PROTOCOLLO BB84 – Q# Holiday Calendar 2022

MICROSOFT

MARIO CUOMO
CLOUD SOLUTION ARCHITECT

Sommario

ABSTRACT

Se da una parte il quantum computing pone una forte sfida alla Church–Turing thesis, mostrando come sia possibile utilizzare il nuovo modello di calcolo per risolvere in maniera efficiente problemi di complessità np, dall'altra offre altrettante sicurezze.
Un esempio è il protocollo BB84 – un protocollo quantico per la distribuzione di segreti condivisi in un canale che può essere non sicuro.

Un segreto condiviso è uno strumento utilizzato nella crittografia per rendere una comunicazione sicura. Spesso questo termine si inserisce nell'ambito della crittografia simmetrica: la situazione in cui le due parti comunicanti utilizzano la stessa chiave per cifrare e decifrare i messaggi che si scambiano. Lo shared secret può essere la chiave di cifratura o un seed secret dal quale generare la vera e propria chiave.

In questo breve articolo è descritto in modo semplice – e in alcuni casi semplicistico – il modello di calcolo del quantum computing fornendo gli strumenti base per comprendere a pieno il funzionamento del protocollo.

INTRODUZIONE AL MODELLO COMPUTAZIONALE QUANTICO

Questo breve articolo non si pone l'obiettivo di spiegare come sia possibile manipolare le particelle quantiche ma piuttosto spiega come sia possibile sfruttarne le caratteristiche fisiche realizzando un nuovo modello di calcolo.

L'unità di calcolo elementare di questo modello è il qubit – un elemento il cui stato è compreso tra i due stati base $|0\rangle$ e$|1\rangle$. Non è possibile conoscerne con certezza il valore fino al momento in cui non si effettua una misurazione su di esso. Si parla in questo caso di decadimento del qubit in uno stato base.
In generale lo stato di un qubit si indica nel seguente modo

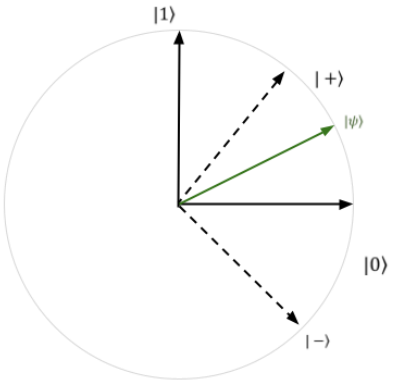
$$|\psi\rangle = a_0|0\rangle + a_1|1\rangle$$

I valori di ampiezza a_0 e a_1 sono dei numeri complessi e rappresentano la probabilità che misurato il qubit esso decade rispettivamente nello stato base $|0\rangle$ oppure $|1\rangle$.
La probabilità che misurando $|\psi\rangle$ esso decade nello stato $|0\rangle$ è $|\alpha_0|^2$.

Quella presentata prende il nome di ket-notation introdotta da Paul Dirac.
In realtà possiamo rappresentare i qubit anche in forma matriciale con la seguente notazione.

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle = \alpha_0\begin{pmatrix}1\\0\end{pmatrix} + \alpha_1\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}\alpha_0\\\alpha_1\end{pmatrix}$$

Per semplicità l'introduzione dei qubit è avvenuta rispetto la base computazionale – ovvero quella composta dai versori $|0\rangle$ e$|1\rangle$. È importante notare che un qubit esiste indipendentemente dalla base rispetto alla quale si sta rappresentando. È possibile infatti rappresentare i qubit in qualsiasi altra base, ovvero rispetto a qualsiasi coppia di versori ortogonali tra loro.
Un'altra base è quella di Hadamard composta di versori $|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ e $|-\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$.



