che l'hardware esegue con rapidità.
  - La criptografia a chiave simmetrica è usata per proteggere i dati in una VPN.

Gestione delle chiavi segrete

- La gestione delle chiavi è rischiosa perché sono uguali.
- La sicurezza di un algoritmo a chiave simmetrica si basa sulla segretezza della chiave.
  - Dopo aver ottenuto una chiave, si possono criptare e decriptare messaggi.
  - Il mittente e il ricevitore devono scambiarsi la chiave segreta usando un canale sicuro, prima di poter criptare (quindi la chiave viaggerebbe in chiaro).

Gestione delle chiavi segrete

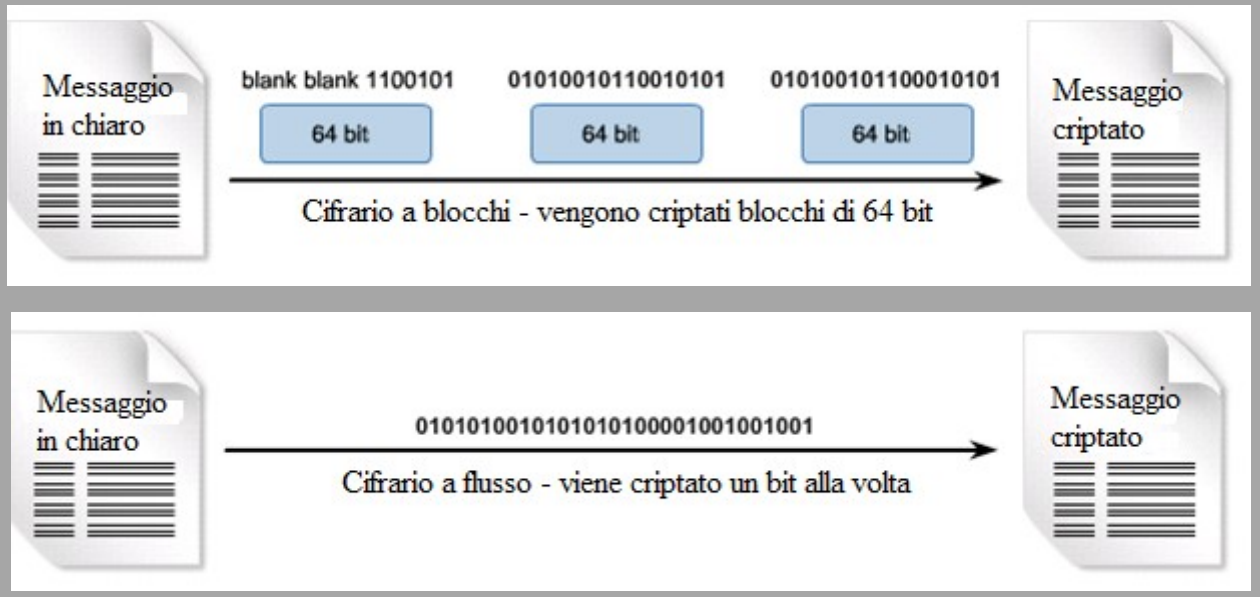
- I più comuni algoritmi che usano la criptografia a chiave simmetrica sono:
  - DES
  - 3DES
  - AES
  - Software Encryption Algorithm (SEAL)
  - Cifrario di Rivest (RC) e le serie (RC2, RC4, RC5 e RC6)
- Altri algoritmi di criptografia a chiave simmetrica sono Blowfish, Twofish, Threefish, e Serpent.
  - Questi comunque non sono stati ancora approvati.

Algoritmi di criptografia a chiave simmetrica

Algoritmi di criptografia a chiave simmetrica	Lunghezza chiave (in bit)	Descrizione
DES	56	Anche se piuttosto vecchio, il DES è ancora usato. Il DES fu progettato per essere realizzato in hardware, quindi è molto lento se usato da un software.
3DES	112 and 168	Cripta i dati tre volte con il DES, quindi si ritiene che sia molto più resistente agli attacchi rispetto al DES. É molto lento in confronto a cifrari a blocchi, quale ad esempio AES.
AES	128, 192, and 256	AES è molto veloce sia nella versione software sia nella versione hardware, è facile da realizzare e richiede poca memoria.
Software Encryption Algorithm (SEAL)	160	SEAL é un algoritmo alternativo a DES, 3DES, ed AES. Usa chiavi di 160 bit e, nella versione software, incide poco sul tempo di calcolo rispetto agli altri algoritmi.
RC	RC2 (40 and 64) RC4 (1 to 256) RC5 (0 to 2040) RC6 (128, 192, and 256)	Gli algoritmi della serie RC sono una combinazioni di algoritmi a chiave simmetrica e asimmetrica progettati da Ron Rivest. RC1 non è mai stato pubblicato e RC3 è stato violato prima ancora di essere usato. RC4 è un cifrario a blocchi ampiamente usato. RC6, un cifrario a blocchi di 28 bit

