Università degli Studi di Milano - Bicocca

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea Triennale in Fisica

Tesi di Laurea Triennale

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

Candidato:
Davide Rinaldi
Matricola 826346

Relatori:
Dr. Andrea Giachero
Prof. Paolo Solinas

Correlatrice:
Dr. Elena Ferri

Anno Accademico 2020-2021

E se sapessi tutte lingue, tutte le scienze e le scritture
Se sapessi profetare e rivelare le cose future
E i segreti delle coscienze e degli animi
Non in ciò è perfetta letizia

Alla mia famiglia

Sommario

Ogni giorno, le sfide che la Scienza deve affrontare sono sempre più complesse. Vari ambiti di ricerca, anche molto diversi tra loro, condividono la stessa necessità: sviluppare un metodo computazionalmente efficiente che permetta di trovare la soluzione di problemi che, fino ad ora, sono stati considerati irrisolvibili, a causa dell'insufficiente capacità di calcolo dei computer *classici* usati normalmente. La modellizzazione dei fenomeni naturali, la simulazione degli stati di atomi e di molecole, lo sviluppo di tecniche di machine learning, la risoluzione di sistemi di equazioni differenziali, l'implementazione dei modelli finanziari e di quelli decisionali, lo studio di nuovi algoritmi di ottimizzazione, l'invenzione di nuovi protocolli per la sicurezza dei dati informatici, sono solo alcuni tra i problemi che, spesso, *non* possono essere risolti in modo efficiente da un computer *classico*.
In questo contesto si inserisce il *Quantum Computing*: una tecnica di computazione basata sulle leggi della Meccanica Quantistica, del tutto diversa dalla computazione classica. In particolare, il Quantum Computing sfrutta tre fondamentali fenomeni della fisica quantistica: la *sovrapposizione degli stati fisici*, l'*Entanglement* e l'*interferenza* tra le funzioni d'onda.
Il Quantum Computing può essere sviluppato secondo diversi modelli, tutti equivalenti tra loro. Il più diffuso è il modello dei *circuiti quantistici*, basato sui *qubit*. Un circuito quantistico è una sequenza di *quantum gate*, analoghe alle porte logiche della computazione classica, ma il cui effetto è quello di determinare trasformazioni reversibili sui qubit del circuito.
Altri modelli di Quantum Computing sono per esempio la Macchina di Turing Quantistica (una generalizzazione di quella classica), il Quantum Computing Adiabatico, ed il Quantum Computing basato sulla misura di un sistema di qubit entangled.
Un computer che permette l'utilizzo del Quantum Computing è detto *computer quantistico*.
Attualmente, varie aziende hanno costruito (e stanno continuamente sviluppando) i propri computer quantistici: fra di esse vi sono IBM, Google, Microsoft, D-Wave, Honeywell e Intel. Anche le tecnologie usate per implementare i dispositivi sono di diverso tipo: i qubit di un computer quantistico sono spesso creati usando *sistemi superconduttivi*, *ioni intrappolati* o *quantum dots*. In generale, tuttavia, un qubit è dato da un generico sistema fisico a due livelli, come per esempio la polarizzazione o lo spin di un fotone, oppure lo spin elettronico o quello nucleare.

Nell'ambito della sicurezza dei dati informatici rientra la *crittografia quantistica*, e questo progetto di tesi è focalizzato su di essa. In particolare, nella tesi viene studiato il *protocollo BB84*. Proposto da C. H. Bennett e G. Brassard nel 1984, esso è il primo protocollo ideato per permettere lo scambio sicuro di chiavi crittografiche attraverso l'uso del Quantum Computing.

Tale protocollo consiste nella trasmissione di qubit preparati inizialmente in quattro possibili stati fisici diversi, corrispondenti ai vettori di stato descritti da due basi ortonormali tra loro. Per esempio, considerando un qubit ottenuto dalla polarizzazione di un fotone, una delle due basi può essere data dai due stati *polarizzazione verticale* e *polarizzazione orizzontale* del fotone; la base ortonormale a questa, dunque, potrebbe essere una coppia di stati aventi una *polarizzazione trasversa*, o intermedia. Ad ogni base è associato un operatore, di cui gli stati della base sono autostati. Ad esempio, gli stati della prima base sono autostati del generico operatore A, mentre quelli della seconda sono autostati del generico operatore B.
La scelta di preparare un qubit in un certo stato è effettuata in modo casuale, con probabilità uniforme associata ad ognuno dei quattro possibili stati. I qubit vengono poi inviati ad un destinatario, che esegue su ogni qubit la misura dell'operatore A o dell'operatore B, operando in modo casuale la scelta dell'operatore da utilizzare.
Per attuare tali scelte, vengono estratte tre stringhe random di bit. Una di esse sarà la stringa dei bit da codificare (ovvero i bit che comporranno la chiave crittografica); una stringa servirà al mittente per scegliere la base di autostati con cui preparare il qubit relativo ad un certo bit codificato, mentre l'altra stringa servirà al destinatario per la scelta dell'operatore con cui misurare il qubit. Ogni misura restituisce un risultato, che può corrispondere ai due possibili valori di un bit: 0 oppure 1.
A fine trasmissione, il mittente ed il destinatario rendono pubbliche le stringhe relative alla scelta delle basi di autostati e a quella degli operatori per le misure. Inoltre *il mittente rende pubblica la prima metà dei bit che ha codificato*.
Viene quindi effettuata l'analisi della prima metà dei dati raccolti, confrontando i risultati delle misure del destinatario con i bit codificati resi pubblici dal mittente. Si conta quante volte il destinatario

ottiene lo stesso valore del rispettivo bit codificato dal mittente, nel solo caso in cui il mittente abbia preparato il qubit in un autostato dell'operatore scelto dal destinatario per fare la misura. In questo caso, si dice che il bit codificato ed il bit corrispondente al risultato della misura sono *correlati*.
Questa analisi a posteriori permette di verificare la sicurezza della trasmissione e di ottenere una chiave crittografica condivisa solamente dal mittente e dal destinatario. Nel caso di trasmissione ideale (senza rumore), infatti, i risultati delle misure dovrebbero essere *sempre* uguali ai relativi bit codificati; la presenza di risultati *diversi*, pertanto, significherebbe che la chiave è stata intercettata da un hacker, e che pertanto il protocollo deve essere interrotto.
Nella tesi viene discussa la teoria a fondamento del protocollo e, sulla base di semplici considerazioni sulle probabilità che entrano in gioco, vengono calcolati i risultati teorici attesi dall'analisi dati a posteriori. Viene introdotto anche l'intervento di un hacker, attraverso l'implementazione di due possibili tipi di attacco: uno basato sulla misurazione diretta dei qubit, e l'altro sulla creazione di stati entangled con i qubit trasmessi. Grazie all'analisi dei dati raccolti dal destinatario, si dimostra che nessuno di questi due attacchi può riuscire, e che il protocollo è del tutto sicuro rispetto a tentativi di intromissione da parte di un agente esterno di questo tipo. Entrambi i tentativi di hackeraggio, infatti, causano il cambiamento dello stato di alcuni tra i qubit trasmessi; come conseguenza osservabile si ha che, analizzando i dati, la percentuale dei risultati diversi risulta pari al 25% dei bit *correlati*.
Il protocollo viene quindi implementato mediante la scrittura di un programma Python che sfrutta *Qiskit*, un framework open-source sviluppato da IBM che permette di scrivere codice per il Quantum Computing.
Si effettua poi una simulazione del codice per provarne il corretto funzionamento e per verificare l'esattezza dei risultati previsti dall'analisi dati. Infine, si esegue l'esperimento sui computer quantistici reali. Sono lanciati dei test preliminari per determinare una buona scelta dei dispositivi messi a disposizione da IBM e per scegliere la configurazione ottimale del circuito quantistico relativo ad ogni esecuzione. Tali test mostrano l'andamento dell'errore sulle misure al variare del numero di qubit utilizzati in parallelo in ogni circuito, relativamente ai diversi dispositivi presi in considerazione. Una volta operata la scelta del dispositivo, si procede con l'esperimento e con l'analisi dati dei risultati, tenendo conto del rumore introdotto dal dispositivo. Tale rumore viene identificato con la percentuale di risultati diversi in assenza dell'hacker (ovvero nel caso in cui, se il dispositivo fosse ideale, tale percentuale sarebbe pari a 0%).
Come previsto, in caso di attacco hacker si ha un aumento della percentuale di risultati diversi, che supera la soglia del rumore presente in assenza dell'hacker; in particolare, la percentuale di bit correlati diversi è circa pari a 25% più la percentuale associata al rumore.

La tesi si divide in 5 capitoli:

1. Nel primo, vengono introdotti i computer quantistici ed i concetti base di Quantum Computing utilizzati nella tesi.

2. Nel secondo capitolo viene presentato il protocollo BB84, i fenomeni quantistici su cui si fonda e il suo funzionamento; si discute inoltre sulla sicurezza di un protocollo crittografico e sulla differenza tra crittografia classica e crittografia quantistica.

3. Nel terzo capitolo viene illustrata una possibile simulazione del protocollo. Viene descritto il codice usato per l'implementazione della simulazione e sono discusse le attese teoriche dei risultati della simulazione, sia in caso di assenza dell'hacker che in caso di attacco. Vengono riportati i risultati della simulazione.

4. Nel quarto capitolo si discute l'esperimento effettuato sui sistemi quantistici reali di IBM, con cui viene messo alla prova il protocollo BB84. Vengono riportati e commentati i risultati dell'esperimento.

5. L'ultimo capitolo è dedicato alle conclusioni.

Indice

Capitolo 1

I computer quantistici

How can we simulate the quantum mechanics? [...]
Can you do it with a new kind of computer - a quantum computer?

Con queste due domande, poste da Richard P. Feynman nel 1981 durante una conferenza al California Institute of Technology [9], viene lanciata al mondo scientifico una sfida a dir poco audace: riuscire a simulare la Natura attraverso strumenti che, intrinsicamente, si comportano come la Natura stessa.
Non si tratta banalmente di *approssimare* il comportamento di fenomeni fisici complessi: questo lo si può fare anche con i normali computer di tutti i giorni (i computer *classici*, cioè basati sulla logica computazionale tradizionale, i cui elementi fondamentali sono i *bit*). Un'approssimazione viene operata facendo ipotesi semplificative sul problema in questione, in modo da renderlo più facilmente approcciabile con simulazioni numeriche: un esempio può essere la scelta di un potenziale opportuno nella descrizione di un atomo a molti elettroni. Spesso in questo modo si riescono a ricavare comunque nuove informazioni sullo stato del sistema (come può essere l'energia di stato fondamentale), ma si tratta pur sempre di un'approssimazione: il sistema fisico reale è molto più complesso di quello descritto dal modello approssimato.

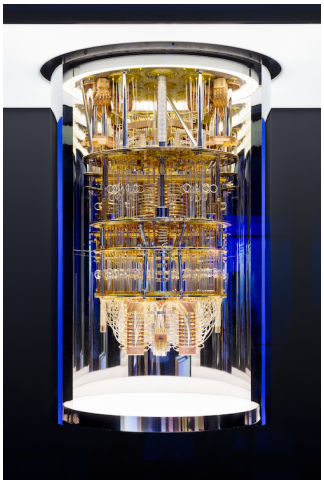


Figura 1.1:
Computer quantistico IBM

Tutt'altra cosa è simulare *esattamente* i fenomeni naturali. Ora l'obiettivo è realizzare un dispositivo che si comporti *esattamente* come si comporta la Natura. Ma la Natura sembra essere essenzialmente *quantistica*, e la Meccanica Quantistica coinvolge un oggetto matematico che pare essere l'unico strumento che permetta di effettuare un qualche tipo di predizione sul mondo fisico: la *probabilità*.
Questo tuttavia sembra un controsenso: com'è possibile predire il risultato di un esperimento che, per sua stessa natura, è *probabilistico*? In effetti, a meno che la probabilità di ottenere un preciso risultato non sia pari a 1, *non è possibile sapere con certezza a priori quale sia il valore del risultato.*
Quello che la Meccanica Quantistica permette di fare è, invece, *predire con quale probabilità il risultato avrà un certo valore.* È la probabilità la chiave di tutto.
Un dispositivo che si comporti come il mondo fisico, pertanto, non può basarsi sulla logica deterministica dei computer classici. Deve essere, al contrario, un computer probabilistico. Questo computer, pur simulando esattamente un certo processo fisico, non otterrà necessariamente lo stesso risultato del fenomeno considerato: infatti il risultato è probabilistico, e come tale non può essere predetto con certezza. La simulazione dunque è *esatta*, ma *non* nel senso che il computer riproduce esattamente un preciso fenomeno, ottenendo il suo stesso risultato; è esatta nel senso che lo riproduce *esattamente con le stesse probabilità associate ai diversi risultati possibili.*
Ripetendo la simulazione svariate volte, si otterranno i diversi risultati, ognuno con una certa frequenza: usando queste frequenze è possibile calcolare la probabilità relativa ad ogni risultato, con una certa accuratezza. Quello che questi dispositivi consentono di trovare, pertanto, è la probabilità con cui si verificano gli eventi legati ad un certo fenomeno fisico, ovvero ciò che conta in Meccanica Quantistica.
È da questa idea che nascono i computer quantistici.

1.1 Breve linea temporale

Il primo a proporre un modello di computer quantomeccanico è Paul Benioff, con un articolo pubblicato nel 1980 [4]. Lo studio del Quantum Computing culmina nel 1998 con la prima dimostrazione sperimentale di un algoritmo quantistico: viene usato un computer quantistico a 2 qubit, prima da Jonathan A. Jones e Michele Mosca alla Oxford University, e poco dopo da Isaac L. Chuang all'IBM's Almaden Research Center [6]. Successivamente, la ricerca nel campo dei computer quantistici porta ad un numero sempre maggiore di nuove scoperte ed alla realizzazione di sistemi quantistici sempre più complessi e potenti. Nel 2016, IBM rende pubblica la prima piattaforma *cloud* adibita al Quantum Computing, ovvero *IBM Quantum Experience*, che permette a chiunque di collegarsi da remoto ai computer quantistici di IBM. Nel 2019 viene annunciato il raggiungimento da parte di Google della *supremazia quantistica* [3] (la dimostrazione che un sistema quantistico programmabile possa risolvere un problema la cui soluzione non possa essere trovata, in un tempo ragionevole, da alcun computer classico). Nello stesso anno, D-Wave rende noto il futuro rilascio, nel 2020, di un computer quantistico a 5000 qubit commerciale [16]. A settembre 2020, la multinazionale Honeywell ottiene un *quantum volume* (un parametro che descrive le prestazioni di un computer quantistico) pari a 128 sul suo computer quantistico [15].

1.2 Tecnologia

L'elemento fondamentale per realizzare un qubit è rappresentato dalla *giunzione Josephson*, ovvero una giunzione costituita da due o più superconduttori separati da una barriera di materiale isolante. In un superconduttore, i portatori di carica non sono i singoli elettroni, ma coppie di elettroni (dette *coppie di Cooper*). A temperature molto basse, gli effetti quantistici nei circuiti diventano rilevanti; per effetto tunnel, si forma una corrente non dissipativa che fluisce attraverso la giunzione Josephson. Le coppie di Cooper hanno spin intero, e pertanto sono bosoni; a temperature basse, diverse coppie possono quindi occupare lo stesso livello energetico, formando il *condensato di Bose-Einstein*. Per descrivere gli elementi circuitali si utilizza quindi la funzione d'onda dello stato condensato, al posto delle grandezze macroscopiche tipiche dei circuiti classici (tensione e corrente). La corrente che fluisce attraverso le giunzioni Josephson è usata per creare induttanze non-lineari, funzionali all'implementazione dei qubit. Esse consentono infatti di realizzare un oscillatore quantistico anarmonico; ciò fa sì che sia possibile considerare solo due stati dell'oscillatore (quelli che andranno a definire il qubit), e non tutti gli stati possibili, come nel caso dell'oscillatore armonico. La frequenza delle oscillazioni che entrano in gioco varia nello spettro delle radiofrequenze (da 20 kHz a 300 GHz).
I computer quantistici di IBM fanno uso di un particolare tipo di qubit, ovvero il *trasmon* qubit. *Trasmon* sta per *transmission line shunted plasma oscillation*. Il trasmon qubit è un particolare tipo di *charge qubit*, ovvero un qubit in cui i diversi stati fisici sono dati dalla presenza o dall'assenza di un eccesso di carica elettrica.
I processori quantistici sono mantenuti a bassissima temperatura, fino a 10 mK, attraverso opportuni apparati detti *criostati*. Tali temperature sono necessarie per ridurre al minimo i disturbi (*decoerenza*).

Tutti i computer quantistici (non soltanto quelli di IBM) necessitano di interfacciarsi con dei computer classici affinché sia possibile l'elaborazione dei dati. Infatti, l'unico modo per ottenere un output da un computer quantistico è fare una misura sullo stato del sistema quantistico stesso. Tale misura restituirà, in generale, un output *classico*, cioè un bit; pertanto, servirà un computer classico che possa analizzare ed elaborare tale risultato.

1.2.1 IBM Quantum Experience

IBM Quantum Experience [17] è una piattaforma online con cui è possibile collegarsi via cloud ai computer quantistici messi a disposizione da IBM. L'iniziativa ha scopo prevalentemente didattico (la versione *open* è accessibile a chiunque si registri sul sito), ma anche di sviluppo e ricerca (per cui è

disponibile la versione *premium*). Attraverso IBM Quantum Experience è possibile eseguire algoritmi quantistici, effettuare simulazioni che coinvolgono computer sia classici che quantistici, e lanciare veri e propri esperimenti sui dispositivi. Vi è la possibilità di scrivere codice direttamente nel cloud, oppure di eseguire programmi da remoto. Sul sito sono disponibili diversi tutorial, nonché il libro di testo *Qiskit Textbook* [18], che fornisce lezioni di Quantum Computing e introduce all'uso di *Qiskit*, un framework opensource che, sfruttando un interfaccia di alto livello in linguaggio Python, permette l'interazione con i dispositivi quantistici.

1.2.2 Quantum Computing con Qiskit

La modalità con cui è possibile sperimentare tecniche di Quantum Computing sulla piattaforma è l'implementazione di *circuiti quantistici*. Qiskit permette di realizzare programmi con i quali è possibile interagire con i computer quantistici IBM; tali programmi possono poi essere eseguiti su simulatori o direttamente sui sistemi quantistici. Esistono quattro diverse partizioni di Qiskit:

- *Qiskit Terra*, per costruire i circuiti quantistici scrivendo codice in un linguaggio vicino al *linguaggio della macchina quantistica*, ovvero inserendo esplicitamente le *quantum gate* nel circuito;
- *Qiskit Aer*, per utilizzare i simulatori classici di piccoli sistemi quantistici;
- *Qiskit Aqua*, che offre tools per l'applicazione del Quantum Computing a diversi ambiti (chimica, finanza, intelligenza artificiale);
- *Qiskit Ignis*, che fornisce tools per la caratterizzazione del rumore nei dispositivi.