È possibile manipolare i qubit attraverso diversi operatori. L'effetto di un operatore non è altro che la rotazione del qubit nello spazio – che è chiamato Spazio di Hilbert.
L'unico vincolo che si ha è che gli operatori devono essere delle unitary transformation, ovvero l'applicazione in successione dell'operatore U e $U^\dagger$ al qubit $|\psi\rangle$ devono restituire $|\psi\rangle$ (o, detto in altri termini, il prodotto $UU^\dagger$ deve essere l'identità I).
Un operatore U è rappresentato da una matrice e $U^\dagger$ è la trasposta della complessa coniugata.

Gli operatori più noti sono quelli di Pauli – bit flip, phase flip, operatore Y e identità I.
Considerando un qubit $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ si ha che

	U	U$\|\psi\rangle$
bit flip (X)	$\begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}$	$\alpha_1\|0\rangle + \alpha_0\|1\rangle$
phase flip (Z)	$\begin{pmatrix}1 & 0\\0 & -1\end{pmatrix}$	$\alpha_0\|0\rangle - \alpha_1\|1\rangle$
operatore Y	$\begin{pmatrix}0 & 1\\-1 & 0\end{pmatrix}$	$\alpha_1\|0\rangle - \alpha_0\|1\rangle$
operatore identità (I)	$\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}$	$\alpha_0\|0\rangle + \alpha_1\|1\rangle$

È anche possibile considerare più qubit assieme.
Quando due o più qubit entrano in contatto tra loro essi devono essere considerati non più come singoli ma come un'unica entità che si influenzano tra loro. È il principio dell'entanglement.
In generale, lo stato di due qubit è descritto nel seguente formalismo

$$\alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

Uno stato entangled usato è lo stato di Bell $|\Phi^+\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$ e mostra con semplicità questo principio: lo stato complessivo del sistema è bilanciato tra gli stati base $|00\rangle$ e $|11\rangle$. Se misuriamo il primo qubit e – per esempio – otteniamo $|1\rangle$ allora inevitabilmente anche il secondo qubit si troverà nello stato $|1\rangle$. È come se il secondo qubit si sia spostato da uno stato di equilibrio a uno stato base come conseguenza della misurazione e decadimento del primo.

Ricapitolando, due qubit possono trovarsi singolarmente ciascuno in uno stato $\alpha_0|0\rangle + \alpha_1|1\rangle$ e $\beta_0|0\rangle + \beta_1|1\rangle$ ma quando sono assieme il loro stato è descritto da $\alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$.
L'operatore che si utilizza è il prodotto tensore $\otimes$.

$$\begin{pmatrix}\alpha_0\\\alpha_1\end{pmatrix} \otimes \begin{pmatrix}\beta_0\\\beta_1\end{pmatrix} = \begin{pmatrix}\alpha_0\begin{pmatrix}\beta_0\\\beta_1\end{pmatrix}\\\alpha_1\begin{pmatrix}\beta_0\\\beta_1\end{pmatrix}\end{pmatrix} = \begin{pmatrix}\alpha_0\beta_0\\\alpha_0\beta_1\\\alpha_1\beta_0\\\alpha_1\beta_1\end{pmatrix} = \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle$$

A due qubit è possibile applicare diversi operatori – sempre con il vincolo di essere delle unitary transformation.
Gli operatori maggiormente utilizzati sono il ControlledNOT e Hadamard. Il primo è l'equivalente del CNOT su bit classici e il secondo è utile per porre in superposition equilibrata un qubit decaduto su uno stato base.
Considerando $|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$

	U
CNOT	$\begin{pmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{pmatrix}$
Hadamard(H_2)	$\frac{1}{\sqrt{2}}\begin{pmatrix}H_1 & H_1\\H_1 & -H_1\end{pmatrix}$

con

$$H_1 = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix}$$

PROTOCOLLO BB84

Il protocollo BB84 è utilizzato per la generazione di un segreto crittografico condiviso tra due parti – che chiamiamo per semplicità Alice e Bob. È importante specificare che il segreto condiviso è generato in modo random e non deciso a priori da una delle due parti: questo è un ulteriore grado di protezione in quanto non è necessario utilizzare uno specifico algoritmo per generarlo.
Il protocollo è proposto da Charles Bennet e Gilles Brassard nel 1984.

