

Appunti Sicurezza su Reti

Anno 2018/2019

Autore: Del Gaudio Giovanni

Docente: Alfredo De Santis

**Capitolo 1 Crittografia e Steganografia**

**Una delle prime tecniche di comunicazione segrete, basata sull'occultamento del messaggio, si chiama *steganografia,*dalle parole greche *steganós*, che significa coperto, e *gráphein*, che significa scrivere. La steganografia, è una tecnica che si prefigge di nascondere la comunicazione tra due interlocutori, generalmente i messaggi nascosti sembrano(o fanno parte di) qualcos’altro. Ad esempio, il messaggio nascosto può essere un inchiostro invisibile tra le linee di una lettera privata. Alcune implementazioni della steganografia che non dispongono di un segreto comune sono forme di**[***security through obscurity***](https://it.wikipedia.org/wiki/Sicurezza_tramite_segretezza)**, e gli schemi steganografici a chiave dipendono dal**[**principio di Kerckhoffs**](https://it.wikipedia.org/wiki/Principio_di_Kerckhoffs)**.**[**[3]**](https://it.wikipedia.org/wiki/Steganografia#cite_note-3)

**La steganografia, al contrario della**[**crittografia**](https://it.wikipedia.org/wiki/Crittografia)**, consente di nascondere un messaggio all'interno di un vettore che possa consentirne il trasporto senza destare sospetti.**

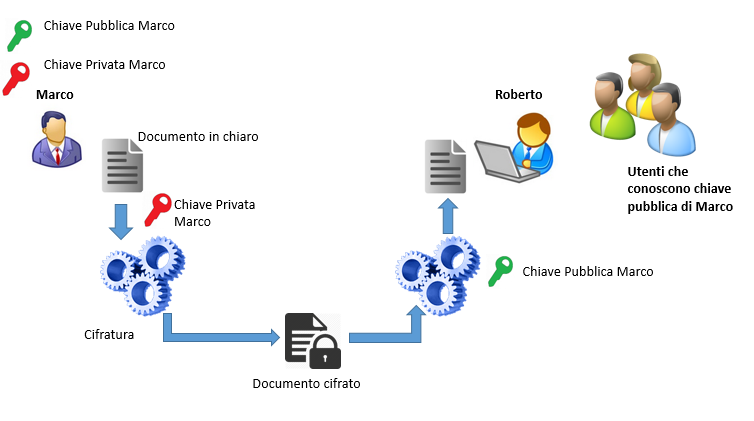
**La steganografia comprende l'occultamento di informazioni all'interno di file di computer. Nella steganografia digitale, le comunicazioni elettroniche possono includere la codifica steganografica all'interno di un livello di trasporto, ad esempio un file di documento, un file di immagine, un programma o un protocollo. I file multimediali sono ideali per la trasmissione steganografica a causa della loro grande dimensione. Ad esempio, un mittente potrebbe inviare un file di immagine innocuo e regolare il colore di un**[**pixel**](https://it.wikipedia.org/wiki/Pixel)**ogni cento per farlo corrispondere a un carattere alfabetico. La modifica è così sottile che qualcuno che non lo cerca in modo specifico è improbabile che noti la modifica.**

**Negli anni sono state impiegate in tutto il mondo innumerevoli  forme di steganografia. Perciò in parallelo con lo sviluppo della steganografia si assisté all'evoluzione della crittografia, dal greco kryptós, che significa nascosto. La crittografia non mira a nascondere il messaggio in sé, ma il suo significato. Per rendere incomprensibile un testo, lo si altera per mezzo di un procedimento concordato a suo tempo dal mittente e dal destinatario. Questi può quindi invertire il procedimento, e ricavare il messaggio originale. Il vantaggio della crittografia è che anche se il nemico intercetta il messaggio, esso risulta incomprensibile e quindi inutilizzabile. Infatti il nemico, non conoscendo il procedimento di alterazione, dovrebbe trovare difficile, se non impossibile, ricostruire il significato.**

**Non tutte le società antiche svilupparono forme di crittografia. La Cina, per esempio, l'unica civiltà antica ad usare una scrittura ideografica, non ne ha mai viste. Le ragioni, a detta degli storici, sono legate alla natura prevalentemente orale delle comunicazioni. Anche se la steganografia e la crittografia sono discipline indipendenti, possono essere impiegate per alterare e occultare il medesimo testo, garantendo un livello di sicurezza molto più alto. Per esempio, il «microdot », cioè la riduzione di uno scritto alle dimensioni di un punto, è una forma di steganografia che ebbe largo impiego durante la seconda guerra mondiale. Tramite un procedimento fotografico, gli agenti tedeschi in America latina trasformavano una pagina scritta, precedentemente crittografata, in una macchia con un diametro inferiore al millimetro, che poteva essere nascosta nel puntino di una « i » in una comunicazione banale. Il primo microdot fu scoperto dall' FBI nel 1941 grazie a una soffiata.**

**Capitolo 1 Crittosistemi Simmetrici**

**Con crittografia simmetrica o crittografia a chiave privata , si intende una tecnica di cifratura. Rappresenta un metodo semplice per cifrare testo in chiaro dove la chiave di crittazione è la stessa chiave di decrittazione, rendendo l’algoritmo molto performante e semplice da implementare. Tuttavia si presuppone che le due parti siano già in possesso delle chiavi, richiesta che non rende possibile uno scambio di chiavi con questo genere di algoritmi. Lo scambio avviene attraverso algoritmi a chiave simmetrica o pubblica, generalmente più complessi sia da implementare che da eseguire ma che permettono questo scambio in modo sicuro. Dopodichè la comunicazione verrà crittata usando solo algoritmi a chiave simmetrica per garantire una comunicazione sicura ma veloce.**

****

**Capitolo 1 Crittosistemi Simmetrici Funzionamento**

**In questo genere di algoritmi si suppone che entrambe le parti conoscono già la chiave con cui crittare e decrittare il messaggio. Il mittente (come in figura Marco) ha un messaggio P(PlaintText o testo in chiaro). Il mittente critta il messaggio P con la chiave pubblica(come in figura chiave verde) con la chiave K(come in figura chiave rossa) usando un algoritmo di crittografia simmetrica chiamato S. Il messaggio risultante sarà il testo in chiaro C(CypherTex o messaggio cifrato). Quindi l’algoritmo viene eseguito in questo modo:**

**Marco e Roberto scelgono per prima cosa una chiave a caso (denominata k Î *K), lo fanno in un posto segreto in cui il nemico non può osservarli, o alternativamente, in posti diversi tramite un canale privato.***

***Supponiamo che il messaggio in chiaro sia una stringa***

***X=* x1x2.... xn.**

**Per qualche interno  n >1, dove ogni simbolo del testo in chiaro xi appartiene a  *P*, 1  < i  < n. Ogni xiè cifrato usando la regola di cifratura ekspecificata dalla predeterminata chiave k. Quindi,Marco calcola yi= ek(xi), 1  <= i  <= n, e la risultante stringa cifrata**

**y =  y1y2.... yn**

**è spedita sul canale pubblico. Quando Roberto riceve y1y2....yn lo decifra usando la funzione di decifratura dk , ottenendo l'originale testo in chiaro x1x2.... xn. Chiaramente, la funzione di cifratura ek è una funzione iniettiva perché altrimenti la decifratura potrebbe avvenire in modo ambiguo. Per esempio, se    y = ek(x1) = ek(x2) dove x1¹x2, allora Roberto non ha modo di sapere se y, decifrato, fornisce x1 oppure x2. Notiamo che se *P* = *C* segue che ogni funzione di cifratura è  una permutazione, nel senso che il testo cifrato è ottenuto come permutazione dei caratteri del testo in chiaro.**

**Se un crittosistema deve essere di uso comune, deve soddisfare alcune proprietà, tra cui:**

* **Ogni funzione di cifratura eked ogni funzione di decifratura dk devono essere calcolabili efficientemente.**
* **Un nemico, guardando la stringa cifrata y, non deve essere capace di determinare la chiave k usata per la cifratura, oppure la stringa del testo in chiaro x.**

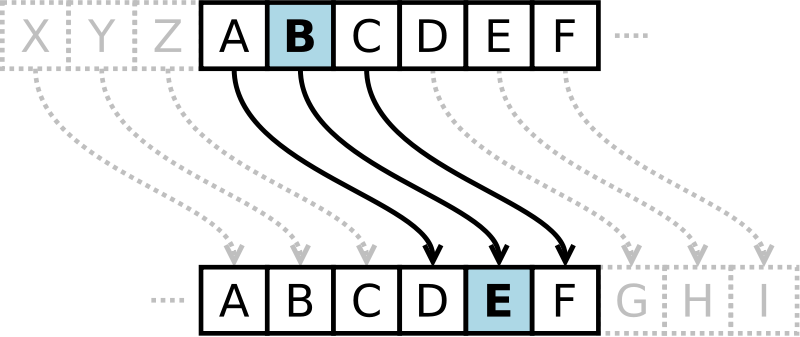
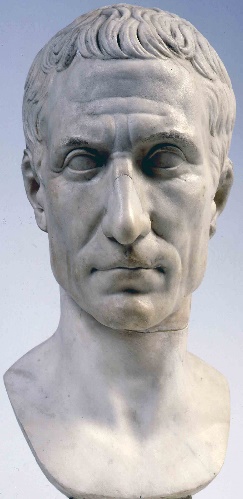
**Un’assunzione da fare è che il nemico conosce sempre l'algoritmo di cifratura ma non la chiave.  Quest’assunzione è fatta per due motivi:**

* **per trovarsi nel caso peggiore**
* **perché  i sistemi che saranno presi in considerazione possono essere utilizzati da utenti diversi, pertanto è improbabile che l'algoritmo possa restare segreto.**

**Capitolo 1.2 : Crittografia Antica Cifrario di Cesare**

**In crittografia il cifrario di Cesare è uno dei più antichi algoritmi crittografici di cui si abbia traccia storica. E’ un cifrario a sostituzione monoalfaberica in cui ogni lettera del testo in chiaro è sostituta del testo cifrato della lettera che si trova un certo numero di posizioni dopo nell’alfabeto. Questi tipi di cifrari sono detti anche cifrari a sostituzione o cifrari a scorrimento a causa del loro modo di operare:**

**la sostituzione avviene lettere per lettera, scorrendo il testo dall’inizio alla fine.**

** **

**Il cifrario di Cesare prende il nome da giulio cesare, che lo utilizzava per proteggere i suoi messaggi segreti. Giulio Cesare utilizzava in genere una chiave di 3 per il cifrario, come nel caso della corrispondenza militare inviata alle truppe comandate da Quinto Tullio Cicerone. Al tempo era sicuro perché gli avversari spesso non erano neanche in grado di leggere un testo in chiaro, ben che mai un testo cifrato; inoltre non esistevano metodi di crittoanalisi in grado di rompere tale codice, per quanto banale.**

**In particolare Cesare utilizzava uno spostamento di 3 posizioni (la chiave, cioè ciò che indica di quanto spostarsi dunque era 3 mostrato sopra in figura), secondo il seguente schema nell’alfabeto latino classico che , aveva 23 caratteri:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Testo in chiaro** | **a** | **b** | **c** | **d** | **e** | **F** | **g** | **h** | **i** | **k** | **l** | **m** | **n** | **o** | **P** | **q** | **r** | **s** | **t** | **v** | **x** | **y** | **z** |
| **Testo cifrato** | **D** | **E** | **F** | **G** | **H** | **I** | **K** | **L** | **M** | **N** | **O** | **P** | **Q** | **R** | **S** | **T** | **V** | **X** | **Y** | **Z** | **A** | **B** | **C** |

**Lo stesso si può fare con l’alfabeto italiano di 21 caratteri:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Testo in chiaro** | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **l** | **m** | **n** | **o** | **p** | **q** | **r** | **s** | **t** | **u** | **v** | **z** |
| **Testo cifrato** | **D** | **E** | **F** | **G** | **H** | **I** | **L** | **M** | **N** | **O** | **P** | **Q** | **R** | **S** | **T** | **U** | **V** | **Z** | **A** | **B** | **C** |

**Per cifrare un messaggio, basta prendere ogni lettera del testo in chiaro e sostituirla con la corrispondente lettera della riga testo cifrato. Per decifrare, viceversa. Ecco un semplice esempio italiano(coerentemente con l’uso antico di omettere gli spazi tra le parole nei papiri, nel testo questi sono omessi; questo aumenta anche la sicurezza del cifrario perché toglie un importante punto di riferimento, cioè la suddivisione in parole , a chi tentasse la decrittazione):**

|  |  |
| --- | --- |
| **Testo in chiaro** | **attaccaregliirriducibiligalliallaorasesta** |
| **Testo crittato** | **DZZDFFDUHLONNUUNGAFNENONLDOONDOODRUDVHVZD** |

**Cifrare (e decifrare) il testo più volte non migliora la sicurezza, in quanto una rotazione di A posti seguita da una di B posti equivale ad una A + B . Matematicamente parlando, la cifratura con le varie chiavi forma un gruppo.**

# **Rompere la Cifratura del cifrario di Cesare**

**Con un diagramma delle frequenze delle lettere nel testo in cifra e nella lingua originale del testo, è facile individuare il valore della chiave osservandone la disposizione. Ad esempio, nella**[**lingua italiana**](https://it.wikipedia.org/wiki/Lingua_italiana)**, le lettere più frequenti sono le vocali E, A, O ed I, con lievi differenze, seguite dalle consonanti L, N, R, S e T, mentre sono rare B, F, Q e Z e praticamente assenti le lettere *straniere* J, K, Y, X e W.**

**Nella**[**lingua inglese**](https://it.wikipedia.org/wiki/Lingua_inglese)**, invece, le lettere più frequenti sono E e T, mentre le più rare sono Q e Z. Anche i computer sono in grado di eseguire simili calcoli senza difficoltà, come pure di effettuare attacchi di tipo**[**brute force**](https://it.wikipedia.org/wiki/Metodo_forza_bruta)**semplicemente provando tutte le chiavi di cifratura possibili finché non appare un testo leggibile.**

**Il cifrario di Cesare è molto più debole di uno in cui la corrispondenza delle lettere sia casuale, come succede in molti**[**giochi enigmistici**](https://it.wikipedia.org/wiki/Enigmistica)**. I luoghi più comuni dove si può trovare questo cifrario è in giochi per bambini oppure nell'algoritmo**[**ROT13**](https://it.wikipedia.org/wiki/ROT13)**, che è fatto apposta per essere facile da decifrare.**

**CAPITOLO 1.3 : CIFRARIO DI Vigenére**

****

**Il cifrario di Vigenére è il più semplice dei cifrari polialfabetici. Si basa sull’uso di un versetto per controllare l’alternanza degli alfabeti di sostituzione , concetto introdotto per la prima volta da Giovan Battista Bellaso. Il metodo si può considerare una generalizzazione del cifrario di Cesare. Invece di spostare sempre dello stesso numero di posti la lettera da cifrare, questa viene spostata di un numero di posti variabile ma ripetuto, determinando in base ad una parola chiave, da concordarsi tra mittente e destinatario, e da scrivere ripetutamente sotto il messaggio, carattere per carattere; la chiave era detta anche verme, per il motivo che m essendo in genere molto più corta del messaggio, deve essere ripetuta molte volte sotto questo, come nel seguente esempio :**

**Testo in chiaro - RAPPORTOIMMEDIATO**

**Verme - VERMEVERMEVERMEVE**

**Testo cifrato - MEGBSMXFUQHIUUEOS**

**Il testo cifrato si ottiene spostando la lettera chiara di un numero fisso di caratteri, pari al numero ordinale della lettera corrispondente del verme. Di fatto si esegue una somma aritmetica tra l’ordine del chiaro(A=0, B=1, C=2) e quello del verme; se si supera l’ultima lettera , la Z si ricomincia dalla A, secondo la logica delle aritmetica finite. Il vantaggio rispetto ai cifrari monoalfabetici(come il cifrario di Cesare o quelli per sostituzione delle lettere con simboli/altre lettere) è evidente: il testo è cifrato con n alfabeti cifranti. In questo modo, la stessa lettera viene cifrata (se ripetuta consecutivamente) n volte: ciò rende quindi più complessa la crittoanalisi del testo cifrato.**

**Si può usare ovviamente una funzione matematica per la criptazione e decriptazione ; in entrambe useremo sempre le stesse lettere:**