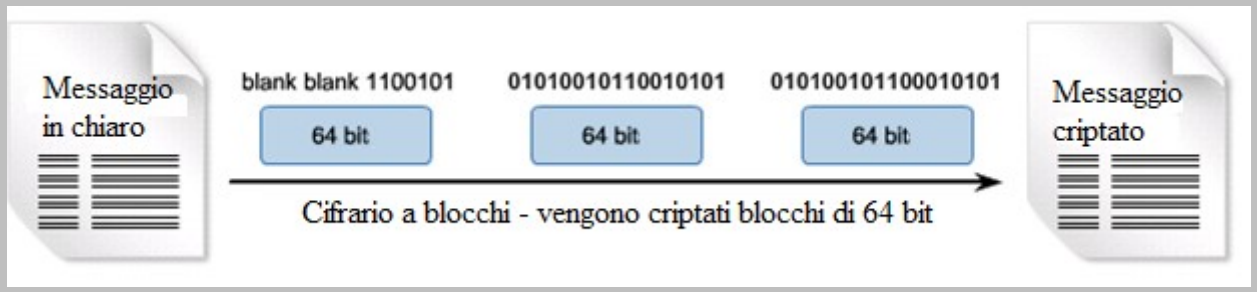
Tecniche di criptografia a chiave simmetrica

- Esistono due metodi di criptografia a chiave simmetrica:
  - Cifrario a blocchi
  - Cifrario sul flusso dei bit (Stream)



Cifrario a blocchi

- Il cifrario a Blocchi trasforma blocchi di testo in chiaro della stessa lunghezza in blocchi di testo cifrato di 64 o 128 bit.
  - La dimensione del blocco è il numero di bit del testo in chiaro che vengono criptati ad ogni passo dell'algoritmo.
  - La lunghezza della chiave, cioè il numero di bit, determina la *robustezza* del metodo di cifratura.
  - Il testo cifrato viene decriptato applicando la trasformazione inversa al blocco cifrato, usando la stessa chiave segreta.
- Esempi di cifrari a blocchi:
  - DES con un blocco di 64 bit
  - AES con un blocco di 128 bit
  - RSA con un blocco di dimensione variabile



Cifrario a blocchi

- In un cifrario a blocchi, il testo in chiaro viene diviso in blocchi di k bit. Se k = 64, il messaggio viene suddiviso in blocchi di lunghezza 64 bit, ed ogni blocco viene cifrato indipendentemente dagli altri.
- Per codificare un blocco, il cifrario usa una corrispondenza biunivoca che viene stabilita tra i blocchi di k bit del testo in chiaro e i blocchi di k bit del testo cifrato. Ad esempio, se k=3, il cifrario stabilisce una corrispondenza tra 3 bit del testo in chiaro e 3 bit del testo cifrato. La tabella mostra un esempio:

Input	Output
000	110
001	111
010	101
011	100
100	011
101	010
110	000
111	001

Cifrario a blocchi

- Esiste una corrispondenza biunivoca tra i codici di input e i codici di output, cioè per ogni codice di output esiste uno ed un solo codice di input e, viceversa, per ogni codice di input esiste uno ed un solo codice di output. Il cifrario viene usato per codificare blocchi di 3 bit del testo in chiaro.

  il testo in chiaro 010110001111

  viene criptato in 101000111001.
- Gli 8 codici che si possono formare con 3 bit, possono essere permutati in 8!=40320 modi, ottenendo, quindi, 40320 possibili tabelle di codifica.
- Una tabella di codifica costituisce la **chiave** con cui due interlocutori possono criptare e decriptare i messaggi che si scambiano.
- Quesito: quanto è lunga questa chiave? Si noti che la prima colonna è una numerazione progressiva, quindi non è necessario memorizzarla

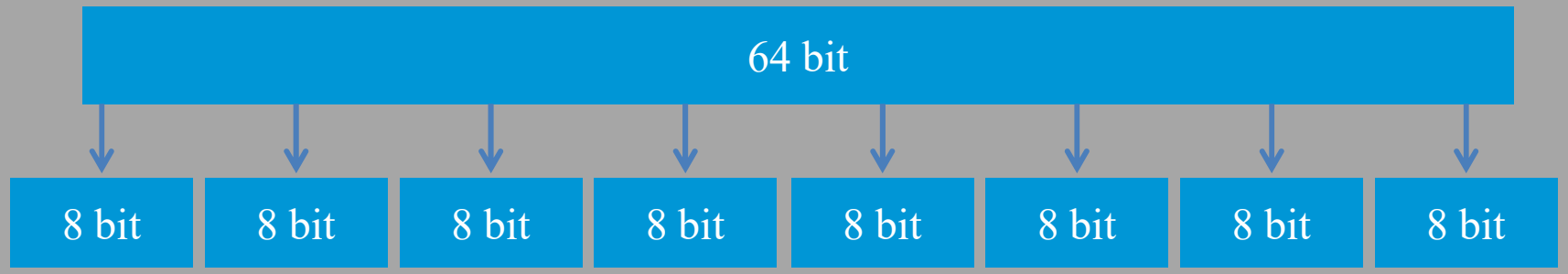
Input	Output
000	110
001	111
010	101
011	100
100	011
101	010
110	000
111	001

Cifrario a blocchi

- L'attacco di forza bruta a questo cifrario consiste nel tentare di decriptare il testo cifrato usando tutte le possibili tabelle di corrispondenza.
- Con il cifrario di esempio, con k=3, non è richiesto molto tempo per provare le 40320 possibili tabelle di codifica. Per contrastare l'attacco di forza bruta, il cifrario a blocchi usa blocchi con k=64 o oltre.
- Il numero di possibili tabelle di codifica, o il numero di possibili chiavi è $2^k!$, che è un numero molto grande anche per piccoli valori di k.
- Per creare un cifrario con blocchi di 64 bit, i due interlocutori devono memorizzare una tabella di corrispondenza con 2^{64} righe: una dimensione improponibile!
- Per cambiare la chiave, si deve rigenerare la tabella di codifica.
- Di conseguenza, memorizzare la tabella completa di un cifrario a blocchi è un'idea da scartare.