Il protocollo si compone principalmente di 3 fasi:

1. Alice genera una superposition di qubit a partire da due sequenze scelte random e la invia a Bob
2. Bob genera una sequenza random e la utilizza per polarizzare i qubit della superposition
3. Alice e Bob si confrontano per ottenere il segreto condiviso

A queste due fasi se ne aggiungono altre due che assicurano la robustezza contro il rumore del canale e un'ipotetica parte malevola – chiamata Eve.

Iniziamo ad analizzare la fase 1.
Alice possiede n qubit negli stati base e genera in modo random due sequenze di bit – α e β – entrambe di lunghezza n.
La sequenza α è una semplice sequenza di bit che rappresenta una superposition di qubit in stati base: se $\alpha_i = 0$ si ha un qubit nello stato base $|0\rangle$, se $\alpha_i = 1$ si ha un qubit nello stato base $|1\rangle$.
La sequenza β indica come polarizzare i qubit: se $\beta_i = 0$ il qubit i è polarizzato con la base computazionale, se $\beta_i = 1$ il qubit i è polarizzato con la base di Hadamard.

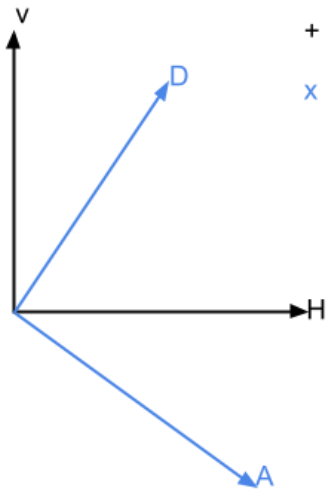
Lo stato complessivo degli n qubit è il prodotto tensore dei qubit $|\psi_{\alpha_i\beta_i}\rangle$.
Alice trasmette la seguente superposition

$$|\psi\rangle = \bigotimes_{i=1}^{n} |\psi_{a_i b_i}\rangle$$

Per avere una leggibilità migliore della superposition attribuiamo dei nomi alle varie forme che i qubit possono assumere.

$\lvert\psi_{\alpha_i\beta_i}\rangle$	Nome simbolico
$\lvert\psi_{00}\rangle = \lvert 0\rangle$	Vertical (V)
$\lvert\psi_{10}\rangle = \lvert 1\rangle$	Horizontal (H)
$\lvert\psi_{01}\rangle = \lvert +\rangle = \frac{1}{\sqrt{2}}\lvert 0\rangle + \frac{1}{\sqrt{2}}\lvert 1\rangle$	Diagonal (D)
$\lvert\psi_{11}\rangle = \lvert -\rangle = \frac{1}{\sqrt{2}}\lvert 0\rangle - \frac{1}{\sqrt{2}}\lvert 1\rangle$	Anti − Diagonal (A)

Quello che ci comunica la tabella è molto semplice.
Quando polarizziamo lo stato $|0\rangle$ rispetto alla base computazionale otteniamo lo stato V.
Quando polarizziamo lo stato $|1\rangle$ rispetto alla base computazionale otteniamo lo stato H.
Quando polarizziamo lo stato $|0\rangle$ rispetto alla base di Hadamard otteniamo lo stato D.
Quando polarizziamo lo stato $|1\rangle$ rispetto alla base di Hadamard otteniamo lo stato A.
In letteratura non è difficile trovare la seguente nomenclatura H(→), V(↑), D(↗), A(↘), base operazionale (+), base di Hadamard (×).



