Schemi di cifratura simmetrici

Indice

Introduzione

- La crittografia è la pratica e lo studio di tecniche per la comunicazione sicura (segreta) in presenza di terzi chiamati avversari.
- Più in generale, la crittografia riguarda la costruzione e l'analisi di protocolli che impediscono a terzi o al pubblico di leggere o alterare messaggi privati.
- Diversi aspetti della sicurezza dell'informazione, come riservatezza e integrità dei dati, l'autenticazione e il non ripudio, sono gli obiettivi principali della crittografia moderna.

La crittografia è onnipresente

- Hashing delle password
- Transazioni sicure con carta di credito su Internet
- WiFi criptato
- Cifratura del contenuto delle memorie econdarie
- Aggiornamenti software con firma digitale
- Bitcoin
- …

Crittografia, Crittoanalisi, ecc…

Alcuni termini chiave:

- **Algoritmo di cifratura**: trasforma un testo in chiaro (**plaintext**, comprensibile a un umano o una macchina) in un testo cifrato (**ciphertext**, incomprensibile).
- **Algoritmo di decifratura**: prende il *ciphertext* e ritorna il *plaintext*.
- Ad ogni algoritmo di cifratura deve corrisponderne uno «inverso» di decifratura.
- Un algoritmo di cifratura/decifratura viene anche chiamato **schema**.

Chiave crittografica

- L'algoritmo di cifratura di solito prende in ingresso una *chiave crittografica*.
- La chiave crittografica:
  - è un'informazione (un parametro) che determina l'output funzionale di un algoritmo crittografico
  - specifica la trasformazione del testo in chiaro in testo cifrato e del testo cifrato in testo in chiaro per gli algoritmi di decrittazione
- La sicurezza di uno schema simmetrico dipende strettamente dalla segretezza della chiave.

Chiavi Crittografiche

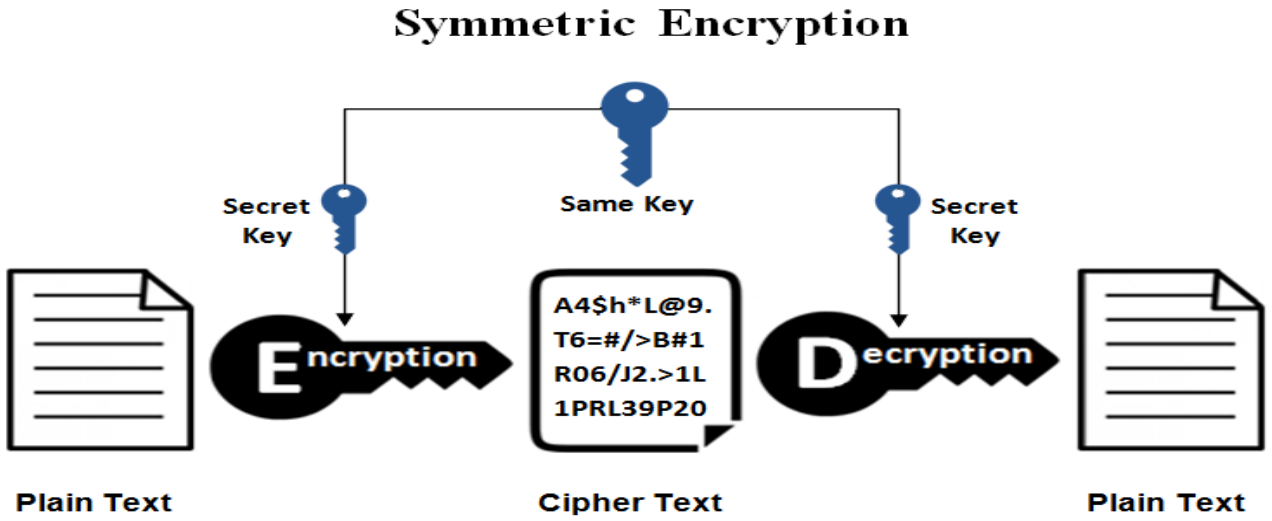
- Due principali metodi:
  - *A chiave simmetrica* – Unica chiave
  - *A doppia chiave* Pubblica–Privata

Crittografia a chiave simmetrica

- Chiamata anche *crittografia convenzionale* o *crittografia a chiave singola*
- Tecnica universale per garantire la riservatezza dei dati trasmessi o memorizzati
- L'unico tipo di cifratura in uso prima della introduzione della cifratura a chiave pubblica (fine anni '70)
- Rimane il più diffuso tra i due tipi di crittografia

Crittografia a chiave simmetrica

Si riferisce a metodi in cui sia il mittente sia il destinatario condividono la stessa chiave.



Sistemi crittografici simmetrici

Gli schemi crittografici si differenziano per:

- **Il tipo di operazione** per trasformate il plaintext (P) nel ciphertext (C):
  - **Sostituzioni:** ogni elemento di P viene sostituito con un elemento di C
  - **Trasposizioni**, vengono effettuati spostamenti tra elementi di P.
- **Il modo di «elaborare» P:**
  - Cifrari a **blocchi:** P è diviso in blocchi ed elaborato dall'algoritmo «un blocco alla volta»
  - Cifrari a **flusso**: P è elaborato considerando un elemento per volta.

Critto analisi

Uno schema di cifratura simmetrica è sicuro se:

- L'algoritmo di cifratura è **robusto** e cioè un attaccante in possesso di un certo numero di ciphertext ma non della chiave non è in grado di inferire i plaintext o la chiave.
- Il mittente e il ricevente **ricevono e mantengono la chiave in maniera sicura** (nessun attaccante deve intercettare chiave).

Si assume che l'algoritmo sia noto e che sia impraticabile decifrare dei messaggi avendo solo ciphertext.

Crittoanalisi

La crittoanalisi è un insieme di tecniche per testare la robustezza dell'algoritmo e della chiave provando a capire la chiave a partire dai ciphertext disponibili.

Tecniche di crittoanalisi possono essere applicate a partire da «ipotesi» diverse sulle informazioni possedute dall'attaccante:

- non conoscere niente, nemmeno l'algoritmo
- Conoscere alcuni ciphertext e l'algoritmo
- Conoscere anche alcuni plaintext.

Conoscenza dell'attaccante

- Ciphertext only: ciphertext e algoritmo di criptaggio
- Known plaintext: ciphertext, algoritmo e una o più coppie di plaintext-ciphertext
- Chosen plaintext: ciphertext, algoritmo, plaintext scelto dal cripto analista e di conseguenza il relativo ciphertext
- Chosen ciphertext: ciphertext, algoritmo, ciphertext scelto dal cripto analista e il relativo plaintext decriptato
- Chosen text: ciphertext, algoritmo, plaintext scelto dall'analista e il relativo ciphertext criptato, ciphertext scelto dall'analista e relativo plaintext decriptato

Tipi di Attacco

- Attacchi Criptonalitici
- Attacchi di forza bruta

Attacchi Criptonalitici

- L’attaccante:
  - Sfrutta la conoscenza dell'algoritmo, più una certa conoscenza delle caratteristiche generali del testo in chiaro ed eventualmente alcune coppie campione di testo in chiaro – testo cifrato.
  - Tenta di dedurre da uno specifico testo in chiaro la chiave utilizzata, in modo da compromettere tutti i messaggi futuri e passati criptati con quella chiave.

Crittoanalisi e attacchi a forza bruta

μs = milionesimo di secondo

Key Size (bits)	Number of Alternative Keys	Time Required at 1 Decryption/μs	Time Required at 10^6 Decryptions/μs
32	$2^{32} = 4.3 \times 10^9$	$2^{31}\mu s = 35.8$ minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55}\mu s = 1142$ years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127}\mu s = 5.4 \times 10^{24}$ years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	$2^{167}\mu s = 5.9 \times 10^{36}$ years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	$2 \times 10^{26}\mu s = 6.4 \times 10^{12}$ years	6.4×10^6 years

Tecniche di cifratura

- Meccanismi di base da applicare ai plaintext per ottenere i ciphertext:
  - la sostituzione
  - la trasposizione.

- La composizione di queste due tecniche permette di costruire algoritmi di cifratura/decifratura.

Tecniche di Sostituzione

- Una tecnica di sostituzione si basa sulla sostituzione di ogni elemento del plaintext con un altro elemento dello stesso alfabeto.
- Le sostituzioni devono essere reversibili, ovvero si deve poter tornare indietro.
- Dato un elemento nel plaintext, la scelta di quale elemento usare per sostituirlo nel corrispondente ciphertext dipende dalla chiave.