1.3 Quantum Computing: concetti base

In questa sezione verranno esposte alcune nozioni basilari di Meccanica Quantistica e di Quantum Computing; in particolare, verranno presentati i concetti di Quantum Computing utilizzati nei capitoli successivi della tesi.

I passaggi fondamentali nell'implementazione di un circuito quantistico sono tre: l'inizializzazione dei qubit, la realizzazione dell'algoritmo attraverso le *quantum gate* e la misura dei qubit.

1.3.1 Il qubit

Un qubit è un *sistema fisico a due livelli*, soggetto alle leggi della Meccanica Quantistica. È l'unità minima di informazione quantistica.
Un *bit classico* è una variabile che può assumere due soli valori: 0 oppure 1. Anche il qubit può restituire in output due soli valori (siano essi 0 oppure 1, *on* oppure *off*, *spin* $\frac{1}{2}$ oppure *spin* $-\frac{1}{2}$, e così via). Qual è la differenza tra bit e qubit, allora?
La differenza sta nel fatto che il bit si trova sempre in uno *stato ben definito*: se viene preparato con il valore 0, ad esempio, qualsiasi lettura di questo bit da parte di un computer restituirà sempre il valore 0. Nel caso del bit, il suo stato coincide con il suo *valore*.
Non è così per il qubit: lo *stato* del qubit può trovarsi in uno stato che è *sovrapposizione* di stati diversi. In altri termini, i risultati di una misura sul sistema non saranno associati alla distribuzione di probabilità descritta da uno solo degli stati sovrapposti, bensì ad una nuova distribuzione di probabilità, data dallo stato *sovrapposizione* (per cosa si intenda più precisamente con *misura*, si veda la sezione ad essa dedicata).
Ad esempio: sia $|\psi\rangle$ lo stato di un qubit. Se si definisce lo stato $|\psi\rangle = |0\rangle$ come lo stato per cui una certa misurazione del qubit restituisce *con certezza* (cioè *con probabilità pari a 1*) il valore 0, e allo stesso modo si definisce $|\psi\rangle = |1\rangle$ come lo stato per cui la stessa misura dà il risultato 1 con certezza, allora è possibile considerare lo stato sovrapposizione:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{1.1}$$

dove α e β sono numeri complessi tali che $|\alpha|^2 + |\beta|^2 = 1$.
Per questo stato, una misura potrebbe dare come risultato sia il valore 0 che il valore 1.
I valori che il qubit può *assumere* dunque sono i risultati di tale misura, e sono gli stessi del bit classico: 0 o 1. Tuttavia il risultato di una misura sul qubit potrà dare in output il valore 0 *con una certa probabilità*, ed il valore 1 *con una certa probabilità*.

È utile soffermarsi sulla notazione utilizzata, per poter specificare adeguatamente cosa si intende con *stato* e con *sovrapposizione*.
Il simbolo $|\psi\rangle$ (notazione di Dirac, o *notazione bra - ket*) indica un *vettore di stato* (in uno *spazio di Hilbert*). In generale, un vettore di stato è descritto dalla *funzione d'onda* del relativo sistema quantistico. In questo caso, tuttavia, si può rappresentare $|\psi\rangle$ come il vettore delle probabilità associate ai diversi possibili risultati di una misura (questo perché si sta considerando un set *discreto* di possibili risultati, ovvero solamente 0 o 1). Di fatto, ogni componente di tale vettore è un valore di probabilità associato ad un possibile risultato.
Si definisce la *base computazionale* o *canonica* come la base di vettori di stato data da:

$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{1.2}$$

Come si può notare, sono vettori bidimensionali: infatti i possibili risultati della misura sono 2 (per definizione di qubit). Inoltre, sono ortonormali: $\langle 0|0\rangle = \langle 1|1\rangle = 1$, mentre $\langle 0|1\rangle = \langle 1|0\rangle = 0$.
Uno stato di sovrapposizione è dato da una combinazione lineare di vettori di stato. In particolare, un qubit si trova in uno stato di sovrapposizione se il suo vettore di stato si può scrivere come:

$$|\psi\rangle = a \cdot |\alpha\rangle + b \cdot |\beta\rangle \tag{1.3}$$

dove a, b sono coefficienti complessi tali che $|a|^2 + |b|^2 = 1$, mentre $|\alpha\rangle$ e $|\beta\rangle$ sono vettori di stato che costituiscono una base ortonormale.

1.3.2 La misura

In Meccanica Quantistica, un'operazione di misura consiste nell'interazione di uno strumento classico con un sistema quantistico. Questo tipo di interazione, per sua natura, è distruttivo: la misura inevitabilmente *perturba* lo stato di un sistema, causando il *collasso della sua funzione d'onda*.
L'interpretazione del collasso della funzione d'onda è stata a lungo controversa, e ancora oggi è oggetto di studio. Non a caso, ci si riferisce a questo fenomeno con l'appellativo di *problema della misura*.

È possibile misurare diverse grandezze (*osservabili fisiche*): ad esempio, la posizione di una particella, il suo momento, oppure la proiezione del suo spin lungo l'asse Z, oppure lungo l'asse X o Y. Nel caso della misura della posizione della particella, il set dei possibili risultati è un insieme *continuo* (in realtà, dovrebbe essere considerato discreto, ma ciò comporterebbe grosse e inutili complicazioni nei calcoli). Nel caso, invece, della misura della proiezione dello spin di una particella lungo l'asse Z, i possibili risultati sono soltanto due: *spin up* oppure *spin down* (insieme *discreto* di possibili risultati). È di fatto un qubit: un sistema a due livelli.

Matematicamente, un'osservabile fisica viene rappresentata con un opportuno *operatore*. Un generico operatore A è un oggetto matematico che, se viene applicato ad uno stato fisico, restituisce un altro stato fisico: $A|\psi\rangle = |\phi\rangle$.
Uno stato $|\psi\rangle$ si dice *autostato* dell'operatore A se $A|\psi\rangle = \lambda|\psi\rangle$, dove λ è l'*autovalore* associato all'autostato $|\psi\rangle$. L'autovalore λ è il *risultato della misura*.
Ne consegue che, se si considerasse il generico vettore di stato $|\psi\rangle = \sum_i c_i |e_i\rangle$ dato dalla combinazione lineare di N vettori di stato $|e_i\rangle$ (che compongono una base ortonormale), l'*operatore associato ad un'osservabile fisica* si potrebbe scrivere come:

$$O = \sum_i \lambda_i |e_i\rangle \langle e_i| \tag{1.4}$$

La somma dei prodotti *ket - bra* $|e_i\rangle\langle e_i|$ non è altro che una matrice identità: infatti, prendendo per esempio il caso bidimensionale, in cui $|e_0\rangle = |0\rangle$ e $|e_1\rangle = |1\rangle$, si ha:

$$\sum_i |e_i\rangle \langle e_i| = |0\rangle \langle 0| + |1\rangle \langle 1| = \begin{pmatrix}1\\0\end{pmatrix}\begin{pmatrix}1 & 0\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix}\begin{pmatrix}0 & 1\end{pmatrix} = \begin{pmatrix}1 & 0\\0 & 0\end{pmatrix} + \begin{pmatrix}0 & 0\\0 & 1\end{pmatrix} = \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix} \quad (1.5)$$

Pertanto, l'operatore *osservabile fisica* è rappresentabile come una matrice diagonale, avente come elementi della diagonale principale i coefficienti λ_i:

$$O = \begin{pmatrix}\lambda_0 & 0\\0 & \lambda_1\end{pmatrix} \quad (1.6)$$

I λ_i sono i *possibili risultati* della misura dell'osservabile O. Come si può notare, gli stati della base computazionale sono autostati dell'osservabile O definita dall'equazione (1.6).
In generale, infatti, si ha che $O|e_i\rangle = \lambda_i |e_i\rangle$. Ad esempio:

$$O|0\rangle = \begin{pmatrix}\lambda_0 & 0\\0 & \lambda_1\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \lambda_0 \begin{pmatrix}1\\0\end{pmatrix} = \lambda_0 |0\rangle \quad (1.7)$$

Il *valore medio* di un'osservabile, in generale, è dato dalla media dei suoi possibili risultati, pesata con le probabilità associate ad ogni risultato:

$$\langle O\rangle = \sum_i \lambda_i \cdot Prob(\lambda_i) \quad (1.8)$$

Calcolando la quantità $\langle\psi| O |\psi\rangle$, si ottiene:

$$\langle\psi| O |\psi\rangle = \langle\psi| \sum_i \lambda_i |e_i\rangle \langle e_i|\psi\rangle = \sum_i \lambda_i \langle\psi|e_i\rangle \langle e_i|\psi\rangle = \sum_i \lambda_i |\langle e_i|\psi\rangle|^2 \quad (1.9)$$

Effettuando l'identificazione $Prob(\lambda_i) = |\langle e_i|\psi\rangle|^2$, si ottiene che il valor medio di O è calcolabile come $\langle O\rangle = \langle\psi| O |\psi\rangle$.

Dunque, è possibile calcolare le *probabilità* associate ad ognuno dei possibili risultati: esse sono $Prob(\lambda_i) = |\langle e_i|\psi\rangle|^2$.

Questa tuttavia è solamente la rappresentazione matematica dei fenomeni. Nella realtà, ciò che permette di estrarre informazione da un certo sistema quantistico è la *misura* di una certa *osservabile*, che viene operata su di esso. Tale misura restituisce certi risultati, e ad ogni risultato è associata una certa probabilità.
Successivamente, per descrivere i risultati e le probabilità tramite la matematica, ogni osservabile viene associata ad un *operatore hermitiano* (quindi diagonalizzabile su una certa base di autostati, e avente autovalori *reali*). Gli autovalori di questo operatore sono i possibili risultati della misura, a cui è associata una probabilità $Prob(\lambda_i) = |\langle e_i|\psi\rangle|^2$.

L'effetto di una misura è il collasso della funzione d'onda del sistema.
Matematicamente, il *collasso della funzione d'onda* è un *cambiamento dello stato del sistema*. Più precisamente, se viene effettuata una misura dell'osservabile O sullo stato $|\psi\rangle$, lo stato del sistema collasserà in un *autostato* di O. Il risultato della misura sarà l'*autovalore* λ_i, associato all'autostato in cui è collassato lo stato del sistema.

Esempio: si consideri l'osservabile $O = \lambda_0 |0\rangle \langle 0| + \lambda_1 |1\rangle \langle 1|$ e lo stato generico $|\psi\rangle$. Gli autostati di O sono $|0\rangle$ e $|1\rangle$: pertanto, una misura dell'osservabile O sullo stato $|\psi\rangle$ causerà il collasso dello stato $|\psi\rangle$ in uno tra gli stati $|0\rangle$ e $|1\rangle$. In particolare, se la misura restituisce il risultato λ_0, lo stato collassa in $|0\rangle$; se restituisce il risultato λ_1, invece, collassa in $|1\rangle$. Per calcolare la probabilità con cui i due risultati possono essere rilevati, è necessario *scomporre* $|\psi\rangle$ *nella base degli autostati di* O: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. A questo punto, le probabilità si calcolano come $Prob(\lambda_i) = |\langle e_i|\psi\rangle|^2$:

$$\begin{aligned} Prob(\lambda_i) &= |\langle 0|\psi\rangle|^2 = |\alpha|^2 \\ Prob(\lambda_i) &= |\langle 1|\psi\rangle|^2 = |\beta|^2 \end{aligned} \quad (1.10)$$

Tuttavia, ancora, questo è solamente mero calcolo: le probabilità sono già state ricavate in precedenza sperimentalmente, effettuando la stessa misura su un gran numero di sistemi quantistici preparati nello stesso stato e osservando la frequenza di ogni risultato.

Tornando al contesto del Quantum Computing, l'operazione di misura è il passaggio fondamentale che permette di ricavare l'informazione sul sistema. In Qiskit, di default, l'unica operazione di misura che si può attuare in un circuito quantistico è la misurazione di Z, un'osservabile speciale, descritta dalla matrice di Pauli σ_z, i cui autostati sono proprio gli stati della base computazionale, ovvero $|0\rangle$ e $|1\rangle$.

1.3.3 Le porte quantistiche

Le porte quantistiche, o *quantum gate*, sono l'analogo delle porte logiche nella computazione classica. Esse consistono in *trasformazioni unitarie* che agiscono sui qubit. Sono pertanto rappresentabili come operatori che, se applicati ai vettori di stato dei qubit, li trasformano in nuovi stati.

Un primo esempio di porte quantistiche *a singolo qubit* (cioè che agiscono su un solo qubit) sono quelle che corrispondono alle *matrici di Pauli* σ_i e alla matrice identità (scritte nella base computazionale):

$$\begin{aligned}
\sigma_x = X &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = |0\rangle\langle 1| + |1\rangle\langle 0| \\
\sigma_y = Y &= \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = -i\,|0\rangle\langle 1| + i\,|1\rangle\langle 0| \\
\sigma_z = Z &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = |0\rangle\langle 0| - |1\rangle\langle 1| \\
\mathbb{1} &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = |0\rangle\langle 0| + |1\rangle\langle 1|
\end{aligned} \tag{1.11}$$

Ognuna di esse ha un preciso effetto su ogni stato della base computazionale. Più in generale, si possono definire queste porte quantistiche come le trasformazioni tali che:

$$\begin{aligned}
&X\,|0\rangle \equiv |1\rangle && X\,|1\rangle \equiv |0\rangle \\
&Y\,|0\rangle \equiv -i\,|1\rangle && Y\,|1\rangle \equiv i\,|0\rangle \\
&Z\,|0\rangle \equiv -\,|0\rangle && Z\,|1\rangle \equiv |1\rangle \\
&\mathbb{1}\,|0\rangle \equiv |0\rangle && \mathbb{1}\,|1\rangle \equiv |1\rangle
\end{aligned} \tag{1.12}$$

Si può notare che gli stati $|0\rangle$ e $|1\rangle$ sono autostati di Z e di $\mathbb{1}$, ma non lo sono di X ed Y.

È interessante interpretare queste porte quantistiche come delle *rotazioni* nello spazio tridimensionale. Tornando a considerare la Sfera di Bloch (Figura 1.2), si può notare che, se il vettore $|+\rangle$ *viene fatto ruotare* di un angolo π attorno all'asse Z, esso si trova a coincidere di fatto con lo stato $|-\rangle$. Al contrario, se il vettore $|0\rangle$ viene fatto ruotare *attorno all'asse Z*, esso non cambia: il qubit rimane ancora nello stato $|0\rangle$.
Queste due trasformazioni hanno lo stesso effetto dell'applicazione dell'operatore Z: infatti, per definizione, $Z\,|0\rangle \equiv -\,|0\rangle$, mentre $Z\,|+\rangle = |-\rangle$. Il fatto che dall'applicazione di Z allo stato $|0\rangle$ esca un *segno negativo* è irrilevante: lo stato ottenuto è equivalente allo stato $|0\rangle$, a meno della *fase globale* $-1 = e^{i\pi}$ che può però essere riassorbita nella definizione stessa dello stato.
L'applicazione di Z è quindi una *rotazione di un angolo π attorno all'asse Z*. Allo stesso modo, l'operatore X rappresenta la *rotazione di un angolo π attorno all'asse X*, mentre Y opera la *rotazione di un angolo π attorno all'asse Y*.

È opportuno ora definire un'altra coppia di vettori di stato: ovvero gli stati $|+\rangle$ e $|-\rangle$, che si ottengono a partire da $|0\rangle$ e $|1\rangle$ nel seguente modo:

$$\begin{aligned}
|+\rangle &\equiv \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\
|-\rangle &\equiv \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)
\end{aligned} \tag{1.13}$$

Essi costituiscono a loro volta una base ortonormale, diversa da quella composta da $|0\rangle$ e $|1\rangle$.
Non solo: $|+\rangle$ e $|-\rangle$ sono gli autostati di X. Pertanto, come si mostrerà in seguito, una misura dell'osservabile X causerà il collasso dello stato del qubit in un autostato dell'operatore X (ovvero $|+\rangle$ oppure $|-\rangle$), mentre una misura dell'osservabile Z farà collassare lo stato del qubit in un autostato dell'osservabile Z (cioè $|0\rangle$ o $|1\rangle$). Sarà su questa peculiarità che si baserà il funzionamento del protocollo BB84.

La *porta di Hadamard* è la trasformazione tale che:

$$\begin{aligned} H|0\rangle &\equiv |+\rangle \qquad H|1\rangle \equiv |-\rangle \\ H|+\rangle &\equiv |0\rangle \qquad H|-\rangle \equiv |1\rangle \end{aligned} \tag{1.14}$$

Nella base computazionale, la forma matriciale della porta di Hadamard è:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{1.15}$$

Altre porte quantistiche a singolo qubit sono:

- R_ϕ, che opera rotazioni di un angolo ϕ attorno all'asse Z;
- S, che è una R_ϕ in cui $\phi = \frac{\pi}{2}$;
- T, una R_ϕ in cui $\phi = \frac{\pi}{4}$;
- $U_3(\theta, \phi, \lambda)$, la più generale porta quantistica a singolo qubit, che effettua rotazioni tramite l'utilizzo dei tre parametri θ, ϕ e λ.

$$R_\phi = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix} \qquad U_3(\theta, \phi, \lambda) = \begin{pmatrix} \cos\frac{\theta}{2} & -e^{i\lambda}\sin\frac{\theta}{2} \\ e^{i\phi}\sin\frac{\theta}{2} & e^{i\phi+i\lambda}\cos\frac{\theta}{2} \end{pmatrix} \tag{1.16}$$

Partendo dalla porta U_3, si definiscono le porte quantistiche $U_2 \equiv U_3(\frac{\pi}{2}, \phi, \lambda)$ e $U_1 \equiv U_3(0, 0, \lambda)$. Sulla piattaforma di IBM Quantum Experience, prima che un circuito venga eseguito su un hardware reale, tutte le quantum gate vengono automaticamente ricompilate sotto forma delle porte U_1, U_2 e U_3, che per questo vengono anche chiamate *porte fisiche*.

