Informazione e computazione quantistica: quali conseguenze?

Author : Redazione

Date : 5 Giugno 2019

Estratto dalla relazione di Alessandro Luongo tenutasi al 10° Cyber Crime Conference 2019

Mi piacerebbe fare un po' di divulgazione scientifica per capire quali siano le conseguenze della comunicazione della computazione quantistica nel campo della Cyber Security. Vorrei spiegare - con pochissima matematica - cosa vuol dire usare la meccanica quantistica per fare dei calcoli (e per questo voglio anche portarvi l'esempio di alcune applicazioni sviluppate nel nostro laboratorio) ovvero come utilizzare dei computer quantistici per fare formal software verification e malware detection.

Vedremo poi, dal lato dell'attaccante, cosa si può fare avendo a disposizione un computer quantistico - probabilmente lo sapete già, si può riuscire a scomporre un numero in fattori primi e rompere la crittografia a chiave pubblica; vedremo cosa questo comporti nell'ambito delle cryptocurrencies - e poi faremo un'overview generale su cosa si possa fare utilizzando della comunicazione quantistica.

Ecco due o tre slide con un po' di matematica: se noi volessimo descrivere matematicamente l'evoluzione di un sistema fisico classico, come per esempio la palla lanciata da questo ragazzo, abbiamo a disposizione le leggi che descrivono il moto e quindi possiamo determinare, per ogni intervallo di tempo, la posizione della palla all'interno del nostro sistema: una cosa molto facile, che abbiamo studiato alle superiori (e che probabilmente abbiamo dimenticato). Se anche avessimo necessità di descrivere due di questi sistemi fisici classici, non è un grosso problema;

Classical mechanics

$$x = x_0 + v_0 t \qquad y = -\frac{1}{2} g t^2 + v_{0_y} t + y_0$$

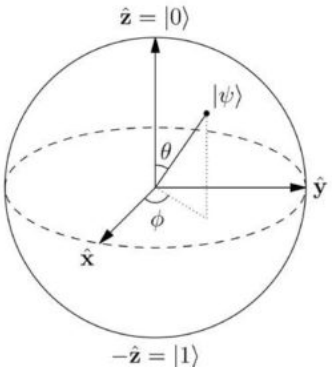
Ma nel caso in cui abbiamo a disposizione un Qubit - la più piccola unità che usiamo per salvare le informazioni all'interno del sistema quantistico - incorriamo in un piccolo problema perché, se vogliamo descrivere due Qubit assieme, la lunghezza del vettore necessario per salvare questa informazione raddoppia ogni volta che aggiungiamo un Qubit. La conseguenza è in questa formula:

Introduction to quantum computing – 1

- **Def**: A qubit is a unit vector in a 2-dimensional space (over complex number)

