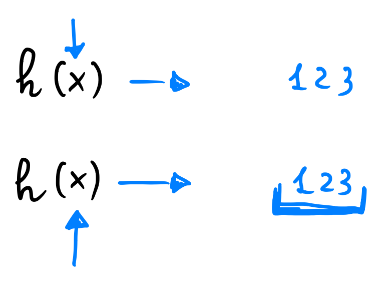
Crypto

HASH FUNCTIONS

Sono funzioni che prendono in input un valore di lunghezza indeterminata e restituiscono in output un valore di lunghezza fissa (DIGEST).

Le funzioni di Hash hanno diverse applicazioni
esempio: salvataggi di password nel DB
Le funzioni Hash non sono reversibili.
Non è possibile risalire al valore di input dato l'output di una funzione hash.

1. Per due input identici la funzione genera sempre lo stesso output
2. Non è garantito che a parità di output, gli input siano gli stessi.



Due valori di input diversi possono dare lo stesso output.
Proprietà:
quando usiamo una funzione hash vogliamo che sia particolarmente difficile:

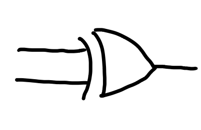
1. Modificare l'input senza che cambi l'hash
2. generare un messaggio che ha un hash specifico
3. trovare due messaggi diversi che producono lo stesso hash

Dalla ① l'ideale è che cambiando anche un singolo bit di input l'output cambi almeno del 50%.
Dalla ② si ricava che l'hash è una funzione unidirezionale.
Dalla ③ si deduce che è molto difficile trovare due messaggi m e m' che generano lo stesso hash.

FUNZIONI PIÙ UTILIZZATE: MD5 SHA256 SHA512

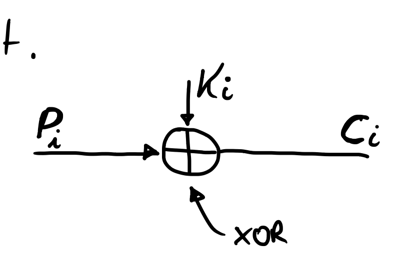
CIFRATURA

L'operazione XOR restituisce vero se i due bit in ingresso sono diversi.



in1	in2	out
0	0	0
0	1	1
1	0	1
1	1	0

Un altro modo per vedere la XOR è in modalità "invertitore programmabile" in cui un bit di input decide se invertire l'altro bit.



In questo schema
P: Plaintext testo in chiaro
K: Key chiave
C: Ciphertext testo cifrato
i: index perché abbiamo a che fare con più di un bit alla volta

In matematica si usa il simbolo ⊕ per indicare la XOR

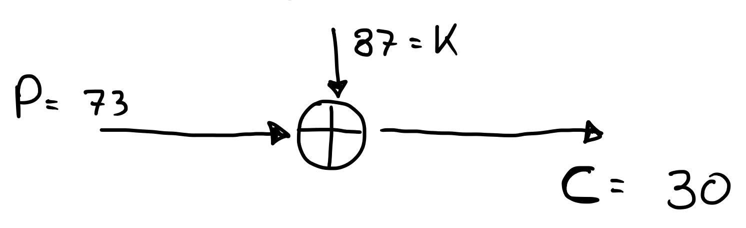
PROPRIETA' DELLA XOR

1. $a \oplus (b \oplus c) = (a \oplus b) \oplus c$
2. $a \oplus b = b \oplus a$
3. $a \oplus a = 0$
4. $a \oplus 0 = a$

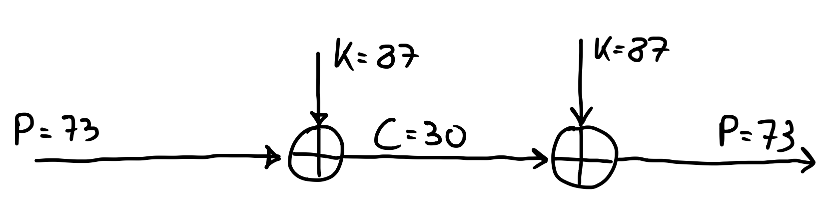
a	a	out
0	0	0
1	1	0

$$\begin{aligned} & a \oplus b \oplus a = && \leftarrow ② \\ & = \underbrace{a \oplus a} \oplus b = && \leftarrow ③ \\ & = 0 \oplus b = && \leftarrow ④ \\ & = b \end{aligned}$$

CASO PRATICO



$$\begin{array}{lcl} 73 & = & [0\ 1\ 0\ 0\ 1\ 0\ 0\ 1]_2 \quad \oplus \\ 87 & = & [0\ 1\ 0\ 1\ 0\ 1\ 1\ 1]_2 \quad = \\ \hline & = & 0\ 0\ 0\ 1\ 1\ 1\ 1\ 0 \\ 30 & = & 16 + 8 + 4 + 2 \end{array}$$



$$\begin{array}{lcl} C = 30 & = & [0\ 0\ 0\ 1\ 1\ 1\ 1\ 0]_2 \\ K = 87 & = & [0\ 1\ 0\ 1\ 0\ 1\ 1\ 1]_2 \\ \hline 73 & = & 0\ 1\ 0\ 0\ 1\ 0\ 0\ 1 \end{array}$$

ONE TIME PAD è uno schema di crittografia che è basato interamente su queste operazioni. Si chiama così perché tutta la sua sicurezza dipende da una sequenza di bit casuali che servono per cifrare il plaintext e devono necessariamente essere usati una volta sola.
Tale meccanismo è unico non solo per la sua semplicità ma anche perché garantisce il grado più forte possibile di sicurezza.
Se un hacker vede il Ciphertext è possibile dimostrare che acquisisce zero informazioni sulla chiave per decifrare.
Questa proprietà è chiamata Perfect Security.
Sfortunatamente molti sistemi che dichiarano di usare ONE TIME PAD non generano numeri puramente casuali.
Così facendo chi legge i messaggi cifrati può ottenere informazioni sulla chiave e sul Plaintext

CONSIDERAZIONI FINALI

ONE TIME PAD è utilizzato raramente perché è molto difficile da mettere in pratica i seguenti motivi:

- la chiave deve essere generata casualmente e deve essere condivisa tra le entità comunicanti
- la chiave deve essere lunga tanto quanto il messaggio che si vuole trasmettere.
- Generare numeri grandi puramente casuali è molto diffile e richiede hardware specializzato.