Esistono anche quantum gate *a più qubit*. In un circuito quantistico, diversi qubit possono essere inizializzati *in parallelo*, ed essere collegati attraverso una o più porte di questo tipo.
Due qubit diversi sono descritti da vettori di stato che appartengono a spazi di Hilbert differenti. Se $|\psi_1\rangle$ è lo stato del primo qubit e $|\psi_2\rangle$ è lo stato del secondo qubit, allora lo stato del *sistema a due qubit* è indicato con:

$$|\psi_1\psi_2\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \tag{1.17}$$

Un esempio di porta quantistica a due qubit è la $CNOT$ *gate* (il nome deriva da *controlled-not* gate). I due qubit collegati dalla $CNOT$ gate sono detti rispettivamente *target qubit* e *control qubit*. La $CNOT$ gate agisce sul *target qubit* in base allo stato in cui si trova il *control qubit*: se il control qubit si trova nello stato $|1\rangle$, il target qubit viene trasformato tramite una X gate (ovvero: se esso è nello stato $|0\rangle$, subisce uno *swap* nello stato $|1\rangle$, mentre se è nello stato $|1\rangle$, viene trasformato nello stato $|0\rangle$). Viceversa, se il control qubit si trova nello stato $|0\rangle$, non viene operata nessuna trasformazione. Scrivendo gli stati $|+\rangle$ e $|-\rangle$ in funzione di $|0\rangle$ e $|1\rangle$ si dimostra che, in questo caso, i ruoli di control qubit e di target qubit vengono invertiti. Ora, infatti, se il *target* qubit è nello stato $|-\rangle$, viene applicata una Z gate al *control* qubit (in questo modo, se il control qubit si trova in $|+\rangle$, viene trasformato in $|-\rangle$, e viceversa), mentre, se il target qubit si trova nello stato $|+\rangle$, non viene cambiato nulla.

$$\begin{aligned} CNOT\,|\psi_{control}\rangle \otimes |\psi_{target}\rangle &= \begin{cases} |\psi_{control}\rangle \otimes |\psi_{target}\rangle \text{ se } |\psi_{control}\rangle = |0\rangle \\ |\psi_{control}\rangle \otimes (X\,|\psi_{target}\rangle) \text{ se } |\psi_{control}\rangle = |1\rangle \end{cases} \\ CNOT\,|\psi_{control}\rangle \otimes |\psi_{target}\rangle &= \begin{cases} |\psi_{control}\rangle \otimes |\psi_{target}\rangle \text{ se } |\psi_{target}\rangle = |+\rangle \\ (Z\,|\psi_{control}\rangle) \otimes |\psi_{target}\rangle \text{ se } |\psi_{target}\rangle = |-\rangle \end{cases} \end{aligned} \tag{1.18}$$

La $CNOT$ gate è usata per creare uno *stato entangled* tra due qubit. Uno stato entangled è uno stato che *non può essere scritto nella forma fattorizzata* dell'equazione (1.17). In seguito si mostrerà come è possibile creare uno stato entangled tra due qubit in un circuito quantistico.

1.3.4 Il generico stato di un qubit

Il generico stato di un qubit può essere rappresentato considerando una generica combinazione lineare di due stati, ovvero $|\psi\rangle = a \cdot |\alpha\rangle + b \cdot |\beta\rangle$. Dato che a, b sono complessi (dunque della forma $z = x + iy$, con x, y reali), lo stato $|\psi\rangle$ dovrebbe essere descritto da due coppie di numeri reali. Tuttavia, uno di questi numeri è fissato dalla normalizzazione $|a|^2 + |b|^2 = 1$, mentre uno è fissato dal fatto che la *fase globale* è inosservabile (cioè, se $e^{i\gamma}$ è la fase globale, si ha che $\left|e^{i\gamma}\right|^2 = 1$). Basta quindi usare soltanto i due numeri reali θ, ϕ per descrivere lo stato generico di un qubit: scegliendo $a = \cos\frac{\theta}{2}$ e $b = \sin\frac{\theta}{2} \cdot e^{i\phi}$, e usando la base computazionale come base di vettori di stato, il generico stato di un qubit è dato da:

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + \sin\frac{\theta}{2} \cdot e^{i\phi}|1\rangle \tag{1.19}$$

Gli stati $|0\rangle$, $|1\rangle$, $|+\rangle$ e $|-\rangle$ sono *vettori di lunghezza unitaria in uno spazio tridimensionale*. Ponendo $|0\rangle$ e $|1\rangle$ paralleli all'asse Z (e con verso opposto), e allineando $|+\rangle$ e $|-\rangle$ lungo l'asse X (facendo ancora in modo che $|+\rangle$ e $|-\rangle$ abbiano verso opposto), si ottiene una descrizione geometrica del generico stato di un qubit: la *Sfera di Bloch*.
Il fatto di usare i due numeri θ e ϕ come parametri reali per descrivere lo stato di un qubit, in questo modo, si rivela un'ottima scelta: identificando θ con l'angolo rispetto all'asse Z e ϕ con l'angolo rispetto all'asse X nel piano X-Y, si può indicare lo stato generico $|\psi\rangle$ come *un punto che si muove sulla superficie della Sfera di Bloch*. Di fatto, θ e ϕ non sono altro che gli angoli che compaiono nelle coordinate sferiche:

$$\begin{cases} x = r\sin\theta\cos\phi \\ y = r\sin\theta\sin\phi \\ z = r\cos\theta \end{cases} \tag{1.20}$$

dove il raggio r è costante ($r = 1$).

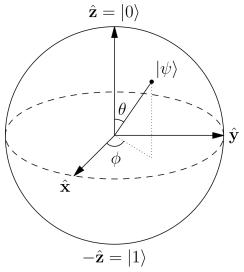


Figura 1.2: La sfera di Bloch

In questo modo, lo stato $|0\rangle$ si può identificare con la terna $(x, y, z) = (0, 0, 1)$, oppure con la coppia $(\theta, \phi) = (0, 0)$. Allo stesso tempo, $|1\rangle$ diventa $(x, y, z) = (0, 0, -1)$ oppure $(\theta, \phi) = (\pi, 0)$.
Inoltre $|+\rangle$ viene indicato con $(x, y, z) = (1, 0, 0)$ o $(\theta, \phi) = (\frac{\pi}{2}, 0)$, mentre $|-\rangle$ è scrivibile come $(x, y, z) = (-1, 0, 0)$ oppure come $(\theta, \phi) = (\frac{\pi}{2}, \pi)$.

Il fatto che lo stato generico di un qubit possa essere descritto attraverso due numeri *reali*, e dunque appartenenti ad un range continuo di numeri, fa sì che un qubit possa trovarsi in un numero *infinito* di stati possibili.
A volte, specialmente nella divulgazione di questi argomenti, è possibile che si faccia confusione sul significato di quanto appena enunciato. È bene quindi evidenziare che ciò *non* significa che la misura di un qubit possa restituire infiniti risultati diversi! Al contrario, un qubit *può essere preparato in infiniti modi differenti*: ovvero una misura su un qubit può restituire solo *due risultati*, ciascuno con la propria probabilità, ed è questa probabilità che può avere un numero infinito di valori diversi tra 0 e 1 (in base a come è stato preparato lo stato del qubit).

1.4 Quantum Computing: dalla teoria alla pratica

1.4.1 Coerenza quantistica

Mentre la fase *globale* di uno stato fisico non è osservabile (ad esempio, $|\psi\rangle$ è equivalente a $e^{i\gamma}|\psi\rangle$, perché nel calcolo delle probabilità, che coinvole un modulo elevato al quadrato, tale fase scompare), al contrario una fase *relativa* può essere osservata. Un esempio di fase relativa è, per esempio, la fase $e^{i\gamma}$ nello stato $|\psi\rangle = \alpha|0\rangle + \beta e^{i\gamma}|1\rangle$: lo stato $|\psi\rangle$ è diverso da $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$, e questa differenza è osservabile.
Si supponga che un sistema fisico sia stato preparato nello stato $|\psi\rangle = \alpha|0\rangle + \beta e^{i\gamma}|1\rangle$. Dopo un certo tempo, l'interazione del sistema con l'ambiente esterno può causare una *perdita di informazione* sulla fase relativa, che potrebbe non essere più osservabile. Tale fenomeno è detto *decoerenza* o *dephasing*. Se un sistema quantistico preserva questa differenza di fase, è detto *coerente*. La perdita di *coerenza quantistica* nel tempo viene chiamata *decoerenza*.
Uno degli obiettivi della ricerca odierna è cercare di ottenere qubit che mantengano la coerenza quantistica il più a lungo possibile nel tempo, in modo da minimizzare la perdita di informazione sul sistema.

1.4.2 Realizzazione di un circuito quantistico con Qiskit

Un circuito quantistico consiste in una serie di operazioni eseguite su uno o più qubit.
Il primo passo nella realizzazione di un circuito è l'*inizializzazione* del qubit. Di default, in Qiskit un qubit è sempre inizializzato nello stato $|0\rangle$; tuttavia, per preparare un qubit in uno stato qualsiasi, basta inserire in successione le opportune porte quantistiche. Ad esempio, inserendo una porta X si ottiene lo stato $|1\rangle$; inserendo solamente una porta H, si ha lo stato $|+\rangle$, mentre, inserendole entrambe, si cambia lo stato del qubit in $|-\rangle$. Di solito, questo passaggio viene concluso da una *barriera*, ovvero una delimitazione che permette di dividere il circuito in parti diverse tra loro.
Segue l'*implementazione dell'algoritmo* vero e proprio. Questa parte dipende unicamente da quale algoritmo debba essere eseguito attraverso il circuito; per svilupparlo, si inseriscono in successione le quantum gate necessarie.
Infine, la *misura dei qubit* conclude il circuito. È il nodo cruciale della computazione quantistica: la misura infatti permette di ottenere il risultato vero e proprio dell'esecuzione del circuito. Si tratta del passaggio da *fenomeno quantistico* ad *output classico*; il sistema quantistico *qubit* viene forzato a restituire un risultato sotto forma di *bit classico*.

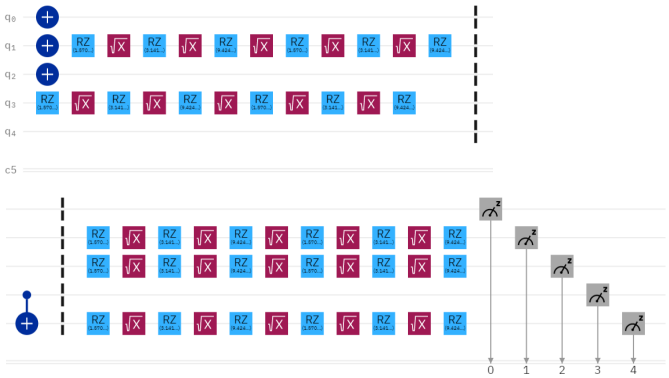


Figura 1.3: Esempio di circuito quantistico, rielaborato sulla piattaforma IBM Quantum Experience dopo che il programma era stato lanciato da remoto. In particolare, questo è l'ultimo circuito eseguito nell'ambito degli esperimenti su dispositivi reali del Capitolo 4.

Quando si effettua una misura in un circuito quantistico, l'output della misura può solo assumere i valori 0 oppure 1. Gli output 0 e 1 sono univocamente relazionati con gli autovalori dell'osservabile. Infatti, i possibili risultati della misura dell'osservabile Z sono gli autovalori $\lambda_0 = 1$ e $\lambda_1 = -1$; ma dato che l'operazione di misura eseguibile di default in Qiskit è proprio quella dell'osservabile Z, i suoi

autovalori possono essere identificati univocamente con gli output 0 ed 1. Ovvero: l'autovalore $\lambda_0 = 1$ è abbinato al valore di bit 0, mentre $\lambda_1 = -1$ è associato all'output 1.

Normalmente, un circuito quantistico viene eseguito diverse volte, ed il risultato della misura viene registrato ogni volta sotto forma di bit. Dopo aver lanciato un sufficiente numero di esecuzioni, dunque, è possibile effettuare un'analisi di tutti i dati raccolti, in modo da poter ricavare effettivamente informazioni sul comportamento del circuito. Tale analisi consiste semplicemente nel contare quante volte il risultato ottenuto è uguale a 0 e quante volte è uguale a 1, nelle diverse configurazioni del circuito.
La frequenza di queste occorrenze permette di calcolare la probabilità relativa ad un certo risultato. Essendo proprio questa probabilità l'unica quantità che la Meccanica Quantistica permette di calcolare e di predire, è possibile verificare le attese della teoria confrontandole con le probabilità ricavate, oppure condurre esperimenti con l'obiettivo di trovare le probabilità che descrivono i risultati di un certo fenomeno fisico.

1.4.3 Fenomeni quantistici alla base del Quantum Computing

Come accennato in precedenza, il Quantum Computing utilizza le peculiarità di tre effetti quantistici in particolare, ovvero:

- Sovrapposizione degli stati fisici;
- Correlazione quantistica o Entanglement;
- Interferenza tra le funzioni d'onda delle particelle.

La *sovrapposizione degli stati* è stata già trattata in precedenza; essa costituisce la base di partenza per ogni algoritmo di Quantum Computing, in quanto grazie ad essa è possibile preparare un qubit in infiniti modi diversi, rendendo centrale il ruolo della probabilità in ogni circuito quantistico.
L'*Entanglement* è il fenomeno per cui, se due sistemi fisici sono fortemente correlati (*entangled*), la misura dello stato di uno dei due sistemi influenza istantaneamente anche l'altro sistema. L'Entanglement è usato per implementare il teletrasporto di informazione (*quantum teleportation*) e le tecniche di *dense coding* e di *crittografia quantistica*.
Infine, l'*interferenza delle funzioni d'onda* è un aspetto della dualità ondulatoria-corpuscolare delle particelle. Come due onde che si compenetrano, le funzioni d'onda di due particelle possono creare interferenza costruttiva o distruttiva. Un esempio celebre è l'esperimento della *doppia fenditura*, effettuato con elettroni o, più recentemente, con molecole di fullerene (*esperimento di Zeilinger* [13]). L'interferenza è usata nel Quantum Computing per la risoluzione di problemi computazionalmente complessi: mentre un computer classico è *costretto* ad analizzare tutti i casi possibili di un problema, un computer quantistico può, sfruttando l'interferenza, *amplificare* la probabilità relativa ai casi più importanti e *smorzare* quella relativa alle casistiche da scartare. In tal modo, l'efficienza del calcolo è di gran lunga maggiore che nel caso della computazione tradizionale.

In questo progetto di tesi, la trattazione coinvolge solamente i primi due fenomeni appena presentati: la sovrapposizione degli stati e l'Entanglement. In particolare, viene analizzato un protocollo di *crittografia quantistica*, ovvero il *protocollo BB84*: tale protocollo basa il suo funzionamento sulla sovrapposizione degli stati fisici e sul collasso della funzione d'onda dopo una misura.

Capitolo 2

Il protocollo BB84

In questa trattazione si prenderà in considerazione soltanto una delle tante applicazioni del Quantum Computing: la *crittografia quantistica*.

2.1 Crittologia: crittografia contro crittoanalisi

La *crittologia* è la disciplina che studia le *scritture nascoste*, ovvero l'offuscamento del significato di un testo scritto o di un messaggio. Tale disciplina si divide in due branche: la *crittoanalisi*, che tratta la decifrazione di testi occultati senza sapere a priori in quale modo il loro significato è stato offuscato, e la *crittografia*, che è volta alla ricerca di metodi sempre più sicuri per occultare il reale significato dei segni che compongono un testo.

2.1.1 Crittografia classica

Nel corso della Storia, fin dall'antichità, sono state sviluppate diverse tecniche di crittografia. Esse erano usate per codificare messaggi importanti che, se intercettati, non dovevano lasciare trapelare informazioni sul loro vero significato, come gli ordini in tempo di guerra. Oggi il campo delle applicazioni è più vasto ancora: basti pensare alla sicurezza informatica, dalla tutela della privacy alla protezione dei dati sensibili di banche ed enti governativi.
Per ogni passo avanti nel campo della crittografia, tuttavia, ve ne è stato uno nell'ambito della crittoanalisi. Nel tempo, la contrapposizione tra le due discipline si è fatta sempre più intensa: quando veniva inventato un nuovo metodo (o *protocollo*) per crittografare un testo, poco dopo veniva trovato un nuovo modo per decrittarlo; quindi qualcuno inventava un protocollo crittografico più sicuro, che veniva inevitabilmente messo in crisi da un metodo di decodifica più efficace.
Ciononostante, nel 1918 viene ideata da G. Vernam una tecnica di crittografia considerata pressoché inattaccabile, nota con il nome di *cifrario di Vernam*. Tale tecnica si basa sul già esistente *cifrario di Vigenère*, che consiste nel sostituire una lettera del testo con un'altra lettera, ottenuta tramite l'utilizzo di una *chiave*. Ad esempio: se la lettera da codificare è A e la chiave è la lettera C, la nuova lettera è data da $A + C = 0 + 2 = 2 = C$ (infatti, la lettera A occupa il posto numero 0 nell'alfabeto, mentre C occupa il posto numero 2). Nel cifrario di Vigenère, la *chiave* è una parola molto piccola, che viene ripetuta in modo da cifrare l'intero messaggio:

Parola:	C	R	I	T	T	O	G	R	A	F	I	A	+
Chiave ripetuta:	C	I	A	O	C	I	A	O	C	I	A	O	=
Parola crittografata:	E	Z	I	H	V	W	G	F	C	N	I	O	

Il cifrario di Vernam sfrutta questo meccanismo, ma pone una condizione in più: *la chiave deve essere lunga quanto il messaggio e non riutilizzabile.*
Nell'articolo *Communication Theory of Secrecy Systems* [7] del 1949, C. E. Shannon dimostrò l'inviolabilità del cifrario di Vernam. La provata inviolabilità fece sì che il cifrario venisse largamente utilizzato dalle spie durante la guerra fredda, le quali usavano scrivere i messaggi su taccuini in cui era

già presente la chiave, condivisa solamente con il destinatario del messaggio. Dopo l'utilizzo di una pagina del taccuino, essa veniva strappata. Dall'appellativo di *taccuino monouso* (*One Time Pad*) deriva l'acronimo OTP, utilizzato oggi per indicare codici di sicurezza basati su questa tipologia di crittografia.

Nonostante il cifrario di Vernam sia inattaccabile, esso presenta tuttavia alcuni punti deboli. Il primo, è il fatto che la chiave non possa essere usata più di una volta (altrimenti il cifrario diventa al contrario decifrabile). Il secondo, è la difficoltà nella *distribuzione* delle chiavi, che potrebbero essere facilmente intercettate, rendendo inutile l'uso del cifrario. È per questo che, a volte, al posto del cifrario di Vernam si utilizzano protocolli a chiavi *pubbliche* (il cifrario di Vernam è invece a chiave *privata*).

2.1.2 Chiavi private e chiavi pubbliche

Un protocollo crittografico in cui il mittente ed il destinatario condividono una chiave conosciuta soltanto da loro è detto *protocollo a chiave privata*. L'esempio citato in precedenza è il cifrario di Vernam: mittente e destinatario posseggono entrambi la stessa chiave di cifratura e di decrittazione. Questa tipologia di tecniche crittografiche è detta *crittografia a chiave privata* o *crittografia simmetrica*.
Il punto debole dei protocolli a chiave privata risiede nel fatto che la chiave debba essere già in possesso di chi manda e di chi riceve il messaggio. In caso contrario, essa dovrebbe essere trasmessa dall'uno all'altro (questo processo si chiama *distribuzione della chiave*), ma ciò comporterebbe l'esposizione della chiave al pericolo di essere intercettata. Inoltre, se questo avvenisse, il mittente ed il destinatario non avrebbero alcun modo di saperlo.