$$|\psi\rangle = [a, b] \qquad |a|^2 + |b|^2 = 1 \qquad a, b \in \mathbb{C}$$

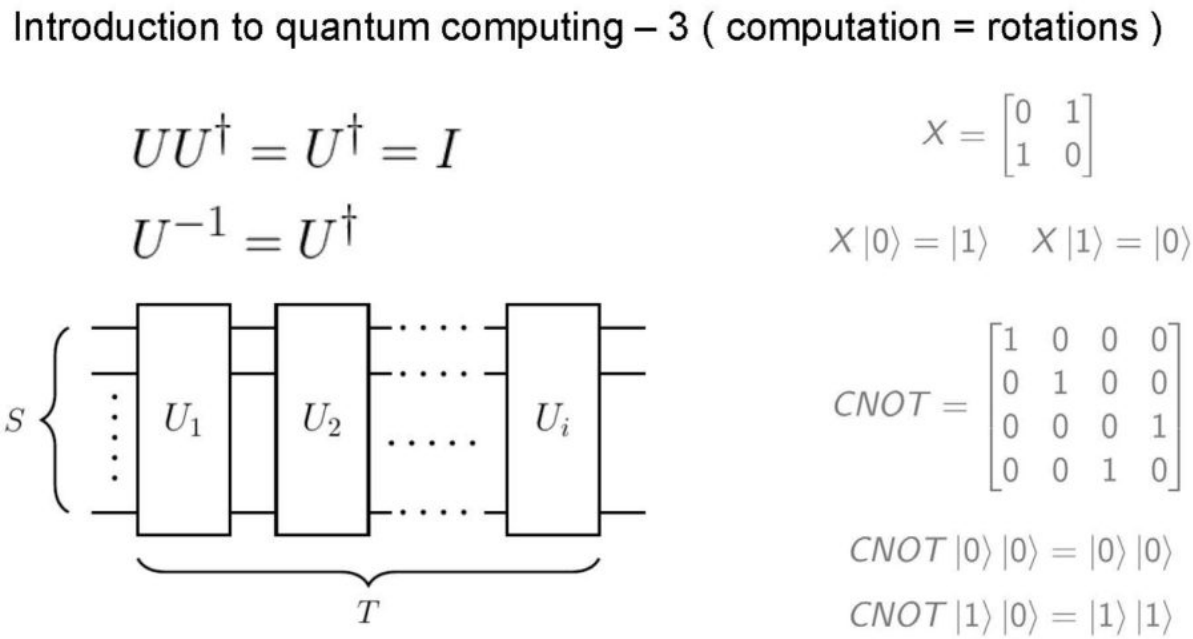
$$|CAT\rangle = \frac{1}{\sqrt{2}}|DEAD\rangle + \frac{1}{\sqrt{2}}|ALIVE\rangle$$



la particolarità è che la dimensione del vettore necessario a salvare tutti questi numeri che descrivono lo stato del sistema quantistico cresce esponenzialmente con il numero di Qubit e, quindi, con un computer classico non riusciamo a simulare in maniera efficiente più di 50 Qubit. Crediamo che, grazie al computer quantistico, riusciamo a fare dei calcoli che prima non riuscivamo a fare, o a svolgerli in maniera più efficiente.

Dal lato della programmazione, programmazione ed evoluzione di un computer vengono descritte da dalle matrici. Qui vedete il caso più semplice, una matrice che rappresenta una sorta di not logico: è facile vedere come mandi un Qubit, che è in coda allo stato zero, nel Qubit che in coda al valore dello stato uno. Possiamo visualizzare un software quantistico semplicemente come un grosso circuito, molto simile a un circuito elettrico; l'evoluzione del tempo è sull'asse orizzontale e sull'asse verticale abbiamo il numero di Qubit ovvero, in qualche modo, quanto sia potente il nostro computer. Giusto per dare un'idea pratica di cosa voglia dire programmare un computer quantistico, ora questa cosa la stiamo facendo in Python: ci sono dei linguaggi Open Source che permettono di "giocare" con dei computer quantistici piuttosto piccoli (fino a 16 Qubit circa).

Potete pensare al computer quantistico come ad un coprocessore, quasi come una GPU, al quale inviate dei calcoli molto specifici da fare. A livello di interfaccia utente, vedete un oggetto che chiamiamo Quantum Circuit, una grossa lista di istruzioni che andiamo a riempire con il nostro codice.



Questa, per esempio, è una funzione che serve semplicemente a fare la somma tra due numeri. Non serve capire bene cosa faccia nel dettaglio - ci vorrebbero ore e ore per spiegarlo - ma, alla

fine, potete vedere che il software genera questo circuito: le linee verticali sono i Qubit, qui vedete i gates che fanno la somma tra due numeri.

Sappiamo benissimo che, grazie all'algoritmo di Shor, riusciamo a "rompere" quasi tutta la crittografia a chiave pubblica che abbiamo oggi a disposizione, - curve ellittiche, logaritmo discreto, il problema del factoring... - con un computer quantistico, riuscire a scomporre in fattori primi un numero del genere potrebbe richiedere una settimana o qualche giorno, dipende dallo sviluppo della tecnologia che implementa effettivamente questi strumenti; non è un processo istantanea, siamo ancora relativamente lontani da avere computer quantistici grandi abbastanza da riuscire a rompere la crittografia di RSA a 2048 bit.

Continua a leggere…

Scarica gratuitamente gli atti del 10° Cyber Crime Conference 2019