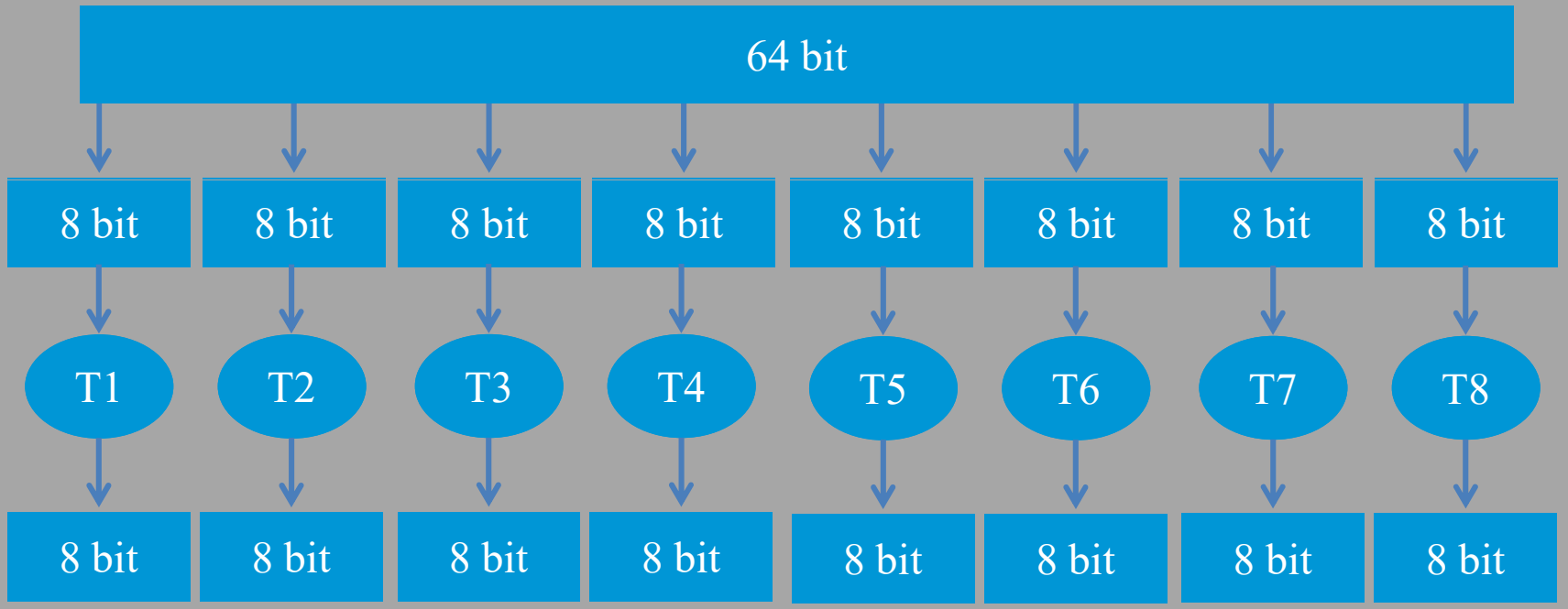
Cifrario a blocchi

- Un cifrario a blocchi, anziché usare una tabella di codifica completa, usa funzioni per calcolare le permutazioni della tabella.
- La funzione scompone un blocco di 64 bit in 8 pezzi di 8 bit. Ogni pezzo di 8 bit è elaborato da un tabella che a ogni gruppo di 8 bit fa corrispondere un altro gruppo di 8 bit. Questa volta la tabella ha dimensioni ragionevoli.