**Numero prima lettera del cifrario (A) = 0**

**Numero ultima lettera del cifrario (Z) = 25**

***L* = Lunghezza del cifrario = Numero elementi dell'insieme (26)**

***a* = Numero della lettera della parola in Chiaro (0-25)**

***b* = Numero della lettera della parola Chiave/Verme (0-25)**

***c* = Numero della lettera della parola Criptata (0-25)**

***Per criptare*: *n* = *a + b (mod 26)***

***Per decriptare*: *n* = *c - b + 26***

**r[parte intera del numero] = n/L**

***F(x)* = *n - ( L \* r )* = Numero della lettera della parola in Chiaro/Criptata (0-25)**

**La funzione si basa semplicemente sulla somma/sottrazione dei numeri delle lettere e dividere per la *lunghezza* del cifrario per ottenere il numero della lettera desiderata. Per ottenere SEMPRE un numero *n* positivo anche per la decriptazione, in quanto una sottrazione, basta ricorrere al semplice aritificio di aggiungere 26, in quanto verrà poi eliminato grazie ad *r*.**

**Esempio di criptazione con le prime due lettere dell'esempio testo/chiave precedente.**

***L* = 26**

***a[R]* = 17**

***b[V]* = 21**

***n* = *17 + 21* = 38**

***r* = *38 / 26 = 1,461...* = 1**

***F(x)* = *38 - ( 26 \* 1 )* = *38 - 26* = 12**

***F(12)* = M**

**Secondo esempio di Criptazione**

***L* = 26**

***a[A]* = 0**

***b[E]* = 4**

***n* = *0 + 4* = 4**

***r* = *4 / 26 = 0,153...* = 0**

***F(x)* = *4 - ( 26 \* 0 )* = *4 - 0* = 4**

***F(4)* = E**

**Esempio di decriptazione con le prime due lettere dell'esempio testo/chiave precedente.**

**Primo esempio di decriptazione**

***L* = 26**

***b[V]* = 21**

***c[M]* = 12**

***n* = *12 - 21 + 26* = 17**

***r* = *17 / 26 = 0,653...* = 0**

***F(x)* = *17 - ( 26 \* 0 )* = *17 - 0* = 17**

***F(17)* = R**

***Secondo esempio di decriptazione***

***L* = 26**

***b[E]* = 4**

***c[E]* = 4**

***n* = *4 - 4 + 26* = 26**

***r* = *26 / 26 = 1* = 1**

***F(x)* = *26 - ( 26 \* 1 )* = *26 - 26* = 0**

***F(0)* = A**

## Tavola di Vigenère

**Per semplificare la cifratura, Vigenère propose l'uso della seguente "matrice" quadrata, composta da alfabeti ordinati e spostati. Se si vuole cifrare, con la chiave dell'esempio precedente, la lettera "R" della parola *rapporto* basterà trovare la lettera "R" nella prima riga, individuando la colonna relativa. Basterà poi trovare la "V" di "verme" nella prima colonna per trovare la riga, individuando, tramite l'incrocio, la lettera corretta da usare.**

**A B C D E F G H I J K L M N O P Q R S T U V W X Y Z**

**B C D E F G H I J K L M N O P Q R S T U V W X Y Z A**

**C D E F G H I J K L M N O P Q R S T U V W X Y Z A B**

**D E F G H I J K L M N O P Q R S T U V W X Y Z A B C**

**E F G H I J K L M N O P Q R S T U V W X Y Z A B C D**

**F G H I J K L M N O P Q R S T U V W X Y Z A B C D E**

**G H I J K L M N O P Q R S T U V W X Y Z A B C D E F**

**H I J K L M N O P Q R S T U V W X Y Z A B C D E F G**

**I J K L M N O P Q R S T U V W X Y Z A B C D E F G H**

**J K L M N O P Q R S T U V W X Y Z A B C D E F G H I**

**K L M N O P Q R S T U V W X Y Z A B C D E F G H I J**

**L M N O P Q R S T U V W X Y Z A B C D E F G H I J K**

**M N O P Q R S T U V W X Y Z A B C D E F G H I J K L**

**N O P Q R S T U V W X Y Z A B C D E F G H I J K L M**

**O P Q R S T U V W X Y Z A B C D E F G H I J K L M N**

**P Q R S T U V W X Y Z A B C D E F G H I J K L M N O**

**Q R S T U V W X Y Z A B C D E F G H I J K L M N O P**

**R S T U V W X Y Z A B C D E F G H I J K L M N O P Q**

**S T U V W X Y Z A B C D E F G H I J K L M N O P Q R**

**T U V W X Y Z A B C D E F G H I J K L M N O P Q R S**

**U V W X Y Z A B C D E F G H I J K L M N O P Q R S T**

**V W X Y Z A B C D E F G H I J K L M N O P Q R S T U**

**W X Y Z A B C D E F G H I J K L M N O P Q R S T U V**

**X Y Z A B C D E F G H I J K L M N O P Q R S T U V W**

**Y Z A B C D E F G H I J K L M N O P Q R S T U V W X**

**Z A B C D E F G H I J K L M N O P Q R S T U V W X Y**

**Chi riceve il messaggio per decifrarlo deve semplicemente usare il metodo inverso (sottrarre invece che sommare); riferendosi all'esempio di sopra si avrà:**

**Testo cifrato - MEGBSMXFUQHIUUEOS**

**Verme - VERMEVERMEVERMEVE**

**Testo chiaro - RAPPORTOIMMEDIATO**

**Usando la tavola quadrata si potrà decifrare la prima "M" ricercandola nella riga della corrispondente lettera del verme, la V; la colonna dove si trova la M ha quindi al primo posto in alto la lettera chiara, la R.**

* **Numero possibili di chiavi = 26t**
* **Resiste all’analisi delle frequenze = Una lettera cifrata corrisponde a più simboli in chiaro**

**CAPITOLO 1.4 Cifrario di Leon Battista Alberti**

**In crittografia il disco cifrante di Leon Battista Alberti descritto nel de cifris intorno al 1467, è il primo sistema di cifratura polialfabetica. L’apparecchio si compone di due dischi concentrici, rotanti uno rispetto all’altro, contenenti un alfabeto ordinato per il testo in chiaro (testo da cifrare) e un alfabeto disordinato (un secondo alfabeto, cioè ove le lettere non si susseguono più nell’ordine con il quale si susseguono nell’alfabeto normale) per il testo cifrato(testo risultante). Permette la sostituzione polialfabetica con periodo irregolare. Lo scorrimento degli alfabeti avviene per mezzo di lettere chiave inserite nel corpo del crittogramma. I quattro numeri possono anche servire per sopra cifrare 336 frasi di codice.**

**Esempio di cifratura – Indice mobile**

**Usa come indice una lettera minuscola scelta nel cerchio interno(mobile).**

**Stabilita la g come lettera indice e avendola giustapposta alla A maiuscola del cerchio esterno (stabile o fisso) lo sviluppo dei due alfabeti è il seguente come in figura**

**disco stabile: ABCDEFGILMNOPQRSTVXZ1234**

**disco mobile: gklnprtvz&xysomqihfdbace**

**Dispaccio da trasmettere : “La guerra si farà”**

**chiaro: LAGVER2RA**

**cifratura: AzgthpmamgQ**

**Il sistema prevede che la lettera chiave (A) sia inserita nel crittogramma all’inizio ed in maiuscolo. La a che cifra il 2 è una “nulla” che si omette nella decifrazione.**

**Dopo aver cifrato alcune lettere si inserisce nel cifrato a fronte della g un’altra lettera maiuscola (Q) ruotando il disco mobile in modo da ottenere le nuove corrispondenze:**

**disco stabile: ABCDEFGILMNOPQRSTVXZ1234**

**disco mobile: ysomqihfdbacegklnprtvz&x**

**La cifratura continuerò così:**

**chiaro: SIFARÀ.**

**cifratura: Qlfiyky.**

**Il risultato finale sarà quindi:**

**chiaro: LAGVER2RASIFARÀ.**

**cifratura: AzgthpmamgQlfiyky.**

**Esempio di cifratura – Indice Fisso**

**Si sceglie una lettera maiuscola dell’anello esterno come indice fisso. In questo esempio dell’indice A viene inizialmente sottoposta la m del cerchio mobile. Per indicare il cambio dell’alfabeto si cifra uno dei quattro numeri:**

**disco stabile: ABCDEFGILMNOPQRSTVXZ1234**

**disco mobile: mqihfdbacegklnprtvz&xyso**

**chiaro: LAGVERA3**

**cifratura: mcmbvfpms**

**A questo punto la presenza della s, cifra il numero 3, segnala la necessita di spostare i dischi portando la s sotto la A. La doppia R è omessa per non facilitare la decrittazione.**

**disco stabile: ABCDEFGILMNOPQRSTVXZ1234**

**disco mobile: somqihfdbacegklnprtvz&xy**

**La cifratura continuerò così:**

**chiaro: SIFARÀ.**

**cifratura: sndhsls.**

**La cifratura di Leon Battista Alberti è una sostituzione polialfabetica con alfabeti mischiati cambiati saltuariamente in modo segreto. Nella letteratura divulgativa questa cifra è a volte definita affine shift oppure Keyword Shift o meccanizzazione del cifrario di Vigenère , ma non ha alcun’affinità con questi sistemi.**

**La cifratura di Cesare è una sostituzione semplice, basata sullo spostamento di un unico alfabeto ordinato rispetto a sé stesso, con chiave fissa. Una volta scoperto il valore di una sola lettera, si conosce anche il valore di tutte le altre. Anche se l'alfabeto fosse mischiato, si potrebbe ugualmente arrivare a una facile soluzione ricorrendo allo studio delle frequenze letterali. Nella cifra di Alberti gli alfabeti sono due, mischiati, e la chiave varia in continuazione durante il messaggio, quindi la scoperta di una sola lettera non permette altri progressi nella decrittazione e lo studio delle frequenze non dà risultati perché la stessa lettera chiara è cifrata sempre in modo diverso. La cifratura Vigenère, oltre ad essere basata come quella di Cesare su soli alfabeti ordinati, è risolvibile dopo aver scoperto il periodo (che è fisso) per mezzo del**[**Metodo Kasiski**](https://it.wikipedia.org/wiki/Metodo_Kasiski)**, il che non è possibile con il sistema di Alberti.**

**CAPITOLO 1.5 CIRFRARIO PERFETTO ONE-TIME-PAD**

**Il cifrario di Vernam è un sistema crittografico basato sul cifrario di Vigenère al quale aggiunge il requisito che la chiave di cifratura sia lunga quanto il testo e non riutilizzabile(per questo viene spesso chiamato OTP acronimo One Time Pad letteralmente “taccuino monouso”).**

**Il cifrario di Vernam è l’unico sistema crittografico la cui sicurezza sia comprovata da una dimostrazione matematica e per questo si è guadagnato il titolo di “cifrario perfetto”. La prima dimostrazione della sua inviolabilità fu pubblicata nel 1949 da Claude Shannon. Si può facilmente capire quanto sia scomodo distribuire in modo sicuro chiavi di tali dimensioni. Ciò nonostante il cifrario di Vernam è stato utilizzato per le comunicazioni con le spie, che venivano equipaggiate di taccuini (pad in inglese) contenenti una lunga chiave per ogni pagina, da poter strappare e gettare una volta utilizzata(appunto deriva one time “un solo uso”).**

**La sua forma più classica è quella in cui la chiave ha la stessa forma del testo(a ogni lettera viene associato il numero corrispondente A=0 B=1 C=2…) e che sfrutta l’aritmetica modulare per operare sulle lettere(quella per cui dopo la lettera Z c’è di nuovo la lettera A, quindi A+C=0+2=2=C, B+C=1+2=3=D, Z+C=25+2=27→1=B, Z+Z=25+25=50→24=Y).**

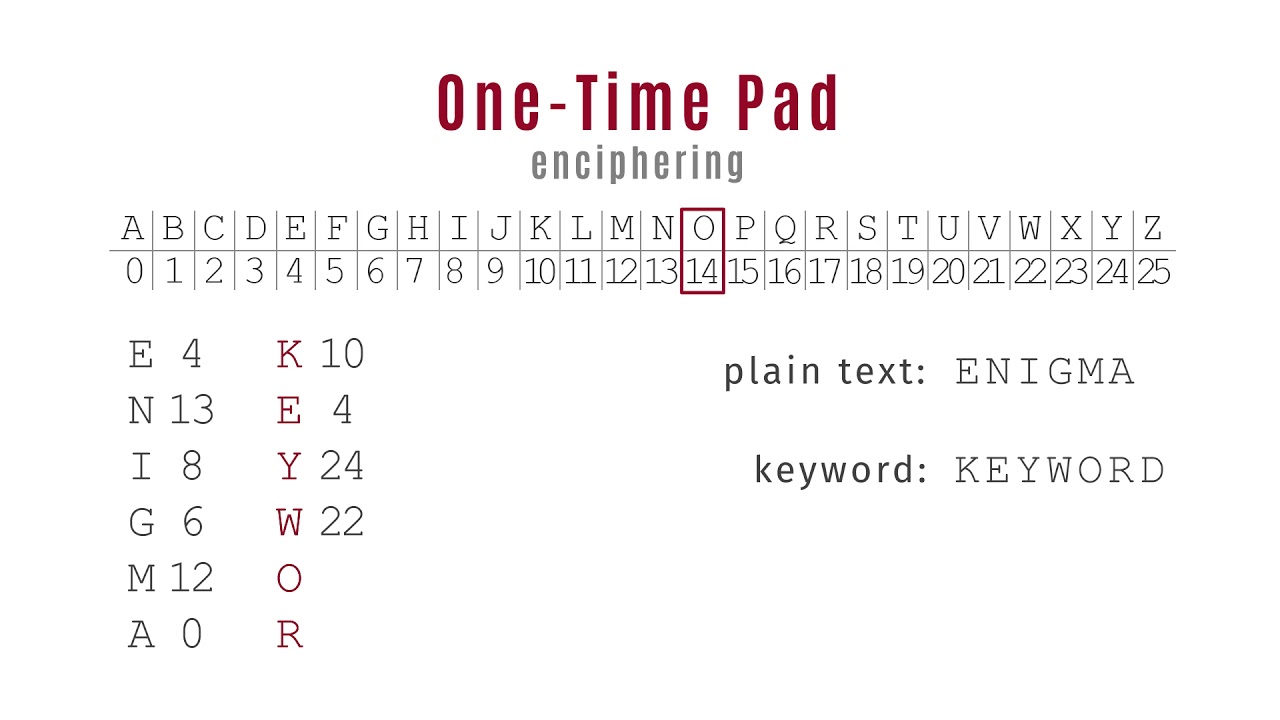
**Testo in chiaro: C I A O**

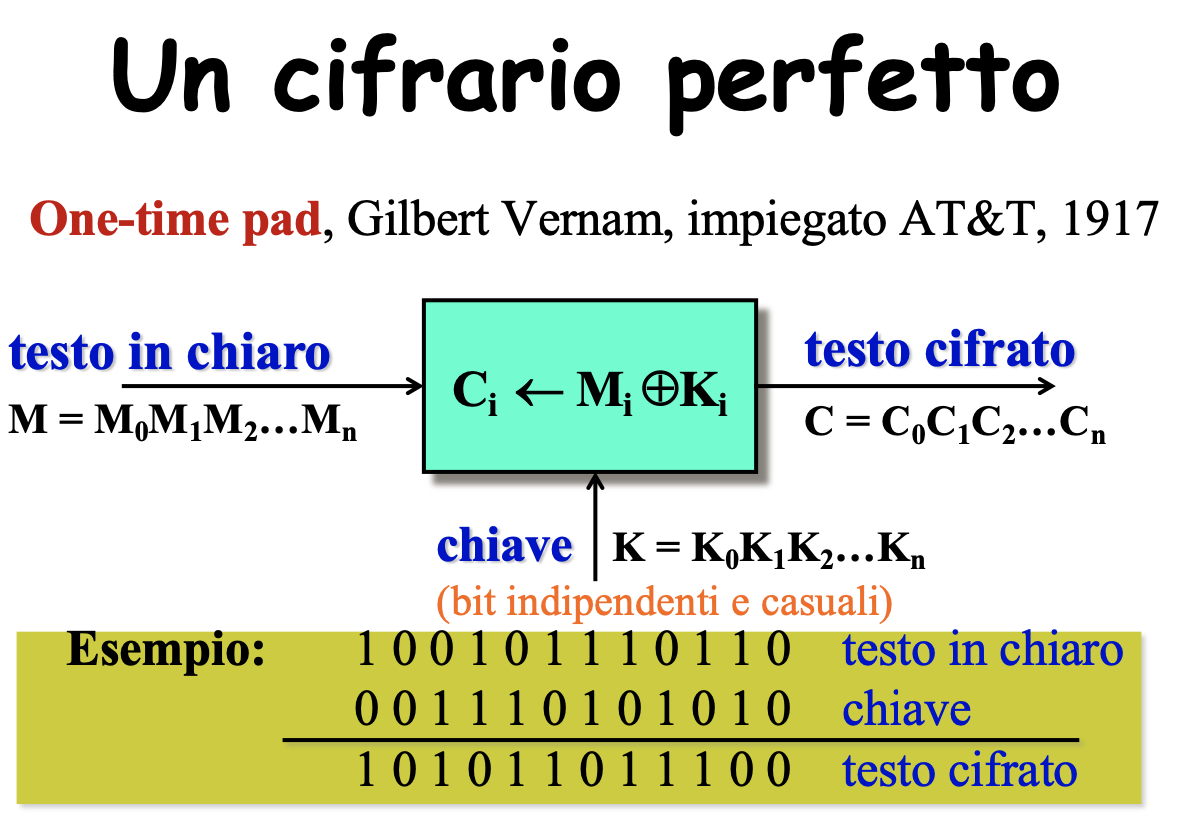
**Chiave: A J R F**

**------------------------**

**Testo cifrato: C R R T**

**E’ tuttavia molto diffusa, specialmente nell’ambito informatico, la forma che fa utilizzo dell’operazioni logica XOR(disgiunzione esclusiva), che del resto non è altro che l’addizione circolare dei singoli bit. E’ importante ribadire che questo tipo di chiave deve essere lunga quanto il messaggio che cifra e può essere utilizzata una sola volta , pena la perdita della validità delle ipotesi iniziali e la riduzione da sistema “inattaccabile” a sistema “facilmente attaccabile”.**