Per ovviare all'inconveniente dell'intercettazione, negli Anni Settanta sono stati sviluppati i *protocolli a chiave pubblica*. Il meccanismo che sta a fondamento di questi protocolli si basa sull'uso di *due chiavi diverse*: una pubblica, usata per crittare il messaggio, ed una privata, usata per decrittarlo.
Un famoso protocollo a chiave pubblica è il *protocollo RSA*, sviluppato nel 1977 da R. Rivest, A. Shamir e L. Adleman. Il suo funzionamento si basa sul fatto che, mentre è molto facile calcolare un numero *moltiplicando fra loro due numeri primi*, al contrario *trovare la fattorizzazione in numeri primi di quello stesso numero* è computazionalmente molto complicato. Nella pratica, il mittente usa la chiave pubblica (condivisa dal destinatario, e dunque visibile a chiunque) per crittare il messaggio, mentre il destinatario decodifica il messaggio usando la chiave privata, nota a lui solo. Tutto ciò viene fatto tramite l'utilizzo, da parte del mittente, di un numero $n = p \cdot q$, la cui fattorizzazione nei numeri primi p e q è conosciuta solamente dal destinatario.

Il protocollo RSA potrebbe sembrare difficile da violare, in quanto un computer normale può impiegare interi anni per trovare i fattori di un numero primo abbastanza grande.
Così tuttavia non è per un *computer quantistico*. Nel 1994 P. Shor ideò un algoritmo quantistico [12] per la fattorizzazione dei numeri primi. Con l'uso di questo algoritmo tramite un computer quantistico, il tempo impiegato per la fattorizzazione viene notevolmente ridotto. Ciò rende molto più debole la crittografia che sfrutta il protocollo RSA.

2.1.3 Crittografia quantistica

Ogni tipologia di protocollo ha i suoi punti di forza ed i suoi punti deboli. Tuttavia, è stata trovata una nuova tecnica di crittografia che supera le precedenti, unendo le peculiarità dei diversi protocolli. Questa nuova tecnica sfrutta le leggi della Meccanica Quantistica, ed è perciò chiamata *crittografia quantistica*.

Il primo protocollo di crittografia quantistica fu inventato nel 1984 da C. H. Bennett e G. Brassard [5], e per questo venne chiamato *protocollo BB84*.
È un protocollo di *Quantum Key Distribution* (*QKD*): la sua funzione non è quella di codificare un messaggio, ma di *distribuire la chiave con cui crittografarlo*. Il protocollo BB84 è un protocollo a chiave *privata*. Oltre ad essere semplice e facilmente implementabile, è *completamente sicuro* rispetto a qualsiasi tentativo di intercettazione della chiave. Dunque, grazie a tale protocollo, è possibile scambiare messaggi crittografati tramite il cifrario di Vernam (e dunque inviolabili), usando una chiave

distribuita (e non inizialmente condivisa) ma del tutto sicura. Inoltre, il protocollo BB84 può essere considerato il prototipo dei protocolli quantistici sviluppati in seguito.

2.2 Il protocollo BB84

In questa sezione viene descritto il funzionamento del protocollo BB84, mostrando i fenomeni quantistici su cui si basa e come essi vengono sfruttati per effettuare la condivisione sicura di una chiave crittografica. Nei capitoli seguenti verrà invece dato spazio all'implementazione del protocollo tramite il Quantum Computing.

2.2.1 Personaggi

In crittografia, ma anche nella Teoria dei Giochi e in Fisica, spesso è d'aiuto assegnare un *nome* ai *personaggi* che entrano in gioco nella trasmissione di un messaggio o in un processo decisionale. È una semplice comodità che permette di seguire la descrizione dei processi più facilmente che usando delle lettere (ad esempio "*A manda un messaggio a B, che...*").
Esiste un vero e proprio gergo convenzionale: ad esempio, *Alice* e *Bob* di solito sono due persone che si vogliono mandare un messaggio; se ci sono più individui che interagiscono, si prosegue in ordine alfabetico (*Carol* o *Charlie*, *Dave*...). Vi è poi *Eve*, che di solito rappresenta uno spione, che *origlia* (dall'inglese *eavesdropper*, origliare). In crittografia quantistica, *Eve* può indicare anche l'ambiente (*environment*).
Nel caso del protocollo BB84, *Alice* vuole spedire un messaggio a *Bob*, mentre *Eve* è l'hacker che tenta di intercettare la loro trasmissione.

2.2.2 Panoramica

Il protocollo si basa sullo scambio di qubit tra Alice e Bob. In particolare, Alice genera una stringa di bit in modo pseudocasuale, in modo che il valore di ogni bit della stringa sia estratto con probabilità uniforme (quindi del 50%, in quanto i valori estraibili sono soltanto 0 oppure 1). La chiave verrà originata da questa stringa.
Quindi Alice codifica questa stringa *preparando un qubit* (secondo un preciso procedimento) per ogni bit della stringa. A questo scopo, Alice estrae una seconda stringa di bit pseudocasuali, funzionali alla scelta dello stato in cui preparare i qubit. Segue poi la *trasmissione* dei qubit a Bob, il quale a sua volta estrae una stringa di bit pseudorandom. In base al valore di ogni bit estratto, Bob *misura* su ciascun qubit una ben precisa osservabile fisica.
Terminate le misure, avviene un particolare *confronto* tra Alice e Bob: entrambi pubblicano *per intero* le stringhe di bit usate per effettuare la preparazione dei qubit e le misure. Inoltre, Alice pubblica la *prima metà* dei bit originariamente estratti, e Bob pubblica la *prima metà* dei risultati ottenuti dalle misure. Come si mostrerà, è grazie a questo confronto che è possibile individuare l'eventuale presenza dell'hacker Eve.

2.2.3 Codifica dei bit e preparazione dei qubit

Si supponga che Alice abbia estratto una stringa di N bit pseudocasuali.
Per codificare i bit attraverso i qubit, Alice prepara ogni qubit in un ben determinato stato fisico $|\psi\rangle$. Questo stato può essere scelto tra quattro possibili stati, cioè quelli appartenenti alla base canonica $\{|0\rangle,|1\rangle\}$ o quelli appartenenti alla base $\{|+\rangle,|-\rangle\}$, definiti nel Capitolo 1. La scelta di usare un determinato stato viene operata estraendo una nuova sequenza di N bit casuali, che vengono poi associati, in ordine, ai bit che Alice vuole trasmettere. In comune accordo con Bob, dunque, Alice decide di preparare i qubit nel seguente modo:

- Se il bit della stringa delle basi è 0 e il bit da codificare è 0, il qubit è preparato nello stato $|0\rangle$;
- Se il bit della stringa delle basi è 0 e il bit da codificare è 1, lo stato associato è $|1\rangle$;

- Se il bit della stringa delle basi è 1 e il bit da codificare è 0, viene preparato lo stato $|+\rangle$;
- Se il bit della stringa delle basi è 1 e il bit da codificare è 1, è inizializzato lo stato $|-\rangle$;

In pratica, se il bit della seconda stringa estratta vale 0, il relativo qubit viene preparato in uno stato della base $\{|0\rangle,|1\rangle\}$ (anche detta *base 0*); se invece il bit della seconda stringa vale 1, il corrispondente qubit viene preparato in uno stato della base $\{|+\rangle,|-\rangle\}$ (chiamata anche *base 1*). Per questo motivo, la seconda stringa estratta da Alice verrà indicata con *stringa delle basi*.

2.2.4 Trasmissione e misure

Una volta che Alice ha codificato un bit, manda a Bob il qubit preparato corrispondente. Questa trasmissione avviene su un *canale pubblico*, ed è pertanto in questa fase che Eve può intromettersi ed intercettare i qubit.

Ogni volta che Bob riceve un qubit, deve effettuare una misura su di esso per poter ricavarne un risultato. Tuttavia Bob *non può* sapere in quale stato è stato preparato il qubit. Sa solamente che il qubit può trovarsi in uno stato della base $\{|0\rangle,|1\rangle\}$ oppure della base $\{|+\rangle,|-\rangle\}$. Pertanto, Bob estrae una stringa di bit pseudocasuali, che ha la stessa funzione della *stringa delle basi* di Alice.
Si ipotizzi che il primo valore della stringa sia 0. Dunque, Bob (in comune accordo con Alice) misura sul primo qubit l'osservabile Z, i cui autostati sono quelli della base $\{|0\rangle,|1\rangle\}$. Se, per esempio, il secondo bit della stringa è 1, Bob misura sul secondo qubit l'osservabile X, i cui autostati corrispondono alla base $\{|+\rangle,|-\rangle\}$. Di fatto, ad un bit uguale a 0 corrisponde una misura di Z (o, equivalentemente, la scelta della base $\{|0\rangle,|1\rangle\}$), mentre a un bit pari a 1 è associata una misura di X (o una scelta della base $\{|+\rangle,|-\rangle\}$).

I risultati che Bob ottiene da ogni misurazione sono dei valori di bit: 0 oppure 1. Questi bit vengono riuniti in una stringa ordinata (il primo bit della stringa è il risultato della misurazione del primo qubit, il secondo bit è il risultato della misura del secondo qubit e così via). Tale stringa sarà indicata come *stringa dei risultati*, e a fine trasmissione sarà composta da N bit.

Bob a priori non può sapere se il qubit su cui effettua la misura si trova in un autostato dell'osservabile misurata oppure no.
Se misurasse un'osservabile su un qubit che si trovasse in un autostato della stessa osservabile, Bob otterrebbe un risultato *certo* (ovvero con probabilità pari a 1); viceversa, se misurasse un'osservabile su un qubit che *non* si trovasse in un autostato di tale osservabile, il risultato sarebbe invece *probabilistico* (cioè, prima della misurazione, il risultato avrebbe avuto il 50% di probabilità di essere ottenuto. Il valore 50% è dovuto a come sono stati definiti gli stati $|0\rangle$, $|1\rangle$, $|+\rangle$ e $|-\rangle$).
Ma questo Bob non lo sa! Dunque lui ottiene un risultato (0 o 1), ma non sa quale probabilità aveva di rilevare quel risultato.

2.2.5 Confronto

Alice e Bob, terminata la trasmissione, pubblicano ciascuno la propria *stringa delle basi* intera. Nel caso di Alice, questa è la stringa usata per codificare i bit iniziali; nel caso di Bob, è la stringa usata per scegliere l'osservabile da misurare.

Viene per prima cosa confrontato l'i-esimo bit di Alice con l'i-esimo bit di Bob. Quindi vengono **scartati** i bit *diversi* (e i relativi risultati delle misurazioni), perché corrispondono a una *diversa scelta della base di autostati*, e dunque a delle misurazioni aventi risultati *probabilistici*.
Ora dunque Bob sa quali risultati sono *certi* e quali no: tutti i qubit preparati nello stato di una certa base e misurati con l'osservabile di cui gli autostati sono gli stati di quella stessa base, restituiscono risultati *certi*. Più semplicemente, se un bit della stringa delle basi di Alice è identico ad un bit della stringa delle basi di Bob, il risultato corrispondente (ovvero il relativo bit della stringa dei risultati) è un risultato certo. Le coppie di bit identici, appartenenti alle due stringhe delle basi, sono detti *bit*

correlati.

Allo stesso modo, tra i bit originali di Alice, cioè quelli che Alice vuole trasmettere, vengono *scartati* tutti quelli che sono stati codificati usando una base *diversa* da quella scelta da Bob per effettuare la misura sul rispettivo qubit.
Pertanto, alla fine di questa prima parte di confronto, Alice e Bob si ritrovano con due stringhe delle basi *identiche* (i bit diversi sono stati tutti scartati). Dato che, statisticamente, si ha che nel 50% dei casi Alice e Bob hanno scelto una base differente, al termine del confronto la lunghezza delle stringhe di bit (sia delle stringhe delle basi, sia di quella dei bit da codificare che di quella dei risultati di Bob) sarà pari a circa $\frac{N}{2}$.

Quindi avviene la seconda parte del confronto. Alice pubblica la *prima metà* dei suoi bit originali, mentre Bob pubblica la *prima metà* della stringa dei risultati delle sue misurazioni. Poi Alice e Bob confrontano, bit per bit, le due metà.
In una trasmissione ideale, senza rumore, ci si aspetta che i risultati di Bob, ottenuti con la misura dello stesso operatore i cui autostati sono stati usati da Alice per preparare il qubit, siano tutti *identici* ai bit originali di Alice. Dunque, se tutto procede correttamente, Alice e Bob ottengono due metà stringhe identiche.

In questo caso, Alice e Bob scartano i risultati pubblicati (infatti, proprio perché *pubblici*, sono inservibili). I risultati non pubblicati (quelli della seconda metà della stringa dei risultati), in quanto *certi*, sono identici ai bit codificati da Alice non pubblicati. Sono questi i risultati che costituiscono la *chiave crittografica* condivisa da Alice e Bob: infatti, per Bob la chiave è la seconda metà dei risultati non scartati, mentre per Alice è la seconda metà dei bit codificati non scartati.
Dato che, ancora, sono stati scartati il 50% dei bit correlati (perché resi pubblici durante il confronto), la chiave condivisa da Alice e Bob sarà una stringa di bit lunga circa $\frac{1}{2} \cdot \frac{N}{2} = \frac{N}{4}$.

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

Tabella 2.1: Un esempio di codifica e decodifica tramite il protocollo BB84. I *bit correlati* sono stati sottolineati: come si può notare, ogni coppia di bit correlati presenta due valori *identici* di bit. I risultati certi e i corrispondenti bit codificati sono in grassetto; al contrario, i risultati probabilistici e i relativi bit codificati sono in corsivo. Inoltre, dopo il confronto, tutti i bit che ricadono nella prima metà (sotto la dicitura *Pubblici*) vengono comunque scartati. La *chiave*, in questo caso, è 100: corrisponde ai risultati (o ai bit codificati) non scartati presenti nella metà sotto la dicitura *Privati*.

2.2.6 Intervento dell'hacker

Nel caso in cui l'hacker Eve voglia intercettare i qubit, potrebbe farlo solamente nella fase di trasmissione tra Alice e Bob. Se tale trasmissione fosse uno scambio di bit *classici*, un hacker potrebbe effettuare diverse operazioni su di essi (per esempio copiarli e spedire a Bob i cloni al posto dei bit originali, o persino misurarli direttamente e leggerne il valore), senza mai essere scoperto. Nel caso di un protocollo di crittografia quantistica come questo, invece, la situazione è più delicata: una misura su un qubit può alterarne lo stato e produrre effetti misurabili sui risultati di misure successive. In questo caso, Eve potrebbe essere scoperta.
Si potrebbe pensare che, per evitare di farsi scoprire, Eve tenti di *clonare* i qubit. In questo modo, ella

potrebbe *copiare* lo stato di un qubit su quello di un qubit di sua proprietà, e mandare a Bob il qubit originale. Tuttavia, questo non è possibile: *clonare* uno stato è *proibito* dal *Teorema di no-cloning.*

Teorema (Teorema di no-cloning)**.** *Non esiste alcuna trasformazione unitaria U che permetta di produrre una copia di un arbitrario qubit ignoto.*

Dimostrazione. Si supponga per assurdo che Eve possa usare una *macchina di clonazione* U in grado di clonare un qubit che si trova in un qualsiasi stato $|\psi\rangle$, copiandolo su un altro qubit, in possesso di Eve, che si trova nello stato $|0\rangle$. Eve può cioè effettuare la trasformazione unitaria U che, per qualunque stato $|\psi\rangle$, agisce in questo modo:

$$|\alpha\rangle = U(|\psi\rangle \otimes |0\rangle) = |\psi\rangle \otimes |\psi\rangle \tag{2.1}$$

Dato che deve essere vero per qualunque stato, allora deve valere anche per un altro stato generico $|\phi\rangle$, diverso da $|\psi\rangle$:

$$|\beta\rangle = U(|\phi\rangle \otimes |0\rangle) = |\phi\rangle \otimes |\phi\rangle \tag{2.2}$$

Siccome l'operatore U è unitario (ovvero $U^\dagger = U^{-1}$), il prodotto scalare tra gli stati $|\alpha\rangle$ e $|\beta\rangle$ deve essere lo stesso, sia *prima* che *dopo* la clonazione:

$$\begin{aligned} \langle\alpha|\beta\rangle &= (\langle\psi| \otimes \langle 0|)U^\dagger U(|\phi\rangle \otimes |0\rangle) = \langle\psi|\phi\rangle \langle 0|0\rangle = \langle\psi|\phi\rangle \\ \langle\alpha|\beta\rangle &= (\langle\phi| \otimes \langle\phi|)(|\psi\rangle \otimes |\psi\rangle) = \langle\phi|\psi\rangle \langle\phi|\psi\rangle = |\langle\psi|\phi\rangle|^2 \end{aligned} \tag{2.3}$$

Dunque $\langle\psi|\phi\rangle = |\langle\psi|\phi\rangle|^2$. Ma quindi $\langle\psi|\phi\rangle = 1$ oppure $\langle\psi|\phi\rangle = 0$, cioè $|\psi\rangle$ e $|\phi\rangle$ fanno parte della stessa base ortonormale di stati.
Ne segue che la clonazione degli stati $|\psi\rangle$ e $|\phi\rangle$ è possibile, ma *solo se* $|\psi\rangle$ *e* $|\phi\rangle$ *appartengono alla stessa base ortonormale di stati.* Questo è in contraddizione con le ipotesi del teorema: se appartengono alla stessa base di stati, i qubit descritti da $|\psi\rangle$ e $|\phi\rangle$ non sono qubit arbitrari. Dunque non esiste una trasformazione unitaria U che permetta di clonare un *generico* stato quantistico. □

Intuitivamente, Eve potrebbe clonare solamente degli stati che appartengano tutti alla base $\{|0\rangle,|1\rangle\}$, oppure che facciano tutti parte della base $\{|+\rangle,|-\rangle\}$. Ma Eve non può sapere a priori quale sia lo stato dei qubit trasmessi, se non facendo su di essi una misura diretta; a questo punto, tuttavia, clonarli sarebbe del tutto inutile.

A Eve non rimane dunque che misurare direttamente i qubit intercettati.
Pertanto, Eve si inserisce nella trasmissione dei qubit e compie su ognuno di essi una misurazione, nello stesso modo che userà Bob dopo di lei: cioè scegliendo di misurare su ogni qubit le osservabili Z o X in base ai valori di una stringa di N bit estratti in modo pseudocasuale (la *stringa delle basi* di Eve). Anche qui, se il bit della stringa delle basi vale 0, Eve misura Z, mentre se vale 1 misura X.
Poi Eve manda i qubit a Bob, che effettua le sue misurazioni.

