Analisi teorica, implementazione ed esecuzione su computer quantistico del protocollo BB84

Università degli Studi di Milano – Bicocca

Laurea Triennale in Fisica

25 febbraio 2021

Candidato: Davide Rinaldi

Matricola: 826346

Relatore interno: Dr. Andrea Giachero

Relatore esterno: Prof. Paolo Solinas

Correlatrice: Dr. Elena Ferri

Quantum Computing – circuiti quantistici e *Qiskit*

Quantum Computing: tecnica computazionale basata sull'utilizzo dei fenomeni quantistici, come:

- la sovrapposizione degli stati fisici
- la correlazione quantistica o Entanglement
- l'interferenza delle funzioni d'onda delle particelle

Un **computer quantistico** è un computer che permette l'esecuzione di algoritmi di Quantum Computing.

Il più diffuso modello di Quantum Computing è il **circuito quantistico**:

1. Preparazione di uno o più **qubit** in un certo stato fisico;
2. Inserimento di **porte quantistiche** (trasformazioni unitarie) che agiscono sui qubit;
3. Misura dello stato del qubit;

- Risultato: è un bit classico (0 oppure 1), a cui è associata una **probabilità** data dallo stato in cui si trova il qubit.

IBM Quantum Experience: è una piattaforma cloud su cui è possibile scrivere programmi di Quantum Computing. Questi programmi sono eseguibili da remoto sui simulatori e sui computer quantistici messi a disposizione da IBM.

Qiskit è un framework open-source sviluppato da IBM, che permette di sfruttare un'interfaccia ad alto livello in linguaggio Python per interagire con i computer quantistici.
Con Qiskit è possibile implementare i circuiti quantistici, per poi eseguirli sui dispositivi di IBM.

Elementi di un circuito quantistico – qubit e misura

Qubit: sistema fisico a due livelli, soggetto alle leggi della Meccanica Quantistica. È l'unità minima di informazione quantistica, e può trovarsi in uno stato $|\psi\rangle$ che è *sovrapposizione* di due stati diversi.

Base canonica o computazionale: base ortonormale data dai due vettori di stato: $|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ e $|1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

$|0\rangle$ e $|1\rangle$ sono gli autostati dell'operatore $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$, i cui autovalori sono 1 e -1.

Altri possibili stati di un qubit sono $|+\rangle$ e $|-\rangle$, così definiti: $|+\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ e $|-\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

$|+\rangle$ e $|-\rangle$ sono gli autostati dell'operatore $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, con autovalori 1 e -1.

In Qiskit, inizialmente un qubit è sempre preparato nello stato $|0\rangle$.

Misura: operazione che permette di estrarre un output classico dallo stato di un qubit.
La misura standard in Qiskit è quella dell'osservabile rappresentata dall'operatore Z.

Possibili risultati: 0 oppure 1 (il valore 0 rappresenta l'autovalore 1, mentre il valore 1 rappresenta l'autovalore -1).

La misura dell'osservabile Z su un qubit dà quindi in output un *bit classico*.

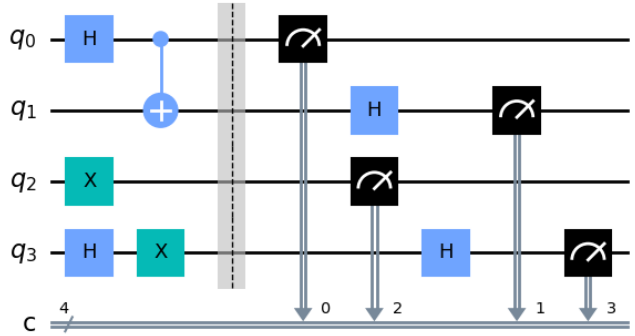
Elementi di un circuito quantistico – porte quantistiche principali

Porta quantistica (quantum gate): trasformazione unitaria che agisce sullo stato iniziale di un qubit, portandolo in uno stato finale. Questa trasformazione è data da un opportuno operatore hermitiano, rappresentabile in forma matriciale.

Porte quantistiche a singolo qubit:

- Porte *X, Y, Z*: l'effetto sugli stati della base canonica è:

$$X|0\rangle \equiv |1\rangle \qquad X|1\rangle \equiv |0\rangle$$
$$Y|0\rangle \equiv -i|1\rangle \qquad Y|1\rangle \equiv i|0\rangle$$
$$Z|0\rangle \equiv -|0\rangle \qquad Z|1\rangle \equiv |1\rangle$$



- Porta di Hadamard: $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ il cui effetto sugli stati $|0\rangle, |1\rangle, |+\rangle$, e $|-\rangle$ è:

$$H|0\rangle \equiv |+\rangle \qquad H|1\rangle \equiv |-\rangle$$
$$H|+\rangle \equiv |0\rangle \qquad H|-\rangle \equiv |1\rangle$$

Porte quantistiche a due qubit:

- *CNOT*: se il *control qubit* si trova nello stato $|1\rangle$, applica una porta X al *target qubit*:

$$CNOT\,|\psi_{control}\rangle \otimes |\psi_{target}\rangle = \begin{cases} |\psi_{control}\rangle \otimes |\psi_{target}\rangle \text{ se } |\psi_{control}\rangle = |0\rangle \\ |\psi_{control}\rangle \otimes (X\,|\psi_{target}\rangle) \text{ se } |\psi_{control}\rangle = |1\rangle \end{cases}$$

Il protocollo BB84 – Quantum Computing per la crittografia quantistica

Il primo protocollo di crittografia quantistica fu ideato nel 1984 da C. H. Bennett e G. Brassard.
Si tratta del **protocollo BB84**.