****

**Probabilmente l’apice del modello di cifratura a chiave infinita è stato raggiunto durante la guerra fredda. Il sistema si basava su un cifrario di parole rappresentato da numeri a 4 cifre, accorpate poi in di cinque cifre e sommate ad un certo numero casuale. Se l’ultimo gruppo di accorpamento a cinque non è completo (cioè ha meno di 5 cifre), la sequenza doveva essere completata a destra con degli zeri(esempio 123 🡺 12300).**

**Esempio 1: Cifrario 1024 = cane, 1056 = gatto , 2345 = lavoro, (spazio bianco) = 3000 , (punto) = 4000 etc…**

**Il cifrario era a disposizione sia del ricevente che del mittente, così come la sequenza dei numeri casuali utilizzati(che venivano tabulati su pagine numerate e il primo gruppo di cifre del codice cifrato trasmesso rappresentava proprio i numeri di pagine e dunque la sequenza dei numeri casuali). In tutto il processo è fondamentale come detto, utilizzare una ed una sola volta una data sequenza di numeri casuali per cifrare i messaggi.**

**Esempio 2 : Meccanismo di accorpamento da 4 a 5 cifre:**

1. **Partiamo dalla frase “canegattolavoro”**
2. **Trasformiamo la frase attraverso il cifrario:10241056234530004000**
3. **Accorpiamo le cifre in gruppi di 5 : 10241 05623 45300 04000**

**La sequenza dei numeri casuali utilizzata andrà sommata ai gruppi di 5 cifre precedentemente preparati partendo dal cifrario.**

**Esempio 3 : Ipotizziamo di avere la sequenza di numeri casuali: 45693, 89765, 77746, 93486**

**Riprendendo la sequenza dell'esempio precedente facciamo la somma dei numeri senza riporto:**

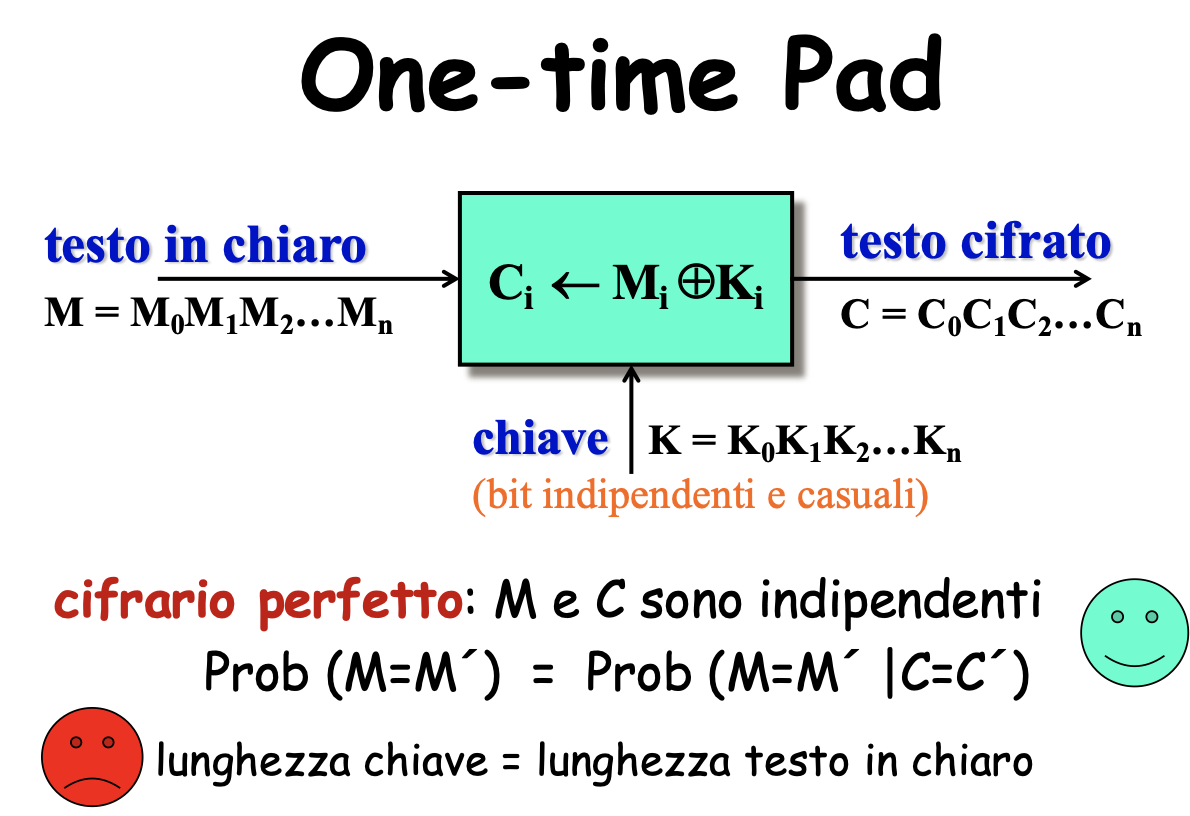
**Derivato da cifrario 10241 05623 45300 04000**

**Numeri casuali 45693 89765 77746 93486**

**Risultato (somma senza riporto) 55834 84388 12046 97486**

**A questo punto il codice è cifrato, ma i sovietici fecero un passo ulteriore con il solo scopo di facilitarne la trasmissione via**[**telefax**](https://it.wikipedia.org/wiki/Telefax)**, ovvero associarono una lettera ai numeri da 0 a 9. Dunque la forma finale del messaggio era una sequenza di gruppi di 5 lettere.**

**È facile dimostrare che questo è un**[**cifrario perfetto**](https://it.wikipedia.org/wiki/Cifrario_perfetto)**, quindi è matematicamente indecifrabile. In moltissimi anni di ricerca (all'incirca 20 –**[**progetto Venona**](https://it.wikipedia.org/wiki/Progetto_Venona)**) gli americani riuscirono a decifrare solo una minima parte dei messaggi registrati (sino all'apertura degli archivi del**[**Cremlino**](https://it.wikipedia.org/wiki/Cremlino)**) e solo perché i mittenti utilizzarono per più di una volta le sequenze di numeri casuali, o perché sequenze delle stesse furono sequestrate a spie arrestate.**

****

**CAPITOLO 1.5 CIFRARIO DI BEALE**

**Il cifrario di beale è considerato uno dei grandi enigmi crittografici ancora irrisolti. Si compone di una serie di tre messaggi lasciati nel 1822 da Thomas j. Beale ad un amico, con l’impegno di leggerli solo se non fosse tornato, che condurrebbero ad un favoloso tesoro sepolto nella Contea di Bedford , in Virginia. Il primo messaggio indicherebbe il luogo del tesoro, il secondo la descrizione ed il terzo i nomi dei compagni di Beale. I messaggi sono composti da una sequenza di numeri e si è compreso che la chiave di lettura è posta in tre libri. I numeri in questione indicano pagine e posizione delle lettere da trovare, una volta unite tutte le singole lettere si ha il messaggio completo . Finora è stato interpretato solo il secondo , a partire dalla dichiarazione di indipendenza degli stati uniti . Gli altri due sono ancora irrisolti.**

## la storia del cifrario

Immagine che contiene testo

Descrizione generata automaticamente

**Nel 1822 Beale affidò a Morris una scatola chiusa a chiave chiedendogli di custodirla , la scatola conteneva documenti cifrati. Se Beale non fosse tornato entro 10 anni, Morris avrebbe dovuto aprirla(la chiave necessaria alla decifratura sarebbe stata recapitata a Morris nel 1832), nel 1845 Morris aprì la scatola in quanto Beale non tornò mai e Morris non ricevette nessuna chiave di decifratura.**

**All’interno c’erano tre crittogrammi e una lettera che svelò che Beale aveva scoperto un giacimento d’oro.**

**Per risolvere gli altri due**[**crittogrammi**](https://it.wikipedia.org/wiki/Crittografia)**si è provato in tutte le maniere: testi giuridici, la**[**Bibbia**](https://it.wikipedia.org/wiki/Bibbia)**, libri in lingua straniera,**[**Shakespeare**](https://it.wikipedia.org/wiki/Shakespeare)**ma tutto questo non portò a nulla. Nel**[**1964**](https://it.wikipedia.org/wiki/1964)**il dottor Carl Hammer di**[**Washington**](https://it.wikipedia.org/wiki/Washington_(distretto_di_Columbia))**programmò elaborati testi statistici per visualizzare le caratteristiche e le singolarità del crittogramma. Analizzò la distribuzione dei numeri, le somme e schemi matematici. Tutto ciò confermò solo che il documento numero uno è stato cifrato con lo stesso sistema del secondo. Ma Hammer non ha risolto il crittogramma.**

**Stando al preambolo, se qualcuno un giorno riuscirà a risolverli diventerà molto ricco, sempre che il tesoro sia ancora al suo posto. Morris tentò per 30 anni di decifrare i crittogrammi , senza successo.**

**Immagine che contiene testo

Descrizione generata automaticamente**

**CAPITOLO 1.6 MACCHINE A ROTORI**

**La macchina a rotori di Hebern fu un dispositivo crittografico elettro-meccanico inventata nel 1917 dallo statunitense Edward Hugh Hebern e da lui brevettata nel 1918. La macchina combinava parti meccaniche ed elettriche di diverse macchine da scrivere dell’epoca, unite da un sistema di rimescolamento delle lettere contenute nei rotori della macchina stessa. La macchina di Hebern fu il primo esempio di macchina a rotori, una classe di dispositivi cifranti che ebbe molta diffusione durante la seconda guerra mondiale ma anche dopo, e che includeva macchine famose quali la più famosa al mondo la tedesca ENIGMA.**

**Immagine che contiene interni, sedendo, tavolo, computer

Descrizione generata automaticamente**

**Enigma fu un dispositivo elettromeccanico per cifrare e decifrare messaggi. Nata da un tentativo di commercializzare poi fallito, fu ampiamente utilizzata dal servizio delle forze armate tedesche durante il periodo nazista e della seconda guerra mondiale. La facilità d’uso e la presunta indecifrabilità furono le maggiori ragioni , ENIGMA cambiava automaticamente ad ogni lettera cifrata.**

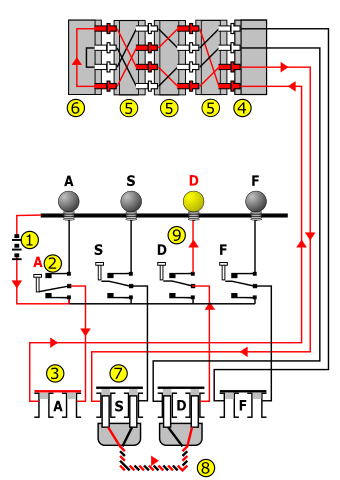
**La struttura di ENIGMA aveva l’aspetto di una macchina per scrivere con due tastiere: la prima inferiore, e la seconda nella quale i tasti erano sostituiti da lettere luminose che si accendevano ogni qualvolta venisse premuto un tasto sulla tastiera effettiva; la sequenza delle lettere che si illuminavano dava il messaggio cifrato. Nella versione a tre dischi, il suo funzionamento si basava su tre dischi cablati, detti “rotori”, che avevano 26 contatti per lato (uno per ogni lettera dell’alfabeto tedesco). I cablaggi interni dei dischi mettevano in comunicazione stabile ciascuna lettera su un lato con una lettera dell’altro lato. Detti “lato L” e “lato R” i due gruppi di contatti di ogni disco, uno dei contatti del “lato L” del primo disco riceveva la tensione del deviatore del tasto premuto, la trasferiva a un contatto, predeterminato dal cablaggio, del suo “lato R” , il quale “toccava” il corrispondente contatto del secondo disco sul “lato L” del medesimo, e il cablaggio del secondo disco trasferiva la tensione su un contatto predeterminato del suo “lato R” e così al terzo disco. I dischi erano collegati fisicamente da un meccanismo simile a un odometro.**

**Immagine che contiene fotografia, computer, diverso, tavolo

Descrizione generata automaticamenteImmagine che contiene schermo, torta

Descrizione generata automaticamente**

**Oltre a questo Enigma poteva essere regolata, per maggior sicurezza, con gli spinotti di un pannello a più prese per scambiare fra loro dieci lettere con altre dieci a scelta prima dell'ingresso nel primo rotore; infine i contatti di ogni rotore da una faccia all'altra potevano venire sfalsati a piacere.**

****

**CAPITOLO 1.7 CIFRARIO DI PLAYFAIR**

**Il cifrario Playfair o quadrato di Playfair è una tecnica di cifratura simmetrica manuale basata su un cifrario monoalfabetico a due lettere. Lo schema fu inventato nel 1854 da fisico inglese sir Charles Wheastone ma prevede il nome del suo amico Lord Playfair Barone di st. Andrews che cercò di divulgarne l’uso. La tecnica cifra coppie di lettere(diagrafi), invece che una singola lettera come nel semplice cifrario a sostituzione di Vigenerè allora in uso. Playfair è quindi significativamente più difficile da forzare poiché l’analisi della frequenza usata per i semplici cifrari a sostituzione non funzionano con esso. L’analisi delle frequenze può essere ancora intrapreso , ma sono possibili 600 digrafi invece di 26 monografi.**

## Funzionamento del Cifrario PlayFair

**Il cifrario si basa sull’uso di una matrice 5x5 contenente una parola chiave. La memorizzazione della chiave a 4 semplici regole sono tutto ciò che è richiesto per creare la tabella 5 per 5 e usare il codice. La tabella è costruita introducendo le lettere della parole chiave(eliminando le lettere duplicate) e poi riempendo gli spazi rimanenti con le lettere non utilizzate dell’alfabeto, in ordine. Essendo 26 le lettere dell’alfabeto inglese e 25 gli spazi nella matrice, occorre escludere una lettera: generalmente viene esclusa la “Q”, ma alcune versione mettendo la “I” e la “J” nello stesso spazio mentre altre escludono “W” , che se necessario può essere cifrata con una doppia “V”. La chiave può essere scritta a partire dalla prima riga della tabella, da sinistra a destra, o con un’altro percorso, un esempio a spirale iniziando e finendo al centro.**

**Per cifrare un messaggio, si deve dividere il messaggio in digrafi(gruppi di 2 lettere) in modo che , un esempio “Messaggio Segreto” diventi “ME SS AG GI OS EG RE TO”. Le lettere di un digrafo individuano un rettangolo nella tabella che ha per vertici opposti le due lettere. Le regole da applicare per ogni coppia di lettere del testo in chiaro sono 4:**

* **Se entrambe le lettere sono le stesse nel digrafo (o se la lettera è da sola), si aggiunga un "X" dopo la prima lettera. Cifrare la nuova coppia di lettere e continuare. Alcune varianti usano la "Q" al posto della "X", ma ogni lettera poco comune andrebbe bene.**
* **Se le lettere appaiono nella stessa riga della tabella, vengono codificate con le lettere alla propria destra (considerando la tabella ciclica).**
* **Se le lettere appaiono nella stessa colonna della tabella, vengono codificate con le lettere immediatamente sotto (considerando la tabella ciclica).**
* **Se le lettere non sono nella stessa riga o colonna, si codificano con le lettere nelle stesse righe rispettivamente ma negli angoli opposti del rettangolo definito dalla coppia originale. L'ordine è importante, la prima lettera della coppia codificata è quella che appartiene alla stessa riga della prima lettera nel messaggio in chiaro.**