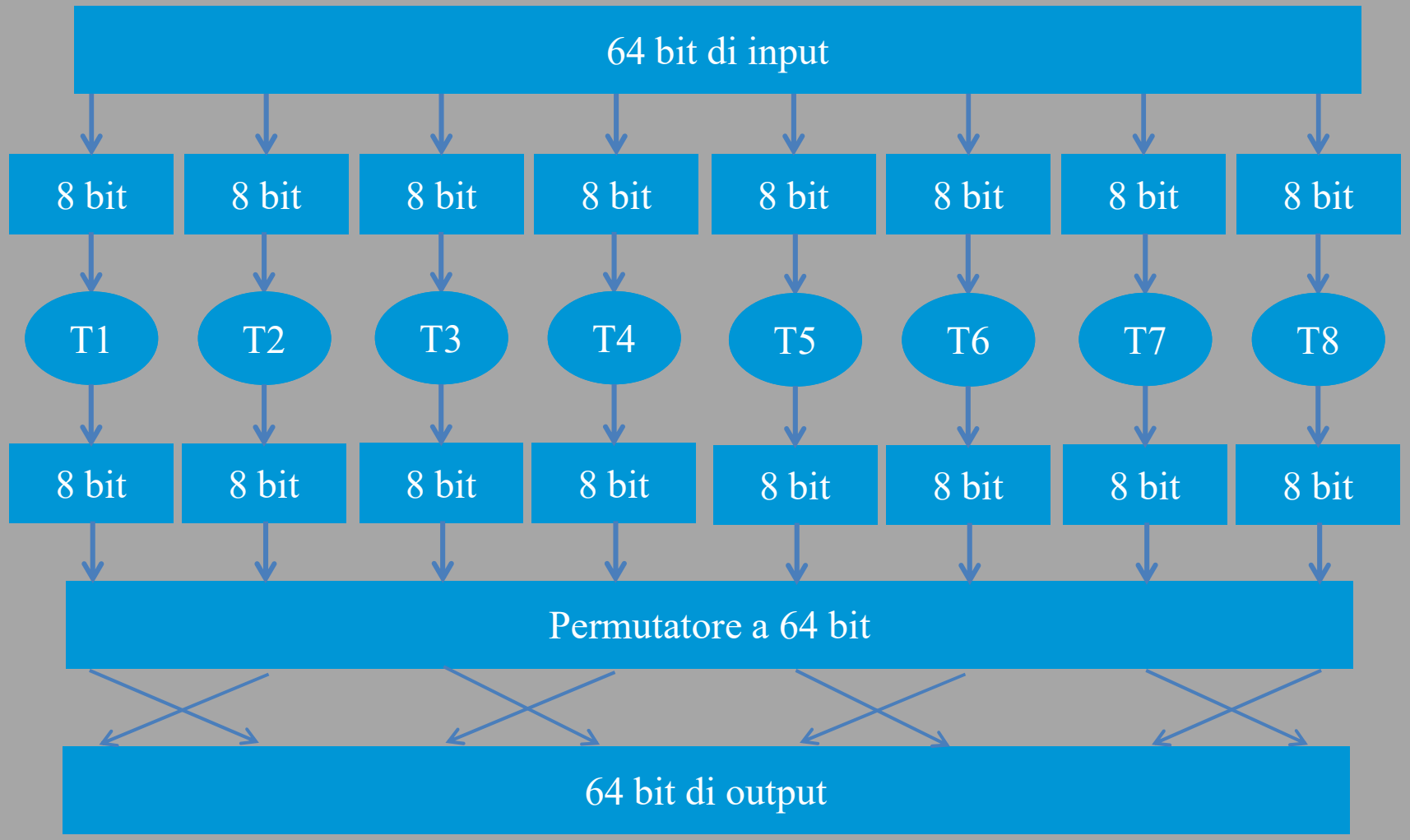
Cifrario a blocchi

- Ad esempio il primo gruppo di 8 bit è elaborato con la tabella T1.
- Si ottengono, quindi, 8 pezzi in output dalle 8 tabelle, che vengono riassemblati per ottenere un blocco da 64 bit.



