

Cifratura simmetrica

1 Operazioni fondamentali

La **cifratura simmetrica**, detta anche *convenzionale*, *tradizionale*, *a chiave privata* o *a chiave singola*, sfrutta algoritmi che si basano su due semplici operazioni:

- **sostituzioni** di un simbolo con uno diverso;
- **trasposizioni** o **permutazioni** dell'ordine dei simboli.

Per analizzare le caratteristiche di queste operazioni è utile studiare la *cifratura classica*, cioè gli algoritmi che venivano impiegati, solitamente a mano, prima dell'introduzione dei calcolatori moderni. I simboli su cui tali algoritmi operavano erano in genere le lettere di un testo, mentre per gli algoritmi moderni i simboli sono le combinazioni di bit (ad esempio i byte) che costituiscono un messaggio digitale, ma i principi di base rimangono gli stessi.

In seguito, si inizieranno a presentare le principali tecniche di sostituzione.

2 Cifrario di Cesare

Il **cifrario di Cesare**, così chiamato perché fu utilizzato da Giulio Cesare per scopi militari, sostituisce ogni lettera dell'alfabeto con la lettera che si trova k posizioni più avanti nell'alfabeto.

Il modo in cui le lettere vengono sostituite per un certo valore di k può essere indicato elencando tutte le lettere dell'alfabeto del testo in chiaro e scrivendo, sotto ciascuna di esse, la corrispondente lettera del testo cifrato; per convenzione, si indicano in minuscolo le lettere del testo in chiaro e in maiuscolo quelle del testo cifrato. Ad esempio, per $k = 3$, la sostituzione è:

```
abcdefghijklmnopqrstuvwxyz  
DEFGHIJKLMNOPQRSTUVWXYZABC
```

Una notazione analoga può essere usata per indicare il testo in chiaro di un messaggio e il corrispondente testo cifrato. Ad esempio, sempre con $k = 3$:

meet me after the toga party
PHHW PH DIWHU WKH WRJD SDUWB

La chiave del cifrario di Cesare è il valore k : se un messaggio è stato cifrato “andando avanti” di k posizioni nell’alfabeto, per decifrarlo bisogna “tornare indietro” di altrettante posizioni.

Se si assegna a ogni lettera un valore numerico da 0 a 25 (dove 0 corrisponde ad a e 25 corrisponde a z), allora gli algoritmi di cifratura e decifratura possono essere formalizzati come segue: date la chiave k e una lettera p del testo in chiaro, la corrispondente lettera C del testo cifrato è data dalla formula

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

e dalla lettera cifrata C si può risalire alla lettera in chiaro p tramite la formula

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

L’uso del modulo in queste formule serve a *garantire la circolarità nel dominio* 0–25:

- se “andando avanti” nell’alfabeto si arriva alla fine, si ritorna all’inizio;
- analogamente, se “andando indietro” si arriva all’inizio, si ritorna alla fine.

Ad esempio, usando le formule con la chiave $k = 5$, il messaggio **howdy** (7 14 22 3 24) viene cifrato e decifrato così:

$E(7, 5) = (7 + 5) \bmod 26 = 12$	(M)	$D(12, 5) = (12 - 5) \bmod 26 = 7$	(h)
$E(14, 5) = (14 + 5) \bmod 26 = 19$	(T)	$D(19, 5) = (19 - 5) \bmod 26 = 14$	(o)
$E(22, 5) = (22 + 5) \bmod 26 = 1$	(B)	$D(1, 5) = (1 - 5) \bmod 26 = 22$	(w)
$E(3, 5) = (3 + 5) \bmod 26 = 8$	(I)	$D(8, 5) = (8 - 5) \bmod 26 = 3$	(d)
$E(24, 5) = (24 + 5) \bmod 26 = 3$	(D)	$D(3, 5) = (3 - 5) \bmod 26 = 24$	(y)

2.1 Attacchi a forza bruta

Il cifrario di Cesare ha solo 26 possibili chiavi ($k = 0, 1, \dots, 25$), e di queste solo 25 sono concretamente utilizzabili, perché $k = 0$ di fatto non esegue alcuna cifratura (sostituisce ciascuna lettera con se stessa). Di conseguenza, è molto facile provarle tutte, cioè eseguire un attacco a forza bruta (purché si sia in grado di riconoscere il testo in chiaro).

3 Cifratura monoalfabetica

Il cifrario di Cesare è un caso particolare di un algoritmo più generale, la **cifratura monoalfabetica**, così chiamata perché usa un unico alfabeto (mapping) per le sostituzioni (una volta fissata la chiave).

Una sostituzione è rappresentata formalmente da una funzione che associa a ogni lettera dell'alfabeto del plaintext una lettera dell'alfabeto del ciphertext. Il cifrario di Cesare considera solo 25 di queste funzioni: quelle che corrispondono a spostamenti di un numero fisso k di posizioni nell'alfabeto, con k compreso tra 1 e 25.