Polarizzare un qubit rispetto a un'altra base è un'operazione molto semplice.
Si consideri il qubit $|1\rangle$ nella base computazionale. Per esprimerlo nella base di Hadamard è sufficiente applicare l'operatore di Hadamard ad esso.

$$H_1|1\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\end{pmatrix} - \frac{1}{\sqrt{2}}\begin{pmatrix}0\\1\end{pmatrix} = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle = |-\rangle$$

Ipotizziamo Alice abbia generato le seguenti stringhe.

$$\alpha = 0111010010100101$$
$$\beta = 0110010100010111$$

Alice polarizza i sui qubit nel seguente modo

α	0	1	1	1	0	1	0	0	1	0	1	0	0	1	0	1
β	+	×	×	+	+	×	+	×	+	+	+	×	+	×	×	×
$\psi_{\alpha\beta}$	↑	↘	↘	→	↑	↘	↑	↗	→	↑	→	↗	↑	↘	↗	↘

A questo punto Alice invia i qubit $|\psi\rangle$ a Bob. Può farlo utilizzando un canale quantico o utilizzare il protocollo del teletrasporto.

Inizia la fase 2.
Bob genera una sequenza random β'di n bit. Quello che vuole fare è cercare di indovinare quali sono le basi di codifica utilizzate da Alice per recuperare il messaggio di partenza α.
Statisticamente parlando, Bob indovina il 50% delle basi.
Ipotizziamo Bob abbia generato la stringa

$$\beta' = 0101010010100100$$

Utilizzando β' Bob polarizza $\psi_{\alpha\beta}$ nel seguente modo

α	0	1	1	1	0	1	0	0	1	0	1	0	0	1	0	1
β	+	×	×	+	+	×	+	×	+	+	+	×	+	×	×	×
$\psi_{\alpha\beta}$	↑	↘	↘	→	↑	↘	↑	↗	→	↑	→	↗	↑	↘	↗	↘
β'	+	×	+	×	+	×	+	+	×	+	×	+	+	×	+	+
α' polarizzato	↑	↘	→	↗	↑	↘	↑	→	↗	↑	↗	→	↑	↘	→	→
α'se misurato	0	1	1	0	0	1	0	1	0	0	0	1	0	1	1	1

Inevitabilmente sono presenti degli errori nelle misurazioni di Bob dovuti alla non conoscenza della stringa β.

Nella fase 3 Bob e Alice si scambiano su un canale classico le basi di codifica utilizzate e scartano i qubit per i quali esse che non combaciano. I valori α_i e α_i' in corrispondenza dei bit β_i e β_i' che combaciano rappresentano il segreto condiviso.
Quello che rimane è un segreto condiviso tra i due!

β	+	×	×	+	+	×	+	×	+	+	+	×	+	×	×	×
β'	+	×	+	×	+	×	+	+	×	+	×	+	+	×	+	+
$shared\ secret$	0	1			0	1	0			0			0	1		

Se la lunghezza del segreto condiviso è minore di $n/2$ è consigliato ricominciare con l'esecuzione del protocollo.

Cosa succede se è presente un utente malevolo sul canale di trasmissione?
Grazie al teorema di no clonazione Eve non può avere una copia dei qubit inviati da Alice a Bob.
L'unica cosa che può fare è un man in the middle attivo allo step 2: Eve acquisisce i qubit da Alice, ne effettua delle misurazioni e li invia a Bob.
Eve si trova nella stessa situazione di Bob perchè deve indovinare le basi scelte da Alice. Lo fa con un successo statistico del 50%. Quando Eve effettua delle misurazioni dei qubit ne perturba lo stato lasciando indirettamente tracce del suo operato.
Vediamo perché.

Immaginiamo che Eve abbia impersonificato Bob e riceve $|\psi\rangle$ da Alice.
Eve genera una stringa β'' di n bit che sfrutta per polarizzare il messaggio.

$$\beta'' = 1010001100100010$$

α	0	1	1	1	0	1	0	0	1	0	1	0	0	1	0	1
β	+	×	×	+	+	×	+	×	+	+	+	×	+	×	×	×
$\psi_{\alpha\beta}$	↑	↘	↘	→	↑	↘	↑	↗	→	↑	→	↗	↑	↘	↗	↘
β''	×	+	×	+	+	+	×	×	+	+	×	+	+	+	×	+
α''polarizzato	↗	→	↘	→	↑	→	↗	↗	→	↑	↗	→	↑	→	↗	→

Eve invia α''polarizzato a Bob che effettua le sue misurazioni con la stringa β'.