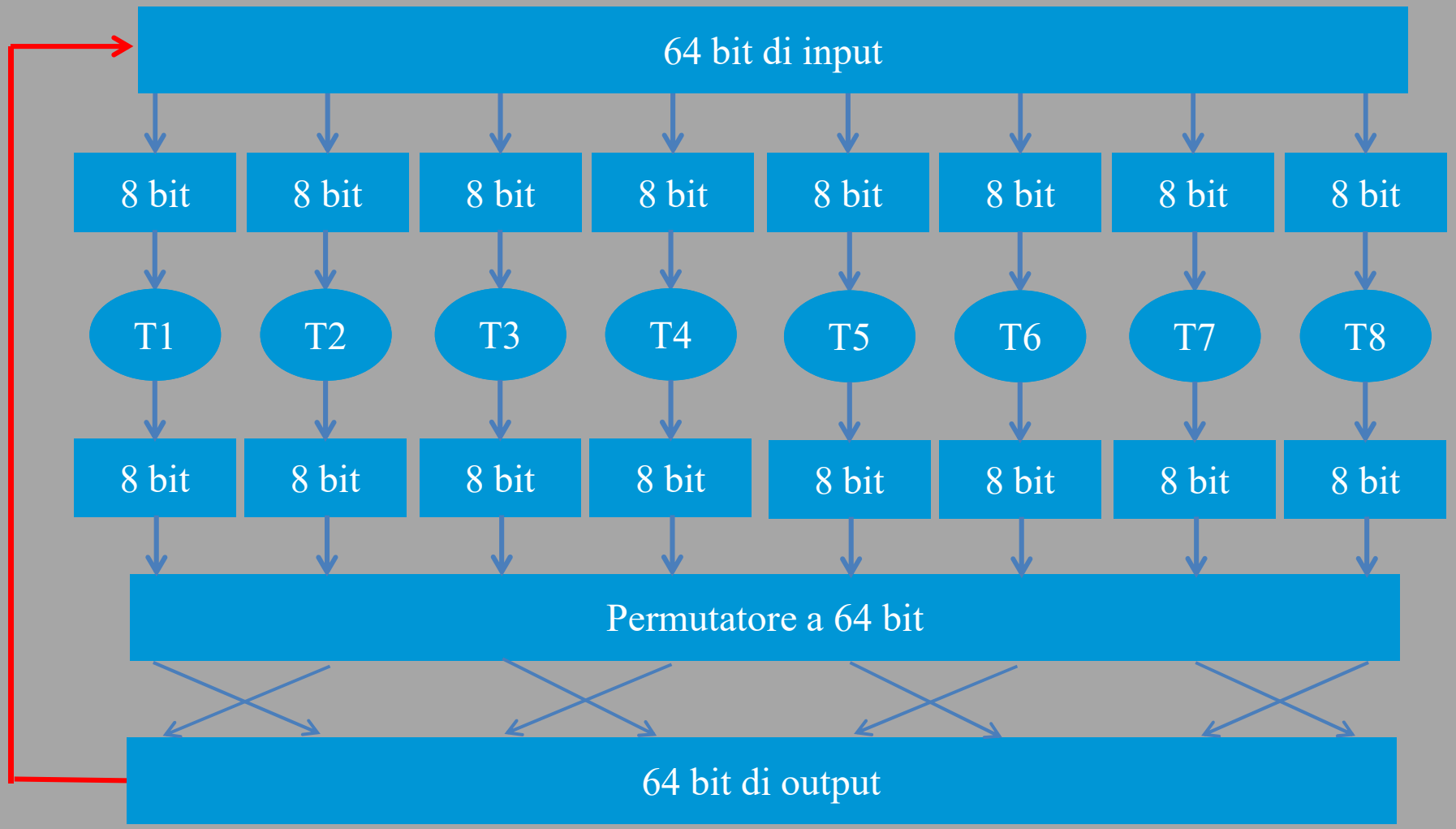
Cifrario a blocchi

- I gruppi di 8 bit nel blocco riassemblato di 64 bit vengono rimescolati (permutati) per ottenere un output a 64 bit.



Cifrario a blocchi

- Questa configurazione viene riportata in input per ripetere le stesse operazioni. Dopo n cicli, si ha il blocco di 64 bit del testo cifrato.



Cifrario a blocchi

- Lo scopo di ripetere ciclicamente le operazioni è quello di fare in modo che ogni bit di ingresso influenzi tutti o almeno la maggior parte dei bit di uscita finali. (Se si eseguisse un solo passaggio, un dato bit di ingresso riguarderebbe solo 8 dei 64 bit di uscita).
- La chiave di questo algoritmo di cifratura a blocchi è costituita dalle 8 tabelle di permutazione (assumendo che la funzione di rimescolamento sia di pubblico dominio).

Cifrario con blocchi concatenati

- Nelle applicazioni di rete, si ha necessità di criptare messaggi molto lunghi. Se a questi messaggi si applica il cifrario a blocchi come descritto, scomponendo semplicemente il messaggio in blocchi di k bit, e criptando indipendentemente ciascun blocco, si corre il rischio di agevolare la scoperta della chiave.
- Si pensi, ad esempio ad un testo in chiaro, in cui si ripetono alcune parole e, di conseguenza, dopo la suddivisione del testo, si hanno alcuni blocchi uguali. Il cifrario a blocchi produrrebbe blocchi cifrati uguali. Un attaccante, vedendo blocchi cifrati uguali, potrebbe intuire a quale testo in chiaro corrispondono, e potrebbe riuscire a decriptare anche l'intero messaggio.
- Per risolvere questo problema, si richiede che a blocchi di testo in chiaro uguali devono corrispondere blocchi di testo cifrati diversi.

Cifrario con blocchi concatenati

- Si indichi con **m(i)** l'i-mo blocco di testo in chiaro, mentre con **c(i)** si denoti l'i-mo blocco cifrato.
- Per specificare l'algoritmo di criptografia con chiave S, si userà la notazione K_S. L'idea è la seguente: il mittente crea un numero casuale **r(i)** di k bit per l'i-mo blocco e calcola:

$$c(i) = K_S(m(i) \oplus r(i)).$$

- Per ogni blocco il mittente genera un nuovo numero casuale di k bit. Il mittente trasmette c(1), r(1), c(2), r(2), c(3), r(3), ecc. Quando il destinatario riceve c(i) e r(i), può risalire al blocco di testo in chiaro calcolando:

$$m(i) = K_S(c(i) \oplus r(i)).$$

  Si può osservare che sebbene r(i) venga trasmesso in chiaro, e quindi può essere intercettato e compreso, un intruso non riesce a risalire al testo in chiaro m(i), perché non conosce la chiave K_S. Inoltre, se due blocchi in chiaro sono uguali, i corrispondenti blocchi cifrati saranno diversi, se sono diversi i numeri casuali, ciò è vero con elevata probabilità

Cifrario con blocchi concatenati

- Per costruire un esempio, si consideri il cifrario a blocchi di 3 bit riportato nella tabella. Si scelga il testo in chiaro: 010010010.
- Se Alice lo cripta direttamente, senza aggiungere la casualità, il testo cifrato risultante sarebbe 101101101.
- Se Trudy intercetta questo testo cifrato, nota subito che i tre blocchi sono uguali e che tali devono essere i tre blocchi del testo in chiaro.
- Questa informazione consente all'intruso di applicare un metodo di criptoanalisi.

Input	Output
000	110
001	111
010	101
011	100
100	011
101	010
110	000
111	001

Cifrario con blocchi concatenati

- Si supponga, adesso, che Alice generi i blocchi casuali r(1) = 001, r(2) = 111, e r(3) = 100 ed usi la tecnica descritta prima per generare i testi cifrati

  c(1) = 101 XOR 001 = 100,

  c(2) = 101 XOR 111 = 010,

  c(3) = 101 XOR 100 = 001.

  Notare che, adesso, i tre blocchi cifrati sono tutti diversi. Alice trasmette c(1), r(1), c(2), e r(2).

