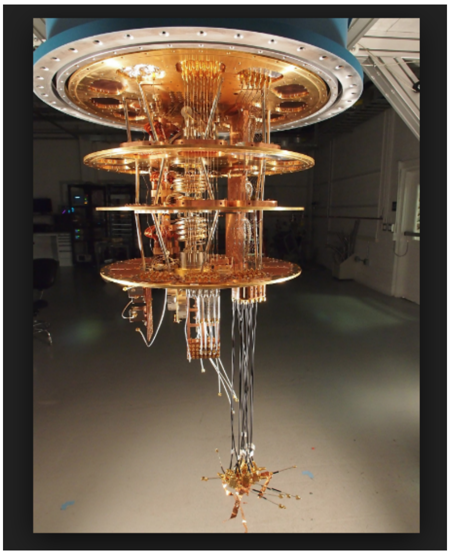
Computer quantistico?



se si realizza un computer quantistico la crittografia che abbiamo studiato fino a ora diventa superata

l'algoritmo di Shor è un algoritmo polinomiale per fattorizzare e trovare il LD su un QC

post-quantum cryptography

- si parla di post-quantum cryptography per crittosistemi che, almeno allo stato attuale, restano sicuri anche con un QC
- ce ne sono di vari tipi
  - lattice-based crypto
  - multivariate crypto
  - code-based crypto
  - e altri
- vediamo qualcosa sul crittosistema di McEliece, basato sui codici correttori di errori

un codice correttore di errori

Pensate a un numero fra 0 e 15 e rispondete alle seguenti domande. Potete mentire una volta.

- è maggiore o uguale a 8?
- è nell'insieme $\{4, 5, 6, 7, 12, 13, 14, 15\}$?
- è nell'insieme $\{2, 3, 6, 7, 10, 11, 14, 15\}$?
- è dispari?
- è nell'insieme $\{1, 2, 4, 7, 9, 10, 12, 15\}$?
- è nell'insieme $\{1, 2, 5, 6, 8, 11, 12, 15\}$?
- è nell'insieme $\{1, 3, 4, 6, 8, 10, 13, 15\}$?

16 risposte vere

$0000000 = c_0$
$0001111 = c_1$
$0010110 = c_2$
$0011001 = c_3$
$0100101 = c_4$
$0101010 = c_5$
$0110011 = c_6$
$0111100 = c_7$
$1000011 = c_8$
$1001100 = c_9$
$1010101 = c_{10}$
$1011010 = c_{11}$
$1100110 = c_{12}$
$1101001 = c_{13}$
$1110000 = c_{14}$
$1111111 = c_{15}$

- queste 16 stringhe formano un codice
- due stringhe differiscono in almeno tre posizioni
- è possibile correggere un errore: se introduciamo un solo errore la parola ottenuta sarà più vicina a quella di partenza che a ogni altra parola del codice
- e rivelarne due
- le 16 stringhe formano uno spazio vettoriale di dimensione 4 su $\mathbb{F}_2 = \mathbb{Z}_2$
- un sistema di generatori è per esempio

$$e_1 = 1000011,$$
$$e_2 = 0100101,$$
$$e_3 = 0010110,$$
$$e_4 = 0001111$$

codice di Hamming

- per codificare l'informazione $x = (x_1, x_2, x_3, x_4)$
- basta considerare la c.l. $x_1 e_1 + x_2 e_2 + x_3 e_3 + x_4 e_4 = c_x$
- la matrice G che contiene i 4 generatori si dice matrice generatrice
- la decodifica è più difficile - assumendo che ci sia un errore
- confrontare la parola ottenuta con le 16 parole del codice
- e vedere quale è più vicina
- la distanza fra due parole è il numero di posizioni in cui differiscono
- il codice che abbiamo descritto di chiama codice di Hamming

codici lineari

- in generale, un codice binario è un sottoinsieme di $\mathbb{Z}_2^n$ ($n = 7$ nell'esempio)
- si dice lineare se è un sottospazio vettoriale di $\mathbb{Z}_2^n$
- in tal caso contiene 2^k parole, se k è la sua dimensione ($k = 4$ nell'esempio)
- la matrice $k \times n$ G che contiene i k generatori si dice matrice generatrice
- nell'ex,

$$G = \begin{pmatrix} 1000011 \\ 0100101 \\ 0010110 \\ 0001111 \end{pmatrix}$$

codici correttori

- corregge e errori se la distanza minima fra due parole del codice è almeno $2e + 1$
- il codice di Hamming dell'esempio corregge un errore
- per un arbitrario codice lineare, la codifica è facile (basta calcolare un'opportuna combinazione lineare)
- ma la decodifica è difficile (è stato mostrato che è un problema NP-completo)
- è difficile per un codice lineare arbitrario: ci sono molti codici (Reed-Muller, Golay, Goppa) per cui la decodifica è facile

decodifica per il codice di Hamming

- per il nostro esempio, la decodifica è facile
- si considera la matrice

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

- se si riceve una parola $p = (p_1 \ldots, p_7)$ bisogna calcolare Hp^T
- si ottiene una terna, che dà la posizione dell'eventuale errore (scrittura binaria)

crittosistema di McEleice

- codifica facile - decodifica difficile
- Bob sceglie un (k, n) codice lineare t-correttore per cui esiste un procedimento di decodifica efficace, e una sua matrice generatrice G - chiave privata
- maschera il codice ottenendo un'altra matrice generatrice G^* che è la chiave pubblica
  - $G^* = SGP$
  - dove S è una matrice casuale $k \times k$ binaria invertibile
  - P è una matrice casuale di permutazione $n \times n$
- i plaintext sono stringhe binarie di lunghezza k
- Alice cifra codificando il messaggio con il codice mascherato e introducendo $\leq t$ errori casuali (randomizzazione)
- per decifrare Bob toglie il mascheramento e poi decodifica (correggendo al tempo stesso gli errori introdotti da Alice)
- il problema è che le chiavi sono enormi (matrici $k \times n$ con $k = 512, n = 1024$ nella proposta originale)