2.2.7 Individuazione dell'hacker

In questo modo, però, le misure di Eve modificano lo stato di alcuni dei qubit trasmessi. Ad esempio, si immagini il caso in cui Alice abbia preparato un qubit usando la base $\{|0\rangle,|1\rangle\}$. Se Eve misurasse sul qubit l'osservabile X, lo stato del qubit cambierebbe: da autostato di Z, esso collasserebbe in un autostato di X. Si ipotizzi poi che Bob effettui la stessa scelta della base di Alice, misurando l'osservabile Z. Egli ora avrebbe il 50% di probabilità di ottenere un risultato *diverso* dal bit codificato da Alice! Questo perché la misura di Z su un qubit che si trova in un autostato di X restituisce con 50% di probabilità il risultato 0 e con il 50% di probabilità il valore 1 (risultato probabilistico).
Al momento del confronto tra Alice e Bob, dunque, essi possono constatare che, tra i bit correlati (e quindi non scartati), sono presenti risultati *diversi* dai bit codificati da Alice. Dato che essi si aspettavano che il 100% dei risultati non scartati fossero uguali ai relativi bit codificati, il rilevamento di risultati *diversi* può significare una cosa sola: che Eve ha intercettato i qubit e li ha misurati.

Alice e Bob possono quindi sapere con certezza che la loro chiave è stata intercettata. Pertanto, devono interrompere la trasmissione e cambiare il canale di comunicazione.

In questo sta la forza del protocollo: l'hacker può sempre intercettare i qubit trasmessi, ma allo stesso modo viene sempre inevitabilmente scoperto.
In realtà, ci sarebbe *una sola eventualità* in cui Eve può intercettare l'intera chiave *senza essere scoperta*: ovvero nel caso in cui la stringa delle basi di Eve coincidesse con la stringa delle basi di Alice. In questo caso, ogni misura di Eve restituirebbe un risultato certo, perché ella misurerebbe sempre l'osservabile di cui lo stato del qubit è autostato. In questo modo, non originerebbe alcuna modifica agli stati dei qubit, passando inosservata attraverso l'analisi del confronto tra le stringhe. Tuttavia, la probabilità che le basi di Alice ed Eve coincidano sempre decresce esponenzialmente con il numero N di bit codificati (cioè la lunghezza delle stringhe estratte). Dunque, ad Alice e Bob basta scegliere di trasmettere un numero N sufficientemente grande di qubit; in tal modo, la probabilità che Eve estragga una stringa delle basi uguale a quella di Alice è esponenzialmente smorzata.

2.2.8 Variazione del protocollo con l'Entanglement

Una interessante variante del protocollo BB84 è quella in cui Eve crea con ogni qubit intercettato uno *stato entangled*, tramite dei qubit da lei preparati nello stato $|0\rangle$. Dopodiché, per ogni coppia di qubit entangled, Eve manda uno dei due qubit a Bob, attende che lui faccia le misure, e poi misura lei stessa i propri qubit entangled. Come si mostrerà nei seguenti capitoli, in questo modo Eve ottiene gli stessi identici risultati che Bob ha ricavato dalle misurazioni. Tuttavia, Eve non potrà creare stati entangled con tutti i qubit trasmessi; soprattutto, anche in questo caso verrà scoperta, grazie al confronto tra le stringhe di Alice e Bob.

Nei seguenti capitoli, il protocollo BB84 viene analizzato dal punto di vista teorico (con il calcolo delle probabilità che entrano in gioco) e dal punto di vista pratico, cioè della sua implementazione prima su un simulatore, e poi sui reali device quantistici.
L'obiettivo delle implementazioni è quello di verificare che l'hacker introduca un effetto tangibile sui risultati delle misurazioni. In più, per avere un maggiore riscontro, vengono confrontati i risultati della simulazione e dell'esperimento reale con le aspettative teoriche. Infine, è stata anche messa alla prova la variante con l'Entanglement.

Capitolo 3

Simulazione

In questo capitolo viene realizzata una simulazione del protocollo BB84. L'obiettivo è scrivere un programma che implementi il protocollo, testandone il corretto funzionamento su un simulatore. Per fare questo, si deve studiare quale sia l'effetto introdotto dall'attacco dell'hacker, per poi verificare che i risultati del programma coincidano con quelli previsti. Se questo avviene, il programma funziona e può essere usato per l'esperimento sui computer quantistici.

3.1 Codice della simulazione

Il programma simula la codifica dei qubit, l'eventuale intercettazione da parte dell'hacker, e le misure operate sul singolo qubit.
Per rendere il codice maggiormente flessibile, sono state definite diverse funzioni a inizio programma; vengono poi eseguite soltanto alcune di esse, in base al tipo di simulazione che viene effettuata.

3.1.1 Misura dell'operatore X

```
# Implementation of the measurement
# in the {|+>, |->} basis:

def x_measure (quantumcircuit, qubit, cbit):
    quantumcircuit.h(qubit)
    quantumcircuit.measure(qubit, cbit)
    return quantumcircuit
```

Nel pacchetto Qiskit, l'unica osservabile misurabile su un qubit è *di default* solamente Z. Pertanto, per poter effettuare una misura dell'operatore X, bisogna costruire esplicitamente una funzione adatta allo scopo. Questo viene fatto inserendo nel circuito una H gate ed effettuando subito dopo una misura standard (ossia misurando l'osservabile Z). Z viene misurata utilizzando la funzione `measure()` presente nel pacchetto Qiskit; dunque nel circuito viene inserita l'operazione di misura, come un vero e proprio elemento circuitale (nella Figura 3.2 a pagina 26 sono mostrati alcuni esempi di circuiti in cui viene misurata X).
Applicare una porta H ed effettuare una misura di Z è equivalente a misurare l'operatore X.
Si può notare infatti che l'inserimento della gate H determina la trasformazione dello stato del qubit $|\psi\rangle$ nel seguente modo:

- se $|\psi\rangle$ inizialmente è un vettore della base $\{|0\rangle, |1\rangle\}$, dopo l'inserimento di H il vettore $|\psi\rangle$ sarà un vettore della base $\{|+\rangle, |-\rangle\}$;
- viceversa, se $|\psi\rangle$ appartiene alla base $\{|+\rangle, |-\rangle\}$, dopo l'inserimento di H sarà un vettore della base $\{|0\rangle, |1\rangle\}$.

Dunque la successiva misura di Z restituirà un risultato:

- probabilistico, se $|\psi\rangle$ era inizialmente un vettore della base $\{|0\rangle, |1\rangle\}$;
- certo, se apparteneva alla base $\{|+\rangle, |-\rangle\}$.

Più precisamente, per misurare X è necessario scomporre lo stato di partenza $|\psi\rangle$ in termini di autostati di X:

$$|\psi\rangle = \alpha\,|+\rangle + \beta\,|-\rangle \tag{3.1}$$

In questo modo, una misura di X può restituire due risultati diversi, uno con probabilità $|\alpha|^2$ e l'altro con probabilità $|\beta|^2$.
Applicando quindi l'operatore H allo stato $|\psi\rangle$, si ottiene:

$$H\,|\psi\rangle = \alpha H\,|+\rangle + \beta H\,|-\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{3.2}$$

Pertanto, ora, una misura di Z può restituire i risultati 0 con probabilità $|\alpha|^2$ o 1 con probabilità $|\beta|^2$, dunque con la stessa distribuzione di probabilità relativa ai risultati della misura di X.

3.1.2 Codifica dei bit di Alice in qubit

```
# Implementation of a function which builds up
# a single qubit circuit based on Alice's strings

def encoding_circuit_builder (alice_bits, alice_basis, nth_qubit):

    i = nth_qubit   # The number n associated to the n-th bit
                    # which Alice wants to encode

    encoding_circuit = QuantumCircuit(1,1)

    if alice_bits[i] == 0 and alice_basis[i] == 0:
        # Alice chooses {|0>, |1>} basis
        pass # Apply I (nothing happens)

    if alice_bits[i] == 1 and alice_basis[i] == 0:
        # Alice chooses {|0>, |1>} basis
        encoding_circuit.x(0) # Apply X-Gate (flip |0> to |1>)

    if alice_bits[i] == 0 and alice_basis[i] == 1:
        # Alice chooses {|+>, |->} basis
        encoding_circuit.h(0) # Apply H-Gate (change |0> to |+>)

    if alice_bits[i] == 1 and alice_basis[i] == 1:
        # Alice chooses {|+>, |->} basis
        encoding_circuit.x(0)
        encoding_circuit.h(0) # Apply X-Gate and H-Gate
                              # (so |0> goes in |->)

    encoding_circuit.barrier()

    return encoding_circuit
```

Questa funzione associa un qubit ad ogni singolo bit che Alice vuole trasmettere. Il bit da codificare viene confrontato con il rispettivo bit della stringa delle basi, cioè la stringa usata per decidere quale base di autostati deve essere utilizzata per preparare il qubit. A seconda dei valori assunti da questa coppia di bit, il relativo qubit viene preparato in un certo stato (come descritto in precedenza). Nella Tabella 3.1 è schematizzata la mappa della codifica:

Tabella 3.1: Mappa della codifica

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

Per preparare il qubit viene creato un circuito quantistico a singolo qubit, a cui viene associato un bit classico che servirà a registrare il risultato di una eventuale misura. Di default, in Qiskit lo stato iniziale del qubit è sempre $|0\rangle$. Per preparare il qubit in uno degli altri stati bisogna dunque inserire le gate X e/o H:

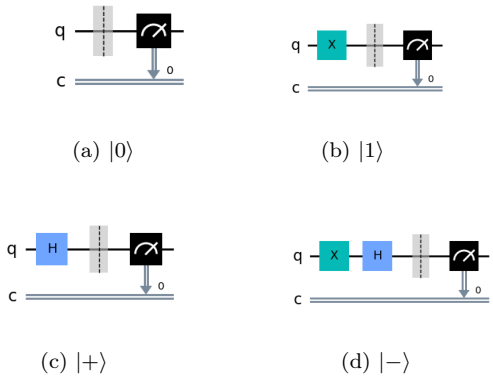
- lo stato $|1\rangle$ viene creato inserendo una X gate;
- lo stato $|+\rangle$ è ottenuto tramite una H gate;
- inserendo la X gate e la H gate in successione, si ha lo stato $|-\rangle$.

Dopo aver inserito una separazione (barriera), utile a scandire le diverse parti del circuito, la funzione ritorna in output il circuito stesso. Nella Figura 3.1 a pagina 25 e nella Figura 3.2 a pagina 26 sono riportati degli esempi di preparazione dello stato di un qubit.

3.1.3 Le misurazioni di Bob

```
# Implementation of the function with which Bob measures Alice's qubit

def circuit_measure (encoding_circuit, bob_basis, bob_measures, nth_qubit):

    i = nth_qubit       # The n-th qubit sent by Alice

    if bob_basis[i] == 0: # Bob chooses {|0>, |1>} basis
        # Measure with the default {|0>, |1>} basis
        encoding_circuit.measure(0,0)

        # To draw the circuit: encoding_circuit.draw("mpl")

# Now we run the circuit ONLY ONE TIME, and memorize
# the result in the bob_measures list.

        backend = Aer.get_backend("qasm_simulator")
        job = execute(encoding_circuit, backend, shots=1, memory=True)
        result = job.result()
        list_of_results = result.get_memory()
        bob_measures.append(list_of_results[0])

    if bob_basis[i] == 1: # Bob chooses {|+>, |->} basis
        # Measure with the {|+>, |->} basis
        x_measure(encoding_circuit, 0, 0)

        # encoding_circuit.draw("mpl")

# Now we run the circuit ONLY ONE TIME,
# and memorize the result in the bob_measures list.

        backend = Aer.get_backend("qasm_simulator")
        job = execute(encoding_circuit, backend, shots=1, memory=True)
        result = job.result()
        list_of_results = result.get_memory()
        bob_measures.append(list_of_results[0])
```

Nella realtà, il qubit inizializzato da Alice verrebbe trasmesso a Bob attraverso un canale di comunicazione. Nei programmi scritti in questa trattazione, invece, la trasmissione non viene implementata: semplicemente, Bob (e, come verrà mostrato, anche Eve) misura il qubit direttamente alla fine dello stesso circuito quantistico creato da Alice.
Bob utilizza la sua stringa di bit, associata alla scelta della base di autostati, per decidere quale operatore misurare. L'i-esimo qubit viene associato all'i-esimo bit della stringa: se esso vale 0, Bob misura l'operatore Z i cui autostati sono quelli della base $\{|0\rangle, |1\rangle\}$; viceversa, se il bit vale 1, Bob misura X, i cui autostati sono $\{|+\rangle, |-\rangle\}$. Nel primo caso, la misura di Z viene effettuata utilizzando semplicemente la funzione `measure()`. Nel secondo caso, viene chiamata la funzione sopra definita per misurare X: nel circuito quindi sono inserite una H gate e un'operazione di misura di Z. Si vedano ancora le Figure 3.1 e 3.2 per degli esempi con misure di Z e di X.
Viene poi simulato il circuito attraverso un backend fittizio, in questo caso *qasm_simulator*. Il circuito viene eseguito **una sola volta**, ed il risultato della misura viene scritto nell'i-esimo bit della stringa che raccoglie i risultati di Bob. In questo modo si ha che **l'i-esimo bit di tutte le stringhe utilizzate** (la stringa dei bit che Alice vuole trasmettere, le due stringhe associate alla scelta delle basi e la stringa dei risultati delle misure di Bob), **è associato allo stesso i-esimo qubit trasmesso**.



(a) $|0\rangle$ (b) $|1\rangle$

(c) $|+\rangle$ (d) $|-\rangle$

Figura 3.1: Preparazione dei qubit e misura dell'operatore Z (senza l'intervento di Eve)

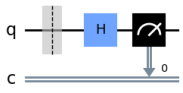
3.1.4 Attacco dell'hacker: misurazione del qubit intercettato

```
def eve_hacking_measure (hacker_activated, encoding_circuit, \
    eve_measures, nth_qubit):

    if hacker_activated == True:

    # Eve measures the n-th qubit sent by Alice.
    # After that, Eve sends it to Bob:

    i = nth_qubit       # The n-th qubit sent by Alice

    if eve_basis[i] == 0: # Eve chooses {|0>, |1>} basis
        # Measure with the default {|0>, |1>} basis
        encoding_circuit.measure(0,0)

        backend = Aer.get_backend("qasm_simulator")
        job = execute(encoding_circuit, backend, shots=1, memory=True)
        result = job.result()
        list_of_results = result.get_memory()
        eve_measures.append(list_of_results[0])

    if eve_basis[i] == 1: # Eve chooses {|+>, |->} basis
        # Measure with the {|+>, |->} basis
        x_measure(encoding_circuit, 0, 0)
        encoding_circuit.h(0)

        backend = Aer.get_backend("qasm_simulator")
        job = execute(encoding_circuit, backend, shots=1, memory=True)
        result = job.result()
        list_of_results = result.get_memory()
        eve_measures.append(list_of_results[0])

    return encoding_circuit

    else:
        pass
```

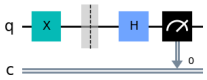
La procedura adottata dall'hacker è la stessa di Bob: su ogni qubit viene misurato l'operatore Z o l'operatore X in base ai valori di una stringa di bit (la strainga delle basi di Eve) che Eve ha, a sua volta, estratto in maniera casuale.
A livello di scrittura del codice, vi è una differenza tra le misure di Eve e quelle di Bob. Ovvero, dopo che Eve ha misurato l'operatore X con la funzione `x_measure()`, deve essere inserita nel circuito una ulteriore porta H. Questo deve essere fatto affinché lo stato del qubit, collassato in un autostato appartenente alla base $\{|0\rangle, |1\rangle\}$ a causa della misurazione di Z, venga poi riportato nello stato $|+\rangle$ o $|-\rangle$, come se fosse effettivamente collassato in uno di essi a seguito della misurazione dell'operatore X. Mentre nel caso delle misure di Bob questo accorgimento sarebbe stato inutile (non essendoci successive misurazioni nel circuito), nel caso delle misure di Eve è invece necessario. Infatti è proprio grazie al collasso dello stato del qubit in uno stato diverso da quello di partenza che Alice e Bob sono

in grado di scoprire l'hacker.

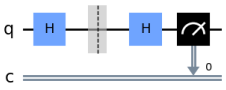
Ogni circuito viene simulato una sola volta. Anche Eve registra i risultati delle sue misurazioni in una propria stringa di bit. Finita l'operazione di misura, la funzione restituisce in output il circuito quantistico, che viene passato a Bob affinché compia le sue misurazioni.
E' presente una variabile booleana a inizio funzione: se la variabile è *True*, la funzione si attiva e viene simulato l'attacco hacker, in caso contrario la funzione non ha alcun effetto (allo stesso modo, si ha un controllo booleano anche per le altre funzioni dedicate alla simulazione dell'attacco di Eve).



(a) $|0\rangle$



(b) $|1\rangle$



(c) $|+\rangle$

(d) $|-\rangle$

Figura 3.2: Preparazione dei qubit e misura dell'operatore X (senza l'intervento di Eve)

3.1.5 Attacco dell'hacker: Entanglement

Questa funzione serve a simulare l'attacco dell'hacker nella variazione del protocollo in cui viene sfruttato l'Entanglement:

```
def eve_hacking_entangle (hacker_activated, encoding_circuit, nth_qubit):

    if hacker_activated == True:

    # Eve ENTANGLES the n-th qubit sent by Alice with a |0> state qubit.
    # After that, Eve sends the entangled qubit to Bob:

        eve_q = QuantumRegister(1, "eve_qubit")
        encoding_circuit.add_register(eve_q)

        encoding_circuit.cx(0, eve_q[0])
        encoding_circuit.barrier()

        # encoding_circuit.draw("mpl")

     return encoding_circuit
```

Con questo processo, Eve inserisce un qubit aggiuntivo nel circuito quantistico intercettato. In Qiskit questa operazione consiste nel creare un *registro quantistico* di un solo qubit, che nel codice viene chiamato *eve_ qubit*. Questo registro viene poi aggiunto al circuito, originando così un sistema di due qubit indipendenti.
Quindi viene inserita una $CNOT$ gate, il cui effetto è quello di applicare una X gate sul *target qubit* (in questo caso, il qubit di Eve) se il *control qubit* (cioè il qubit di Alice) si trova nello stato $|1\rangle$. Ad esempio:
Se lo stato del sistema di due qubit (non entangled) $|0\rangle$ e $|+\rangle$ è descritto da:

$$|0\rangle\,|+\rangle = |0+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |01\rangle) \tag{3.3}$$

allora, applicando al *target qubit* $|0\rangle$ una $CNOT$ gate, avente $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ come *control qubit*, si ha la trasformazione $|00\rangle \to |00\rangle$ e $|01\rangle \to |11\rangle$. Infatti, se si denota con κ il *control qubit* e con

τ il *target qubit*, lo stato prodotto diretto dei due sarà $|\tau\kappa\rangle = |0\kappa\rangle$ (in questo caso, perché $\tau = 0$). Dunque, se $\kappa = 0$, lo stato prodotto diretto non subisce trasformazioni, ma se $\kappa = 1$, il target qubit viene trasformato: $\tau = 0 \to \tau = 1$ (se τ fosse uguale a 1, si avrebbe allo stesso modo $\tau = 1 \to \tau = 0$). Tornando al caso dell'esempio, dunque, si ottiene:

$$CNOT\,|0+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{3.4}$$

che è uno stato entangled. Nella Figura 3.3 a pagina 28 è mostrato un esempio di circuito quantistico in cui viene applicata una $CNOT$ gate, in modo da creare lo stato entangled (3.4).