Il protocollo BB84 è un protocollo a **chiave privata** di *Quantum Key Distribution* (QKD).

La **funzione del protocollo** è permettere la condivisione sicura di una chiave crittografica tra due interlocutori.

La chiave viene condivisa usando dei **qubit** preparati in stati diversi per codificare l'informazione da trasmettere.

Il protocollo basa la sua efficacia su due fenomeni quantistici in particolare:
- la sovrapposizione degli stati fisici;
- il collasso della funzione d'onda di un sistema dopo una misura.

Il protocollo è completamente sicuro, perché permette di individuare sistematicamente una eventuale intercettazione della chiave da parte di un hacker.

Infatti, un hacker:
- non può clonare lo stato dei qubit trasmessi, copiandolo su altri qubit, per il **Teorema di no-cloning**;
- se compie una **misura** sui qubit trasmessi, **perturba lo stato** di alcuni di essi e viene individuato.

Il protocollo BB84 – Funzionamento

Alice vuole condividere una chiave crittografica con Bob tramite il protocollo BB84.

1. Alice estrae una stringa di N bit pseudocasuali. Sono i ***bit da codificare*** .
2. Alice estrae un'altra stringa di N bit. È la *stringa delle basi*, che serve per la **codifica dei bit**.

1	0	1	0	1	1	0	0
0	1	1	0	0	0	1	0

3. Alice codifica i bit della prima stringa preparando un qubit per ogni bit. Ogni qubit viene preparato in questo modo:

Bit della stringa delle basi:	**0**			**1**		
Bit codificati in qubit:	**0**	$\mapsto$	$\|0\rangle$	**0**	$\mapsto$	$\|+\rangle$
	1	$\mapsto$	$\|1\rangle$	**1**	$\mapsto$	$\|-\rangle$

1	0	1	0	1	1	0	0
0	1	1	0	0	0	1	0
$\|1\rangle$	$\|+\rangle$	$\|-\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|+\rangle$	$\|0\rangle$

4. I qubit sono poi trasmessi a Bob, il quale estrae a sua volta una stringa di N bit (*stringa delle basi*). Questi bit sono usati per la **scelta dell'osservabile da misurare**:

0	0	1	1	0	1	1	0

- Se il bit estratto è 0, Bob misura l'osservabile Z (cioè misura *nella base* $\{|0\rangle, |1\rangle\}$).
- Se il bit estratto è 1, Bob misura l'osservabile X (cioè misura *nella base* $\{|+\rangle, |-\rangle\}$).

5. I risultati delle misure di Bob sono una stringa di N bit.

1	0	1	1	1	0	0	0

6. Terminata la trasmissione, Alice e Bob **pubblicano le rispettive *stringhe delle basi*.**
7. Le confrontano e ***scartano*** tutte le coppie di bit **diversi** (diversa scelta di base, quindi risultati *probabilistici*).
8. Alice pubblica *la prima metà* dei bit che ha codificato, mentre Bob pubblica la *prima metà* dei risultati delle misure. Confrontandoli, ci si aspetta che questi bit (detti ***bit correlati***) siano tutti **identici**.

Il protocollo BB84 – Intervento e individuazione dell'hacker

- La scelta di due basi diverse porta a risultati *probabilistici.*
- La stessa scelta di base porta a risultati *certi.*

I risultati probabilistici vengono quindi scartati; pertanto ci si aspetta che i risultati salvati siano tutti certi, e dunque **tutti identici** ai bit codificati inizialmente.

	Pubblici				Privati			
Stringa dei bit codificati da Alice:	**1**	*0*	**1**	*0*	**1**	*1*	**0**	**0**
Stringa delle basi di Alice:	0	1	1	0	0	0	1	0
Qubit:	$\|1\rangle$	$\|+\rangle$	$\|-\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|+\rangle$	$\|0\rangle$
Stringa delle basi di Bob:	0	0	1	1	0	1	1	0
Stringa dei risultati di Bob:	**1**	*0*	**1**	*1*	**1**	*0*	**0**	**0**
Salvati (✓) o scartati (✗):	✓	✗	✓	✗	✓	✗	✓	✓

Un eventuale hacker (chiamato Eve) può agire esattamente come Bob, **misurando direttamente i qubit intercettati**:

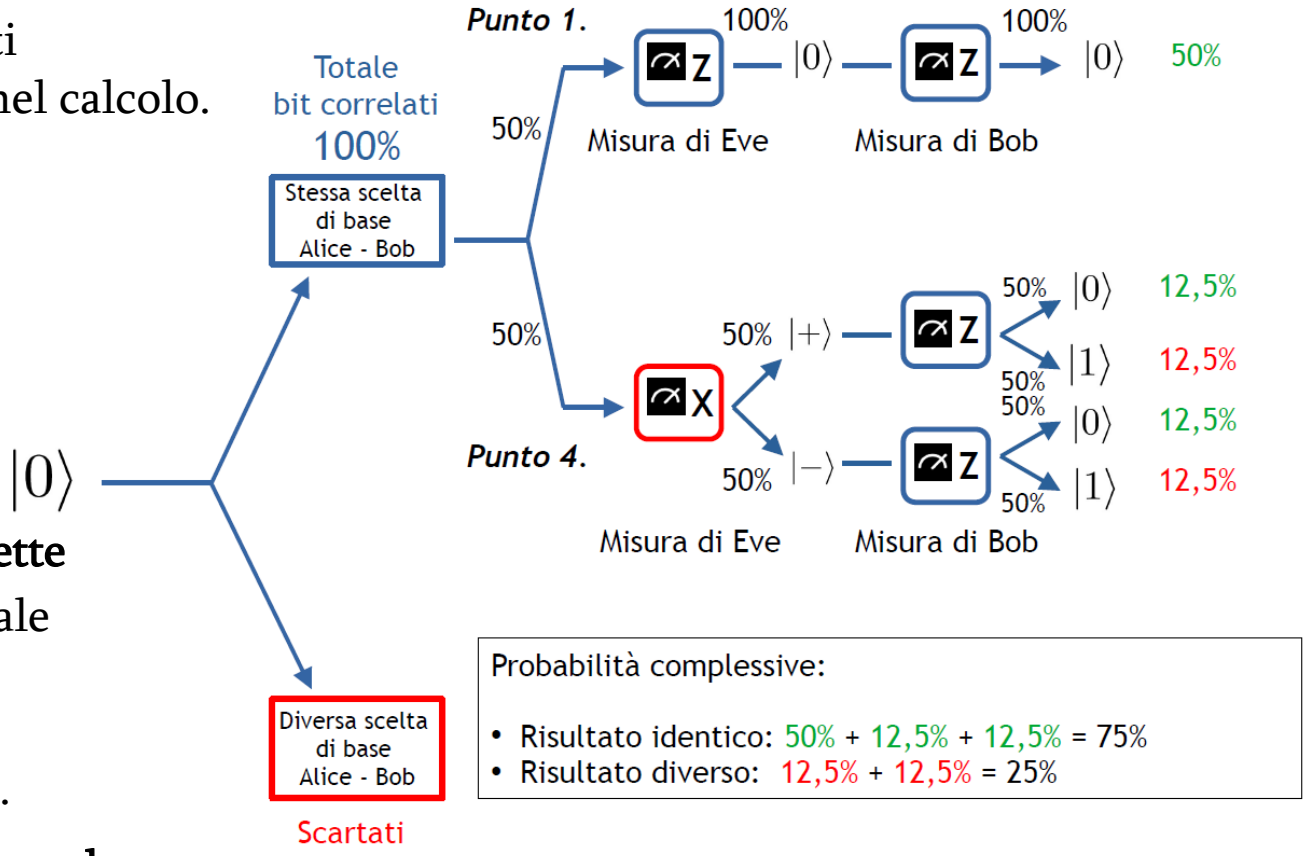
- estrae una stringa di N bit (stringa delle basi);
- misura Z se il bit estratto è 0, misura X se il bit estratto è 1.

Tuttavia, nel caso in cui Alice e Bob effettuano la **stessa scelta di base**, ma Eve misura l'osservabile associata **all'altra base di stati**, tale misura cambia lo stato del qubit e introduce una perturbazione nei risultati di Bob.
Esempio: se Alice prepara il qubit nello stato $|0\rangle$, ma Eve misura l'osservabile X, lo stato del qubit collassa in un autostato di X (per esempio $|+\rangle$). Se Bob misura Z, dunque, il risultato è **probabilistico**; pertanto, potrà **essere diverso dal bit codificato da Alice.** Eseguendo quindi il confronto finale, **l'hacker viene individuato**.

Variazione: Eve può sviluppare un attacco hacker basato sull'Entanglement. Tuttavia viene comunque individuata.

Il protocollo BB84 – Attese teoriche

Per il calcolo delle seguenti percentuali, sono considerati soltanto i *bit correlati*. I risultati scartati non rientrano nel calcolo.



1. Scenario 1 – assenza dell'hacker

In questo caso, ci si aspetta che **tutti i risultati** di Bob non scartati siano **identici** ai bit codificati da Alice.

$P_{identici} = 100\%$

$P_{diversi} = 0\%$

$|0\rangle$

2. Scenario 2 – attacco hacker basato su misurazioni dirette

Se l'hacker ha intercettato i qubit trasmessi, la percentuale di risultati **diversi** dai bit codificati **non è nulla**:

$P_{identici} = 75\%$

$P_{diversi} = 25\%$ Per il calcolo esplicito, si veda lo schema.

3. Scenario 3 – variazione: attacco hacker basato sull'Entanglement

L'hacker può creare stati entangled tra i qubit trasmessi e altri qubit in suo possesso. Tuttavia, il fatto di interagire in questo modo con i qubit trasmessi introduce comunque una perturbazione nei risultati di Bob.
Le percentuali considerate sono, come nello Scenario 2, pari a $P_{identici} = 75\%$ e $P_{diversi} = 25\%$.

Il fatto che $P_{diversi}$ **non sia uguale a zero** è dunque indice della presenza dell'hacker.

Il protocollo BB84 – Codice

Tramite Qiskit è possibile scrivere un codice Python che implementi il protocollo BB84.

Il programma è così strutturato:

1. Vengono estratte 2 stringhe di bit pseudocasuali: la prima contiene i bit da codificare, la seconda è la stringa delle basi di Alice.
2. Viene creato un circuito quantistico, in cui ogni bit è preparato in uno tra gli stati $|0\rangle, |1\rangle, |+\rangle$ o $|-\rangle$ attraverso l'uso di opportune porte quantistiche, secondo lo schema del protocollo. In particolare:

- Per preparare un qubit nello stato $|0\rangle$, non viene inserita alcuna porta;
- Per preparare lo stato $|1\rangle$ viene inserita la porta *X;*
- Per preparare lo stato $|+\rangle$ viene applicata la porta *H;*
- Per preparare lo stato $|-\rangle$ vengono inserite in successione le porte *X* ed *H*.