**Per decifrare, si usano le inverse di queste quattro regole e eliminando ogni "X" (o "Q") non necessaria nel messaggio finale.**

**CAPITOLO 1.8 CIFRARIO DI Hill**

**Nella crittografia classica, il cifrario di Hill è un cifrario a sostituzione polialfabetica basato sull’algebra lineare. Ideato da Lester S. Hill nel 1929, è stato il primo cifrario polialfabetico in cui era possibile nella pratica(anche se con difficoltà) operare con più di 3 simboli alla volta. La seguente discussione presuppone una conoscenza basilare della teoria delle matrici.**

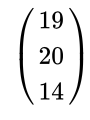
## Funzionamento della Cifratura

**Nella cifratura, ogni lettera è per prima cosa codificata in numero. Lo schema usato più di frequente è semplicemente : A=0, B=1,….Z= 25, ma questa non è una caratteristica essenziale del cifrario. Un blocco *n* di lettere è quindi considerato come uno**[**spazio vettoriale**](https://it.wikipedia.org/wiki/Spazio_vettoriale)**di**[**dimensione**](https://it.wikipedia.org/wiki/Dimensione_(spazio_vettoriale))***n*, e moltiplicato per una**[**matrice**](https://it.wikipedia.org/wiki/Matrice)***n* x *n*, modulo 26 (se si usa un numero più alto di 26 nel modulo base, allora si può usare uno schema numerale diverso per codificare le lettere, ed è possibile anche utilizzare spazi e punteggiatura). L'intera matrice è considerata la chiave del cifrario e deve essere casuale, a patto che sia invertibile in Z26n**

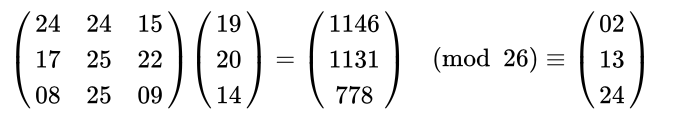
**Z26n (per assicurare che la decrittazione sia possibile). Considerando il messaggio 'TUO' e la seguente chiave (che in lettere sarebbe Y Y P R Z W I Z J):**

****

**Considerando che ‘T' è ‘19', ‘U' è ‘20' e ‘O' è ‘14', il messaggio è il vettore:**

****

**Conseguentemente il vettore cifrato è dato da:**

****

**che corrisponde al testo crittato ‘CNY'.**

**Il calcolo è dato dalla moltiplicazione matrice × testo in chiaro. Dopo aver effettuato questo calcolo, al risultato si toglie il modulo (in questo caso 26) tante volte quante ne servono per avere come differenza un numero inferiore del modulo stesso e che dia, quindi, in risultato una lettera ben definita.**

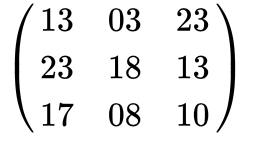
**C 🡪 02 = 24 • 19 + 24• 20 + 15 • 14 (mod 26)**

**N🡪 13 = 17 • 19 + 25 • 20 + 22 • 14 (mod 26)**

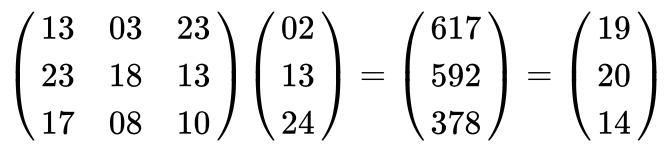
**Y 🡪 24 = 08 • 19 + 24 • 20 + 09 • 14 (mod 26)**

## Funzionamento della Decifratura

**Per decifrare riscriviamo il testo codificato come vettore, che poi moltiplicheremo semplicemente per la matrice inversa della matrice chiave (IFKVIVVMI in lettere). Ci sono metodi standard per calcolare la matrice inversa; si veda**[**matrice invertibile**](https://it.wikipedia.org/wiki/Matrice_invertibile)**per i dettagli. Si trova che la matrice inversa in Z26n  di quella dell’esempio precedente è :**

****

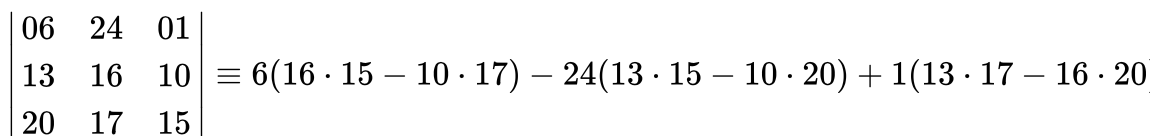
**E quindi :**

****

**che ci fa tornare a 'TUO', come si voleva.**

**C'è ancora da discutere una complicazione presente nella scelta della matrice di codifica: non tutte le matrici sono**[**invertibili**](https://it.wikipedia.org/wiki/Matrice_invertibile)**. Una matrice ha un inverso se e solo se il suo**[**determinante**](https://it.wikipedia.org/wiki/Determinante_(algebra))**non è zero e non ha fattori in comune con il modulo base. Perciò, se si lavora modulo 26 come sopra, il determinante non deve essere zero e non deve essere divisibile per 2 o 13. Se il determinante è zero, o ha fattori in comune con il modulo base, allora la matrice non può essere usata nel cifrario di Hill e deve essere scelta un'altra matrice (altrimenti non sarebbe possibile decrittare). Fortunatamente, matrici che soddisfano le condizioni per essere usate nel cifrario di Hill sono piuttosto comuni.**

**Riferendosi all'esempio fatto, il determinante è:**

****

**Così, modulo 26, il determinante è 25. Poiché questo non ha fattori comuni con 26, la matrice può essere usata nel cifrario di Hill. Il rischio del determinante di avere fattori comuni con i moduli può essere eliminato utilizzando come modulo un**[**numero primo**](https://it.wikipedia.org/wiki/Numero_primo)**. Di conseguenza un'utile variante del cifrario aggiunge tre simboli ulteriori (ad esempio spazio, punto e punto di domanda) per portare il modulo a 29.**

## Funzionamento della Sicurezza

**Per quanto riguarda la sicurezza, sfortunatamente, il cifrario di Hill nella sua versione di base è vulnerabile ad un known-plaintext attack (KPA) in quanto è completamente lineare. Un avversario che intercetti n² coppie di caratteri in chiaro e cifrati può impostare un sistema lineare che può (di solito) essere risolto facilmente; può capitare che il sistema sia indeterminato, ma in questo caso basta aggiungere alcune altre coppie di caratteri in chiaro e cifrati per poterlo risolvere. Calcolare la soluzione con un algoritmo standard di algebra lineare richiede poco tempo.**

**Mentre limitarsi a moltiplicare per una matrice non costituisce un cifrario sicuro, può comunque essere un passaggio utile quando combinato con altre operazioni non lineari, perché genera**[**diffusione**](https://it.wikipedia.org/wiki/Confusione_e_diffusione)**. Ad esempio, una matrice scelta in modo opportuno può far sì che piccole differenze del vettore iniziale (testo in chiaro) diano origine a differenze molto maggiori dopo la moltiplicazione (testo cifrato). A questo scopo alcuni cifrari moderni usano al loro interno moltiplicazioni per matrici. Un esempio è dato dal passo MixColums di**[**AES**](https://it.wikipedia.org/wiki/Advanced_Encryption_Standard)**. Anche la funzione *g* in [Twofish](https://it.wikipedia.org/wiki/Twofish" \o "Twofish) è una combinazione di**[**S-box**](https://it.wikipedia.org/wiki/S-box)**non lineari e della moltiplicazione per una matrice (MDS).**

**CAPITOLO 2 CIFRARI A BLOCCHI**

**In crittografia un algoritmo di cifratura a blocchi (dall’inglese: block cipher) è un algoritmo a chiave simmetrica operante su un gruppo di bit di lunghezza finita organizzati in un blocco. A differenza degli algoritmi a flusso che cifrano un solo elemento alla volta, gli algoritmi a blocco cifrano un blocco di elementi contemporaneamente.**

## Funzionamento della Cifratura a Blocchi

**Molti algoritmi sono basati su semplici funzioni che vengono combinate e iterate più volte. Ogni iterazione viene chiamata round e normalmente gli algoritmi utilizzano più di 4 round ma meno di 64 per singolo blocco da cifrare. Molti algoritmi basati sulla rete di Feistel o più generale sulla rete a sostituzione e permutazione. Operazioni aritmetiche, operazioni logiche (specialmente XOR) S-box e varie tecniche e permutazioni sono tra i componenti più comuni di queste reti.**

****

**Un cifrario a blocchi è un sistema crittografico che suddivide il testo in blocchi di lunghezza fissa e applica la funzione di cifratura o decifratura attraverso l’uso di una chiave. Quindi dato un messaggio M si ha :**

**M = M1, M2, M3, … Mn**

**E una chiave k, viene generato un messaggio cifrato C**

**C = C1, C2, C3, … Cn**

**Nel modo seguente si ha :**

**C1 = E(M1, k)  
C2 = E(M2, k)  
…  
Cn = E(Mn, k)**

**Il primo cifrario a blocchi fu LUCIFER, ed era una implementazione del cifrario di Feistel.**

## Cifrario a Blocchi Ideale

**Idealmente, un cifrario a blocchi trasforma un blocco di testo in chiaro lungo N bit in un blocco di testo cifrato da N bit in modo che la relazione tra i due blocchi di dati sia completamente casuale. Ma, per poter essere invertibile, ovvero decifrabile, ogni blocco in ingresso deve poter essere mappato in un blocco in uscita. Questa mappatura, ovvero la corrispondenza tra un i-esimo blocco di input e un j-esimo blocco in output, è la chiave.**

**Esistono 2N blocchi possibili in input e 2N blocchi possibili in output. Una mappatura può essere rappresentata come una sequenza di blocchi cifrati che corrispondono ai blocchi in input (ordinati). Ad esempio, se si impiega N=3 si ha:**

Immagine che contiene tavolo

Descrizione generata automaticamente

**La chiave di cifratura è quindi 100 010 110 000 001 111 101 011. Ricevendo un testo cifrato seguente: 011 001 100 110, si può utilizzare la chiave di cifratura per ricostruire il testo in chiaro. Controllando nella chiave, il primo blocco 011 si trova nell'ultima posizione, quindi corrisponde a 111, il secondo blocco 001 è nella quinta posizione e corrisponde a 100, e così via.**

**Per evitare gli attacchi a forza bruta, il numero di bit di ogni blocco deve essere elevato. Il numero di possibili mappature, ovvero di chiavi, è 2n!, poiché corrisponde al numero di permutazioni possibili dei blocchi di input, che sono in totale 2n.**

**Il problema consiste nella lunghezza della chiave, che è data da N x 2N bit. Nel caso di N=16 si ha: 16 x 216 = 1048576 bit = 128 Kbyte. Con N=32 si sale a: 32 x 232 = 137438953472 bit = 17179869184 byte = 16 Gbyte.**

## Cifrario a Blocchi non Ideale

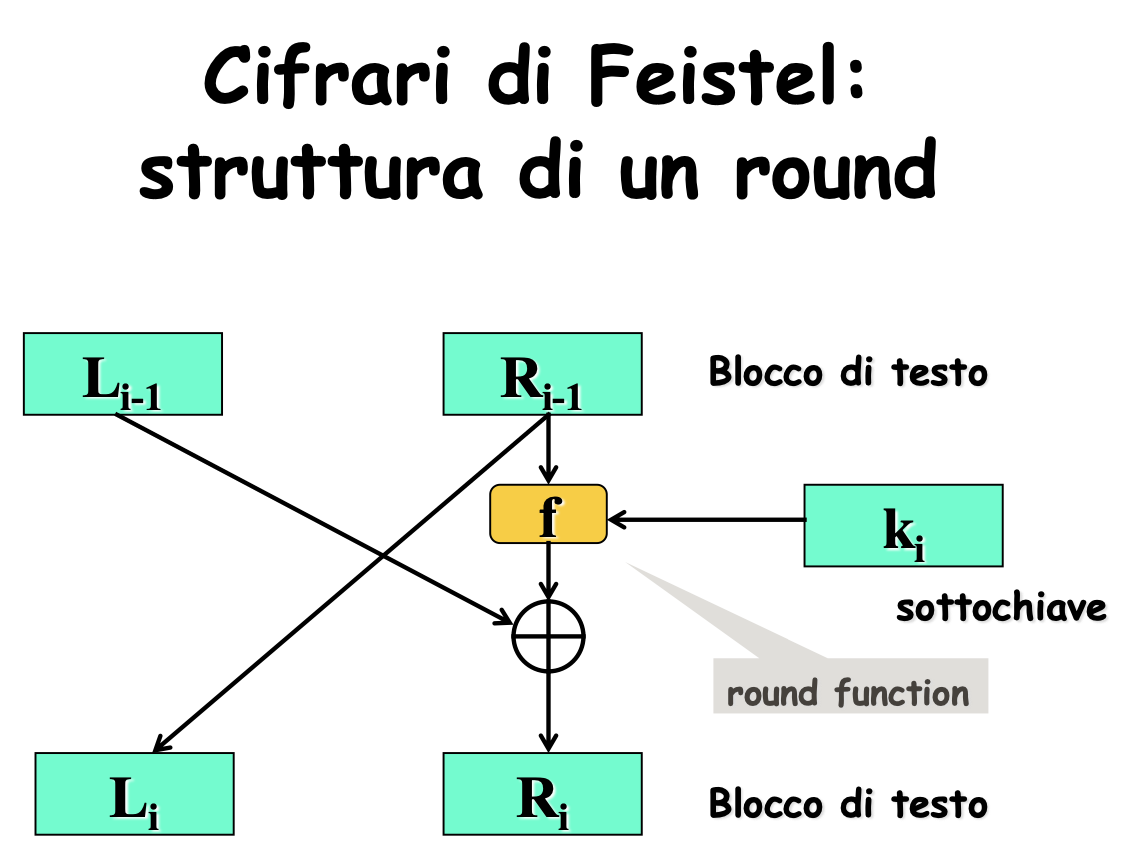
**Un cifrario a blocchi non ideale approssima un cifrario ideale. Mentre in quest'ultimo sono possibili 2n! chiavi diverse, nel cifrario non ideale si utilizza una chiave di lunghezza k, che determina la mappatura tra i blocchi di input e i blocchi di output. Il numero di chiavi possibili è 2k, e la mappatura non è completamente casuale. Un cifrario a blocchi non ideale di tipo**[**cifrario di Feistel**](http://alessiorolleri.wikidot.com/cifrario-feistel)**, implementato nel**[**DES**](http://alessiorolleri.wikidot.com/crypto-des)**e nelle**[**sue varianti**](http://alessiorolleri.wikidot.com/crypto-des-varianti)**. Un cifrario a blocchi non derivante dal cifrario di Feistel è l'algoritmo**[**AES**](http://alessiorolleri.wikidot.com/crypto-aes)**.**

**CAPITOLO 2.1 CIFRARI Di FEISTEL**

**Horst Feistel (IBM) ha introdotto uno schema di cifrario nei primi anni 70. Il DES è un cifrario di Feistel. La cifratura e la decifratura funzionano essenzialmente allo stesso modo. In un cifrario di Feistel, i blocchi hanno lunghezza m = 2w :**

* **a ogni iterazioni, ogni stato è diviso in due blocchi di lunghezza w (Li,Ri). Servono R chiavi di round (k1….kr).**

**In Particolare , il blocco diinput viene suddiviso in due parti dette L e R . La parte destra viene lasciata immutata, mentre la parte sinistra viene combinata con una operazione di xor all’output di una funzione. Questa funzione, detta funzione di Feistel , riceve in input la parte destra e un certo blocco di dati chiamato sottochiave. Dalla chiave di cifratura vengono generate tante sottochiavi quanti sono i round dell’algoritmo. Successivamente le due parti vengono scambiate, e si ripete il processo. Ogni passo del cifrario detto round. Al termine di tutti i round, le due parti formano il blocco output.**

****

## Cifratura di feistel

**Nella cifratura, sia L0 e R0 le due parti del blocco in chiaro. Al termine del primo round si hanno due blocchi di dati L1 e R1 tali che:**