Nel caso generale della cifratura monoalfabetica, invece, si possono considerare sostituzioni arbitrarie. Allora, per indicare la sostituzione scelta, non è più sufficiente un numero da 1 a 25, ma piuttosto bisogna indicare come viene sostituita ciascuna lettera. Ad esempio:

```
abcdefghijklmnopqrstuvwxyz  
DKVQFIBJWPESCXHTMYAUOLRGZN
```

Data questa sostituzione, si può cifrare un messaggio:

```
if we wish to replace letters  
WI RF RWAJ UH YFTSDVF SFUUFYA
```

La chiave di questa tecnica di cifratura è rappresentata dall'alfabeto usato per la sostituzione: elencare le lettere di quest'ultimo nell'ordine delle lettere dell'alfabeto del plaintext (cioè, nel caso dell'esempio precedente, scrivere DKVQFIBJWPESCXHTMYAUOLRGZN) è sufficiente a descrivere una qualsiasi funzione di sostituzione. Allora, le possibili chiavi sono tutte le permutazioni (senza ripetizione) delle 26 lettere, cioè esistono $26! \approx 4 \cdot 10^{26}$ chiavi distinte. Questo è uno spazio delle chiavi molto più ampio rispetto a quello del cifrario di Cesare, il che risolve il problema degli attacchi a forza bruta: anche facendo un milione di tentativi ogni microsecondo, ci vorrebbero in media $6.4 \cdot 10^6$ anni per trovare la chiave giusta.

3.1 Crittoanalisi

L'ampiezza dello spazio delle chiavi potrebbe far pensare che lo schema di cifratura monoalfabetica sia del tutto sicuro, ma in realtà esso è molto vulnerabile ad attacchi di crittoanalisi, poiché *non nasconde i pattern presenti nel plaintext*.

Ad esempio, in ogni lingua ci sono lettere che compaiono più frequentemente di altre: ad esempio, in inglese, la lettera più frequente è la E, mentre alcune lettere come la X sono usate raramente. Siccome la cifratura monoalfabetica sostituisce tutte le occorrenze

di una lettera nel plaintext con la stessa lettera dell'alfabeto del ciphertext, queste frequenze vengono preservate, e possono essere confrontate con una tabella delle frequenze previste¹ per fare ipotesi su come vadano decifrate le lettere: ad esempio, nel caso di un messaggio in inglese, la lettera più frequente nel ciphertext sarà probabilmente quella che corrisponde alla E nel plaintext.

Inoltre, siccome anche l'ordine delle lettere viene mantenuto, si possono fare ragionamenti analoghi sulla frequenza dei *digrammi* (coppie di lettere), dei *trigrammi* (triple di lettere), ecc.

Nel caso di messaggi digitali, potrebbero non essere presenti pattern dovuti alle regolarità del linguaggio naturale, ma ci sarebbero invece strutture come header, ecc.

4 Sostituzione con più omofoni

Un modo per nascondere le frequenze dell'alfabeto originale è sostituire la stessa lettera con più sostituiti, chiamati **omofoni**, assegnati a rotazione o casualmente: se il numero di omofoni associati a ciascuna lettera è proporzionale alla sua frequenza, allora si cela la frequenza della singola lettera. Rimangono però possibili attacchi di crittoanalisi basati sull'analisi della frequenza di digrammi e trigrammi.

Ci sono due famiglie di soluzioni per associare più omofoni alle lettere:

- i **poligrammi** eseguono le sostituzioni su gruppi di lettere, invece che sulle singole lettere;
- la **cifratura polialfabetica** utilizza più alfabeti, cioè funzioni di sostituzione diverse.

4.1 Playfair

Un esempio di poligramma è lo schema **Playfair**, adottato dall'esercito inglese durante la prima guerra mondiale. Esso è in particolare un digramma, cioè cifra due lettere alla volta. Ogni lettera ha più omofoni, scelti in base al digramma di appartenenza.

Gli algoritmi di cifratura e decifratura usano una matrice di 5×5 lettere, contenente tutte le lettere dell'alfabeto. Siccome l'alfabeto (inglese) ha 26 lettere, mentre la matrice ha solo 25 celle, è necessario mettere insieme due lettere, tipicamente I e J: esse vengono trattate come se fossero la stessa lettera, cioè usate in modo intercambiabile.

Per costruire la matrice:

¹Se non è già disponibile, questa tabella può essere facilmente calcolata da un qualunque testo di dimensioni sufficienti, ad esempio un dizionario.

1. si inserisce la parola chiave, procedendo da sinistra a destra e dall'alto verso il basso (cioè nell'ordine di lettura di un normale testo) e omettendo le lettere duplicate;
2. si riempiono le restanti celle della matrice con tutte le altre lettere dell'alfabeto, in ordine alfabetico.

Ad esempio, con la chiave MONARCHY si costruisce la seguente matrice:

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

Poi, il testo in chiaro viene cifrato due lettere alla volta, applicando regole diverse in base a dove tali lettere si trovano nella matrice:

- Se entrambe le lettere ricadono nella *stessa riga*, si sostituisce ciascuna lettera con quella alla sua destra nella matrice (eventualmente ripartendo da sinistra, se si supera la fine di una riga). Ad esempio, il digramma *nr* viene cifrato come *AM*:

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

- Se entrambe le lettere ricadono nella *stessa colonna*, si sostituisce ciascuna lettera con quella sottostante nella matrice (eventualmente ripartendo da sopra, se si supera la fine di una colonna). Ad esempio, il digramma *mu* viene cifrato come *CM*:

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

- Se le due lettere non sono nella stessa riga né nella stessa colonna, allora costituiscono due vertici di una sottomatrice, e vengono sostituite con i vertici opposti. Più precisamente, ciascuna lettera viene sostituita con la lettera che si trova sulla sua stessa riga, ma nella colonna occupata dall'altra lettera. Ad esempio, il digramma *ea* viene cifrato come *IM* (o *JM*, visto che *I* e *J* sono considerate equivalenti):

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

- Se le due lettere sono uguali, si aggiunge al plaintext una lettera di riempimento prestabilita (ad esempio *x*) tra le due lettere, in modo da separarle in due digrammi. Ad esempio, la parola *balloon* viene suddivisa nei digrammi *ba lx lo on*.

Per la decifrazione si eseguono semplicemente le operazioni inverse.