3. A seconda dello scenario considerato:

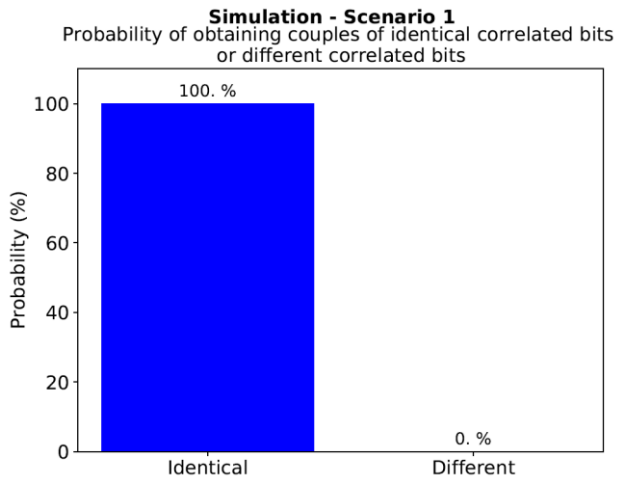
- (Scenario 2) viene introdotta la misura diretta di Eve;
- (Scenario 3) viene applicata la porta *CNOT* ad un qubit posseduto da Eve (target) e ad un qubit trasmesso (control). Il qubit di Eve è sempre preparato nello stato $|0\rangle$. Pertanto, viene creato uno stato entangled solamente se il control qubit si trova in uno stato della base $\{|+\rangle, |-\rangle\}$.

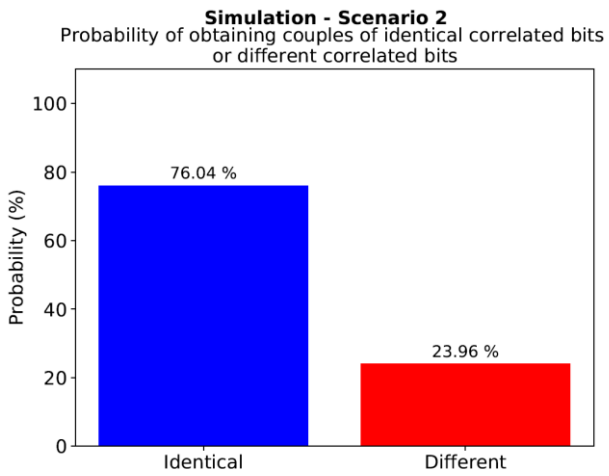
4. Infine viene implementata la misura di Bob. In generale, per misurare l'osservabile *Z*, nel circuito viene inserita un'operazione di misura standard. Se invece deve essere misurata l'osservabile *X*, viene inserita una porta *H* e, in successione, un'operazione di misura di *Z*.

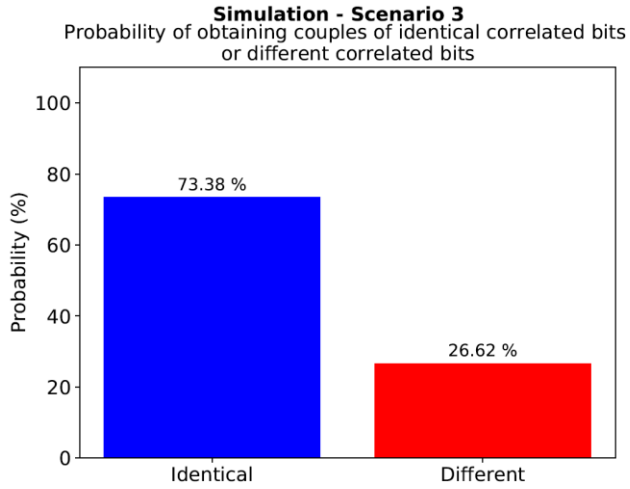
Tramite un altro programma si effettua infine il confronto tra i bit di Alice e Bob e l'analisi delle percentuali.