Il Cifrario di Cesare

Il primo esempio di cifrario a sostituzione è quello di Giulio Cesare che prevede di sostituire ogni lettera con la corrispondente lettera dell'alfabeto che sta «*tre spazi*» più in là; quando l'alfabeto finisce ricomincia da capo. Ad esempio:

```
Plaintext:  meet me after the     toga  party
Ciphertext: PHHW PH DIWHU WKH     WRJD  SDUWB
```

In termini matematici abbiamo che ogni lettera C del ciphertext a partire dalla corrispondente p è data da:

$$C=E(3,p) = (p+3) \bmod 26.$$

Il Cifrario di Cesare

Il primo esempio di cifrario a sostituzione è quello di Giulio Cesare che prevede di sostituire ogni lettera con la corrispondente lettera dell'alfabeto che sta «*tre spazi*» più in là; quando l'alfabeto finisce ricomincia da capo. Ad esempio:

PLAINTEXT	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**
CIPHERTEXT	D	E	F	G	H	I	J	K	L	M	N	O	P
PLAINTEXT	**n**	**o**	**p**	**q**	**r**	**s**	**t**	**u**	**v**	**w**	**x**	**y**	**z**
CIPHERTEXT	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

attack

diventa

dwwdfn

Cifrario di Cesare

- Un altro esempio:

  Plaintext: meet me after the toga party

  Ciphertext: PHHW PH DIWHU WKH WRJD SDUWB

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Con l'associazione a destra, la codifica numerica del testo cifrato C in corrispondenza del numero p è data da:

$C = E(3,P) = (P+3) \bmod 26.$

Limiti del Cifrario di Cesare

Da un punto di vista della crittoanalisi, il cifrario di Cesare è debole, fondamentalmente per tre motivi:

1. L'algoritmo di cifratura è noto (quasi sempre così)
2. Ci sono solo 25 chiavi possibili da provare
3. Il linguaggio in chiaro è facilmente riconoscibile (ovvero è molto semplice accorgersi quando la chiave è giusta).

Cifrario Monoalfabetico

- Un cifrario può essere rinforzato permettendo *sostituzioni arbitrarie ad esempio* introducendo la possibilità di non solo spostare le lettere ma di permutarle arbitrariamente.
- Per un insieme di n elementi, abbiamo n! permutazioni. Dato S= {a,b,c} esistono sei permutazioni di S: *abc, acb, bac, bca, cab, cba*.
- Nel cifrario di Cesare, (S = 26 - caratteri alfabeto) si usa una specifica permutazione che sposta di 3 ogni carattere.
- Se un messaggio fosse cifrato spostando il plaintext in una delle possibili permutazioni (sono 26!) sarebbe (molto) più difficile portare attacchi di forza bruta.

Cifrario monoalfabetico - Permutazioni

Consideriamo la seguente permutazione delle 26 lettere (che poi è la chiave!):

- ABCDEFGHIJKLMNOPQRSTUVWXYZ
- DXUTNAVWKZFQGSIOYJBPLHCERM

Con questa permutazione il plaintext «arrivano rinforzi» diventa:

- ARRIVANORINFORZI
- DJJKHDSIJKSAIJMK

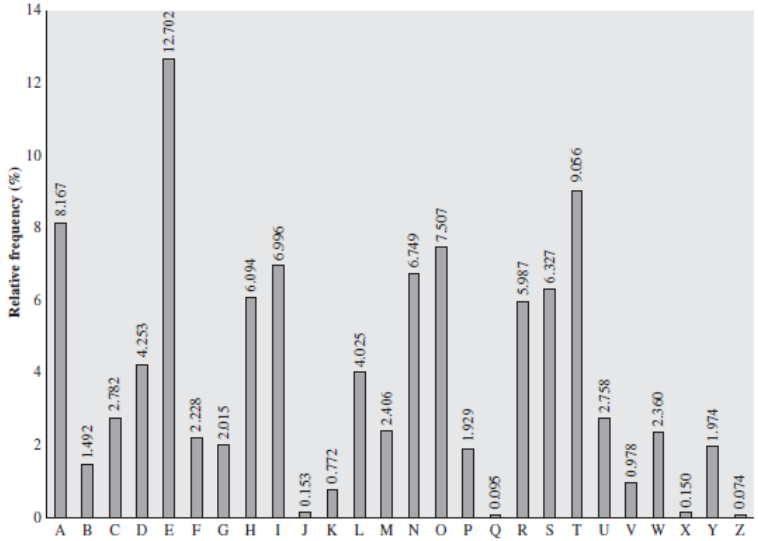
Un attacco a forza bruta richiederebbe provare, nel caso peggiore **26!** chiavi, invece che 25 chiavi come nel caso del cifrario di Cesare.

Cifrario monoalfabetico - Permutazioni

Le permutazioni rendono difficili attacchi a forza bruta, ma si possono sfruttare le *regolarità del linguaggio quali*:

- frequenza di singole lettere cifrate
- coppie di lettere vicine per parole comuni come «of» in inglese

e confrontarle con la frequenza delle lettere o delle coppie nella lingua di riferimento.



Frequenza delle lettere nella lingua inglese

Lettere più frequenti in diverse lingue

ENGLISH		GERMAN		FINNISH		FRENCH		ITALIAN		SPANISH	
	%		%		%		%		%		%
E	12.31	E	18.46	A	12.06	E	15.87	E	11.79	E	13.15
T	9.59	N	11.42	I	10.59	A	9.42	A	11.74	A	12.69
A	8.05	I	8.02	T	9.76	I	8.41	I	11.28	O	9.49
O	7.94	R	7.14	N	8.64	S	7.90	O	9.83	S	7.60
N	7.19	S	7.04	E	8.11	T	7.26	N	6.88	N	6.95
I	7.18	A	5.38	S	7.83	N	7.15	L	6.51	R	6.25

Cifrari polialfabetici

- Cifrari Polialfabetici sono un'alternativa a quelli mono-alfabetici, e si basanoi su alfabeti di sostituzione multipli.
- Proposta per complicare l'analisi basata sulla frequenza delle lettere in chiaro
  - Se P è la lettera più frequente in un testo cifrato di un testo in chiaro inglese, si potrebbe sospettare che P corrisponda a E,
  - Questo non è possibile se E è cifrata come lettere diverse in punti diversi del messaggio.
- Il Cifrario di Vigenère è l'esempio più noto di cifrario polialfabetico.

Il cifrario di Vigenère

- La chiave è ora una stringa, non solo un carattere, e ad ogni carattere viene assegnato un numero in base alla posizione nell'alfabeto (a:0, b:1, c:3, d:4, e:5, f:6, ..., z:25)
- Per cifrare, si sposta ogni carattere in chiaro della quantità dettata dal carattere corrispondente della chiave
  - Ciclare se necessario
- Per decodificare invertire il procedimento
- Ha ispirato "macchine a rotore" come Enigma usate nella seconda guerra mondiale

```
tellhimaboutme
cafecafecafeca
veqpjiredozxoe
```

La chiave è: `cafe`

Tecniche di Trasposizione

Con la **trasposizione** il ciphertext è ottenuto *modificando l'ordine delle lettere*.