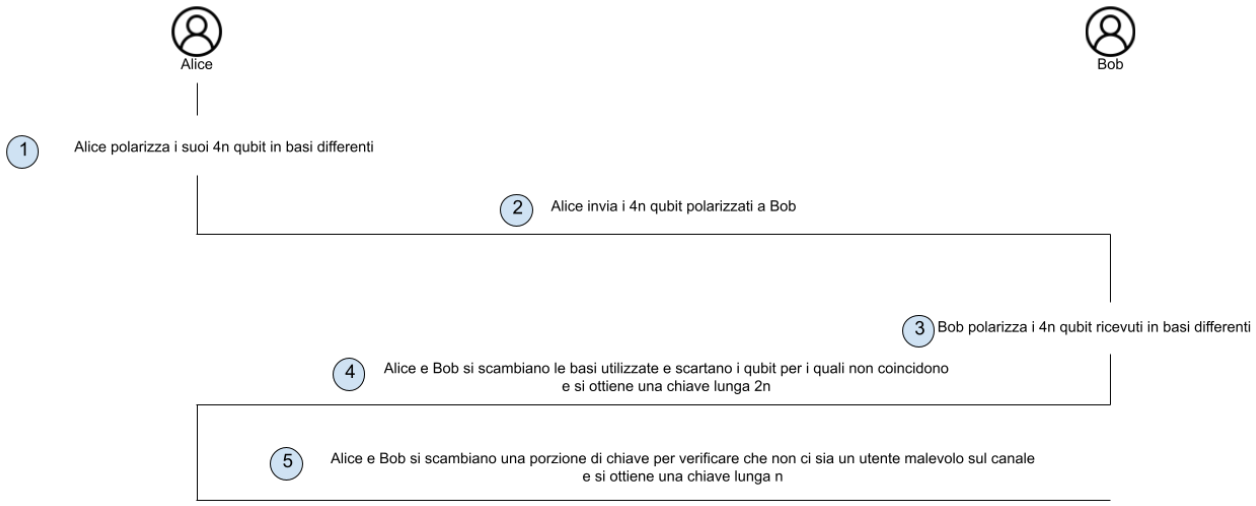
α	0	1	1	1	0	1	0	0	1	0	1	0	0	1	0	1
β	+	×	×	+	+	×	+	×	+	+	+	×	+	×	×	×
$\psi_{\alpha\beta}$	↑	↘	↘	→	↑	↘	↑	↗	→	↑	→	↗	↑	↘	↗	↘
β''	×	+	×	+	+	+	×	×	+	+	×	+	+	+	×	+
α''polarizzato	↗	→	↘	→	↑	→	↗	↗	→	↑	↗	→	↑	→	↗	→
β'	+	×	+	×	+	×	+	+	×	+	×	+	+	×	+	+
α'polarizzato	→	↗	→	↗	↑	↗	→	→	↗	↑	↗	→	↑	↗	→	→
α'se misurato	1	0	1	0	0	0	1	1	0	0	$\underline{0}$	1	0	0	1	1

Alice e Bob si scambiano correttamente le basi utilizzate per acquisire lo shared secret.

β	+	×	×	+	+	×	+	×	+	+	+	×	+	×	×	×
β'	+	×	+	×	+	×	+	+	×	+	×	+	+	×	+	+
$shared\ secret\ per\ Alice$	0	1			0	1	0			0			0	1		
$shared\ secret\ per\ Bob$	1	0			0	0	1			0			0	0		

Come si nota le due chiavi segrete non coincido ma Alice e Bob ancora non lo sanno.
Decidono di scambiarsi $n/4$ bit – per esempio i primi – del segreto condiviso per un check.
Statisticamente parlando il bit error rate è lo stesso tra sequenza trasmessa e quella non trasmessa.
Se gli errori presenti sono in numero minore di $n/8$, Alice e Bob possono decide di utilizzare i restanti $n/4$ bit come segreto condiviso.
A questo punto gli errori sul segreto condiviso possono essere mitigati utilizzando la tecnica Privacy Amplification – un metodo che permette ad Alice e Bob di generare una chiave completamente segreta sapendo che Eve ha informazioni parziali sulla stringa di partenza.