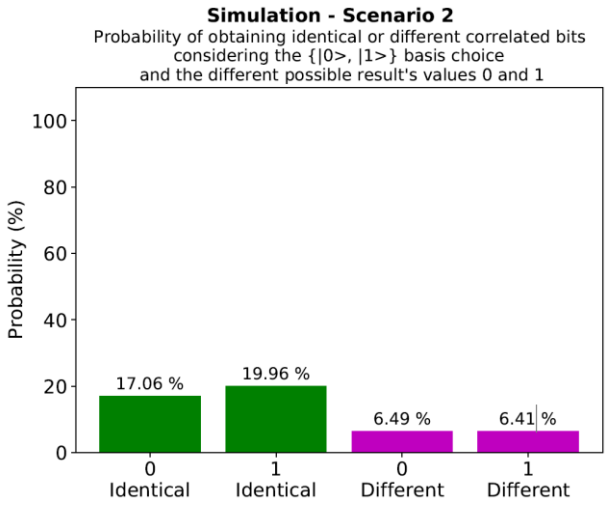
Simulazione del protocollo

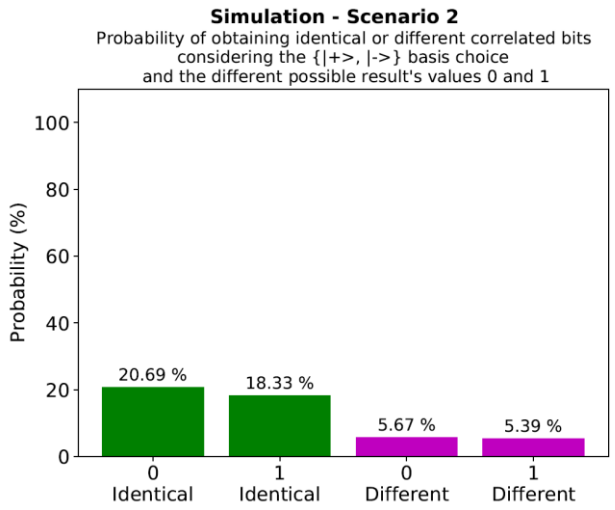
Il programma è stato eseguito su un simulatore di IBM per verificarne il corretto funzionamento.
I risultati sono riportati nei seguenti grafici:

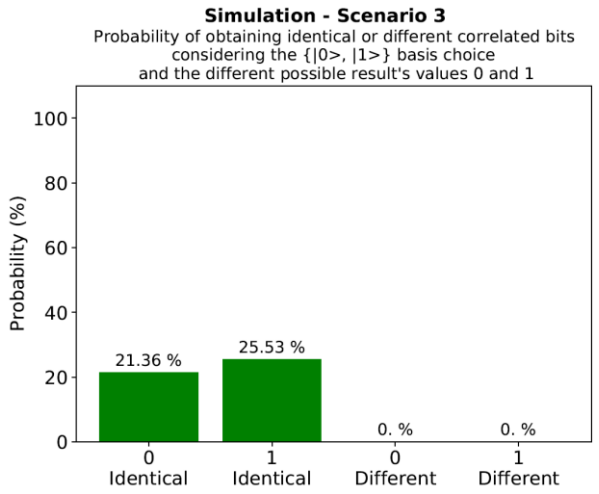


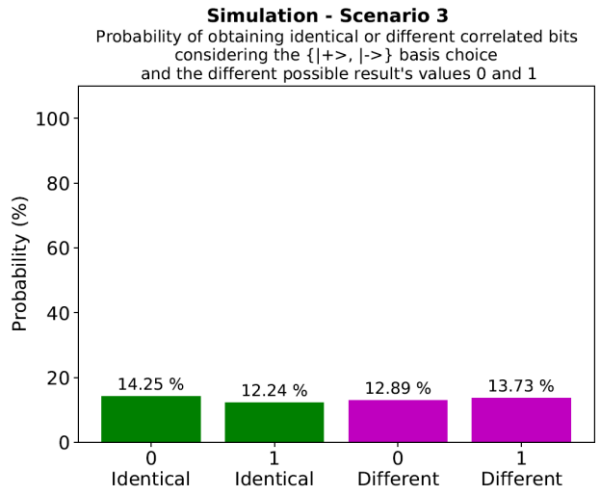












Esperimento su computer quantistico – scelta dei dispositivi e rumore

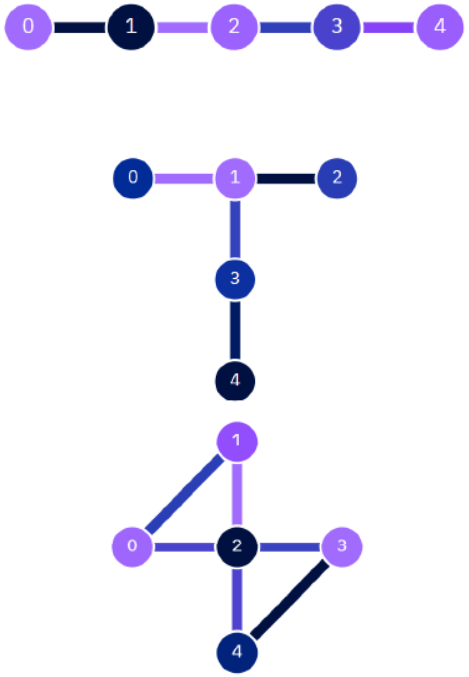
Sono state prese in considerazione le diverse caratteristiche dei dispositivi di IBM per scegliere quali di essi utilizzare.
I dispositivi scelti sono: **ibmq_santiago**, **ibmq_vigo** e **ibmq_5_yorktown**.