**L1 = R0**

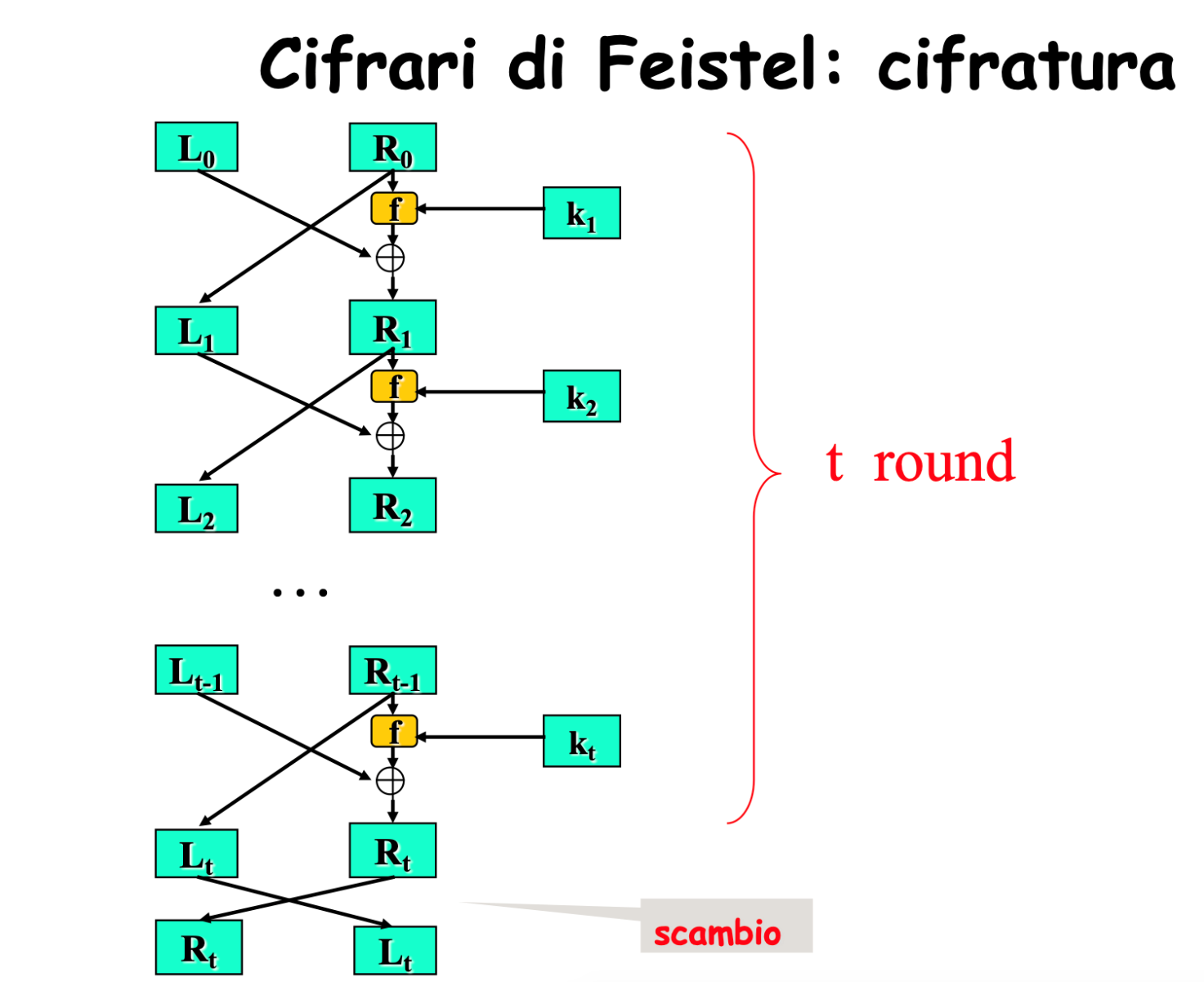
**R1 = L0⊕F(R0,k1)**

**dove k1 è la sottochiave generata per il primo round. In generale, all'i-esimo round si avranno i seguenti dati:**

**Li = Ri-1**

**Ri = Li−1⊕F(Ri−1,ki)**

**Supponendo che si effettuino 16 round, allora il testo cifrato è costituito da L16e R16.**



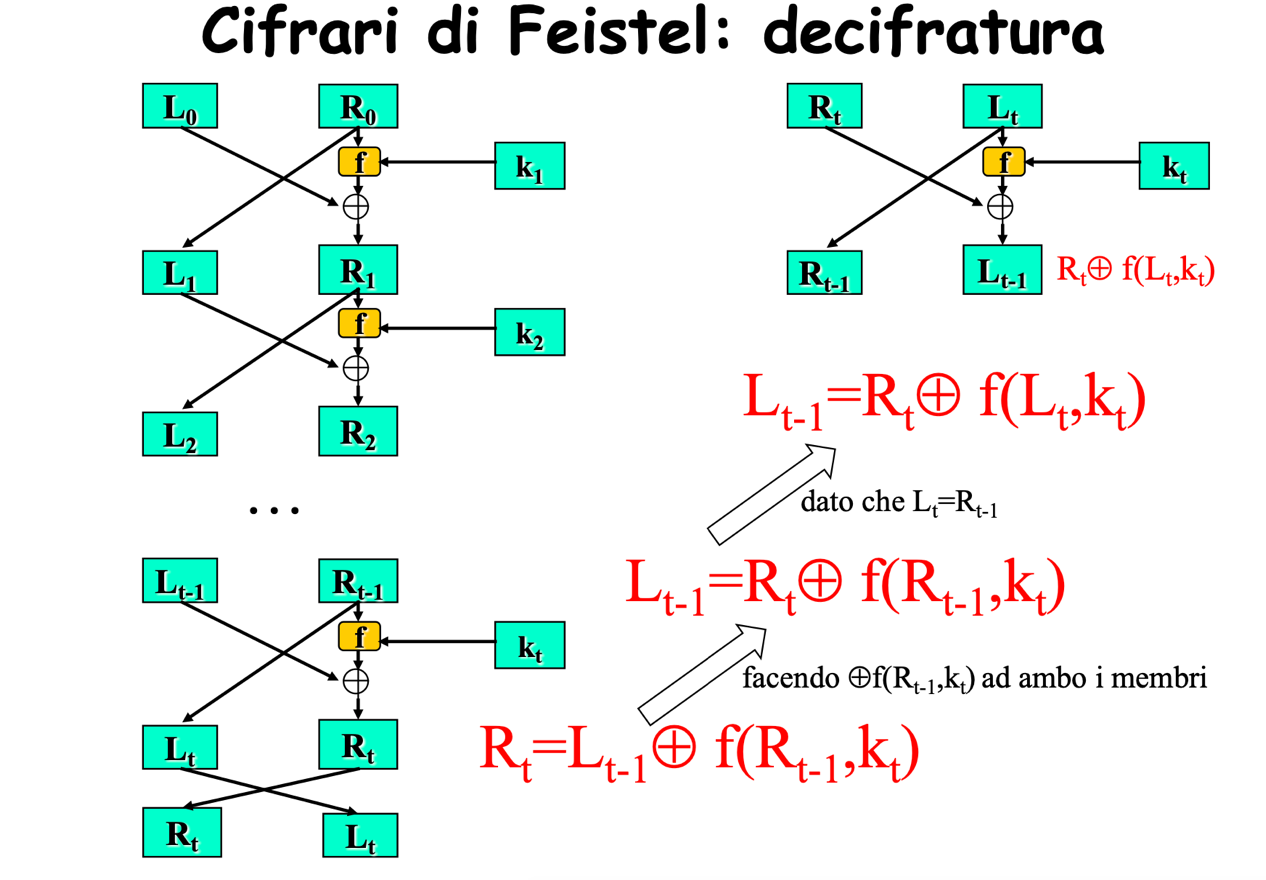
## Decifratura di Feistel

**Notiamo che la F non deve necessariamente essere iniettiva: la funzione di round è comunque invertibile.  
da come esito :**

**Li =Ri−1,**

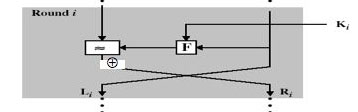
**Ri =Li−1⊕F(Ri−1,ki), si ha Li−1 =Ri ⊕F(Li,ki)**

**Ri−1 =Li.**



## Funzione di Round

**La funzione di round ha la seguente forma: input: (Li−1, Ri−1, ki ), output (Li , Ri )**

****

**dove Li = Ri−1, Ri =Li−1⊕F(Ri−1,ki).**

## Caratteristiche Principali sulla Decifratura di Feistel

* **Dimensione del blocco L’ aumento della grandezza del blocco aumenta la sicurezza ma rallenta la velocità di cifratura e decifratura**
* **Dimensione della chiave L’ aumento della lunghezza della chiave aumenta la sicurezza e rende la ricerca esaustiva più difficile ma rallenta l’algortimo**
* **Numero di round L’aumento del numero di round aumenta la sicurezza ma rallenta l’algoritmo**
* **Funzione F L’aumento di complessità rende l’analisi più difficile ma rallenta l’algoritmo**

**CAPITOLO 2.2 Data Encryption Standard (DES)**

**La storia del Data Encryption Standard(DES) è un algoritmo di cifratura scelto come standard dal Federal Information Porcessing Standard(FIPS) e dal governo degli stati uniti d’America nel 1976 e in seguito diventato di utilizzo internazzionale. Questo algoritmo ha suscitato molte discussioni per via della sua chiave di codifica corta e per via di alcune scelte progettuali che erano classificate. Si supponeva che dietro queste scelte vi fosse la National Decurity Angecy(NSA) e l’inserimento di Backdoor.**

**Il DES è considerato insicuro per moltissime applicazioni. L a sua insicurezza deriva dalla chiave utilizzata per cifrare i messaggi che è di soli 56 bit.**

**Infatti il DES cifra blocchi di 65-bit usando chiavi lunghe su 56bit, inoltre oggi giorno ci sono macchine molto complesse in grado di cifrare tutte le possibili chiavi e decifrarle in un messaggio in meno di 24 ore. Per rendere sicuro il DES nel 1999 il NIST(National Bureau of Standards) pubblicò una nuova versione chiamata triple DES(3DES) o anche chiamata TDEA. Ma il DES è stato molto più longevo del previsto (si credeva 10-15 anni).**

## Caratteristiche Principali del DES

**Il cifrario simmetrico a blocchi diviso principalmente in :**

* **Cifrario a blocchi : il messaggio viene suddiviso in blocchi di n bit, i blocchi vengono cifrati indipendentemente l’uno dall’altro**
* **Cifrari simmetrici: Cifratura e decifrazione con la stessa chiave.**

**Principalmente le basi del DES si focalizzano di un algoritmo di cifratura secondo shannon , hanno che prevedeva due caratteristiche importanti:**

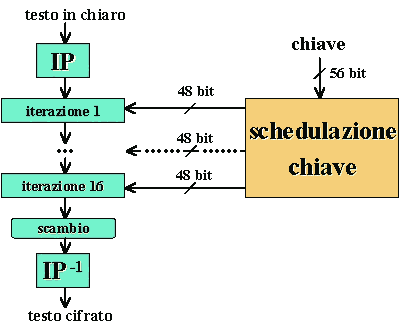
**Diffusione: Alterare la struttura del testo in chiaro “spargendo” i caratteri su tutto il testo cifrato.**

**Confusione: Combinare in modo complesso il messaggio e la chiave, per non permettere al crittoanalista di separare le due sequenze mediante l’analisi del crittogramma.**

**Nel DES diffusione e confusione sono soddisftatte attraverso una serie di permutazioni e combinazioni del messaggio con la chiave.**

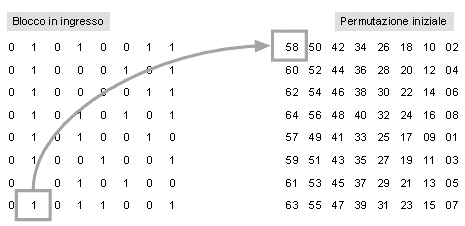
## La struttura del DES

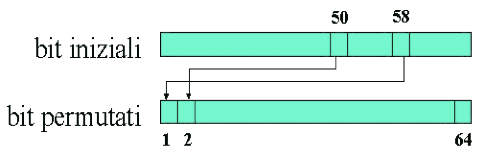
**Un blocco di testo in chiaro(64 bit) viene permutato dal blocco IP, il risultato di questa permutazione andrà in ingresso alla prima iterazione. In ogni iterazione il blocco di 64 bit viene opportunamente mescolato con una K-iesima chiave. In tutto sono 16 iterazioni, dopo le 16 iterazioni al testo di 64 bit dopo aver cifrato viene applicata le permutazioni iniziale inversa IP-1.**



## IP : Permutazione Iniziale del DES

**Nella permutazione iniziale i bit vengono scambiati in base alla matrice di permutazione IP, ovvero in posizione [1,1] andrà il bit in posizione 58 del test in chiaro, in seconda posizione il bit 50 e così via. Al termine della permutazione il blocco viene suddiviso in due sottoblocchi da 32 bit ciascuno S0 e D0, questi saranno l’input della prima iterazione. Al termine della permutazione il blocco viene suddiviso in due sotto-blocchi da 32 bit ciascuno S0 e D0, questi saranno l’input della prima iterazione.**



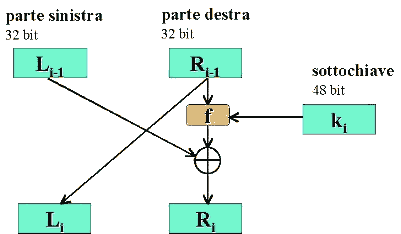
## Iterazione Iniziale del DES

**Ogni iterazione ha la seguente struttura:**

**Input : due blocchi Si-1 e Di-1 da 32 bit e una chiave Ki da 48bit**

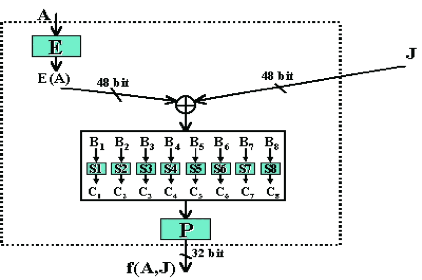
**Elaborazione: Combinazione di Si-1 e Di-1 con Ki.**

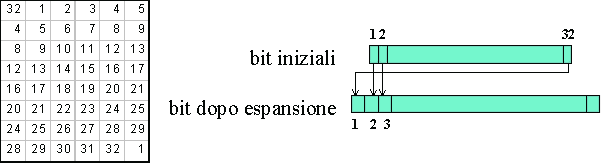
**Output: Due nuovi blocchi Si e Di da 32 bit.**

**La funzione F combina il vecchio blocco di destra RI-1 con la chiave Ki**  

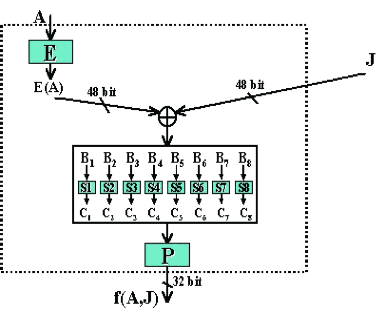
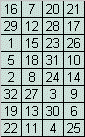
## La funzione F

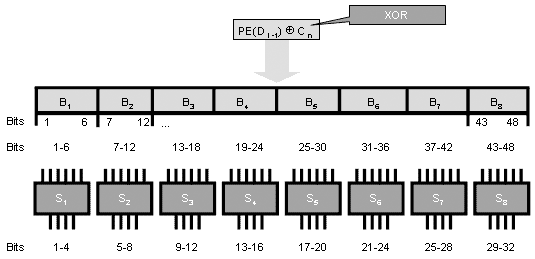
**Inizialmente i 32 bit del vecchio blocco di destra Ri-1 vengono espansi a 48 bit dal blocco E mediante la matrice E, semplicemente alcuni bit vengono ripetuti, i 48 bit ottenuti dall’espansione della chiave vengono inseriti in XOR con la Ki-esima chiave anch’essa di 48bit. La nuova sequenza B di 48 bit ottenuta viene suddivisa in 8 sottoblocchi da 6 bit ciascuno. Le stringhe Bi-esime di 6 bit vengono date in ingresso alle otto funzioni S-BOX.**





**Le S-Box hanno il compito di trasformare un blocco in ingresso Bi di 6 bit in un blocco in uscita di 4 bit. Dalla figura successiva possiamo vedere come ciò avviene: dal primo elemento e l’ultimo bit del blocco Bi in input ricaviamo l’indice di riga della SBOX, dai 4 bit centrali restanti invece ricaviamo l’indice di colonna. Otteniamo così l’elemento S[i,j] = 1 1 ; trasformandolo in binario otteniamo i 4 bit di output. Mettendo in sequenza gli output delle otto SBOX otteniamo la stringa id 32 bit, questa verrà sottoposta ad un’ultima permutazione P.**