  (Si verifichi che Bob può ottenere il messaggio in chiaro originale usando la chiave condivisa K_S, cioè la tabella di corrispondenza)

Input	Output
000	110
001	111
010	101
011	100
100	011
101	010
110	000
111	001

Cifrario con blocchi concatenati

- Prestando attenzione, si osserva che l'introduzione della casualità risolve un problema a costo di raddoppiare il volume del traffico trasmesso.
- Infatti, per ogni bit del testo cifrato, si deve spedire anche un bit del numero casuale, raddoppiando così l'occupazione di banda.
- I cifrari a blocchi usano una tecnica denominata Concatenazione dei Blocchi Cifrati (CBC).
- L'idea base è quella di trasmettere solo il primo valore casuale insieme al primo messaggio, poi il mittente ed il destinatario usano i blocchi di codifica calcolati al posto dei successivi numeri casuali.

Cifrario con blocchi concatenati

- Più esattamente, CBC funziona così:
- Prima di criptare il messaggio (o il flusso di dati), il mittente genera una stringa di k bit casuali, chiamata il **vettore di inizializzazione** (**IV**). Questo vettore di inizializzazione viene indicato con c(0). Il mittente trasmette l'**IV** al ricevitore, in chiaro.
- Per il primo blocco, il mittente calcola

$$m(1) \oplus c(0)$$

cioè calcola l'OR esclusivo del primo blocco del testo in chiaro con **IV**. Il risultato viene elaborato attraverso l'algoritmo del cifrario a blocchi per ottenere il corrispondente blocco cifrato:

$$c(1) = K_S(m(1) \oplus c(0)).$$

Il mittente trasmette il blocco cifrato c(1) al ricevitore.

- Per l'i-mo blocco, il mittente genera l'i-mo blocco cifrato da

$$c(i) = K_S(m(i) \oplus c(i - 1)).$$

Cifrario con blocchi concatenati

- Riassumendo:
- il ricevitore riesce a risalire al messaggio originale, infatti quando il destinatario riceve c(i), lo decripta con k_S per ottenere

$$s(i) = m(i) \oplus c(i-1);$$

- Poiché il ricevitore conosce anche c(i-1), ricava il testo in chiaro del blocco da

$$m(i) = s(i) \oplus c(i-1).$$

- Anche se due blocchi di testo in chiaro sono identici, i corrispondenti blocchi di testo cifrato sono (con elevata probabilità) diversi.
- Nonostante il mittente trasmetta l'**IV** in chiaro, un intruso non riuscirà a decriptare i blocchi cifrati, perché non conosce la chiave segreta K_S.
- Il mittente trasmette solo un blocco di dati in più, l'**IV**, che costituisce un aumento trascurabile dell'occupazione di banda, se il messaggio è molto lungo.

Cifrario con blocchi concatenati

- Come esempio, usando il cifrario riportato nella tabella precedente, si determini il testo cifrato per i tre blocchi da 3 bit:

  010010010 e con **IV** = c(0) = 001.

  Il mittente usa l'**IV** per calcolare

  $$c(1) = K_S(m(1) \oplus c(0)) = 100.$$

Il mittente poi calcola

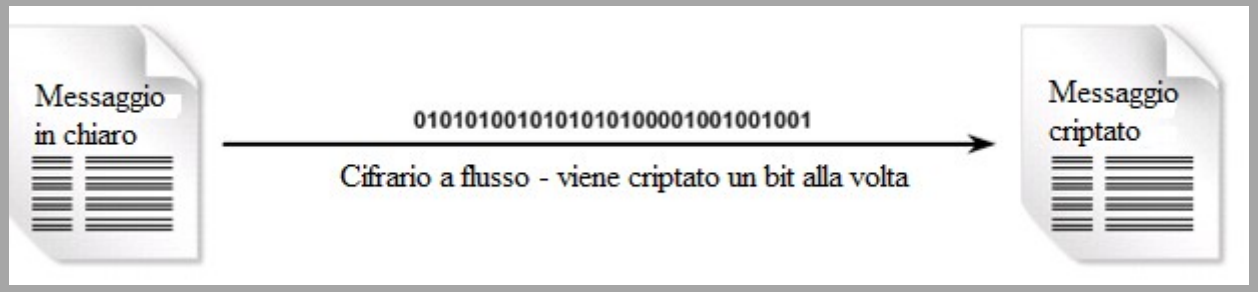
$$c(2) = K_S(m(2) \oplus c(1)) = K_S(010 \oplus 100) = 000,$$

$$\text{e } c(3) = K_S(m(3) \oplus c(2)) = K_S(010 \oplus 000) = 101.$$

Come esercizio, si verifichi che il ricevitore, conoscendo l'**IV** e K_S può risalire al testo in chiaro.

Cifrari a flusso

- I cifrari a flusso criptano un byte o un bit alla volta del messaggio.
  - Funziona come un cifrario a blocchi in cui la lunghezza del blocco è 1 bit.
  - Il cifrario di Vigenère è un esempio.
  - Può essere molto più veloce di un cifrario a blocchi e non incrementa la lunghezza del messaggio (aggiungendo dati ridondanti).
  - È una criptografia usata nelle reti wireless.
- Esempi:
  - A5 usato per criptare le comunicazioni GSM nella telefonia cellulare.
  - Cifrario RC4.
  - Il DES può essere usato nella modalità cifrario a stream.