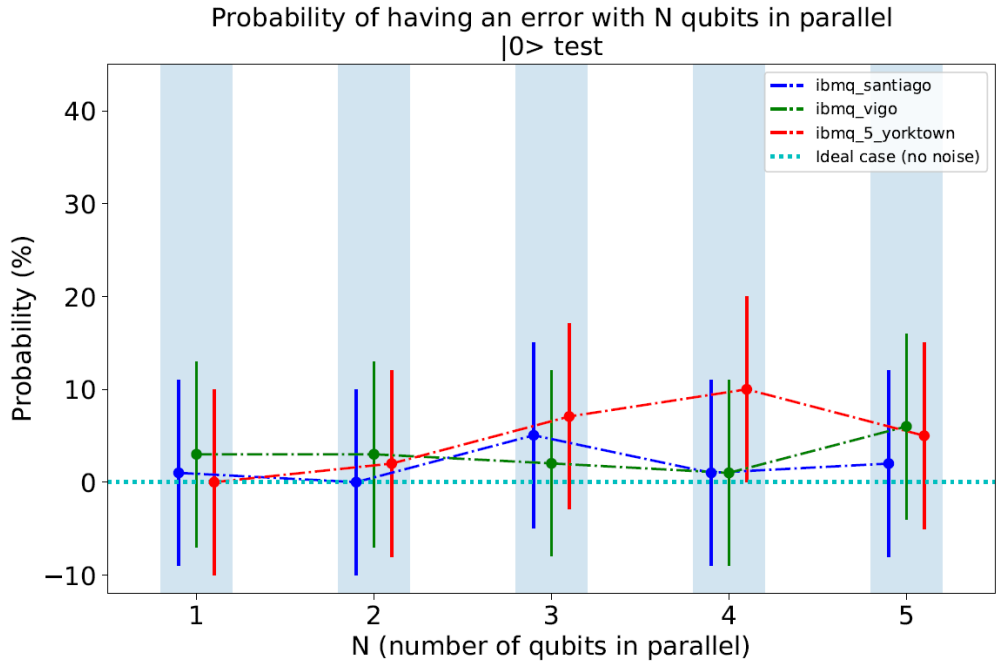
I computer quantistici non sono dispositivi ideali, ma hanno un rumore di fondo. Questo comporta la possibilità di errore nelle misurazioni.
Ogni dispositivo considerato può realizzare circuiti quantistici con un massimo di 5 qubit in parallelo.
Pertanto, sono stati effettuati semplici test per verificare che il rumore non dipenda dal numero di qubit in parallelo.

ibmq_santiago	
Numero di qubit	5
Quantum volume	32
Average Readout Error	$1,952 \cdot 10^{-2}$
Average $CNOT$ Error	$1,027 \cdot 10^{-2}$

ibmq_vigo	
Numero di qubit	5
Quantum volume	16
Average Readout Error	$3,340 \cdot 10^{-2}$
Average $CNOT$ Error	$8,757 \cdot 10^{-3}$

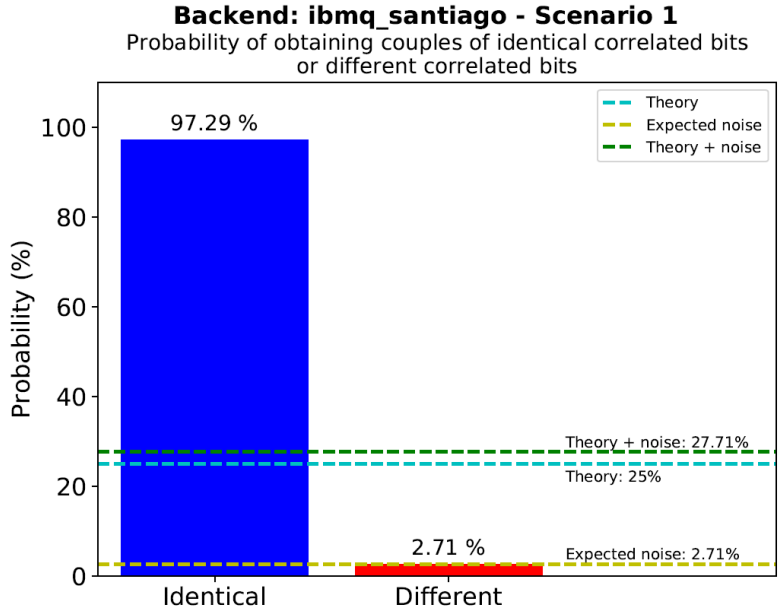
ibmq_5_yorktown	
Numero di qubit	5
Quantum volume	8
Average Readout Error	$4,620 \cdot 10^{-2}$
Average $CNOT$ Error	$1,626 \cdot 10^{-2}$