In questo modo, **per la metà dei qubit intercettati**, viene creato uno stato entangled con ognuno di essi ed uno di quelli di Eve. In particolare:

- se il control qubit è $|+\rangle$, si origina lo stato complessivo $CNOT\,|0+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$;
- se il control qubit è $|-\rangle$, si ottiene lo stato $CNOT\,|0-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$.

Questi sono stati entangled, perché non sono scrivibili come prodotto diretto di due stati indipendenti. Tuttavia:

- se il control qubit è $|0\rangle$, lo stato del sistema di due qubit è $|00\rangle$;
- se il control qubit è $|1\rangle$, lo stato che si ha è $|11\rangle$.

Ma questi **non sono stati entangled**, perché sono fattorizzabili nei prodotti diretti $|0\rangle\otimes|0\rangle$ e $|1\rangle\otimes|1\rangle$. Dunque Eve riesce a creare stati entangled **solo in 2 casi su 4**, ovvero nella metà dei casi.

Infine la funzione restituisce in output, come in precedenza, il circuito quantistico.

Di fatto però, quando Bob effettuerà le sue misure, considererà soltanto il qubit di Alice, senza sapere che, nel 50% dei casi, esso è in realtà un qubit entangled, legato ad un qubit che è rimasto in possesso dell'hacker Eve. L'obiettivo di quest'ultima è effettuare le misure sui qubit soltanto dopo che Bob avrà eseguito le sue. In questo modo, se Bob ha compiuto una misurazione su un qubit entangled, la misurazione di Eve sul suo stesso qubit entangled le restituirà lo stesso identico risultato ottenuto da Bob.
Per visualizzare intuitivamente questa conseguenza dell'Entanglement, si consideri ancora lo stato entangled:

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(|0\rangle_{Eve}\otimes|0\rangle_{Bob} + |1\rangle_{Eve}\otimes|1\rangle_{Bob}) \tag{3.5}$$

in cui sono stati indicati, a pedice, i due spazi di Hilbert distinti in cui vivono i due qubit: quello di Eve e quello di Bob.
Ora si supponga che Bob abbia effettuato una misurazione di Z (allo stesso modo si può ragionare misurando X), e che il suo qubit sia collassato nello stato $|1\rangle_{Bob}$. Ciò che ne consegue è:

$$\begin{aligned}
&{}_{Bob}\langle 1|\,[\frac{1}{\sqrt{2}}(|0\rangle_{Eve}\otimes|0\rangle_{Bob} + |1\rangle_{Eve}\otimes|1\rangle_{Bob})] = \\
&= \frac{1}{\sqrt{2}}(|0\rangle_{Eve}\otimes_{Bob}\langle 1|0\rangle_{Bob} + |1\rangle_{Eve}\otimes_{Bob}\langle 1|1\rangle_{Bob}) = \\
&= \frac{1}{\sqrt{2}}(|0\rangle_{Eve}\cdot 0 + |1\rangle_{Eve}\cdot 1) = \\
&= 0\cdot|0\rangle_{Eve} + \frac{1}{\sqrt{2}}\cdot|1\rangle_{Eve} \equiv \\
&\equiv |1\rangle_{Eve}
\end{aligned} \tag{3.6}$$

dove nell'ultimo passaggio la fase globale $\frac{1}{\sqrt{2}}$ è stata riassorbita nella definizione di $|1\rangle_{Eve}$. Ma questo significa che il qubit di Eve ha probabilità 0 di essere nello stato $|0\rangle_{Eve}$, e probabilità 1 di essere nello stato $|1\rangle_{Eve}$, identico a quello in cui è collassato il qubit di Bob.

Terminate le misurazioni di Bob, l'hacker esegue le sue proprie misure, simulate tramite la seguente funzione:

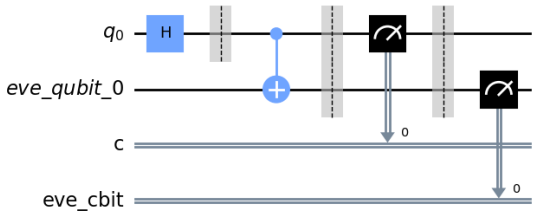


Figura 3.3: Creazione e misurazione di uno stato entangled tra $|+\rangle$ e $|0\rangle$

```
def eve_entangled_measurement (hacker_activated, encoding_circuit, \
eve_measures, nth_qubit):

    if hacker_activated == True:

    # Eve measures all her entangled qubits in the {|0>, |1>} basis:

        i = nth_qubit   # The number n associated to the n-th qubit

        eve_c = ClassicalRegister(1, "eve_cbit")
        encoding_circuit.add_register(eve_c)

        encoding_circuit.barrier()
        encoding_circuit.measure(1,1)

        # encoding_circuit.draw("mpl")

        backend = Aer.get_backend("qasm_simulator")
        job = execute(encoding_circuit, backend, shots=1, memory=True)
        result = job.result()
        list_of_results = result.get_memory()
        eve_measures.append(list_of_results[0])
```

Il processo di misura è lo stesso di quello sopra descritto, con la differenza che in questo caso Eve misura soltanto l'operatore Z. Per registrare l'output della misura, si inserisce nel circuito un *registro classico*, che corrisponde all'aggiunta di un nuovo bit classico su cui scrivere il risultato delle misurazioni effettuate sul qubit di Eve.

3.1.6 Esecuzione del programma ed export dei risultati

```
# Number of bits (and then qubits) that Alice is going to use:
number_of_qubits = 10000

# Alice generates n random bits (some of these bits will form the key)

alice_bits = []
for i in range (number_of_qubits):
    alice_bits.append(randint(0,1))

print("\nAlice's bits (first 20 bits):\n", alice_bits[0:19])

# Alice randomly chooses the bases in which she is going to measure

alice_basis = []
    for i in range (number_of_qubits):
    alice_basis.append(randint(0,1))
print("\nAlice's basis (first 20 bits):\n", alice_basis[0:19])

# Bob also randomly chooses the bases in which he is going to measure
```

```
bob_basis = []
    for i in range (number_of_qubits):
    bob_basis.append(randint(0,1))
print("\nBob's basis (first 20 bits):\n", bob_basis[0:19])
```

Dopo aver impostato il numero totale di qubit da preparare, cioè *number_of_qubits* (in questo caso fissato a **10000**), vengono estratte **3 stringhe di bit random**: la prima corrisponde ai bit che Alice vuole trasmettere (registrati in *alice_bits*), mentre la seconda (*alice_basis*) e la terza (*bob_basis*) sono le stringhe associate alla scelta delle basi, rispettivamente di Alice (per la codifica) e di Bob (per le misure). A titolo di controllo, per verificare che l'estrazione sia stata eseguita correttamente, vengono stampati i primi 20 bit di ogni stringa.

Nel programma è presente la possibilità di scegliere quale tipo di simulazione deve essere eseguita: ovvero se la trasmissione avviene:

1. Senza alcun tentativo di intercettazione da parte dell'hacker (**Scenario 1**);
2. Con un attacco dell'hacker basato sulla misurazione diretta dei qubit (**Scenario 2**);
3. Con un attacco basato sull'Entanglement (**Scenario 3**).

La scelta determina l'assegnazione del valore *True* alle variabili booleane che, a seconda del caso, attivano le funzioni che simulano la presenza di Eve:

```
print("\nChoose an option [digit 1, 2 or 3]:")
print("\n1. Transmission without hacker's attack")
print("\n2. Transmission with a measurement-based hacker's attack" \
"\n3. Transmission with an entanglement-based hacker's attack\n")
choice = input()

if choice == "1":
    hacker_activated1 = False
    hacker_activated2 = False
if choice == "2":
    hacker_activated1 = True
    hacker_activated2 = False
if choice == "3":
    hacker_activated1 = False
    hacker_activated2 = True
if choice != "1" and choice != "2" and choice != "3":
    print("\nTry again (digit only 1, 2 or 3)")
```

Nel caso in cui si sceglie di simulare l'attacco basato sulle misurazioni dei qubits di Alice da parte dell'hacker, viene estratta una ulteriore stringa di bit random (*eve_basis*), funzionali alla scelta della base di autostati che userà Eve nelle misure:

```
# Eve randomly chooses the bases in which
# she is going to measure (like Bob)
if hacker_activated1 == True:
    eve_basis = []
    for i in range (number_of_qubits):
        eve_basis.append(randint(0,1))
    print("\nEve's basis (first 20 bits):\n", eve_basis[0:19])
```

A questo punto, ha luogo la **simulazione vera e propria**:

```
# For each classical bit which Alice wants to encode
# and transmit to Bob, they proceed as it follows:

bob_measures = []
eve_measures = []

for n in range(number_of_qubits):

    # Alice codes the n-th bit of her initial string
    # as a qubit and sends it to Bob
    qubit = encoding_circuit_builder(alice_bits, alice_basis, n)

    # Bob measures the qubit with his own basis:
```

```
    # but what if Eve is hacking the message?
    eve_hacking_measure (hacker_activated1, qubit, eve_measures, n)
    eve_hacking_entangle (hacker_activated2, qubit, n)
    circuit_measure (qubit, bob_basis, bob_measures, n)
    eve_entangled_measurement (hacker_activated2, qubit, eve_measures, n)
```

Al variare dell'indice n in un ciclo for, che inizia con $n = 0$ e termina con $n = number\_of\_qubits - 1$, avviene la simulazione della **codifica**, della **trasmissione**, dell'eventuale **intercettazione** e della **misura** dei ***number_ of_ qubits* = *10000* qubit**, secondo la seguente procedura.

1. Viene costruito un solo qubit con la funzione `encoding_circuit_builder()`;
2. Vengono eventualmente chiamate le funzioni che implementano l'attacco di Eve, cioè `eve_hacking_measure()` o `eve_hacking_entangle()`, in base al valore assunto dalle relative variabili booleane;
3. È simulata la misurazione del qubit da parte di Bob;
4. Se è avvenuto un attacco basato sull'Entanglement, viene simulata la misurazione, da parte di Eve, del qubit entangled associato al qubit trasmesso.

In questo modo, si simula la trasmissione di **un solo qubit per volta**: Alice codifica un qubit, lo spedisce a Bob e lui effettua la sua misurazione, registrando il risultato. Eventualmente, per ogni qubit viene simulata anche l'azione di Eve.

Infine, sempre per verificare il corretto funzionamento del codice, vengono stampati i primi 20 bit delle stringhe associate ai risultati delle misurazioni. Il programma conclude la simulazione esportando un file di dati che contiene i valori dei bit delle stringhe *alice_bits*, *alice_basis*, *bob_basis* e *bob_measures*, ovvero rispettivamente i bit che Alice vuole trasmettere, le due stringhe associate alla scelta delle basi, e la stringa dei risultati delle misurazioni di Bob:

```
# Let's see the results of the measurements!

print("\nBob's measurements (first 20 measurements):\n")
print(bob_measures[0:19])

if hacker_activated1 == True or hacker_activated2 == True:
    print("\nEve's measurements (first 20 measurements):\n")
    print(eve_measures[0:19])

# Now we export the results in a text file:

data_file = open("bb84_data.txt", "w")

for i in range(number_of_qubits):
    data_file.write(str(alice_bits[i]))
    data_file.write("\t")
    data_file.write(str(alice_basis[i]))
    data_file.write("\t")
    data_file.write(str(bob_basis[i]))
    data_file.write("\t")
    data_file.write(str(bob_measures[i]))
    data_file.write("\n")

    data_file.close()

plt.show()
```

3.2 Attese teoriche

La possibilità di rilevare l'intercettazione dei bit trasmessi si fonda sull'**analisi dei dati ricavati dai risultati delle misurazioni di Bob**. E' fondamentale quindi stimare con quanta probabilità hanno luogo certi tipi di risultati, sia nel caso in cui la trasmissione avvenga senza tentativi di intromissione da parte dell'hacker, sia nel caso in cui, invece, venga messo in atto un attacco.
Ciò permette anche di distinguere, nel caso della simulazione, i tre diversi scenari implementati nel codice; tuttavia, durante un esperimento su hardware *reale* (non un simulatore, bensì un sistema quantistico vero e proprio), entra in gioco un nuovo fattore, molto importante nella trasmissione dell'informazione: il **rumore del device**. *Nella simulazione, al contrario, il rumore è totalmente assente*, e basta una minima variazione dei risultati dell'analisi dati dalle attese teoriche per poter affermare con sicurezza che il canale non è sicuro. In seguito si dovrà quindi stimare come il rumore influisca sulla possibilità di individuare l'intromissione dell'hacker.

Ad ogni modo, dato che il numero di prove è finito, le probabilità che verranno calcolate dall'analisi dei dati non saranno precisamente uguali a quelle attese, nemmeno nella simulazione. Infatti, per semplicità, in questa sezione si tratterà il limite asintotico per un numero infinito di prove, e dunque le previsioni sono da intendersi come valori ideali (comunque realistici per un numero di prove alto).

Innanzitutto, ci si aspetta che, quando viene estratto un bit **random** (per esempio, quando vengono estratti i valori delle stringhe associate alla scelta delle basi), esso assuma i valori 0 oppure 1 con la **stessa probabilità**. Questo accade perché la funzione `randint(0,1)` di default estrae 0 oppure 1 con una probabilità uniforme associata ad entrambi i valori del bit. In questo modo, si avrà che circa la metà dei qubit saranno preparati in un autostato di Z (in quanto associati al valore 0 di un bit della stringa), e che la restante metà sia preparata in un autostato di X (perché associata a bit di valore 1).
Dunque, *la probabilità di scegliere una certa base di autostati sarà del* ***50%***. Sia Alice che Bob quindi avranno il 50% di probabilità di usare una determinata base ed il relativo operatore. Ovvero, detta $P_A(0)$ la probabilità che Alice ha di estrarre la base associata a 0, chiamando $P_A(1)$ la probabilità che ha di estrarre la base associata a 1, e analogamente definendo le stesse probabilità $P_B(0)$ e $P_B(1)$ per Bob, si ha che:

$$P_A(0) = P_A(1) = P_B(0) = P_B(1) = 50\%. \tag{3.7}$$

In questo modo, è possibile calcolare la probabilità che i due scelgano la **stessa base**:

$$P_{stessa\ base} = P_A(0) \cdot P_B(0) + P_A(1) \cdot P_B(1) = \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2} = 50\% \tag{3.8}$$

Da cui, la probabilità di scegliere una **base diversa** è ancora:

$$P_{base\ diversa} = 1 - \frac{1}{2} = \frac{1}{2} = 50\% \tag{3.9}$$

Quanto appena discusso vale in tutti e tre i casi, sia quando Eve è presente, sia quando non c'è stato alcun tentativo di attacco.

3.2.1 Caso senza l'attacco hacker

Una volta avvenuta la trasmissione, Alice e Bob *confrontano bit per bit le stringhe associate alla scelta delle basi*, e **scartano i bit (e i qubit) relativi alla scelta di basi diverse**. In base a quanto discusso sopra, la metà dei qubit verrà quindi scartata, perché nel 50% dei casi saranno state estratte due basi differenti.
In seguito alla pubblicazione della prima metà dei bit originali di Alice, i bit utili alla trasmissione saranno soltanto la metà della metà (cioè $\frac{1}{4}$) dei bit veramente utilizzabili per lo scambio della chiave crittografica. Da ciò discende la necessità di dover usare **4N qubit per trasmettere in modo sicuro N bit**.

Si considerino ora i bit **non scartati**. In seguito, ci si riferirà ad essi indicandoli come *bit (o qubit) correlati* oppure come *bit (o qubit) relativi alla stessa scelta di base di autostati*. Infatti essi

sono quelli associati ad una uguale scelta della base di autostati da parte sia di Alice che di Bob. Questo significa che *Bob misura su ogni qubit l'operatore tale per cui il qubit si trova in un autostato dell'operatore stesso.* In altri termini, ***ogni singolo qubit è stato preparato da Alice in un autostato di un certo operatore, e Bob misura su ogni qubit proprio quello stesso operatore***.

La misura di un operatore su un suo autostato restituisce un risultato **certo**. Vale la pena enfatizzare il significato di questa affermazione: essa significa che, *preso un sistema costituito da un numero elevato di qubit, tutti preparati nello stesso autostato dell'operatore generico A, e misurando su ogni qubit lo stesso operatore A,* ***il risultato sarà uguale per tutte le misurazioni***. Statisticamente, dunque, il risultato della misura dell'operatore A su un suo autostato sarà **sempre** un ben determinato valore. Ovvero, **la probabilità** che tale misura dia, come risultato, quel preciso valore, **è pari a 1** (risultato **certo**).

Ciononostante, nel caso di Bob, lui **non può** sapere a priori quali dei suoi risultati sono certi. Infatti, quando Bob riceve un qubit, può misurare *solamente quel qubit*, non un sistema di 10000 copie di tale qubit, tutte preparate nello stesso stato!
L'unico modo per poter sapere che il risultato è **certo**, senza dovere misurare ogni volta 10000 qubit tutti uguali, è capire se la misura dell'operatore A sia stata effettuata su un qubit preparato in un autostato dello stesso operatore A. È il caso dei *qubit correlati*: se Alice e Bob hanno effettuato la stessa scelta della base di autostati, allora Bob ha misurato l'operatore su un qubit preparato in un autostato di quell'operatore. Dunque il risultato è certo: è un ben preciso valore, con probabilità 1.
Ecco perché Alice e Bob confrontano le stringhe delle basi: in questo modo, sono sicuri che ad ogni coppia di *bit correlati* corrisponderà un risultato **certo** della misura di Bob; ma soprattutto, dato che è un risultato certo, esso sarà **identico** al bit codificato da Alice (questo perché Alice ha appositamente preparato il qubit in modo che, se su di esso viene misurato quello specifico operatore, il risultato della misura sarà con certezza il valore di quel bit).

Pertanto, quando Alice pubblica la **prima metà dei suoi bit originali**, e Bob li confronta uno ad uno con la **prima metà dei bit da lui ottenuti con le misurazioni**, ci si aspetta che le due stringhe di bit confrontate siano **perfettamente identiche**.
Per quantificare tale attesa, si definisce la probabilità di trovare identici gli i-esimi bit delle due stringhe (condizionata dal fatto di aver considerato solo i qubit per cui la codifica e la misurazione sono state associate alla stessa scelta di base) nel seguente modo:

$$P_{identici} = \frac{Numero\ di\ bit\ identici}{Numero\ di\ bit\ considerati} \tag{3.10}$$

Si noti che il numero di bit considerati è la metà del numero dei qubit spediti, perché solo la metà dei qubit è relativa alla stessa scelta della base di autostati.