Criteri di scelta di un algoritmo di criptografia

- L'algoritmo è affidabile?
  - Gli algoritmi che hanno resistito agli attacchi e non sono stati violati sono i preferiti.
- L'algoritmo protegge adeguatamente contro gli attacchi di forza bruta?
  - Con una appropriata lunghezza della chiave, questi attacchi sono generalmente considerati irrealizzabili.
- L'algoritmo ammette chiavi lunghe e di lunghezza variabile?
- L'algoritmo è soggetto a limitazioni territoriali? (gli algoritmi commissionati da autorità militari non possono essere usati per comunicazioni che superino i confini dello stato)

Criteri di scelta di un algoritmo di criptografia

	DES	3DES	AES
L'algoritmo è accettato dagli addetti ai lavori?	È stato sostituito dal 3DES	Sì	Non ancora giudicato
L'algoritmo protegge adeguatamente dagli attacchi di forza bruta?	No	Sì	Sì

Data Encryption Standard (DES)

- Lo standard più diffuso.
  - Sviluppato da IBM
  - Negli anni ‘70 si riteneva inviolabile
  - Chiave condivisa per criptare e decriptare
- DES converte blocchi di 64 bit del testo in chiaro in testo cifrato usando un algoritmo di criptografia.
  - L’algoritmo di decriptografia sul terminale remoto estrae il testo in chiaro dal testo cifrato.
  - Assumendo che un computer possa provare 255 chiavi al secondo, sono richiesti 6.4 giorni per scoprire la chiave

Valutazione della sicurezza DES

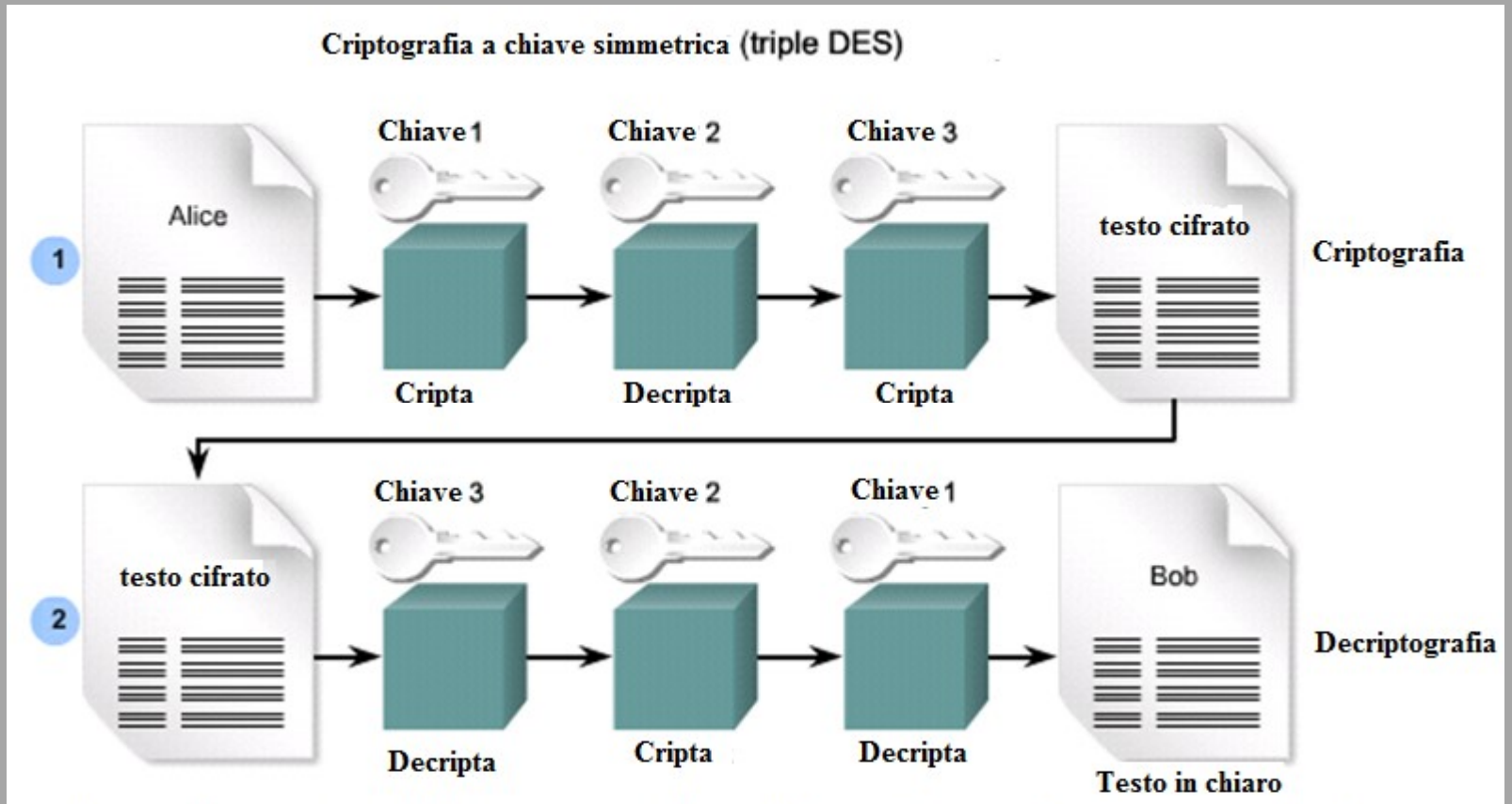
- Per la sua chiave molto corta, il DES é considerato un buon protocollo di protezione dei dati, almeno per un breve periodo.
  - 3DES è una scelta migliore per proteggere i dati, perché è un algoritmo molto affidabile.
- Raccomandazioni:
  - Cambiare frequentemente la chiave per impedire gli attacchi di forza bruta.
  - Usare un canale sicuro per comunicare la chiave DES dal mittente al destinatario.