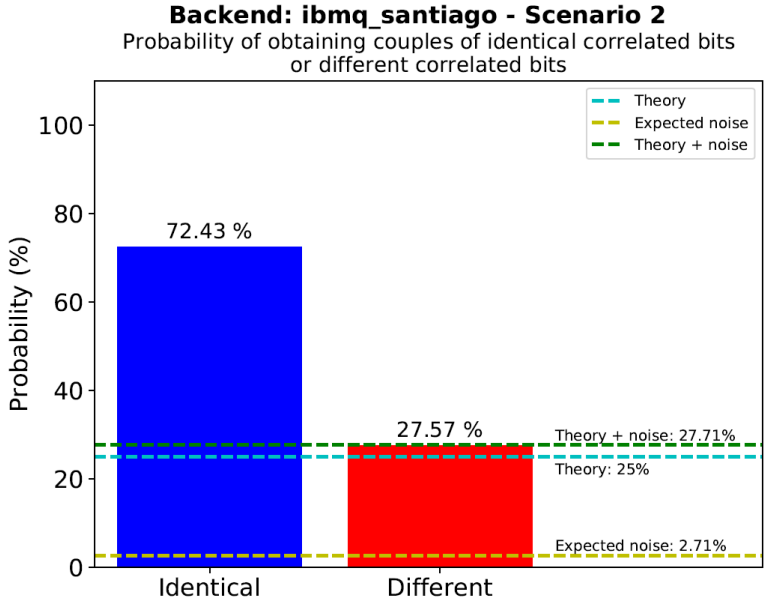


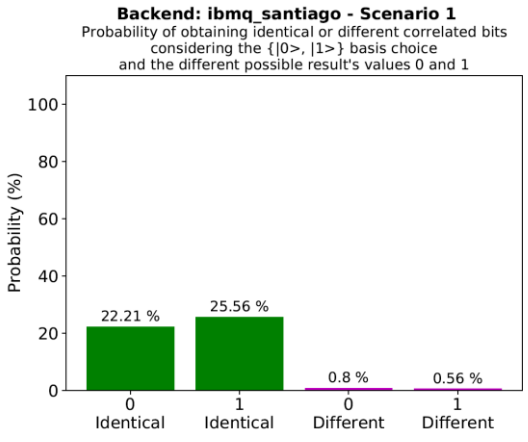


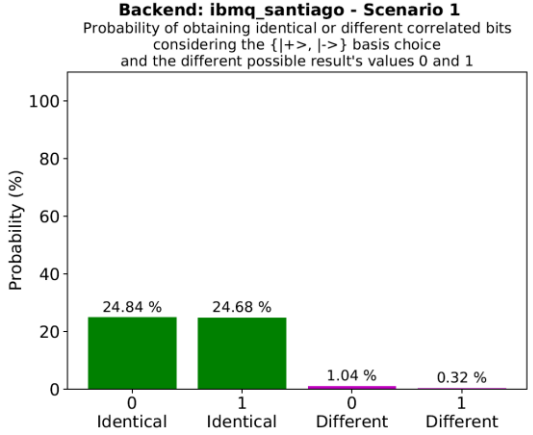
Esperimento su computer quantistico – Scenario 1 e Scenario 2

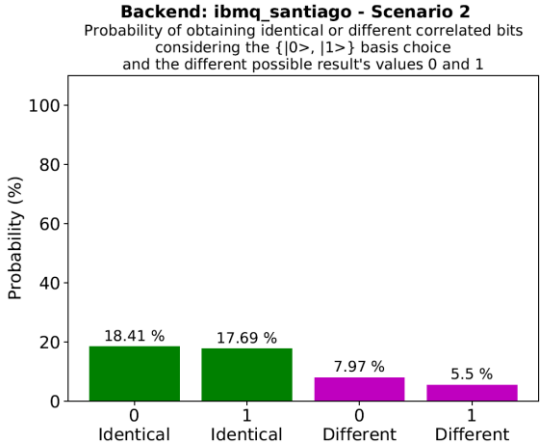
La **soglia di rumore** è stata considerata pari alla **percentuale di bit correlati diversi nello Scenario 1.**
L'obiettivo è verificare che, in presenza dell'hacker, questa percentuale sia superiore alla soglia di rumore.