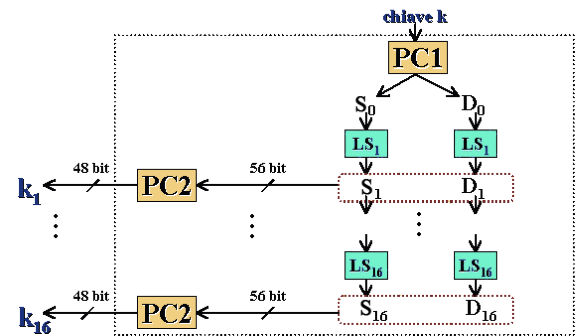
 



## Generazioni delle chiavi

**Le chiavi vengono generate da uno scheduler. Esso prende in input la chiave di 64 bit (generata in modo random automaticamente dal sistema una volta per l’intero procedimento di cifratura), lo scheduler interviene ad ogni iterazione del DES producendo una sotto-chiave diversa di 48 bit per ciascun round. Dalla chiave di partenza vengono dunque prodotte 16 sotto-chiavi diverse. Lo scheduler si compone di diverse box, attraverso le quali la chiave viene ridotta di dimensioni(in termini di bit) e rimaneggiata per produrre le diverse sotto-chiavi gli elementi fondamentali sono :**

* **PC1(Permutate Choise 1 )**
* **LSi (Left Shift i-esimo)**
* **PC2(Permutate Choise 2)**

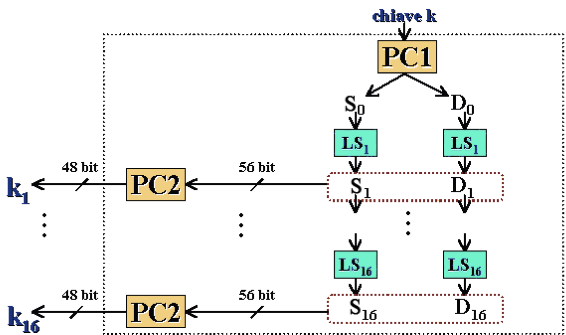


## Spiegazione PC1

**Accetta in ingresso la chiave di 64 bit assegnata al processo di cifratura. Tale chiave contiene 8 bit per il controllo di parità (uno per ogni byte). Non contenendo informazione i bit di parità possono essere eliminati, portando la chiave da 64 a 56 bit. In una rappresentazione matriciale della chiave (matrice 8x8) i bit da scartare sono quindi 8,16,24,32,40,48,56,64. La matrice uscente è quindi 8x7.**

**Esegue una permutazione di bit secondo lo schema in figura. Il bit 57 ad esempio viene mappato nel bit 1 della nuova matrice, il bit 49 nel bit 2 etc..**

**La matrice così ottenuta viene suddivisa in due (per righe), andando a produrre due sotto-chiavi di 28 bit ciascuna, etichettate come *S0* e *D0* (sotto-chiave Sinistra e sotto-chiave Destra)**

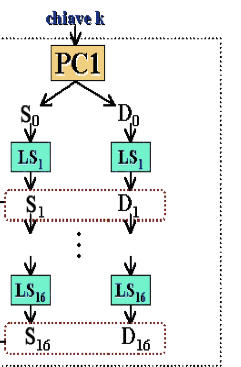
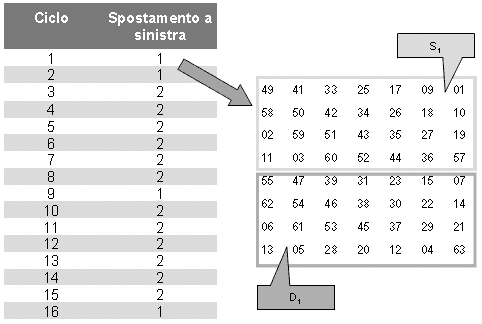


## Spiegazione ls

**Ad ogni iterazione le due semichiavi *Si* e *Di* di 28 bit vengono elaborate dal blocco *LSi* in parallelo. *LSi* effettua uno shift verso sinistra di tutti i bit delle semichiavi ad ogni iterazione in accordo con la figura successiva. Come si può notare lo spostamento è di una posizione nelle itarazioni 1-2-9-16, di due posizioni in tutte le altre**

**se sommiamo i numero totale di spostamenti otteniamo un numero di bit pari a 28. Ciò significa che lo shift è ciclico, ovvero alla sedicesima itarazione il primo bit della matrice di partenza è tornato in posizione 1. Questo dettaglio ci sarà utile nella fase di decifrazione.**

***LSi* produce in output due nuove semichiavi *Si+1* e *Di+1*.**

## Spiegazione PC2

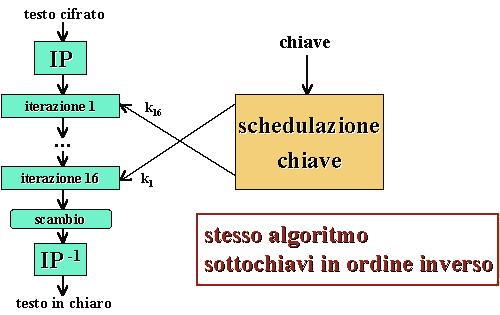
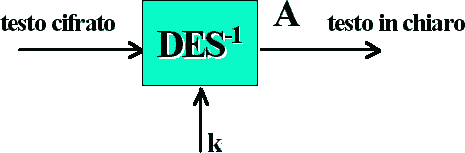
**PC2 opera ad ogni iterazione per produrre in output una chiave da 48 bit. Prende in ingresso le due sotto-chiavi Si e Di di 28bit ciascuna e le ricompone in una da 56bit. Effettua una seconda scelta permutata che rimescola e comprime la chiave in input(vengono eliminati i bit in posizione 9,18 ,22, 25, 35,43, e 54) per ottenere in output una chiave da 48 bit. La chiave ki così ottenuta va quindi in ingresso alla funzione f dell’i-esimo round del DES.**

**Immagine che contiene testo, orologio, segnale

Descrizione generata automaticamente**

## Decifratura del DES

**La decifratura del DES è esattamente come la cifratura ma con le schedulazioni delle chiavi inverse stesso algoritmo stesso sottochiavi in ordine inverso**



## Proprietà XOR

**La proprietà XOR è un’operazione reversibile : reciproco annullamento di due XOR identici, non comporta perdita di informazione.**

**Immagine che contiene testo

Descrizione generata automaticamente**

**Consideriamo il messaggio cifrato *R16 L16* a meno della permutazione finale IP^(1).**

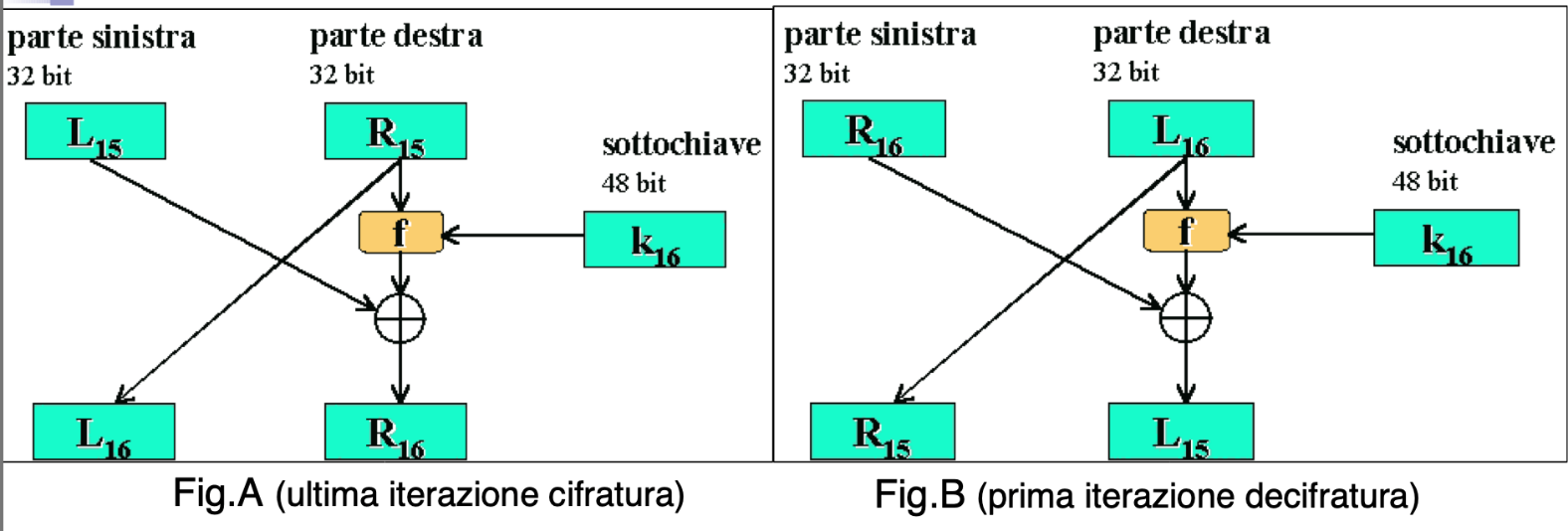
**Dalla Figura A si ricavano le seguenti relazioni**

***L16*=*R15* e *R16*=*L15*⊕*f*(*R15*,*K16*)**

**Da qui riscrivendo le due equazioni possiamo ricavare i *valori precedenti*: *R15* =*L16* e *L15*=*R16*⊕*f*(*R15*,*K16*)=*R16*⊕*f*(*L16*,*K16*)**

***L16*=*R15* e *R16*=*L15*⊕*f*(*R15*,*K16*)**

**In conclusione: da *R16* e *L16* abbiamo ricavato *R15* ed *L15* (Figura B)**

****

## Chiave Schedulazione Inversa

**Dal momento che per la fase di decifratura le chiavi devono essere schedulate in ordine inverso rispetto alla cifratura, dobbiamo effettuare delle variazioni al dispositivo di schedulazione definito in precedenza.**

**In particolare dobbiamo fare in modo che pur prendendo in input la chiave *K*, tale dispositivo produca le chiavi nell'ordine desiderato:**

* **Per ottenere ciò invertiamo lo shift che veniva applicato alle semi- chiavi *Li* e *Ri* effettuandolo verso destra anziché verso sinistra**
* **Applichiamo la tabella di *LS* in ordine inverso**
* **In questo modo al primo passo di decifrazione ottengo da *LS* la sedicesima sotto-chiave di cifratura.**

**Per ottenere la i-esima chiave non devo ricalcolare tutte le i-1 che la precedono!**

## Conclusioni finali del DES e Tipi di Attacchi

**DES non è un cifrario perfetto. Il crittogramma, in assenza della chiave, dà informazioni sul messaggio . La chiave è più corta del messaggio DES non è un cifrario ideale . La distribuzione statistica del testo in chiaro dipende dalla distribuzione statistica del crittogramma**

## Debolezze del DES

1. **Chiavi deboli**

**Generano un’unica sottochiave. La sottochiave viene usata per 16 round. Esistono 4 chiavi di questo tipo**

* **Proprietà: Ek( Ek( x ) ) = x Ovvero: se ricifriamo un testo cifrato con una weak key otteniamo il testo in chiaro**

1. **Chiavi Semi-Deboli**

**Generano 2 sottochiavi diverse ,ognuna utilizzata in 8 round. Esistono 6 coppie di chiavi di questo tipo**

* **Proprietà: Ek1( Ek2( x ) ) = x**

**Ovvero: alcune coppie di chiavi trasformano un messaggio in un unico crittogramma. Una chiave della coppia può decrittare un messaggio crittato con l’altra chiave**

1. **Chiavi particolari**

**Generano 4 sottochiavi diverse . Ognuna usata in 4 round. Esistono 48 chiavi di questo tipo ,nessuna proprietà attualmente scoperta**

* **Lunghezza: una chiave di 56 bit è troppo piccola per essere sicura**

## Tipi di Attachi del DES

**Forza bruta**

**Un attacco brute force a DES è un problema NP-completo .Il DES ha una chiave a 56 bit .Questo attacco esegue una ricerca esaustiva sullo spazio delle chiavi . Complessità temporale dell'ordine di 2^56 (2^55 in media)**

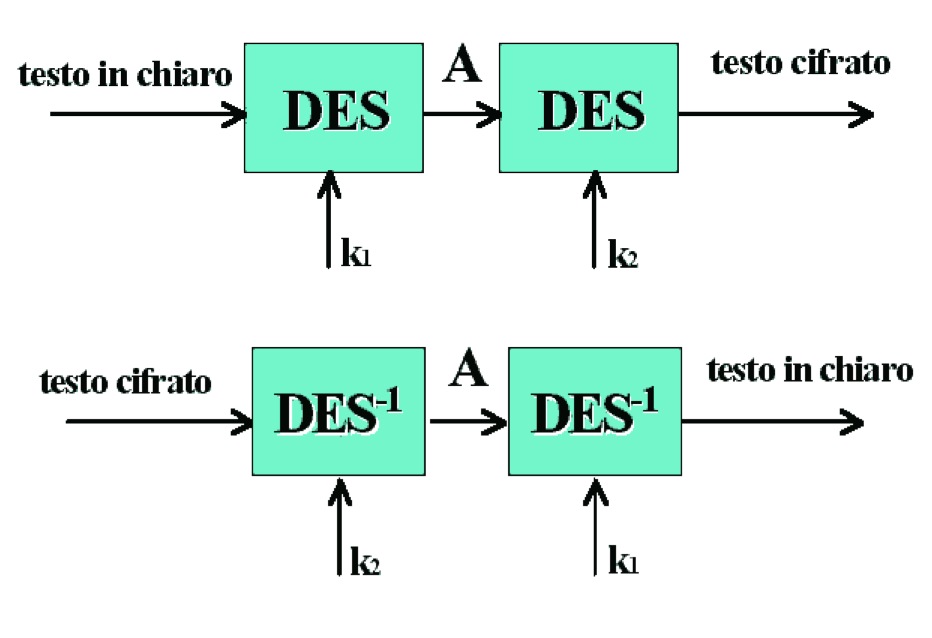
**Complessità spaziale costante**

**CAPITOLO 2.3 Data Encryption Standard (DES) Doppio(2DES)**

**Una possibile soluzione è utilizzare un DES DOPPIO. Ovvero suddividere sempre il testo da inviare in blocchi di 64 bit ciascuno e utilizzare due chiavi k1 e K2 da 56 bit ciascuno. Quindi cifrare una prima volta i 64bit del testo in chiaro utilizzando la prima chiave K1 e l’output rappresenterà l’input del nuovo modulo DES che prenderà la seconda chiave K2. Una possibile attacco a tale cifrario è del tipo Known Plaintext Cipher(conosco delle parti di testo in chiaro). Tale attacco viene definito anche Meet in the Middle. Supponendo di conoscere una coppia X,Y (X=testo cifrato, Y = testo cifrato) è possibile effettuare tutte le 256 decifrature per le k2 per Y inserisco i valori nella tabella e cerco le corrispondenze.**

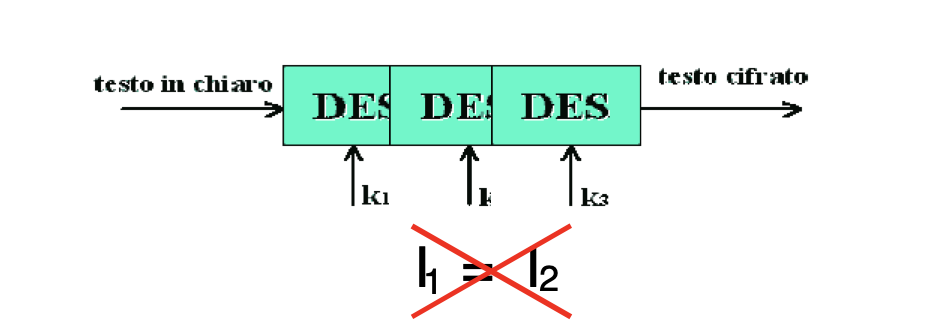
**Spazio : 256 per inserire i valori cifrati da Y**

**Tempo: 256 per cifrare i valori cifrati da X( e costruire la tabella), 256 per decifrare Y, tempo per cercare le corrispondenze che se uso, mi permette di impiegare tempo 1 altrimenti almeno 56 se è un array ordinato. Quindi con tale attacco al più impiego 256 e l’attacco a forza bruta è di 2112 quindi equivalentemente al cifrario con chiave 56 bit.**

****

**CAPITOLO 2.3 Data Encryption Standard (DES) Triplicato(3DES)**

**Viene aumentata la stessa tecnica del DES DOPPIO solo 3 volte e con 3 chiavi differenti. Anche in questo chiaro è attuabile un attacco Meet in the Middle che può essere effettuato in due modalità:**

****

**Prendiamo il testo in chiaro X e lo cifriamo prima una volta con le 226 possibili K1 e poi l’output viene cifrato con 256**  **possibili K2, ne costruiamo la tabella ed effettuiamo la decifratura del testo cifrato Y con 226 possibili K3.**

**SPAZIO 2112**

**TEMPO 2112** **per ottenere Z, 256  per decifrare y, 256 per trovare l’associazione**

**E’ quindi dimostrato che è equivalente ad un cifrario che prende una chiave** di **2112 bit e non 2168**

**Il DES TRIPLO viene utilizzato l’algoritmo DES per tre volte. La prima e l’ultima viene attuata una cifratura utilizzando una chiave K, mentre la seconda volta viene applicato un DES-1 utilizzando una chiave K’. Quindi la dimensione totale della chiave dovrebbe essere 112bit.**

**CAPITOLO 3 Advanced Encryption Standard (AES)**

**E’ uno standard ideato nel 2001 successivo al DES, progettato da Rinijidael su proposta del NIST che aveva indotto una challenge per la scrittura di un algoritmo di cifratura che non avesse le stesse problematiche del DES(problemi relativi alla chiave o progettazioni non spiegate). L’AES a differenza di molti altri algoritmi simmetrici a blocchi non utilizza la struttura di Feistel ma possiede una struttura molto più complessa**.