- Se il plaintext è scritto in sequenze di diagonali e poi letto in riga, il messaggio «meet me after the toga party» diventa:

```
m e m a t r h t g p r  y
  e t e f e t e o a a t
```

  Che letto riga per riga è MEMATRHTGPRYETEFETEOAAT

- Un'alternativa è scrivere in un quadrato il messaggio riga per riga, e leggerlo colonna per colonna fissando l'ordine di lettura tra le colonne

```
Key:        4 3 1 2 5 6 7
Plaintext:  a t t a c k p
            o s t p o n e
            d u n t i l t
            w o a m x y z
```

Leggendo le colonne nell'ordine imposto dalla chiave abbiamo

TTNAAPTMTSUOAODWCOIXKNLYPETZ

Cifrari a blocco e a flusso

- Block cipher elaborano i messaggi in blocchi, ognuno dei quali viene poi cifrato o decifrato
  - Essenzialmente una tecnica di sostituzione polialfabetica con un set di caratteri molto grande (64 bit o più) abbinato a tecniche di permutazione
- Stream ciphers elaborano i messaggi un bit o un byte alla volta durante la cifratura o la decifratura
  - Il flusso della chiave è combinato con il testo in chiaro utilizzando or esclusivo (XOR)
- Molti cifrari attuali sono basati su tecniche di cifratura a blocchi
  - Analizzati meglio e con una più ampia gamma di applicazioni

Cifrari a flusso

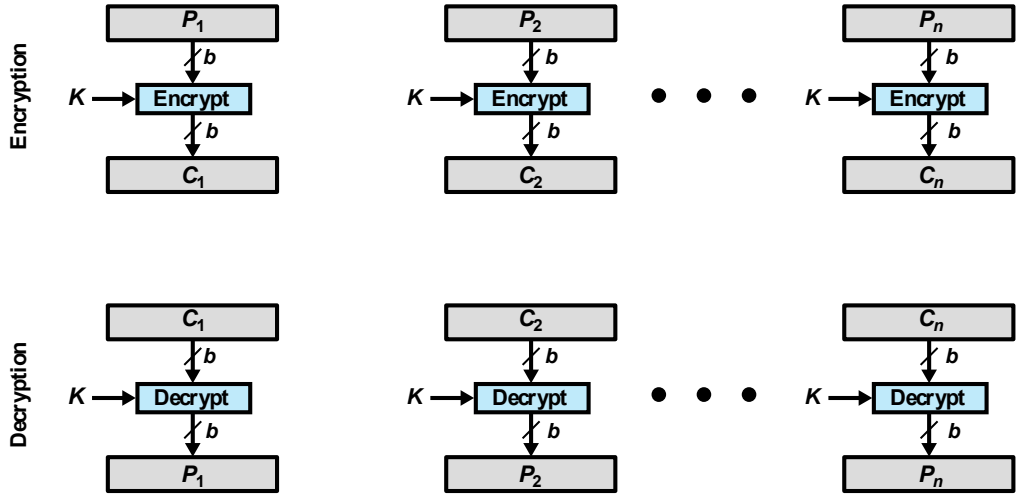
- I cifrari a flusso cifrano un flusso di dati digitali un bit o un byte alla volta
- Le cifre in chiaro sono combinate (XOR-ed) con un flusso di cifre pseudorandom (keystream) per ottenere un bit o un byte di testo cifrato
- Il generatore di flusso di bit è una procedura algoritmica utilizzata da entrambi i partner per cifrare e decifrare.
- I due partner, per produrre il keystream, devono solo condividere la chiave per la generazione pseudocasuale.

Cifrari a blocchi

- Un cifrario a blocchi opera su un blocco di n bit in chiaro per produrre un blocco cifrato di n bit.
- Un blocco di testo in chiaro viene usato nel suo insieme per produrre un blocco di testo cifrato di uguale lunghezza
- Si utilizzano blocchi di dimensioni tipiche di 64 o 128 bit (n = 64 o 128)
- Gli utenti condividono una chiave di crittografia simmetrica
- Ci sono 2^n possibili diversi blocchi di testo in chiaro e, affinché la crittografia sia reversibile, ognuno deve produrre un unico blocco di testo cifrato.

Cifrari a blocchi

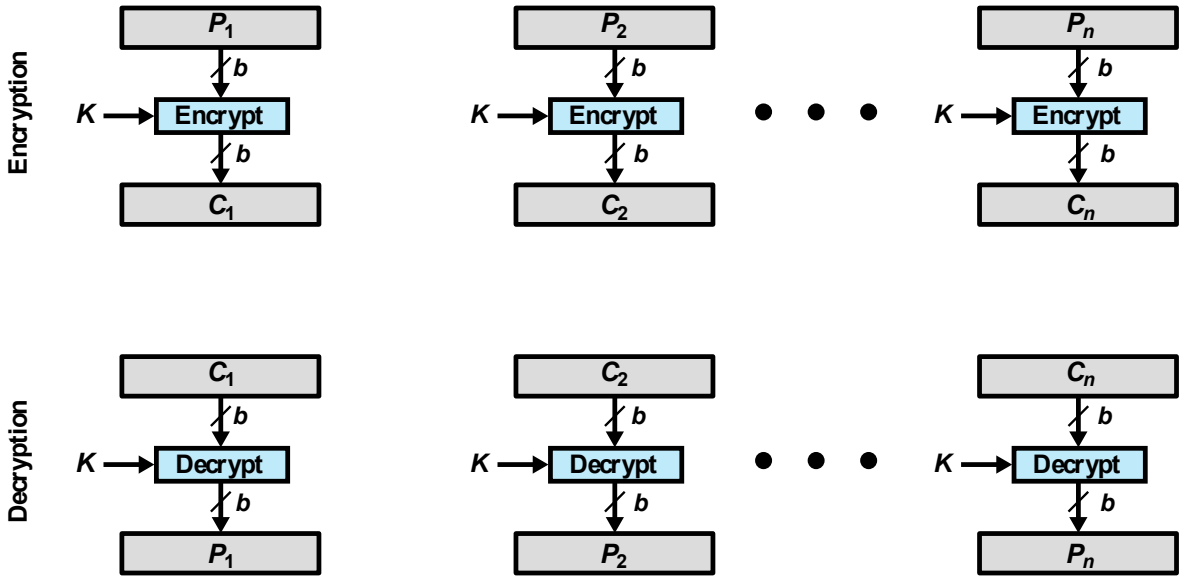
Un dato plaintext di lunghezza n*b è diviso in b blocks of n bits



Ogni blocco è criptato utilizzando lo stesso algoritmo e la stessa chiave, per produrre una sequenza di b blocchi di n-bit di testo cifrato.

Cifrari a blocchi

- Un dato plaintext di lunghezza n*b viene diviso in n blocchi di b-bit ciascuno



1. Ogni blocco è criptato utilizzando lo stesso algoritmo e la stessa chiave di cifratura
2. Viene prodotta una sequenza di n blocchi di bit b-bit di testo cifrato

Cifrari a blocchi

- I messaggi vengono divisi in blocchi di lunghezza n prefissata e trasformati in blocchi di ciphertext della stessa lunghezza secondo una data associazione.
- Per decifrare il ciphertext e tornare al plaintext originale la trasformazione deve essere reversibile.

Reversible Mapping

Plaintext	Ciphertext
00	11
01	10
10	00
11	01

Irreversible Mapping

Plaintext	Ciphertext
00	11
01	10
10	01
11	01

Da: W. Stalling: Cryptography and Network Security, Int'l Edition, Pearson

Qui n = 2

Cifrari a blocchi- Trasformazioni

- Un blocco di 4 bit produce 16 configurazioni di ingresso che possono essere (reversibilmente) mappate su una qualsiasi delle 16 configurazioni di uscita.
- Una delle possibili mappature è riportata a destra.
- La chiave condivisa (di 64 bit) è la colonna del ciphertext

Plaintext	Ciphertext
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

Il cifrario di Feistel

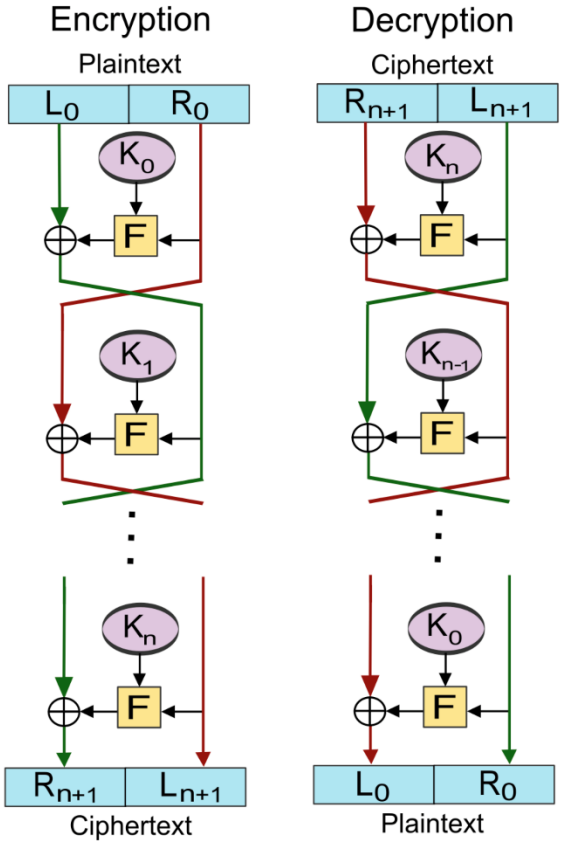
- Molti cifrari a blocchi, tra cui il noto Data Encryption Standard (DES), utilizzano come base il cifrario di Feistel
- Le operazioni di cifratura e decifratura sono simili, in alcuni casi identiche, solo le chiavi vengono utilizzate in ordine inverso.
- Si basa sulla combinazione di semplici trasformazioni come la sostituzione (S-box), la permutazione (P-box) e l'aritmetica modulare.
- Implementa il concetto di confusion and diffusion di Shannon attraverso sostituzioni e permutazioni, suddividendo i blocchi di input in due metà ed eseguendo diversi round.