La probabilità di ottenere dei risultati differenti dai bit originali di Alice, nonostante la stessa scelta della base, è di conseguenza:

$$P_{diversi} = 1 - P_{identici} \tag{3.11}$$

In base a quanto affermato sopra, il risultato atteso è:

- $P_{identici} = 100\%$
- $P_{diversi} = 0\%$

Infine, per distinguere le probabilità relative ai diversi risultati ottenuti da Bob, si definiscono le seguenti probabilità del tipo $P_i^{identici}(j)$, con $i, j = 0, 1$, tali che:

- $P_0^{identici}(0)$ = probabilità di ottenere il risultato 0 avendo scelto la base associata al bit 0
- $P_0^{identici}(1)$ = probabilità di ottenere 1 con la base associata al bit 0
- $P_1^{identici}(0)$ = probabilità di ottenere 0 con la base associata al bit 1
- $P_1^{identici}(1)$ = probabilità di ottenere 1 con la base associata al bit 1

contando il numero di risultati di un certo valore (0 oppure 1) misurati con un certo operatore (Z oppure X), associato ad una delle due diverse basi di autostati, e dividendo per il numero di qubit considerati (quelli non scartati).

In pratica, ognuna di queste probabilità corrisponde ad aver misurato il qubit in uno dei quattro stati $|0\rangle$, $|1\rangle$, $|+\rangle$ o $|-\rangle$ dopo aver operato la stessa scelta della base, trovando un risultato identico al bit codificato. Ad entrambe le scelte di base, corrispondono due possibili risultati; perciò le probabilità sopra definite sono tutte pari a:

$$P_i^{identici}(j) = P_{base\ i} \cdot P_{risultato\ j} \cdot P_{identici} = 50\% \cdot 50\% \cdot 100\% = \frac{1}{2} \cdot \frac{1}{2} \cdot 1 = \frac{1}{4} = 25\% \tag{3.12}$$

Il fatto di aver scelto di considerare soltanto i risultati identici ai rispettivi bit codificati da Alice è utile per controllare, di nuovo, che non ci siano perturbazioni nei risultati introdotte da un eventuale attacco hacker. In questo modo, infatti, se dall'analisi dati risulterà che tali probabilità sono tutte pari a 25%, si avrà un'ulteriore conferma che, ad una scelta della stessa base di autostati, corrispondono con certezza risultati identici ai bit codificati (infatti, se sommate, esse danno una probabilità totale del 100%).

Ciò permette di definire le analoghe probabilità di ottenere un certo risultato, misurando un certo operatore, avendo però ottenuto un risultato *diverso* dal rispettivo bit codificato da Alice. Queste probabilità, indicabili per esempio con $P_i^{diversi}(j)$, con $i, j = 0, 1$, sono tutte **nulle** se i risultati di Bob sono del tutto identici ai bit codificati da Alice. Tuttavia, se così non fosse, queste probabilità (la cui somma deve dare esattamente $P_{diversi}$) sarebbero non nulle, ed indicherebbero l'avvenuta intercettazione dei qubit trasmessi.

$$P_i^{diversi}(j) = P_{base\ i} \cdot P_{risultato\ j} \cdot P_{diversi} = 50\% \cdot 50\% \cdot 0\% = \frac{1}{2} \cdot \frac{1}{2} \cdot 0 = 0\% \tag{3.13}$$

Tutto quanto è stato appena illustrato vale nel caso in cui la trasmissione avvenga senza interferenze da parte di un agente esterno.
Si ipotizzi ora che Eve abbia intercettato i qubit spediti da Alice.

3.2.2 Attacco basato sulla misurazione diretta dei qubit

In questo caso, **Eve misura l'operatore Z o l'operatore X sul qubit intercettato**, in base ai valori dei bit della sua stringa associata alla scelta delle basi di autostati.
Così facendo, *lo stato del qubit collassa in un autostato dell'operatore misurato*, e ciò determina l'insorgere di risultati inattesi nel momento in cui Bob effettua le sue misure sui qubit già passati tra le mani di Eve.

Possono avere origine quattro situazioni:

1. Alice prepara il qubit nell'autostato appartenente alla base associata ad un generico operatore A. Eve misura lo stesso operatore, e anche Bob misura l'operatore A;
2. Alice prepara il qubit nell'autostato di A, Eve misura A, ma Bob misura l'operatore B, diverso da A;
3. Alice prepara il qubit nell'autostato di A; tuttavia, sia Eve che Bob misurano l'operatore B;
4. Alice prepara il qubit nell'autostato di A, Eve misura B, ma Bob misura l'operatore A.

I punti **2.** e **3.** possono essere **trascurati**: infatti, in questi casi, una volta che Alice e Bob pubblicano le rispettive stringhe associate alla scelta delle basi, i due capiscono di aver scelto operatori *diversi* per effettuare la codifica e la misura. Pertanto, ogni qubit che ricade in uno dei due casi indicati, e i relativi bit delle stringhe, vengono **scartati a priori**. I casi rilevanti sono quindi soltanto due: quelli descritti dai punti **1.** e **4.**

Nel caso **1.**, la misura di Eve **non cambia lo stato del sistema**. Ad esempio, se Alice aveva preparato il qubit nello stato $|0\rangle$ (oppure in $|1\rangle$), autostato dell'operatore Z, quando Eve misura Z, il qubit rimane nello stesso autostato $|0\rangle$ (oppure $|1\rangle$). Infine, quando Bob misura a sua volta l'operatore Z, egli ottiene un risultato certo, che è lo stesso di Eve ed è anche lo stesso di Alice. Dunque, se Alice pubblica il bit codificato con il rispettivo qubit nello stato $|0\rangle$, Bob non può fare altro che constatare di aver ottenuto un risultato identico al bit che Alice aveva codificato. Ed è veramente così; soltanto che, ora, anche Eve è in possesso di quel bit e, al pari di Bob, sa che è associato ad una misura che restituisce un risultato certo.
Essendo questo un caso sui due possibili, la probabilità associata ad esso è del **50%**: ovvero **la probabilità che Eve riesca a misurare senza perturbare lo stato del qubit** (e dunque senza farsi scoprire) **è pari a 50%**.

Il punto **4.** è quello fondamentale, su cui si fonda la sicurezza del protocollo. Eve in questo caso *sbaglia la scelta dell'operatore da misurare*, e dunque **lo stato del qubit collassa in un autostato non dell'operatore** A (come quello in cui era stato preparato da Alice), **ma dell'operatore** B **misurato da Eve**. Anche in questo caso, la probabilità che ciò avvenga è del **50%**. Tuttavia, quando Bob effettua la sua misurazione con l'operatore A, dato che ora il qubit si trova in un autostato dell'operatore B, il risultato che otterrà non sarà certo, ma **probabilistico**.

Come sopra, anche qui è utile chiarire cosa si intende con ***risultato probabilistico***:
Preso un sistema costituito da un elevato numero di qubit, preparati tutti in uno stesso autostato dell'operatore A, e misurando su ognuno di essi ***l'operatore*** *B* ***diverso da*** *A,* ***i risultati delle misure non saranno tutti uguali fra loro****. Al contrario, i risultati di ogni misura potranno assumere* ***due valori diversi*** *(0 e 1 per esempio). Contando quante misure danno come risultato il primo valore, e quante il secondo, è possibile calcolare la* ***probabilità*** *con cui occorrono i due diversi risultati.*
Il termine *probabilistico* pertanto descrive il carattere aleatorio del risultato di queste misure, in contrasto con il temine *certo* che definisce l'impossibilità di ottenere due risultati diversi.

Tornando al caso del protocollo, si può dire che, equivalentemente, su k qubit che hanno subito questo processo (preparazione dell'autostato di A, misura di B e misura di A), Bob otterrà un numero m di risultati di valore 0, ed un numero $l = k - m$ di risultati di valore 1, pur misurando lo stesso operatore di Alice.
La cosa notevole è che, se non ci fosse stata l'intromissione di Eve, che ha cambiato lo stato del sistema, il risultato di Bob sarebbe dovuto essere *certo*: su k qubit preparati in un autostato di A e su cui è stato misurato l'operatore A, Bob avrebbe dovuto ottenere un numero k di risultati di valore 0, e 0 risultati di valore 1 (o, equivalentemente, k risultati di valore 1, e 0 risultati di valore 0). È questa differenza nei risultati che permette di determinare la presenza dell'hacker.

Per quantificare le probabilità associate ai due risultati, devono essere considerate le relazioni tra gli stati in cui può essere preparato il qubit:

$$\begin{aligned} |+\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |-\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\ |0\rangle &= \frac{1}{\sqrt{2}}(|+\rangle + |-\rangle) \\ |1\rangle &= \frac{1}{\sqrt{2}}(|+\rangle - |-\rangle) \end{aligned} \tag{3.14}$$

Per come sono stati definiti questi stati, la misura dell'operatore A su un autostato dell'operatore B dà come risultato 0 oppure 1 con **probabilità del 50% ciascuno**.

Non solo: quando Alice e Bob confrontano i bit codificati e i risultati ottenuti da Bob nei casi descritti dal punto 4., scoprono che il **50%** delle volte il risultato è **diverso** dal bit codificato; *tuttavia, la restante metà delle volte trovano che il bit ed il risultato sono* ***identici****.* Questi non sono altro che i casi *fortuiti* in cui, pur essendoci stata l'intrusione di Eve, Bob ottiene un risultato sì probabilistico, ma di fatto identico a quello del bit che Alice aveva codificato. Eve in questo frangente ha fortuna, e

di nuovo non viene scoperta; ma *non nel caso in cui i risultati differiscono dai bit codificati.*

Calcolando, come nella parte relativa alla trasmissione senza tentativo di attacco, la probabilità di ottenere risultati identici e quella di ottenere risultati diversi, a condizione che siano considerati soltanto i qubit corrispondenti alla stessa scelta della base di autostati, si ottiene:

- $P_{identici} = P_1 + P_4 \cdot P_{stesso\ risultato} = \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{3}{4} = 75\%$
- $P_{diversi} = P_4 \cdot P_{risultato\ diverso} = 1 - P_{identici} = \frac{1}{4} = 25\%$

dove:

- $P_1 = 50\%$ = probabilità che si verifichi il caso descritto nel punto 1;
- $P_4 = 50\%$ = probabilità del caso descritto nel punto 4.
- $P_{stesso\ risultato} = P_{risultato\ diverso} = 12,5\%+12,5\% = 50\%$ = probabilità di ottenere un risultato identico o uno diverso dal bit codificato da Alice, **considerando il solo punto 4.** (in cui la misura di Eve cambia lo stato del qubit)

Si faccia riferimento allo **schema di Figura 3.4** per visualizzare più chiaramente il ragionamento.

Si noti che queste probabilità sono le stesse indipendentemente dal fatto che si identifichino gli operatori A o B con Z o X: questo è dovuto al fatto che è possibile ripetere lo stesso ragionamento del punto 4. scambiando A con B, e non cambierebbe nulla. Dunque, sia che si consideri la base $\{|0\rangle, |1\rangle\}$ (ed il relativo operatore Z), sia che si consideri la base $\{|+\rangle, |-\rangle\}$ (associata a X), si ha sempre che $P_{identici} = 75\%$ e che $P_{diversi} = 25\%$. Questo, come si vedrà, non è valido nel caso dell'attacco basato sull'Entanglement.

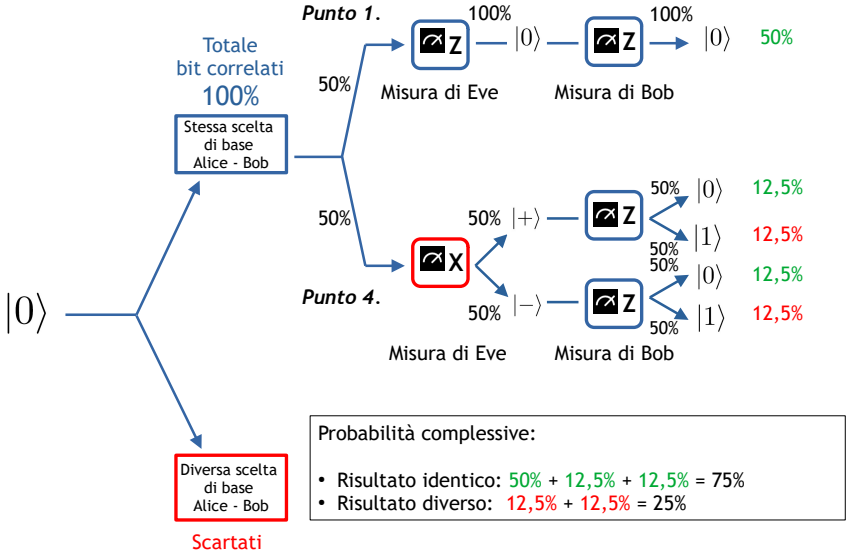


Figura 3.4: Schema per il calcolo delle probabilità (Scenario 2). Come esempio è stato usato il qubit $|0\rangle$, ma gli altri tre possibili stati sono del tutto equivalenti. Si noti inoltre che il totale a cui si riferiscono le percentuali **non** è il numero dei bit trasmessi, ma soltanto il **numero di bit correlati**. Dunque per il calcolo **non** viene presa in considerazione la divisione tra il caso con stessa scelta di base ed il caso con scelta differente.

Come in precedenza, possiamo definire le probabilità di ottenere un certo valore (0 oppure 1), dopo aver misurato con un certo operatore (Z oppure X), ed avere trovato tale risultato identico al bit

codificato da Alice: $P_i^{identici}(j)$ con $i, j = 0, 1$. Senza l'interazione di Eve, si aveva:
$P_i^{identici}(j) = 25\% \quad \forall i, j$
Ad esempio, in assenza dell'hacker, il 25% delle volte Bob otteneva il risultato 0 misurando l'operatore associato alla base 0, e nel 100% dei casi i suoi risultati coincidevano con i corrispondenti bit codificati da Alice nella base 0.
Ora, nel 25% dei casi, Bob otterrà sempre il risultato 0 effettuando la misura dell'operatore associato alla base 0; tuttavia, si avrà che, mentre il 75% di questi risultati saranno ancora identici ai bit codificati da Alice, per il restante 25% saranno invece ***diversi***. Lo stesso vale per gli altri casi, riassunti dagli indici $i, j = 0, 1$:

- $P_i^{identici}(j) = P_{base\ i} \cdot P_{risultato\ j} \cdot P_{identici} = 50\% \cdot 50\% \cdot 75\% = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{3}{4} = \frac{3}{16} = 18,75\%$
- $P_i^{diversi}(j) = P_{base\ i} \cdot P_{risultato\ j} \cdot P_{diversi} = 50\% \cdot 50\% \cdot 25\% = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{4} = \frac{1}{16} = 6,25\%$

3.2.3 Attacco basato sull'Entanglement

Come mostrato nella parte riguardante il codice della simulazione, *Eve riesce a creare stati entangled* ***soltanto con la metà dei qubit trasmessi da Alice***. Dunque la probabilità di creare stati entangled è pari al **50%**. Non solo: Eve crea stati entangled tramite dei qubit da lei preparati unicamente nello stato $|0\rangle$.
L'azione di Eve pertanto si applica in due sole situazioni:

1. Alice prepara il qubit nello stato $|+\rangle$
2. Alice prepara il qubit nello stato $|-\rangle$

ovvero nel solo caso in cui Alice sceglie la base $\{|+\rangle, |-\rangle\}$.

Nel primo caso, lo stato entangled creato da Eve è $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, nel secondo è $\frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$.
Si prenda in esame la **situazione 1.** (la 2. è del tutto analoga). Quando Bob misura il qubit entangled speditogli da Eve, si aprono ancora **due possibilità** (ciascuna con il **50%** di probabilità):

1. Bob misura l'operatore Z;
2. Bob misura l'operatore X.

Nel primo caso, il risultato viene **scartato**, in quanto misurare l'operatore Z significa aver scelto la base $\{|0\rangle, |1\rangle\}$, mentre Alice aveva scelto la base $\{|+\rangle, |-\rangle\}$.
Nel secondo, la base scelta è la stessa sia per Bob che per Alice, ovvero $\{|+\rangle, |-\rangle\}$; dunque il risultato viene preso in considerazione. Tuttavia, il qubit si trova nello stato entangled $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. In termini degli autostati $|+\rangle$ e $|-\rangle$, esso è:

$$\begin{aligned}
&\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|1\rangle) = \\
&= \frac{1}{\sqrt{2}}[\frac{1}{\sqrt{2}}(|+\rangle + |-\rangle)(\frac{1}{\sqrt{2}}(|+\rangle + |-\rangle) + \frac{1}{\sqrt{2}}(|+\rangle - |-\rangle)(\frac{1}{\sqrt{2}}(|+\rangle - |-\rangle)] = \\
&= \frac{1}{\sqrt{2}}[\frac{1}{2}(2 \cdot |++\rangle + 2 \cdot |--\rangle)] = \\
&= \frac{1}{\sqrt{2}}(|++\rangle + |--\rangle)
\end{aligned} \tag{3.15}$$

Dunque, quando Bob effettua la misurazione di X sul suo qubit, ottiene il risultato 0 il 50% delle volte, e trova 1 per la restante parte delle misure: il risultato è ***probabilistico***, *nonostante Alice e Bob abbiano scelto la stessa base di autostati.*
Quindi, nel 50% dei casi, il risultato sarà identico al bit codificato da Alice, mentre negli altri casi sarà differente.

Per calcolare le probabilità $P_{identici}$ e $P_{diversi}$, si deve quindi **distinguere tra la scelta della base** $\{|0\rangle, |1\rangle\}$ **e la scelta della base** $\{|+\rangle, |-\rangle\}$; se Alice e Bob scelgono la base $\{|0\rangle, |1\rangle\}$, Eve **non riesce a creare stati entangled**, per cui in questo caso ci si aspetta che tutto avvenga *come se non*

ci fosse alcun attacco hacker.

Eve riesce a creare stati entangled **solo se sia Alice che Bob scelgono la base** $\{|+\rangle, |-\rangle\}$.
Le altre eventualità, in cui Alice e Bob scelgono basi diverse (per esempio, nel punto 2., Alice sceglie la base $\{|+\rangle, |-\rangle\}$, ma Bob sceglie $\{|0\rangle, |1\rangle\}$) **non influiscono sui conti**: infatti, avendo scelto una base diversa, i bit e i qubit corrispondenti alla misura di Z da parte di Bob vengono scartati a priori, mentre le probabilità di interesse sono calcolate sempre sul numero di bit per cui si ha la stessa scelta di base (*bit correlati*).