**A differenza del DES , AES è una rete a sostituzione e permutazione, non una rete di Feistel, che appunto applica il principio crittografico di Shannon “confusione e diffusione”. AES è veloce sia se sviluppato in software sia se sviluppato in Hardware è offre un buon livello di protezione/sicurezza, motivi che complessivamente l’hanno preferito agli altri algoritmi preposti**.

## Requisiti del AES

**Il nuovo standard di cifratura sta sostituendo i precedenti standard e la sua diffusione continua ad aumentare. Formalmente, AES non è equivalente al Rijndael (sebbene nella pratica siano intercambiabili) dato che il Rijndael gestisce differenti dimensioni di blocchi e di**[**chiavi**](https://it.wikipedia.org/wiki/Chiave_crittografica)**. Nell'AES il blocco è invece di dimensione fissa (128**[**bit**](https://it.wikipedia.org/wiki/Bit)**) e la chiave può essere di 128, 192 o 256 bit mentre il Rijndael specifica solo che il blocco e la chiave devono essere un multiplo di 32 bit con 128 bit come minimo e 256 bit come massimo.**

**AES opera utilizzando matrici di 4×4**[**byte**](https://it.wikipedia.org/wiki/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**](https://it.wikipedia.org/wiki/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.**

**C'è un passaggio iniziale:**

1. **AddRoundKey : Ogni byte della tabella viene combinato con la chiave di sessione, la chiave di sessione viene calcolata dal**[**gestore delle chiavi**](https://it.wikipedia.org/wiki/Gestore_della_chiave)**.**

**Successivamente per cifrare sono previsti diversi round o cicli di processamento: ogni round (fase) dell'AES (eccetto l'ultimo) consiste dei seguenti quattro passaggi:**

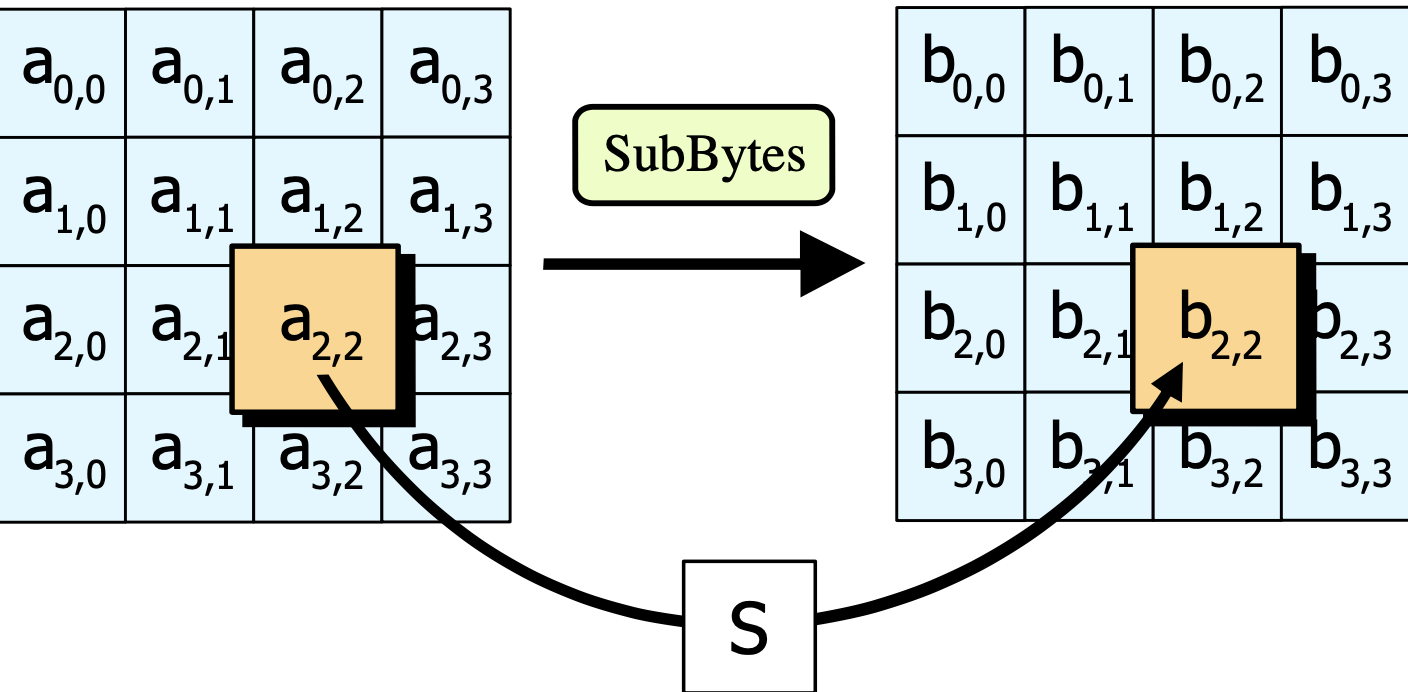
1. **SubBytes – Sostituzione non lineare di tutti i byte che vengono rimpiazzati secondo una specifica tabella.**
2. **ShiftRows – Spostamento dei byte di un certo numero di posizioni dipendente dalla riga di appartenenza.**
3. **MixColumns – Combinazione dei byte con un'operazione lineare, i byte vengono 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.**

**Il numero di round o cicli di processamento/elaborazione crittografica dei quattro passaggi precedenti è 10 con l'ultimo round che salta il passaggio MixColumns. A seguito la descrizione di ogni singolo passaggio.**

**La fase di decifratura non è identica a quella di cifratura dal momento che gli step sono eseguiti in ordine inverso. Tuttavia, si può definire un cifrario inverso equivalente ai passi dell'algoritmo usato per la cifratura, usando la funzione inversa a ogni step e un differente key schedule. Funziona siccome il risultato non cambia quando si scambiano la fase di SubBytes con quella di ShiftRows, e quella di MixColumns con una fase aggiuntiva di AddRoundKey.**

## SubBytes del AES

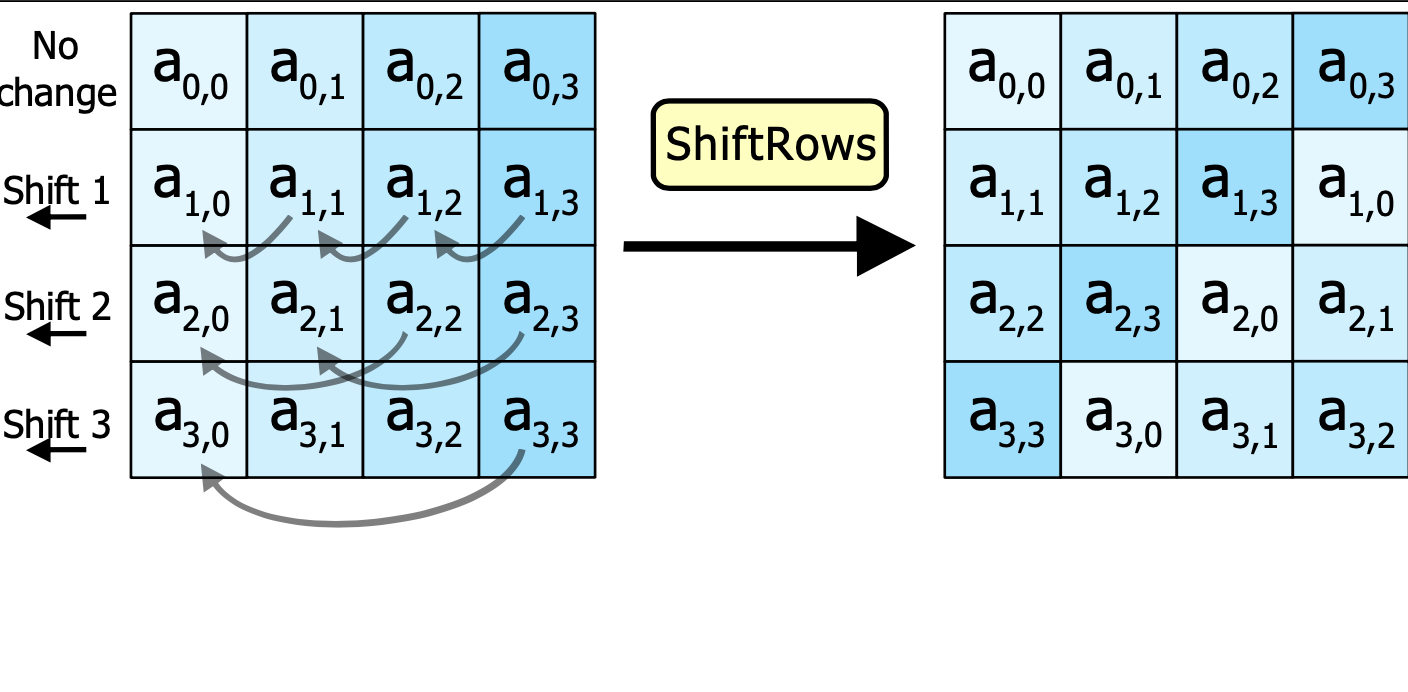
**Nel passaggio SubBytes ogni byte della matrice viene modificato tramite la**[**S-box**](https://it.wikipedia.org/wiki/S-box)**a 8 bit. Questa operazione provvede a fornire la non linearità all'algoritmo. La S-box utilizzata è derivata da una funzione inversa nel**[**campo finito**](https://it.wikipedia.org/wiki/Campo_finito)[**GF**](https://it.wikipedia.org/wiki/Campo_di_Galois)**(*28*), 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**](https://it.wikipedia.org/wiki/Trasformazione_affine)**invertibile. La S-box è stata scelta con cura per non possedere né**[**punti fissi**](https://it.wikipedia.org/wiki/Punto_fisso)**né punti fissi opposti.**

****

## ShiftRows del AES

**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***](https://it.wikipedia.org/wiki/Diagonale)**della matrice in uscita. (Rijndael utilizza un disegno leggermente diverso per via delle matrici di lunghezza non fissa.)**

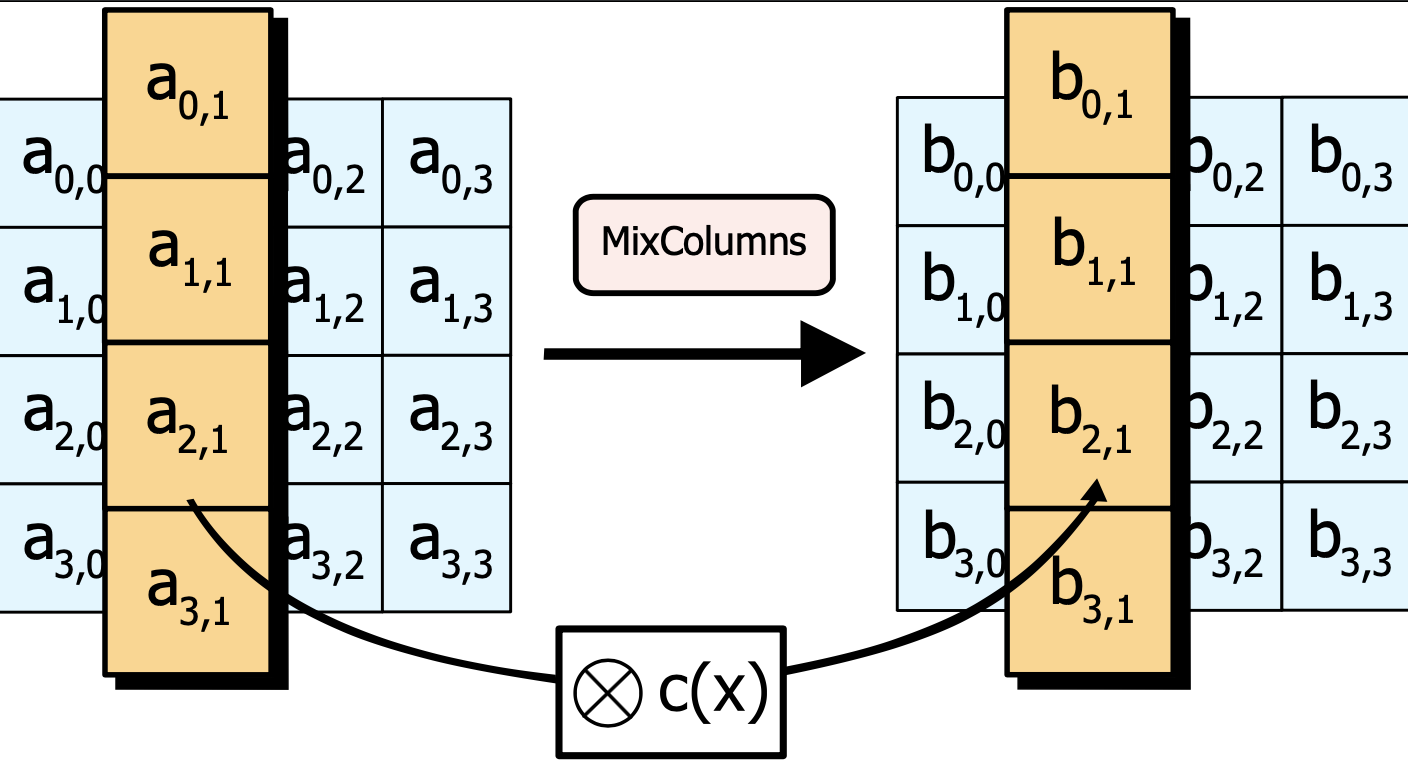
**Tutte le operazioni sono effettuate utilizzando l'indice della colonna “**[**modulo**](https://it.wikipedia.org/wiki/Aritmetica_modulare)**” il numero di colonne.**

****

## MixColumns del AES

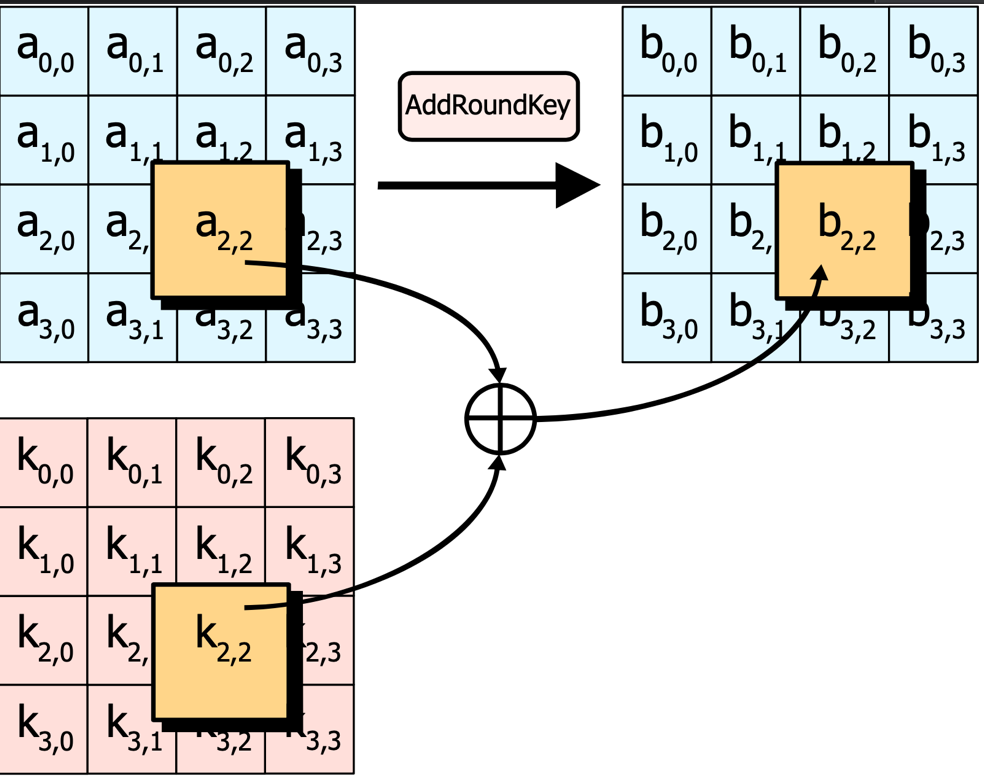
**Il passaggio MixColumns prende i quattro byte di ogni colonna e li combina utilizzando una trasformazione lineare invertibile. Utilizzati in congiunzione, ShiftRows e MixColumns provvedono a far rispettare il criterio di**[**confusione e diffusione**](https://it.wikipedia.org/wiki/Confusione_e_diffusione)**nell'algoritmo (**[**teoria di Shannon**](https://it.wikipedia.org/wiki/Confusione_e_diffusione)**). Ogni colonna è trattata come un polinomio in**[**GF(28)**](https://it.wikipedia.org/wiki/Campo_di_Galois)**e viene moltiplicata modulo x4+1x4+1 per polinomio fisso c(x)=3x3+x2+x+2.**