Per generare un segreto condiviso di n bit è necessario quindi utilizzare $4n$ qubit.
Di seguito uno schema riassuntivo di ciò che succede nel protocollo.



IMPLEMENTAZIONE IN Q#

È possibile usare un simulatore quantico per testare localmente le applicazioni che utilizzano qubit e Microsoft ne mette a disposizione uno nel Microsoft Quantum Developer Kit.
Un simulatore quantum non è altro che un software eseguito su un calcolatore tradizionale ma che permette di simulare il comportamento dei qubit.

Di seguito si utilizza uno script python che interagisce con uno script .qs.
Sono state implementate due funzioni – `BB84WithoutEve e BB84WithEve` – che restituiscono il segreto condiviso percepito da Bob. Come suggerisce il nome, solo nel primo caso il segreto coincide con quello posseduto da Alice.

A solo titolo di esempio consideriamo la funzione `BB84WithoutEve`.

```
operation BB84WithoutEve(a : Int[], b : Int[], c : Int[], d : Int[]): String
```

a	È un array di 16 bit. Rappresenta la stringa α di Alice.
b	È un array di 16 bit. Rappresenta la stringa β di Alice.
c	È un array di 16 bit. Rappresenta la stringa β' di Bob.
d	È un array di 16 bit. Rappresenta il momento in cui Alice e Bob si scambiano le chiavi. `d[i]==1` se `b[i]==c[i]`

Per prima cosa si crea un vettore contenente 16 qubit negli stati base $|0\rangle$ e $|1\rangle$ in accordo ai valori nell'array a.

```
use qubits = Qubit[16];
for i in 0 .. 15 {
     let bit = a[i];
     if bit==1 {
          X(qubits[i]);
     }
}
```

Inizialmente i 16 qubit si trovano tutti nello stato $|0\rangle$.
Con l'operatore X è possibile trasformarli nello stato $|1\rangle$.

A questo punto Alice polarizza i qubit in accordo ai valori nell'array `b`.

```
for i in 0 .. 15 {
   let _base = b[i];
   if _base == 1 {
       H(qubits[i]);
   }
}
```

Bob fa lo stesso in accordo ai valori nell'array `c`.

```
for i in 0 .. 15{
   let _base = c[i];
   if _base == 1 {
       H(qubits[i]);
   }
}
```

Alice e Bob si scambiano le loro basi e mantengono solo i valori tali per cui `d[i]==1`.

```
mutable shared_secret = "";
for i in 0 .. 15 {
  let common_base = d[i];
  if common_base == 1 {
     let misure = M(qubits[i]);
     mutable x = "0";
     if misure == One {
       set x = "1";
     }
     set shared_secret = shared_secret + x;
  }
}
```

Prima di concludere la funzione è buona norma pulire il contenuto dei qubit.

```
for i in 0 .. 15 {
   Reset(qubits[i]);
}
```

RISORSE

Il mio interesse per il quantum computing nasce grazie alle lezioni del corso universitario Next Generation Computing Models all'università Roma Tre.
Di seguito alcuni miei articoli riguardo l'argomento

- *https://github.com/mariocuomo/Microsoft-Q-Advent-Calendar-2021*
  (No-Cloning Theorem)
- *https://medium.com/@mariocuomo/teletrasporto-non-solo-fantascienza-f4663b5a1c3a*
  (Teleportation protocol)
- *https://medium.com/@mariocuomo/superdense-coding-due-bit-al-prezzo-di-un-qubit-a341457352d5*
  (Superdense coding protocol)

Il codice presentato è disponibile a *https://github.com/mariocuomo/Microsoft-Q-Holiday-Calendar-2022*

Per qualsiasi chiarimento, correzione o dubbio `cuomomario@hotmail.com`