Il cifrario di Feistel: principi

- Tecniche :
  - Sostituzione: Ogni elemento in chiaro è sostituito in modo univoco da un corrispondente elemento cifrato
  - Permutazione: Nessun elemento viene aggiunto, rimosso o sostituito nella sequenza, ma viene cambiato l'ordine in cui gli elementi appaiono.
- Principi di Shannon :
  - Diffusione: dissipa la struttura statistica del testo in chiaro sulla maggior parte del testo cifrato
  - Confusione: ogni bit del testo cifrato dipende da diverse parti della chiave, oscurando le connessioni tra le due.

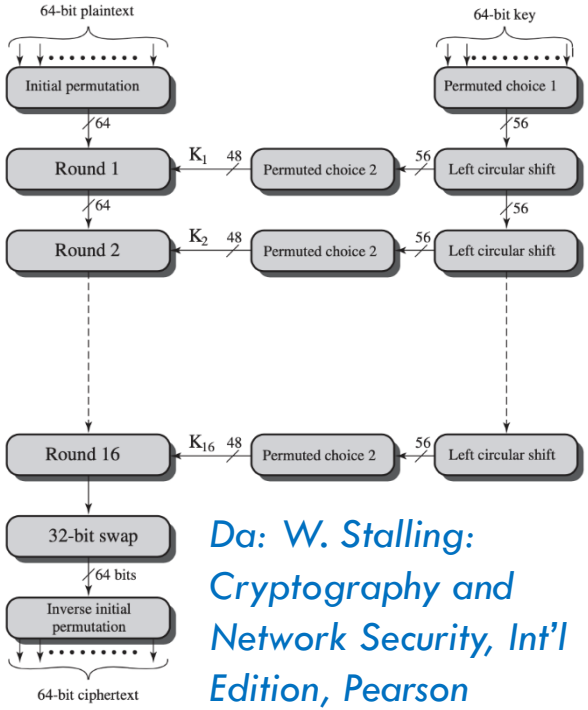
Il cifrario di Feistel

- Il testo cifrato viene calcolato a partire dal testo in chiaro mediante l'applicazione ripetuta della stessa trasformazione.
- Il testo da cifrare è diviso in due. La funzione circolare F viene applicata ad una metà usando una sottochiave e l'output di F è XOR-ed con l'altra metà.
- Le due metà vengono poi scambiate. Ogni round segue lo stesso schema ma nell'ultimo round non c'è scambio
- Cifratura e decifratura sono strutturalmente identiche, ma le sottochiavi sono usate in ordine inverso per cifratura e decifratura.



DES (Data Encryption Standard)

- Utilizza chiavi di 56 bit, divide il testo in chiaro in blocchi di 64 bit, effettua delle permutazioni iniziali e finali ed un ciclo di 16 iterazioni di permutazioni e xor (Feistel network, tecniche di confusione e diffusione).
- La chiave è in realtà 64 bit, ma ogni 8 bit ce n'è uno per controllo di parità; quindi, solo 56 dei 64 bit sono significativi.
- L'algoritmo originariamente aveva una chiave a 128 bit, ma le dimensioni della chiave sono state ridotte dalla NSA (per qualche ragione)



Da: W. Stalling: Cryptography and Network Security, Int'l Edition, Pearson

Debolezze di DES

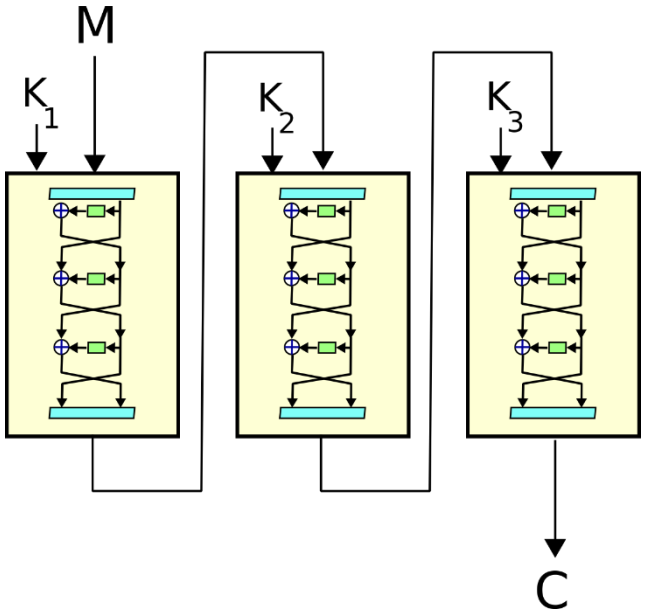
- DEA l'algoritmo alla base di DES e delle sue varianti - Triple DES e Advanced ES - è l'algoritmo di crittografia più studiato
- Nonostante i numerosi tentativi, nessuno ha finora segnalato una debolezza fatale dell'algoritmo
- Però con una chiave di 56 bit, ci sono 2^{56} (circa $7{,}2 * 10^{16}$) possibili chiavi. Ora sono possibili attacchi di forza bruta
- La Electronic Frontier Foundation (EFF) ha annunciato nel luglio 1998 di aver rotto una crittografia DES.
- Se l'unica forma di attacco a un algoritmo di cifratura è la forza bruta, allora «utilizzate chiavi più lunghe!»

Dopo DES

- A partire dal 1999, il DES è considerato insicuro a causa delle ridotte dimensioni delle sue chiavi
- I cifrari simmetrici più recenti che hanno sostituito il DES lo sono:
  - Triple-DES - triplica la dimensione della chiave DES
  - Blowfish - dimensioni variabili delle chiavi da 32 bit fino a 448 bit
  - International Data Encryption Algorithm (IDEA) – chiavi di 128 bit
  - Advanced Encryption Standard (AES) - chiavi di 128, 192 o 256 bit

Triple DES (3DES)

Ripete l'algoritmo DES di base tre volte utilizzando una, due o tre chiavi, per una dimensione della chiave di 168 bit



- Con 3 chiavi, la robustezza contro gli attacchi di forza bruta aumenta.
- Ma non permette codice software efficiente
- Utilizza blocchi di 64 bit (ma per sicurezza meglio blocchi più grandi)

Advanced Encryption Standard (AES)

Necessaria un'alternative a 3DES

- 3DES non ragionevole per un uso duraturo

NIST ha fatto un bando per un nuovo AES in 1997

- Più robusto
- Molto più efficiente
- Symmetric block cipher
- Blocchi da 128 e chiavi da 128/192/256 bit

Nel Novembre 2001è stao scelto Rijndael

- Primo giro scelti 15 algoritmi
- Secondo giro ridotti a 5
- Publicato come FIPS 197

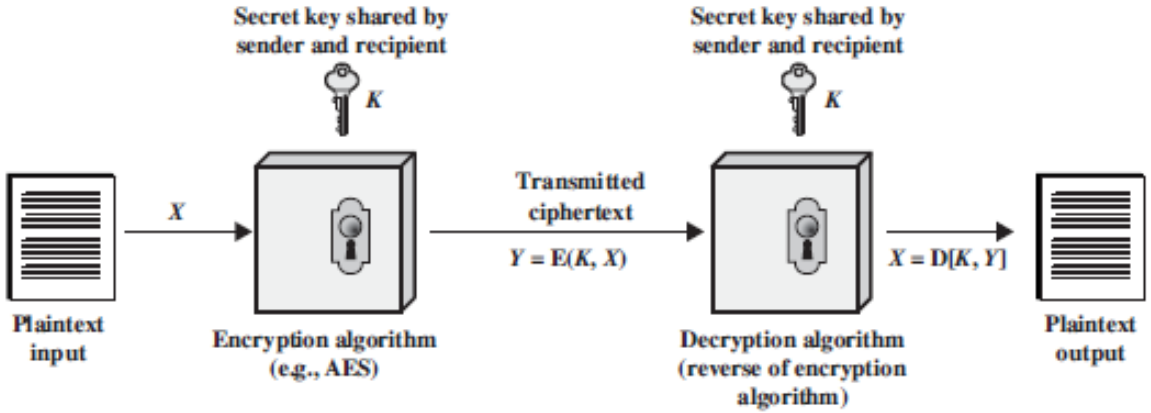
Tempi per gli attacchi

Key size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 decryptions/sec Personal Computer	Time Required at 10^{13} decryptions/sec Super Computer
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	2^{55} ns = 1.125 years	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	2^{127} ns = 5.3×10^{21} years	5.3×10^{17} years
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	2^{167} ns = 5.8×10^{33} years	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	2^{191} ns = 9.8×10^{40} years	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	2^{255} ns = 1.8×10^{60} years	1.8×10^{56} years