Triple DES (3DES)

- 3DES è 256 volte più robusto del DES.
- Prende un blocco di dati di 64 bit ed applica tre operazioni DES in sequenza:
  - Cripta, decripta, e cripta.
  - Richiede maggior tempo di elaborazione.
  - Può usare 1, 2, o 3 chiavi differenti (quando usato con una chiave, si comporta come il DES).
- Il software 3DES non può uscire dagli USA.

Assumendo che un computer possa provare 255 chiavi al secondo, sono richiesti 4.6 miliardi di anni per scoprire la chiave

3DES



1. Il testo in chiaro originato da Alice è criptato con la chiave 1. Tale testo cifrato viene decriptato con una chiave diversa, chiave 2. Infine tale testo viene cifrato con un'altra chiave, la 3.

2. Quando viene ricevuto il testo cifrato con 3DES, si esegue il processo inverso. Il testo cifrato viene prima decifrato con la chiave 3, criptato con la chiave 2 e decriptato con la chiave 1.

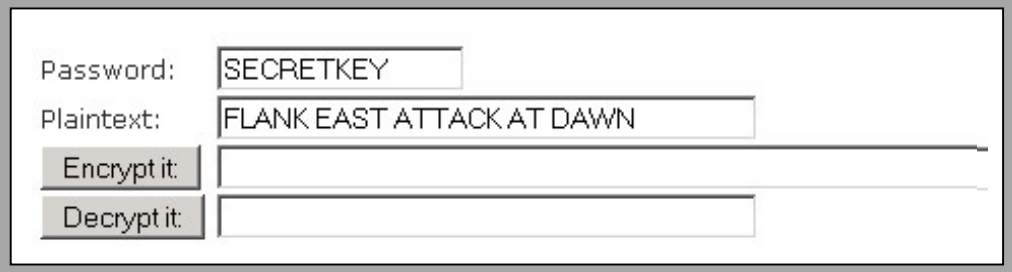
Valutazione della sicurezza 3DES

- Sebbene il 3DES sia molto sicuro, spreca molte risorse e al suo posto viene preferito l'algoritmo AES.
  - È stato provato che AES è sicuro quanto 3DES, ma è molto più veloce.

Advanced Encryption Standard (AES)

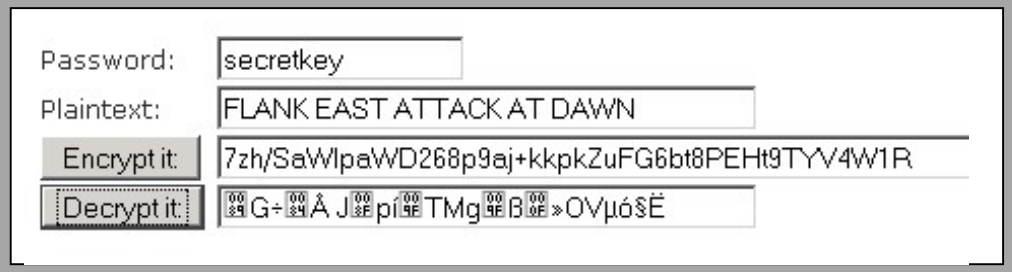
- AES é un algoritmo di criptografia estremamente sicuro.
  - Sviluppato da Rijndael (“Rhine dahl”).
  - Usa chiavi di lunghezza 128, 192, o 256 bit per criptare blocchi di 128, 192, o 256 bit.
    - Sono ammesse tutte le 9 combinazioni di lunghezza della chiave e di lunghezza dei blocchi.

Esempio AES

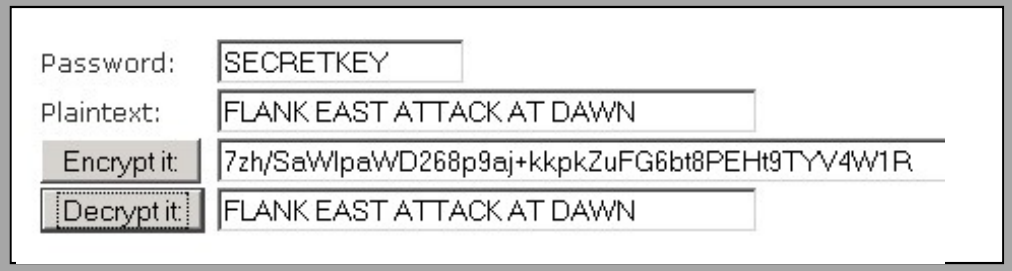


In questo esempio vengono inseriti la chiave SECRETKEY e il testo cifrato.

Vengono criptati con AES a 128 bit.



Si tenta di decifrare il messaggio usando una chiave sbagliata.



Un secondo tentativo, che fa uso della chiave corretta, produce il testo in chiaro,

AES

- AES fu adottato come sostituto del DES per varie ragioni:
  - La lunghezza della chiave AES è molto più robusta del DES.
  - A parità di hardware, AES è molto più veloce del 3DES.
  - AES è più efficiente del DES e del 3DES, sullo stesso hardware.
  - AES è consigliato quando il flusso dei dati è elevato, in particolare, quando la criptografia è realizzata in software.
- AES è un algoritmo recente, gli algoritmi più vecchi, che hanno resistito per molti anni, sono ritenuti affidabili.
- 3DES, quindi viene scelto per la sua affidabilità, infatti è stato sperimentato per 35 anni.