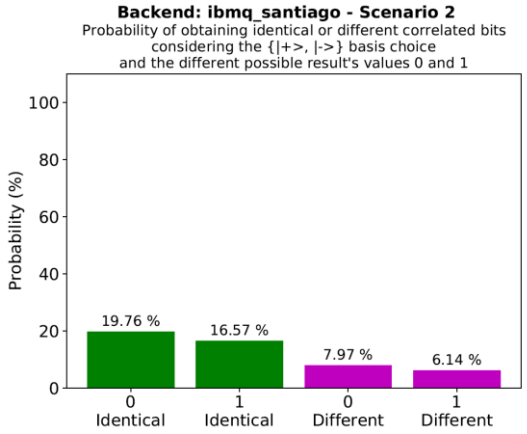






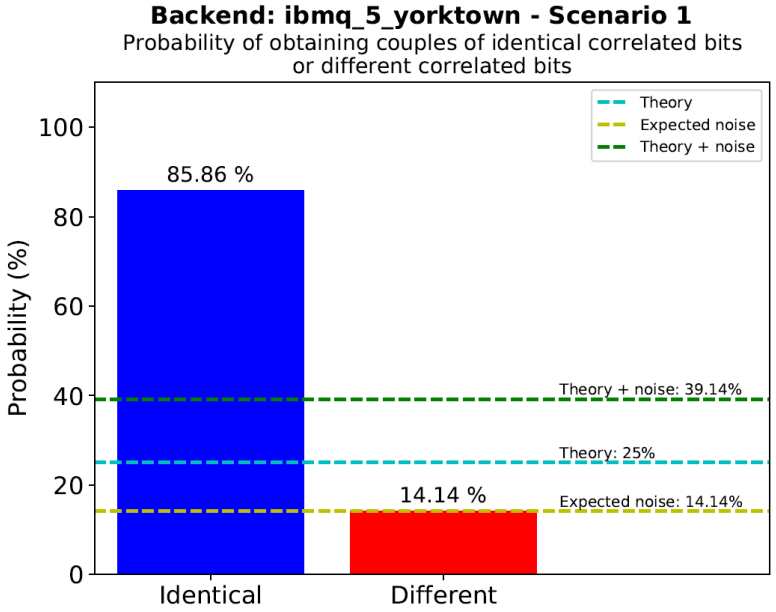


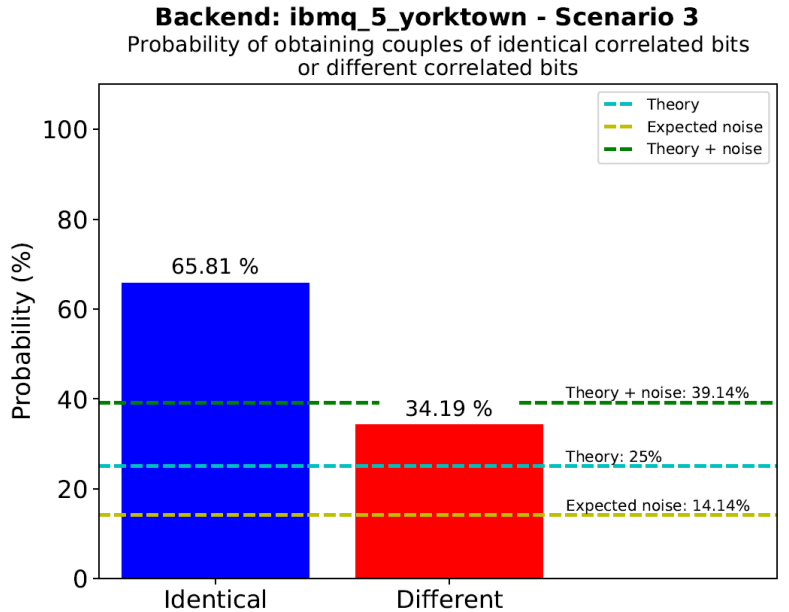


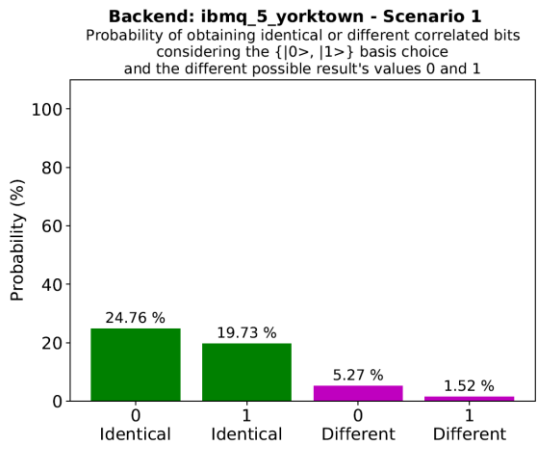


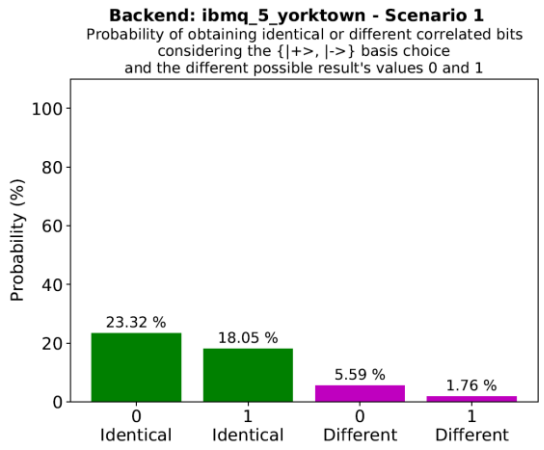
Esperimento su computer quantistico – Scenario 1 e Scenario 3

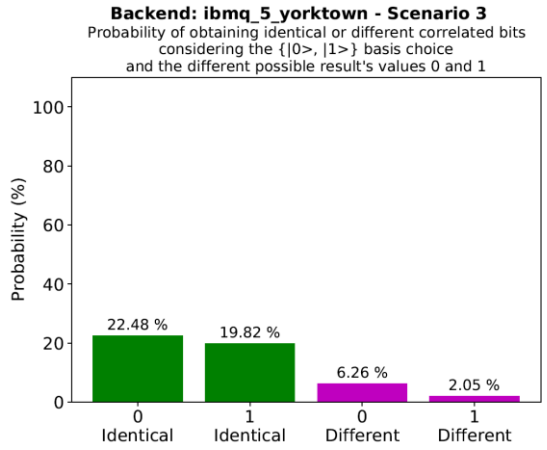
Per lo Scenario 3 è stato usato solamente il dispositivo ibmq_5_yorktown.

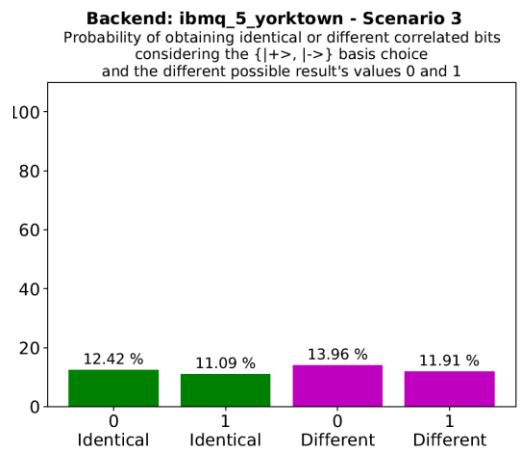












Conclusioni

Il protocollo BB84:

- rende sicura la condivisione di una chiave crittografica privata;
- permette l'individuazione di qualsiasi tipo di intromissione da parte di un hacker;
- non è basato sulla complessità della codifica, ma sulle proprietà di particolari fenomeni quantistici.

In particolare:

- in assenza dell'hacker, la percentuale di bit correlati diversi è **nulla**;
- in presenza dell'hacker, sia in caso di un attacco basato sulle misurazioni dirette dei qubit intercettati, sia in caso di attacco basato sull'Entanglement, la percentuale di bit correlati diversi è pari a circa **25%**.

In un dispositivo reale, il rumore introduce una percentuale di bit correlati diversi non nulla anche in assenza dell'hacker. Tuttavia, la percentuale di bit diversi introdotta dall'hacker è indipendente dal rumore, e quindi si somma ad esso.

La soglia di riferimento è fissata pari al rumore misurato in assenza dell'hacker.
Grazie all'esperimento sui computer quantistici si è mostrato che, in caso di attacco hacker, la percentuale di bit diversi supera abbondantemente tale soglia.

L'hacker viene quindi sempre individuato, ed il protocollo BB84 risulta essere completamente sicuro.