Approfondimento:
Il modello a chiave simmetrica /1

- Uno «schema» a chiave simmetrica ha 5 ingredienti:
  - Il **Plaintext**: il messaggio originale «intelligibile» ovvero che «trasmette informazione» per fare un paragone con quanto detto nelle lezioni precedenti.
  - **L'algoritmo di cifratura**: definisce ed esegue sostituzioni e trasformazioni sul plaintext per trasformarlo nel ciphertext.
  - **La chiave segreta**: è un *input* dell'algoritmo di cifratura ed è un valore indipendente sia dall'algoritmo che dal plaintext. L'algoritmo di cifratura, a partire da un dato plaintext produce ciphertext diversi per chiavi diverse.
  - **Ciphertext**: il messaggio «illeggibile» prodotto dall'algoritmo di cifratura a partire dalla chiave e da plaintext.
  - **Algoritmo di decifratura**: l'inverso dell'algoritmo di cifratura. Permette di ricostruire il corretto plaintext a partire dal corrispondente ciphertext e della chiave segreta

Approfondimento:
Il modello a chiave simmetrica /2



- L'immagine precedente rappresenta le relazioni dei cinque attori nelle fasi di cifratura e decifratura:
  - una sorgente produce un plaintext $X=[X_1, X_2, \ldots, X_M]$, dove le X sono lettere in un qualche alfabeto. Una chiave $K=[K_1, K_2, \ldots, K_J]$ è costruita e *in qualche modo* inviata alle due parti.
  - E è uno algoritmo di cifratura che prende K e X e restituisce un ciphertext **Y=E(K,X)**.
  - D è l'algoritmo di decifratura, che prende Y e sempre K e restituisce **X=D(K,Y).**

Il modello a chiave simmetrica /3

- Affinché uno schema di cifratura simmetrica sia «sicuro» ci sono due requisiti fondamentali da soddisfare:
  - L'algoritmo di cifratura deve essere **robusto**, nel senso che un attaccante in possesso di un certo numero di ciphertext ma non della chiave non sia in grado di inferire i plaintext o la chiave.
  - Il mittente e il ricevente devono *ricevere la chiave in maniera sicura*, ovvero occorre avere la sicurezza che nessun attaccante possa intercettare ed usare la chiave.
  - Inoltre si assume che sia molto impraticabile decifrare dei messaggi solo se si hanno i ciphertext e si conosce l'algoritmo. In sostanza, **l'algoritmo è NOTO**. Studi dimostrano che non vi è alcun guadagno a rendere non noto l'algoritmo, quindi tutti gli schemi sono noti.
  - In definitiva, la sicurezza di uno schema simmetrico dipende strettamente dalla segretezza della chiave.

Approfondimento: Sicurezza degli Schemi

- In funzione di quali attacchi può subire uno schema può essere:
  - **Debole**, se risulta facile violarlo (e in tempi ragionevoli) tanto che il guadagno ottenuto dalle informazioni ottenute sia sensibilmente maggiore dello sforzo per ricavarle.
  - **Incondizionatamente sicuro**, se i ciphertext che genera non forniscono mai informazioni sufficienti a ricavare il plaintext, indipendentemente dal numero di ciphertext di cui l'attaccante può disporre.
  - **Computazionalmente sicuro** se lo schema soddisfa uno dei seguenti criteri:
    - Il costo per ricavare il plaintext supera il valore delle informazioni contenute.
    - Il tempo richiesto per violare lo schema supera il tempo utile affinché' le informazioni contenute siano utilizzabili.

Approfondimento: Attacchi a Forza Bruta

- Oltre a analisi sottili basati su quello che l'attaccante conosce, è anche possibile fare attacchi "**a forza bruta**" dove l'attaccante *prova tutte le combinazioni* di chiavi possibili fino a quando non trova quella giusta.
- L'approccio dell'attaccante è *provare ogni chiave possibile su un certo insieme di ciphertext* fino a quando non ottiene plaintext intelligibili applicando l'algoritmo di decifratura.
- Il successo dell'approccio a forza bruta dipende dalla *lunghezza* della chiave in bit. Banalmente, maggiore è il numero di bit di una chiave, maggiori sono le combinazioni che l'attaccante dovrà provare. Ad esempio, se una chiave è di N bit, ci saranno 2^N combinazioni da provare!!
- E' chiaro che l'approccio sarà automatizzato da parte dell'attaccante: tuttavia, quanto tempo si impiega mediamente? La tabella seguente da' una idea dei tempi a grandi linee per chiavi di lunghezza diversa (con algoritmi diversi).

Approfondimenti:
Il Cifrario di Cesare

Il primo esempio di cifrario a sostituzione è stato usato da Giulio Cesare. L'espediente, usato durante le guerre galliche, era di sostituire ogni lettera con la corrispondente lettera dell'alfabeto che sta «*tre spazi*» più in là. Ad esempio:

- Plaintext : meet me after the toga party
- Ciphertext: PHHW PH DIWHU WKH WRJD SDUWB

È possibile verificare che ogni lettera del ciphertext corrisponde alla lettera del plaintext spostata di tre. Chiaramente, quando l'alfabeto finisce ricomincia da capo. Aggiungendo un po' di matematica ed usando i simboli che abbiamo introdotto nella lezione scorsa, diciamo che ogni lettera C del ciphertext a partire dalla corrispondente p e' data da: $C=E(3,p) = (p+3) \bmod 26$.

Ragionando sulla simbologia che abbiamo usato viene da se che 3 è la chiave che determina per il dato P quale è il C corrispondente calcolato dall'algoritmo. Possiamo generalizzare il cifrario di Cesare per una chiave generica *k* (quindi spostiamo di k, non più di 3) e otteniamo le seguenti formule generiche per la cifratura e la «simmetrica» decifratura:

$C = E(k, p) = (p+k) \bmod 26$ **$p = D(k,C) = (C-k) \bmod 26$**

Da un punto di vista della crittoanalisi, il cifrario di Cesare è debole, fondamentalmente per tre motivi: (i) L'algoritmo di cifratura è noto; (ii) Ci sono solo 25 chiavi possibili da provare; (iii) Il linguaggio in chiaro è facilmente riconoscibile (ovvero è molto semplice accorgersi quando la chiave è giusta).

Approfondimenti: Cifrario Monoalfabetico/1

- I limiti del cifrario di Cesare sono evidenti. Tuttavia, si possono ottenere risultati decisamente migliori permettendo *sostituzioni arbitrarie*.
- È utile il concetto di **permutazione**. La permutazione di un insieme finito di elementi S è una *sequenza ordinata di tutti gli elementi di S.* Ad esempio, dato S= {a,b,c} esistono sei permutazioni di S: *abc, acb, bac, bca, cab, cba*. Per un insieme di n elementi, abbiamo n! permutazioni.
- Tornando al cifrario di Cesare e dato S come l'insieme dei 26 caratteri dell'alfabeto inglese, tale cifrario usa una sola permutazione di tali caratteri, quella che sposta ogni carattere in quello «+3». Se invece, per ogni messaggio, si potesse cifrare spostando il plaintext in una delle possibili permutazioni (sono 26!), il numero di chiavi possibili aumenterebbe a dismisura e sarebbe più robusto contro attacchi a forza bruta!
- Facciamo un esempio. Consideriamo la seguente permutazione delle 26 lettere (che poi è la chiave!):