**C(x) = 3x 3+ x2 + x + 2.**

****

## AddRoundKey del AES

**Il passaggio AddRoundKey combina con uno**[**XOR**](https://it.wikipedia.org/wiki/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 al**[**Key Scheduler**](https://it.wikipedia.org/wiki/Rijndael_key_schedule)**.**

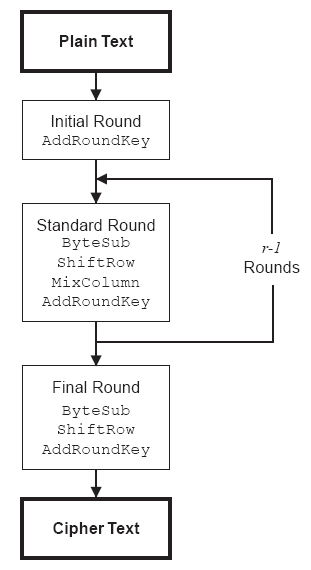
****

## Espansione della Key nel AES

**La fase dell’algoritmo indicata solitamente con “Key Schedule”, è la parte di AES che definisce l’utilizzo della chiave di cifratura.  
Delle quattro operazioni di base che compongono un round di cifratura, soltanto l’operazione AddRoundKey implica l’utilizzo della chiave di cifratura. Essa infatti effettua uno XOR bit a bit tra le colonne della matrice State e quelle di un’altra matrice indicata come “Round Key”, diversa ad ogni applicazione della AddRoundKey.**

**Le Round Keys non sono altro che una serie di matrici formate da un numero di parole di 4 byte pari al valore di Nb, ovvero al numero di colonne (e quindi di parole di 4 byte) che compongono un blocco di ciffratura (nello standard AES, che prevede solo blocchi di 128 bit, Nb è sempre uguale a 4). Le Round Keys sono ottenute dalla chiave di cifratura tramite l’operazione di “Key Expansion”. Questa operazione produce la cosiddetta “Expanded Key”, ovvero un array lineare, identificato con W, contenente tutte le Round Keys che verranno utilizzate nel corso dell’algoritmo.**

**Il numero di parole di cui è composta la Expanded Key è dovuto alla configurazione dell’algoritmo: come è stato visto in precedenza, infatti, ad ogni esecuzione dell’operazione AddRoundKey vengono “consumate” un numero di parole pari a Nb. Inoltre, come si può vedere dal diagramma di flusso dell’algoritmo di cifratura, il numero di applicazioni della AddRoundKey è pari a Nr + 1.**



**La funzione di espansione della chiave è stata progettata per resistere ad attacchi tipo:**

* **attacchi in cui il crittoanalista conosce una parte della chiave di cifratura**
* **attacchi in cui la chiave è interamente conosciuta o può essere scelta dal crittoanalista**
* **attacchi related-key, ovvero in cui il crittoanalista è in possesso di testi cifrati con più chiavi di cifratura, di cui conosce solo le differenze (ad esempio sa che due chiavi di cifratura differiscono solo in un bit).**

**In particolare una condizione necessaria per resistere ad attacchi di tipo related key è quella di non avere chiavi di cifratura differenti che producano un certo numero di round key uguali. L’espansione della chiave gioca inoltre un ruolo molto importante nell’eliminazione della simmetria introdotta dalla cifratura:**

* **simmetria all’interno dei round di trasformazione: tutti i byte dello state vengono trattati in maniera simile nel corso delle operazioni di un round**
* **simmetria tra i round: le trasformazioni applicate durante un round sono le stesse per tutti i round**

## Attacchi e Sicurezza nel AES

**S-Box che usa l’inverso spezza sia attacchi lineari, che differenziali, che di interpolazione.**

**ShiftRow serve a resistere a due attacchi denominati *differenziali troncati* e *quadrati(square)*.**

* **MixColumn serve a spargere rapidamente la chiave tra i vari bits**
* **Il numero di iterazioni(10,12,14)serve in quanto ci sono attacchi migliori della forza bruta per 6 iterazioni. 10 è un valore di tranquillità.**

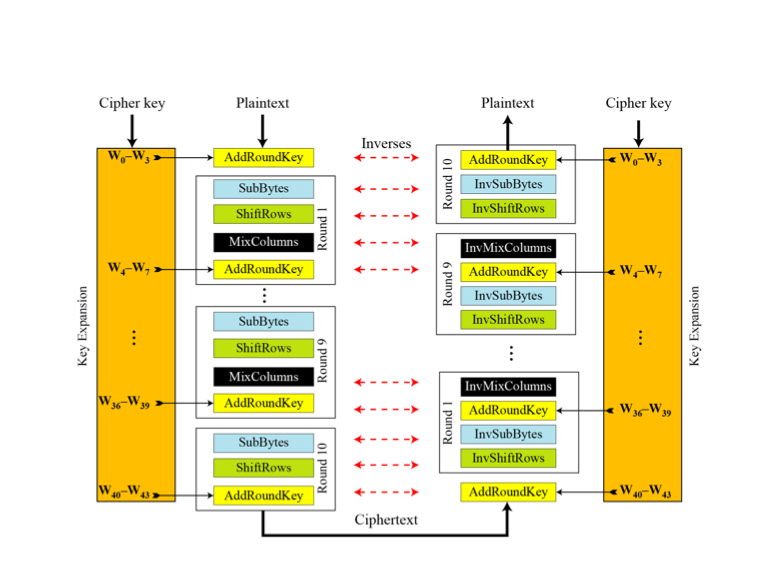
**Anche qui ci sta bene un laboratorio avanzato. Sugli attacchi effettuati e/o su un tentativo di attacco di forza bruta magari su AES con meno livelli (3/4).**

## Decifratura nel AES

**Si può lavorare sulla decifrazione in modo da renderla un pò più simile (ma non uguale) alla cifrazione. Il fatto che non siano uguali è un punto di forza. Usa una diversa sequenza di trasformazioni, precisamente sono trasformazioni inverse**

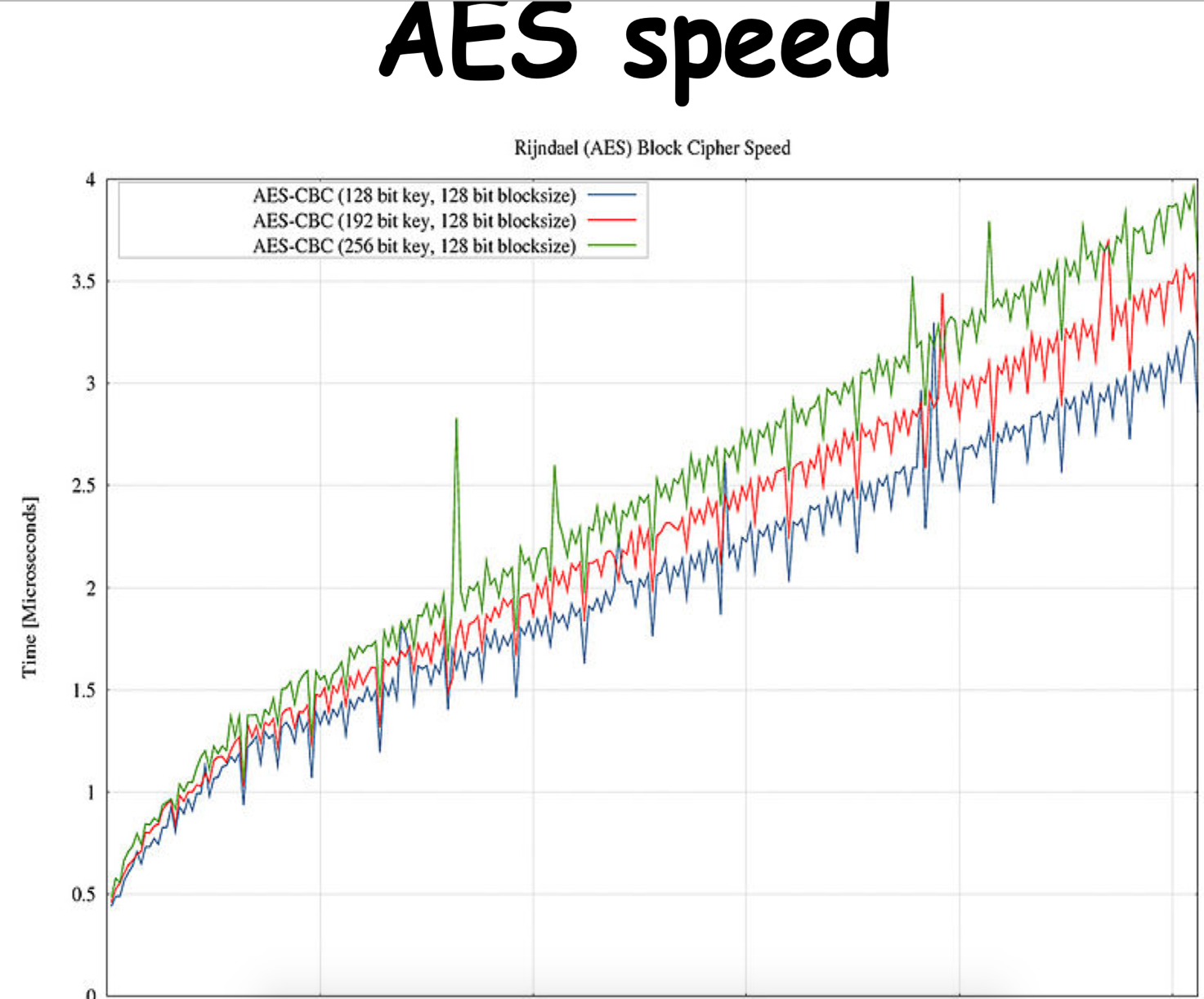
**Svantaggio sta nella necessità di una doppia implementazione.**

**Esiste anche un algoritmo di decifratura che ha la stessa struttura di quello di cifratura. Necessita di un cambiamento nella schedulazione della chiave.**



**Immagine che contiene testo

Descrizione generata automaticamente**

****

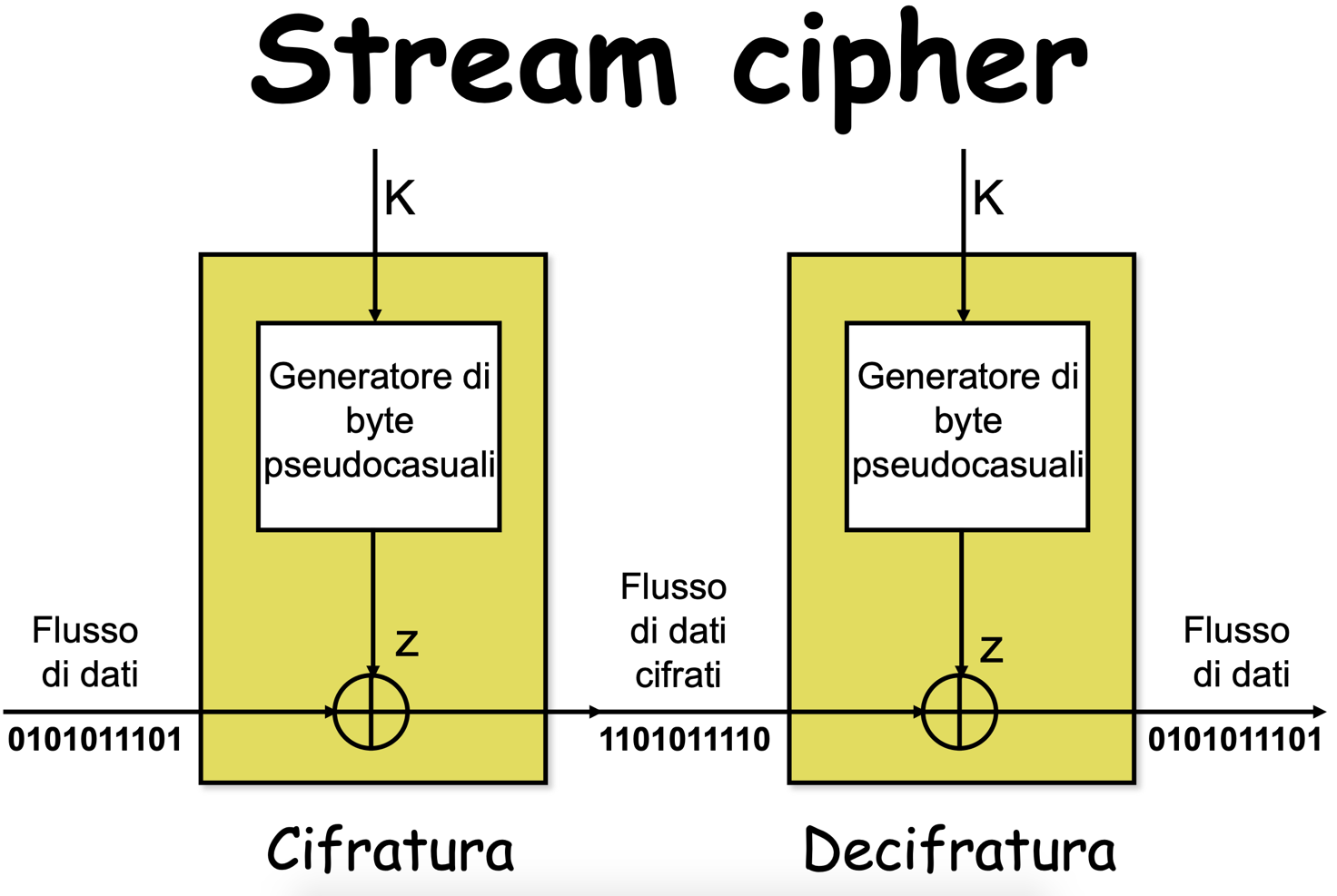
**CAPITOLO 4: Stream Ciphers**

**I cifrari possono essere divisi in due principali categorie :**

* **Cifrari a Blocchi: Cifrari in cui la trasformazioni avviene in grandi blocchi del testo in chiaro**
* **Stream Ciphers: In cui la trasformazioni, dipendente dal tempo di singoli caratteri del testo in chiaro.**

**Lo stream Ciphers(o meglio cifrario a flusso) detto anche cifrario a caratteri è un cifrario simmetrico nel quali i simboli (i bit) che codificano il testo in chiaro sono cifrati indipendente l’uno dall’altro e nel quale la trasformazione dei simboli successivi varia con il procedere della cifratura. Un altro termine usato per tale cifrario è cifrario a stati, termine che ricorda che la cifratura di ogni simbolo dipende da uno stato corrente. Tipicamente nella pratica i simboli sono singoli bit o byte.**

**I cifrari a flusso costituiscono un approccio alla cifratura simmetrica differente da quello dei cifrari a blocchi: questi prevedono che si effettui su successivi estesi blocchi di simboli una determinata trasformazione che non cambia da blocco a blocco. Questa distinzione in realtà non è sempre netta: alcune modalità operative dei cifrari a blocchi effettuano una cifratura a blocchi primitiva in un modo tale che il suo effetto corrisponde a quello di un cifrario a flusso. Tipicamente i cifrari a flusso sono eseguiti a velocità superiori a quelle dei cifrari a blocchi e si servono di hardware meno complesso. Per contro i cifrari a flusso possono essere suscettibili a seri problemi di sicurezza , se non sono utilizzati correttamente: I documenti nella National Security Agency talora usano il termine combiner-type algorithms(algoritmi di tipo combinatorio) per denotare algoritmi che si servono di qualche opportuna funzione per combinare un flusso di testo in chiaro con un generatore di numeri pseudo-casuali(pseudo-random number generator o PRNG).**

****

**Principalmente nello Stream Cipher è possibile descrivere 5 categorie principali che sono LSFR(Linear FeedBack Shift Register), A5(e GSM), RC4,Salsa20,ChaCha20.**

## LSFR (Linear FeedBack Shift Register)