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
DXUTNAVWKZFQGSIOYJBPLHCERM
```

- Con questa permutazione il plaintext «arrivano rinforzi» diventa:

ARRIVANORINFORZI

DJJKHDSIJKSAIJMK

- Un attacco a forza bruta per trovare la chiave giusta richiederebbe provare, nel caso peggiore **26!** Chiavi, invece che 25 chiavi come nel caso del cifrario di Cesare. Lo spazio delle chiavi aumenta in maniera considerevole!

Approfondimenti:
Cifrario monoalfabetico /2

Nonostante lo spazio delle chiavi possibili aumenti considerevolmente, portando praticamente ad una impossibilita' di un attacco a forza bruta, i crittoanalisti possono sfruttare le *regolarità del linguaggio* per provare a trovare la permutazione vincente. Consideriamo ad esempio il seguente messaggio cifrato con cifrario monoalfabetico:

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ

- Un attaccante potrebbe calcolare la frequenza delle singole lettere cifrate e confrontarla con la frequenza delle lettere nella lingua di riferimento (nell'ipotesi che questa informazione sia data.
- Più è lungo il messaggio, maggiore è la probabilità che la distribuzione delle lettere cifrate corrisponda alle corrispondenti lettere in chiaro. Vediamolo meglio nel seguito.
- Attenti crittoanalisti possono sfruttare queste informazioni per trovare corrispondenze tra le lettere più frequenti, inoltre possono cercare coppie di lettere vicine per parole comuni come «of», …
- Dopo attente ricerche e analisi, il risultato ottenuto è:

```
it was disclosed yesterday that several informal but
   direct contacts have been made with political
    representatives of the viet cong in Moscow
```

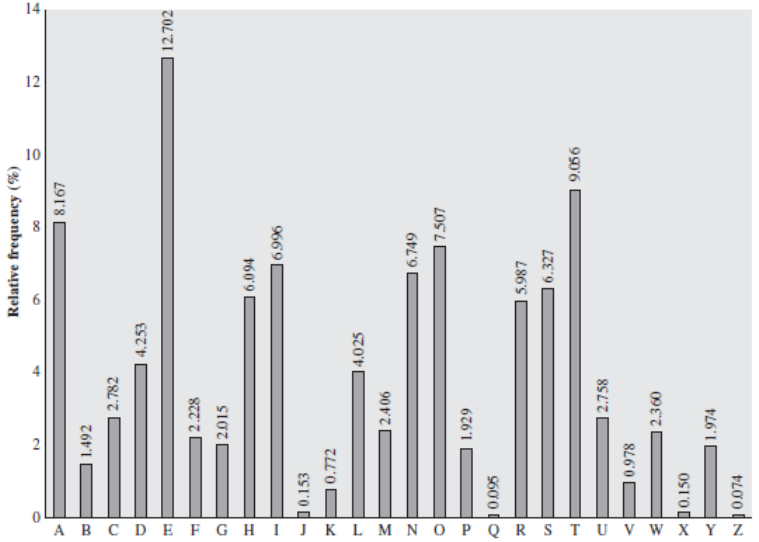
- *Lo studente è invitato a provare a fare l'analisi e a verificare la correttezza della soluzione proposta.*
- Il cifrario monoalfabetico può essere esteso usando più alfabeti. Lo studente ricerchi e studi il *cifrario polialfabetico di Vigenere*.

Approfondimenti:
Cifrario monoalfabetico permutazioni

- Ad esempio se calcoliamo la frequenza delle lettere nel ciphertext otteniamo:

P 13.33	H 5.83	F 3.33	B 1.67	C 0.00
Z 11.67	D 5.00	W 3.33	G 1.67	K 0.00
S 8.33	E 5.00	Q 2.50	Y 1.67	L 0.00
U 8.33	V 4.17	T 2.50	I 0.83	N 0.00
O 7.50	X 4.17	A 1.67	J 0.83	R 0.00
M 6.67				

- Mentre le frequenza delle cifre nella lingua inglese sono riportate qui a fianco →



Approfondimento:
Tecniche di Trasposizione

Come abbiamo visto, le tecniche di sostituzione portano al cambiamento di una lettera del plaintext con un'altra per ottenere il ciphertext. Un altro approccio è l'utilizzo di tecniche di **trasposizione** dove il ciphertext è ottenuto *modificando l'ordine delle lettere* del plaintext.

Ad esempio, il sistema più semplice è quello del rail fence (recinto ferroviario) dove il plaintext è scritto in sequenze di diagonali e poi letto in riga. Per capire con un esempio, il messaggio «meet me after the toga party» può essere scritto come segue:

```
m e m a t r h t g p r y
  e t e f e t e o a a t
```

ed essere letto riga per riga, diventando: MEMATRHTGPRYETEFETEOAAT

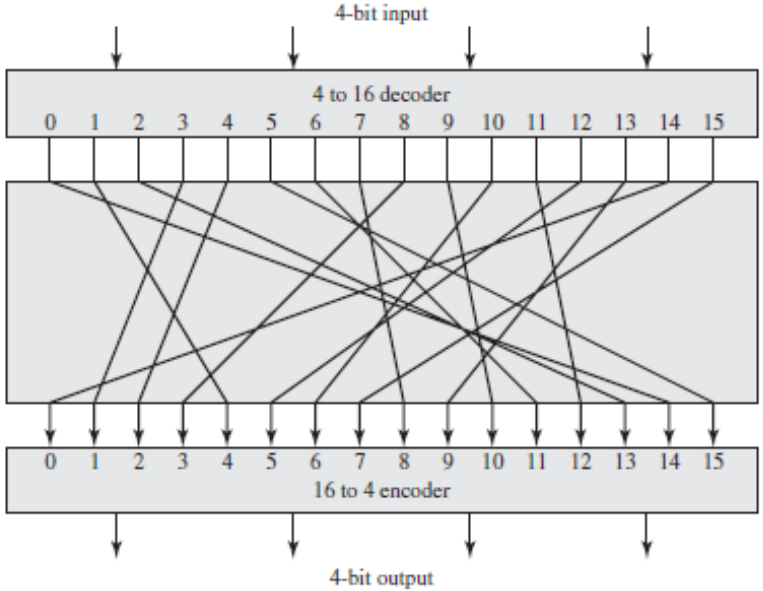
Tuttavia, per un crittoanalista la soluzione della trasposizione precedente è banale. Una trasposizione migliore potrebbe essere scrivere in un quadrato il messaggio riga per riga, e poi leggerle colonna per colonna fissando un certo ordine di lettura tra le colonne (questo ordine è la chiave!). Prendiamo ad esempio:

```
Key:          4 3 1 2 5 6 7
Plaintext:    a t t a c k p
              o s t p o n e
              d u n t i l t
              w o a m x y z
```

- Otteniamo il ciphertext TTNAAPTMTSUOAODWCOIXKNLYPETZ leggendo le colonne nell'ordine imposto dalla chiave data.

Approfondimento:
Cifrario ideale di Feistel

- Su n bit, sono possibili $2^n!$ trasformazioni *reversibili*. Come mai proprio questo numero? L'idea è che dato il primo plaintext possiamo scegliere 2^n trasformazioni. Per il secondo, per motivi di *reversibilità*, dobbiamo scartare quello già scelto e possiamo quindi scegliere tra $2^n - 1$, e cosi via… .



Approfondimento: Cifrario ideale di Feistel /2

L'idea è che un blocco di input di 4 bit produce potenzialmente 16 configurazioni di input (banalmente $16=2^4$ ovvero tutte le configurazioni possibili su 4 bit!) che possono essere mappati su una di 16 configurazioni di output sui 4 bit di uscita. Un esempio di mapping completo di cifratura/decifratura su 4 bit è dato dalle tabelle qui sotto (*W. Stalling: Cryptography and Network Security, International Edition, Pearson)*:

Plaintext	Ciphertext
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

Ciphertext	Plaintext
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

Ma tale cifrario è usabile? Ci sono problemi? No, perché:

- Blocchi di piccole dimensioni sono soggetti all'analisi di frequenza di n sistemi a sostituzione
- Il cifrario ideale è irrealizzabile in pratica
- Serve pertanto una approssimazione del cifrario a blocchi, in modo che sia facilmente realizzabile.

Approfondimento:
Cifrario ideale di Feistel /3

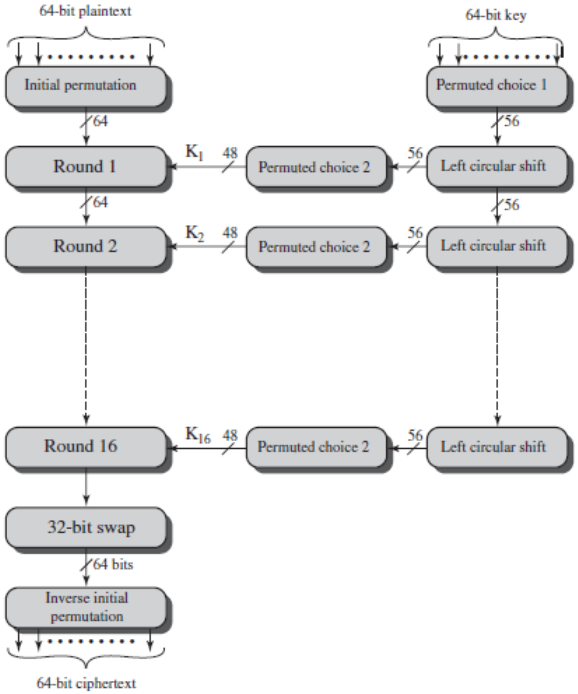
- Data l'inapplicabilità del cifrario ideale, lo stesso Feistel ha proposto di utilizzare una approssimazione del cifrario ideale con un cifrario ottenuto facendo il prodotto di due cifrari più semplici. In questo caso con prodotto intendiamo l'esecuzione, in sequenza di due cifrari. Il primo cifrario genere un ciphertext intermedio che viene dato in input ad un secondo cifrario.
- In questo modo è possibile ottenere un cifrario a blocchi di n bit che lavora su una chiave di k bit (k<n).
- Tale tipo di cifrario si basa su 2 trasformazioni: le sostituzioni (che abbiamo già visto in precedenza) e le permutazioni, dove un plaintext di n bit viene trasformato in una delle sue possibili permutazioni, senza aggiunte di alcun tipo.
- Il cifrario ottenuto permette di soddisfare due proprietà fondamentali per gli schemi crittografici, ovvero la **diffusione** e la **confusione**.
  - **Diffusione:** serve per dissolvere la struttura statistica del plaintext in statistiche lungo range maggiori nel ciphertext. Ogni cifra del plaintext influenza più cifre o simboli del ciphertext. Esempio: facendo una media
  - **Confusione:** cerca di rendere complicata la relazione tra le statistiche del ciphertext e la chiave per criptare. Esempio: uso di un algoritmo di sostituzione COMPLICATO.

Approfondimento:
Data Encryption Standard (DES)

- Il DES è stato l'algoritmo di cifratura standard in America a partire dal 1977.
- È un algoritmo di cifratura a blocchi che lavora su blocchi di *64 bit* con una chiave di *56 bit*. Banalmente, la stessa dimensione di chiave e blocchi è utilizzata nella fase di decifratura.
- Il DES ha avuto uno sviluppo enorme ma è stato anche oggetto, in 40 anni, di molte critiche legate alla sua potenziale efficacia e sulla sicurezza effettivamente garantita.
- A fine anni '60, l'IBM ha investito in un progetto di crittografia, dando a Henry Feistel la direzione. Il progetto, terminato nel 1971, ha portato alla definizione di un algoritmo chiamato LUCIFER , un cifrario a blocchi con blocchi di 64 bit e chiave di 128 bit.
- Dato il successo di LUCIFER, IBM iniziò a lavorare ad una versione commercial dello stesso, in grado di "girare" su singolo chip. Tale sforzo si concretizzò (anche con l'aiuto di alcuni esperti della NSA) in una versione con blocchi a 64 bit ma con una chiave di «soli» 56-bit.
- Nel 1973, la National Bureau of Standard indisse una gara per definire uno standard di cifratura nazionale. Vinse il nuovo LUCIFER commerciale. E venne adottato, sotto il nome di DES, a partire dal 1977.
- Ad eccezione della permutazione iniziale e finale, il DES è a tutti gli effetti un cifrario di Feistel.
  - La chiave viene inizialmente *permutata*
  - Per ogni round, una sottochiave (K_i) è prodotta combinando uno *spostamento* (shift) *a sinistra* e una *permutazione*.
  - La funzione di permutazione è *identica* per tutti i round, ma viene comunque prodotta una chiave sempre diversa a causa dello *shift* a sinistra.

Approfondimento: cifratura DES

- La figura qui a lato riassume il processo di cifratura a blocchi del DES.
- Partendo da sinistra vediamo che il processo di cifratura avviene tramite tre fasi:
  - Inizialmente il plaintext di 64 bit subisce una *permutazione iniziale* (IP);
  - All'input permutato viene applicata, per *16 volte*, una funzione di cifratura che contiene sia permutazioni che sostituzioni ed è funzione sia del plaintext che della chiave. Alla fine viene prodotto un *pre-output* ottenuto tramite swap delle due metà ottenute dal 16° round.
  - Il ciphertext è calcolato applicando al *pre-output* la permutazione iniziale inversa.



Approfondimento:
AES: cenni storici

- Dopo 20 anni di onorata carriera, il DES stava finendo la sua carriera, anche per il fatto che calcolatori più veloci iniziavano a rendere possibile attacchi a forza bruta alla chiave di 56 bit, rischiando di rendere il cifrario obsoleto per le generazioni di applicazioni future.
- Per questo, a metà degli anni 90 il NIST decise che era giunta l'ora di trovare un nuovo algoritmo standard che avrebbe preso il nome di Advanced Encryption Standard (AES).
- Pertanto, indi' un concorso affinché crittografi da ogni parte del mondo proponessero cifrari nuovi, tra i quali trovare il nuovo standard AES. Gli unici vincoli posti per gli schemi candidati erano i seguenti:
  - Lo schema doveva essere a blocchi e simmetrico.
  - L'intero schema deve essere reso pubblico
  - Lo schema avrebbe dovuto supportare chiavi di lunghezza 128, 192 e 256 bit (giusto per difendersi da attacchi a forza bruta in futuro)
- Nell'ottobre 2000, il candidato vincitore risultò essere il **Rijndael**, dai nomi dei due ricercatori belgi *Joan Daemen* e *Vincent Rijimen* che lo avevano proposto.
- Come il DES, il Rijndael si basa su sostituzioni e permutazioni e su diversi round di cifratura.
- Il numero di round dipendono dalla lunghezza della chiave e dei blocchi, in un intervallo di round comunque compreso tra 10 e 14.
- A differenza del DES, il *Rijndael* (che d'ora in poi chiameremo **AES** anche se la versione adottata è leggermente diversa da quella originale):
  - è una *rete a sostituzione e permutazione*, non una rete di *Feistel* ma implementa comunque i principi di confusione e diffusione.
  - Lavora su interi *byte*, anziché bit come il DES risultando molto più efficiente sia per implementazioni a livello software che a livello hardware.

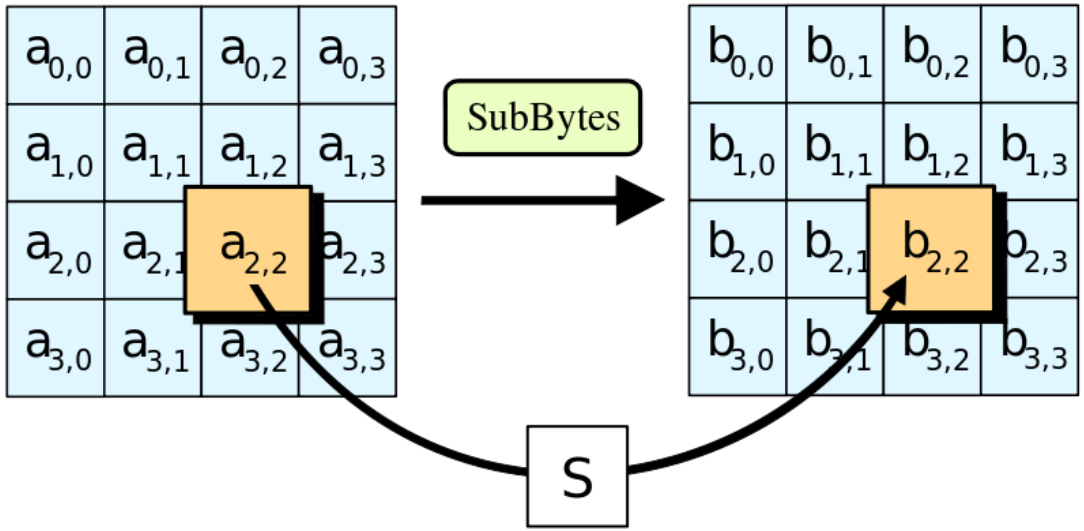
Approfondimento: l'algoritmo AES/1

- Nella sua formulazione finale, l' AES opera utilizzando matrici di 4×4 byte chiamate stati (states).
- Quando l'algoritmo ha blocchi di 128 bit in input, la matrice State ha 4 righe e 4 colonne; se il numero di blocchi in input diventa di 32 bit più lungo, viene aggiunta una colonna allo State, e così via fino a 256 bit. In pratica, si divide il numero di bit del blocco in input per 32 e il quoziente specifica il numero di colonne.
- E' previsto un passaggio iniziale detto AddRoundKey, più altri 4 passaggi che vengono ripetuti per ogni round: *SubBytes*, *ShiftRows*, *MixColumns* (non eseguito nell'ultimo round) e *AddRoundKey* (nuovamente).
- L'AES si basa su una **aritmetica polinomiale su un GF(2^8).**
- Passaggio iniziale:
- 0. **AddRoundKey** – Ogni byte della tabella viene combinato con la chiave di sessione, la chiave di sessione viene calcolata dal gestore delle chiavi. Il gestore delle chiavi è quello che per il DES abbiamo chiamato key scheduling, ovvero quell'algoritmo che calcola le sottochiavi a partire dalla chiave di sessione, nei vari round.
- Il round di cifratura consiste nei seguenti quattro passaggi:

1. **SubBytes** – Sostituzione non lineare di tutti i byte che vengono rimpiazzati secondo una specifica tabella.
2. **ShiftRows** – Spostamento dei byte di un certo numero di posizioni dipendente dalla riga di appartenenza
3. **MixColumns** – Combinazione dei byte con un'operazione lineare, i byte vengono trattati una colonna per volta.
4. **AddRoundKey** – Ogni byte della tabella viene combinato con la chiave di sessione, la chiave di sessione viene calcolata dal gestore delle chiavi.

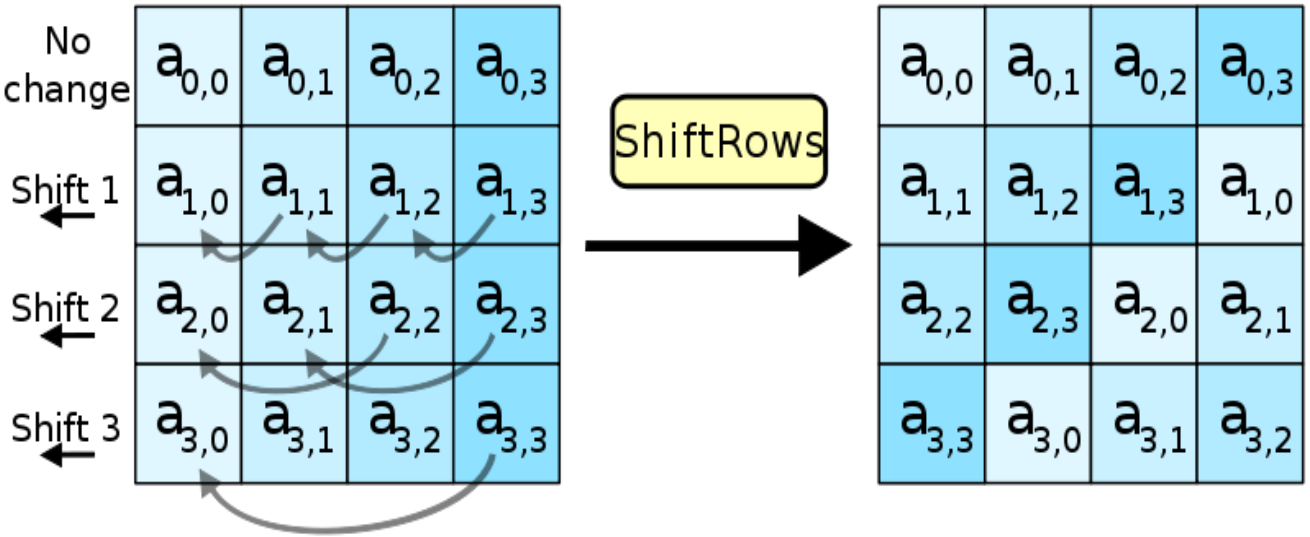
Approfondimento: l'algoritmo AES/2

- Nel passaggio *SubBytes* ogni byte della matrice viene modificato tramite una S-box a 8 bit. Questa operazione provvede a fornire la *non linearità* all'algoritmo. La S-box utilizzata è derivata da una funzione inversa nel campo finito GF(2^8), conosciuta per avere delle ottime proprietà di non linearità. Per evitare un potenziale attacco basato sulle proprietà algebriche la S-box è costruita combinando la funzione inversa con una trasformazione affine invertibile. La S-box è stata scelta con cura per non possedere né punti fissi né punti fissi opposti.



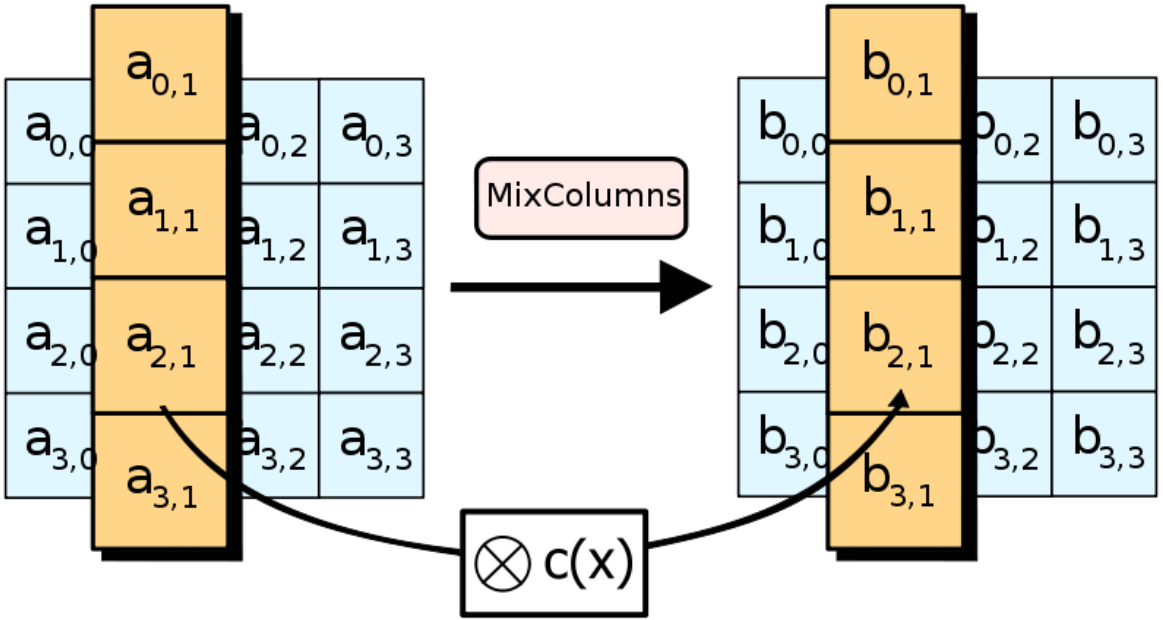
Approfondimento: l'algoritmo AES/3

- Il passaggio ShiftRows provvede a scostare le righe della matrice di un parametro dipendente dal numero di riga. Nell'AES la prima riga resta invariata, la seconda viene spostata di un posto verso sinistra, la terza di due posti e la quarta di tre. In questo modo l'ultima colonna dei dati in ingresso andrà a formare la diagonale della matrice in uscita. (Il Rijndael originale utilizza un disegno leggermente diverso per via delle matrici di lunghezza non fissa.) Tutte le operazioni sono effettuate utilizzando l'indice della colonna "modulo" il numero di colonne.



Approfondimento: l'algoritmo AES/4

- Il passaggio *MixColumns* prende i quattro byte di ogni colonna e li combina utilizzando una trasformazione lineare invertibile. Utilizzati in congiunzione, ShiftRows e MixColumns provvedono a far rispettare il criterio di *confusione* e *diffusione*. Ogni colonna è trattata come un polinomio in GF(28) e viene moltiplicata modulo x^4+1 per un polinomio fisso $3x^3 + x^2 + x + 2$.



Approfondimento: l'algoritmo AES/5

- Il passaggio AddRoundKey combina con uno XOR la chiave di sessione con la matrice ottenuta dai passaggi precedenti (State).
- Una *chiave di sessione* viene ricavata dalla chiave primaria ad ogni round (con dei passaggi più o meno semplici, ad esempio uno *shift* di posizione dei bit) grazie all'algoritmo di schedulazione della chiave.