Dunque:

- $P_{identici} = P_{base\ \{|0\rangle,|1\rangle\}} \cdot P^{base\ \{|0\rangle,|1\rangle\}}_{stesso\ risultato} + P_{base\ \{|+\rangle,|-\rangle\}} \cdot P^{base\ \{|+\rangle,|-\rangle\}}_{stesso\ risultato} =$

  $= 50\% \cdot 100\% + 50\% \cdot 50\% = \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot \frac{1}{2} = \frac{3}{4} = 75\%$

- $P_{diversi} = P_{base\ \{|0\rangle,|1\rangle\}} \cdot P^{base\ \{|0\rangle,|1\rangle\}}_{risultato\ diverso} + P_{base\ \{|+\rangle,|-\rangle\}} \cdot P^{base\ \{|+\rangle,|-\rangle\}}_{risultato\ diverso} =$

  $= 50\% \cdot 0\% + 50\% \cdot 50\% = \frac{1}{2} \cdot 0 + \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4} = 25\%$

Con riferimento alla parte in cui i qubit non sono stati resi entangled, essi hanno il 100% di probabilità di presentare risultati identici ai bit codificati, e 0% di probabilità di portare a risultati diversi (naturalmente, scartando i qubit relativi ad una diversa scelta della base).
Viceversa, quando i qubit sono stati resi entangled, nel 50% dei casi i risultati saranno uguali ai bit codificati, mentre nel restante 50% saranno diversi.
Per maggior chiarezza, si faccia riferimento allo schema di **Figura 3.5**.

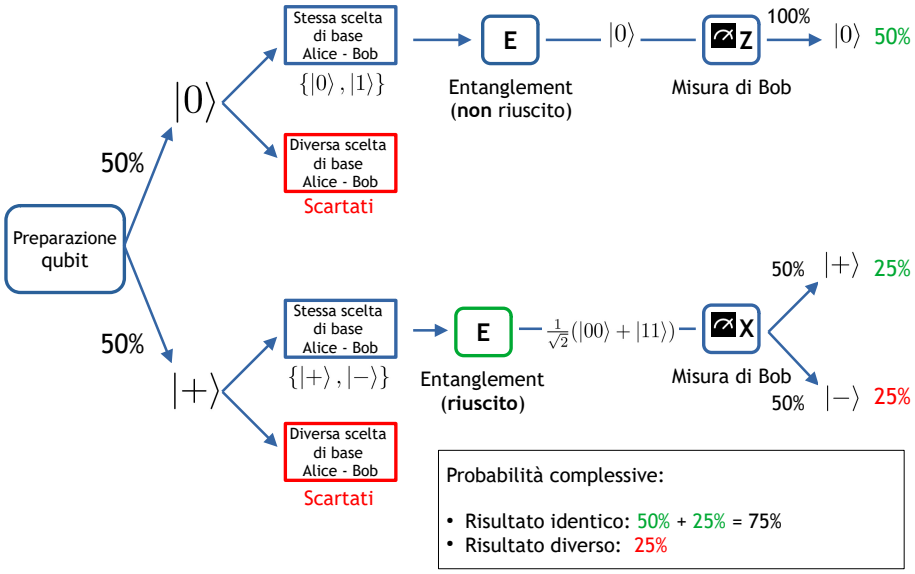


Figura 3.5: Schema per il calcolo delle probabilità (Scenario 3). In questo caso viene distinta la scelta della base $\{|0\rangle, |1\rangle\}$ da quella della base $\{|+\rangle, |-\rangle\}$, perché, come discusso in precedenza, esse **non sono equivalenti** (nel secondo caso vengono formati stati entangled, nel primo no). Si noti che, come esempio, sono stati considerati gli stati $|0\rangle$ e $|+\rangle$; tuttavia essi **sono equivalenti** rispettivamente agli stati $|1\rangle$ e $|-\rangle$, per cui la distribuzione di probabilità è la stessa. Ancora, i qubit relativi ad una diversa scelta di base non sono stati considerati, e non rientrano nel calcolo delle probabilità.

Per il calcolo delle $P_i^{identici}(j)$ e delle $P_i^{diversi}(j)$, è sufficiente distinguere i quattro casi che si originano dalla preparazione degli stati $|0\rangle$, $|1\rangle$, $|+\rangle$ e $|-\rangle$. Facendo riferimento allo schema di **Figura 3.5**: se, invece di distinguere solamente tra le due possibili scelte di basi, si considerassero i quattro diversi

stati, si avrebbe che ognuno di essi viene preparato con il 25% di probabilità. È come se, nello schema, agli stati esemplificativi $|0\rangle$ e $|+\rangle$ venissero affiancati anche gli stati $|1\rangle$ e $|-\rangle$. Dunque il primo ramo del diagramma presenterebbe una probabilità del 25% (mentre prima era del 50%).

Pertanto, si ottiene che, scegliendo la base $\{|0\rangle, |1\rangle\}$ (detta *base 0*), i risultati 0 e 1 sono sempre identici ai bit codificati, con probabilità del 25% sul totale dei bit correlati; scegliendo la base $\{|+\rangle, |-\rangle\}$ (detta *base 1*), invece, i bit 0 e 1 avranno uguale probabilità di essere identici o diversi dai bit codificati, con probabilità associata del 12,5%:

- $P_0^{identici}(j) = 25\%$
- $P_0^{diversi}(j) = 0\%$
- $P_1^{identici}(j) = 12,5\%$
- $P_1^{diversi}(j) = 12,5\%$

$\forall j = 0, 1$.

Questo significa anche che i risultati dovuti alla misura dell'**operatore** Z (i cui autostati sono quelli della base 0, cioè $\{|0\rangle, |1\rangle\}$) **non presentano cambiamenti** dovuti all'azione di Eve; al contrario, **la misura di** X presenta uguali probabilità di ottenere il risultato (0 o 1) **sia identico che diverso** dal bit codificato da Alice.

Questo è un modo non solo per capire che Eve ha intercettato i qubit, ma anche per **distinguere il tipo di attacco messo in atto**. Il fatto che l'analisi dati dia risultati **diversi** per le misure di Z e per quelle di X, infatti, denota il fatto che **Eve ha creato stati entangled**, legando i qubit di Alice e dei qubit (di Eve) preparati in un autostato di Z.

3.2.4 Commento sull'efficacia degli attacchi di Eve

In entrambi gli scenari in cui vi è un attacco hacker, Eve può essere scoperta grazie all'analisi dei risultati delle misure di Bob; questo fa sì che il protocollo BB84 sia estremamente sicuro, a patto comunque di trasmettere una chiave abbastanza lunga. Vi è infatti una sola eventualità, relativa allo Scenario 2, in cui Eve può intercettare l'intera chiave **senza introdurre perturbazioni** nei risultati ottenuti da Bob: ovvero *il caso in cui la stringa delle basi di Eve coincide con la stringa delle basi di Alice.* Tuttavia, questo evento è tanto **meno probabile quanto più la chiave** (e dunque ogni stringa di bit ad essa associata) **è lunga**. La probabilità associata a questo caso tende a zero esponenzialmente, al crescere della lunghezza della chiave.

3.2.5 Tabelle dei valori attesi

Si riportano ora, per chiarezza, i risultati dei calcoli delle precedenti sezioni. Con "Scenario 1." si intende il caso senza hacker; con "Scenario 2" ci si riferisce all'attacco basato sulla misurazione diretta dei qubit, mentre con "Scenario 3." si indica l'attacco basato sull'Entanglement. Si ricorda anche che:

- $P_i(j)$ indica $P_{base}(risultato)$, dove $i, j = 0, 1$;
- la *base 0* è la base $\{|0\rangle, |1\rangle\}$, la *base 1* è la base $\{|+\rangle, |-\rangle\}$;
- il *risultato* è il valore restituito dal qubit dopo la misura, e può essere 0 oppure 1.

Tabella 3.2: Attese teoriche - Probabilità complessive (%)

Scenario	$P_{stessa\ base}$	$P_{base\ diversa}$	$P_{identici}$	$P_{diversi}$
1	50	50	100	0
2	50	50	75	25
3	50	50	75	25

Tabella 3.3: Attese teoriche - Probabilità dei singoli risultati $P_i(j)$ (%)

	$P_i^{identici}(j)$				$P_i^{diversi}(j)$			
Scenario	$P_0(0)$	$P_0(1)$	$P_1(0)$	$P_1(1)$	$P_0(0)$	$P_0(1)$	$P_1(0)$	$P_1(1)$
1	25	25	25	25	0	0	0	0
2	18,75	18,75	18,75	18,75	6,25	6,25	6,25	6,25
3	25	25	12,25	12,25	0	0	12,25	12,25

3.3 Run delle simulazioni e analisi dati

A questo punto, si può eseguire il programma e analizzarne i risultati. Per ognuno dei tre scenari possibili, sono state effettuate **10 simulazioni** e altrettante analisi dati. Ogni run del programma ha simulato la trasmissione di **10000** qubit. Le probabilità relative ad uno stesso scenario sono state calcolate tramite la media delle probabilità ricavate da ogni singola simulazione, e come errore su tale media è stata scelta la deviazione standard divisa per $\sqrt{10}$. Tale errore verrà indicato con σ.

3.3.1 Scenario 1: assenza dell'hacker

Tabella 3.4: Simulazioni - Probabilità dello scenario in cui non viene messo in atto alcun attacco (%)

	$P_{stessa\ base}$	σ	$P_{base\ diversa}$	σ	$P_{identici}$	σ	$P_{diversi}$	σ
Complessive	50,1	0,1	49,9	0,1	100	0	0	0
$P_i(j)$	$P_0(0)$	σ	$P_0(1)$	σ	$P_1(0)$	σ	$P_1(1)$	σ
Risultati identici	25,5	0,3	24,8	0,3	24,8	0,2	24,8	0,4
Risultati diversi	0	0	0	0	0	0	0	0

I risultati delle simulazioni mostrano una piena compatibilità con le attese teoriche. Si noti che **non vi è alcun errore nella decodifica dei bit** (per esempio, la probabilità di trovare tutti i risultati non scartati esattamente identici ai bit codificati da Alice è pari a 100%, con massima precisione). Su un harware reale, non vi saranno risultati così esatti a causa del rumore.

3.3.2 Scenario 2: attacco basato sulle misurazioni dirette

Nel caso dell'attacco basato sulle misurazioni dirette sui qubit spediti da Alice, si vede subito che ci sono dei risultati diversi dai bit codificati da Alice. Le probabilità rispecchiano quelle attese; inoltre, le probabilità associate ai diversi valori misurati da Bob sono uniformi, sia nel caso di risultati identici (teoricamente 18,75%), che nel caso di risultati diversi (teoricamente 6,25%).

3.3.3 Scenario 3: attacco basato sull'Entanglement

In questo ultimo caso, è possibile capire che Eve ha creato degli stati entangled con i qubit di Alice: anche soltanto guardando i grafici, si nota che, mentre per le misurazioni dell'operatore Z, i risultati

Tabella 3.5: Simulazioni - Probabilità dello scenario in cui viene messo in atto l'attacco basato sulle misurazioni dirette dei qubit (%)

	$P_{stessa\ base}$	σ	$P_{base\ diversa}$	σ	$P_{identici}$	σ	$P_{diversi}$	σ
Complessive	49,9	0,1	50,0	0,2	75,2	0,2	24,8	0,2
$P_i(j)$	$P_0(0)$	σ	$P_0(1)$	σ	$P_1(0)$	σ	$P_1(1)$	σ
Risultati identici	18,8	0,3	19,0	0,3	18,8	0,2	18,7	0,2
Risultati diversi	6,4	0,1	6,1	0,2	6,4	0,1	5,9	0,1

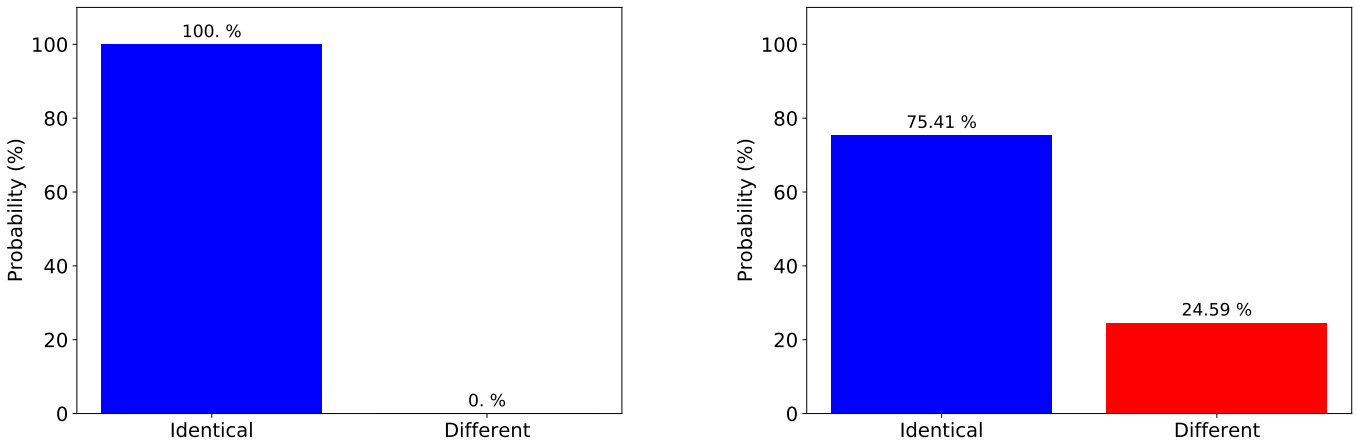
Tabella 3.6: Simulazioni - Probabilità dello scenario in cui viene messo in atto l'attacco basato sull'Entanglement (%)

	$P_{stessa\ base}$	σ	$P_{base\ diversa}$	σ	$P_{identici}$	σ	$P_{diversi}$	σ
Complessive	49,9	0,1	50,1	0,1	74,9	0,2	25,1	0,2
$P_i(j)$	$P_0(0)$	σ	$P_0(1)$	σ	$P_1(0)$	σ	$P_1(1)$	σ
Risultati identici	24,8	0,3	24,9	0,3	12,6	0,2	12,6	0,2
Risultati diversi	0	0	0	0	12,5	0,1	12,6	0,2

sembrano non aver subito modifiche, al contrario per le misure dell'operatore X si nota che si hanno le stesse probabilità di ottenere i valori 0 o 1, identici o diversi dai bit codificati da Alice. Anche qui, si riscontra una piena compatibilità tra i risultati delle simulazioni e le probabilità attese dalla teoria.

3.3.4 Grafici delle simulazioni

Di seguito sono mostrati i grafici relativi ad una sola simulazione tra quelle effettuate (è stata scelta l'ultima per ogni scenario). I due grafici presentati in questa sezione sono quelli relativi al confronto tra Scenario 1 e Scenario 2, ovvero di fatto quelli più importanti. Il confronto tra Scenario 1 e Scenario 3 è pressoché identico, perché le probabilità complessive in gioco sono le stesse. I grafici relativi allo Scenario 3 e quelli relativi alle diverse scelte di base per entrambi gli scenari sono riportati, per completezza, nell'**Appendice A**.



(a) Scenario 1

(b) Scenario 2

Figura 3.6: Confronto tra le probabilità complessive negli Scenari 1 e 2. Nei grafici sono riportate la percentuale di coppie di *bit correlati* identici (*Identical*) e quella delle coppie di bit *diversi* (*Different*)

Capitolo 4

L'esperimento su computer quantistico

4.1 Panoramica dell'esperimento

Dopo aver analizzato la teoria alla base del protocollo BB84 e aver implementato una simulazione che permetta di verificare le predizioni teoriche e, al contempo, di testare il funzionamento del codice senza dover ricorrere ai dispositivi reali, è possibile eseguire l'esperimento vero e proprio. Si tratta di mettere alla prova il protocollo BB84 sui dispositivi reali che IBM rende fruibili attraverso il sito *IBM Quantum Experience*.

Lo **scopo dell'esperimento** è mostrare l'efficacia del protocollo BB84 sui dispositivi reali. Di fatto, il protocollo è efficace se, in caso di intercettazione del messaggio, la presenza dell'hacker viene rilevata. Per individuare l'hacker è sufficiente considerare la percentuale relativa alle misure con risultato *diverso* dal bit codificato: se questa percentuale è più alta di una certa *soglia* fissata (che, nel caso ideale, è pari a 0%; nel caso reale, invece, verrà fissata uguale al *rumore di fondo*), ciò significa che è avvenuta l'intercettazione della trasmissione.
Pertanto, l'esperimento consiste nel determinare la soglia di riferimento in assenza dell'hacker (Scenario 1), e verificare che, in presenza dell'hacker (Scenario 2), tale soglia *venga superata*.
In più, viene effettuato un confronto con i valori previsti dalla teoria. Questo non è lo scopo principale dell'esperimento, ma è utile per avere un riscontro maggiore nei risultati.
Infine, viene effettuato l'esperimento anche con la variazione del protocollo in cui l'hacker sfrutta l'Entanglement (Scenario 3).

Nell'esperimento vi è un unico passaggio che viene solamente *simulato*: la *trasmissione* dei qubit. Infatti i qubit sono preparati e misurati, ed eventualmente intercettati dall'hacker, nello stesso computer quantistico. Non vi è una vera trasmissione di qubit tra un computer e un altro. Tuttavia, per questa trattazione, è sufficiente prendere in considerazione l'inizializzazione e la misura (con relativo risultato) di ogni qubit: queste due operazioni sono realmente eseguite attraverso il computer quantistico. Anche l'interazione dell'hacker con i qubit è effettivamente reale: nel computer quantistico viene realizzato l'effetto della misura da parte dell'hacker sui qubit preparati, così come vengono creati stati entangled tra i qubit inizializzati ed altri qubit preparati nello stato $|0\rangle$.

Essendo reali, i dispositivi presentano un **rumore di fondo** ineliminabile, dovuto a diverse cause, per esempio ad errori nella misurazione dei qubit, oppure al fatto che le porte quantistiche, non essendo ideali, possano trasformare degli stati in modo sbagliato, oppure ancora al rumore ambientale, che distrugge l'informazione quantistica (*decoerenza*).
Prima di effettuare l'esperimento definitivo, dunque, sono stati condotti dei **test preliminari** per quantificare, almeno approssimativamente, l'entità del rumore nei diversi dispositivi. Unitamente ai parametri di calibrazione dei dispositivi forniti da IBM, questo ha permesso di scegliere quali dispositivi utilizzare e di individuare la configurazione ottimale dei qubit in ogni circuito quantistico.

L'esperimento, dunque, si articola in due step: nel primo, viene eseguito il protocollo nello **Scenario 1**, ovvero senza l'interazione dell'hacker. Nel secondo, invece, avviene l'esecuzione del protocollo con la presenza dell'hacker. A sua volta, questo step si divide in due casi: in uno, l'hacker misura diretta-

mente i qubit inizializzati (**Scenario 2**), nell'altro crea degli stati entangled tra i qubit (**Scenario 3**). Il primo step è fondamentale: grazie ad esso è possibile *identificare il rumore di fondo con il numero di misure che, nonostante siano relative alla stessa scelta della base di autostati di Alice e Bob, restituiscono invece un risultato diverso dal bit codificato da Alice.* Dato che il rumore è indipendente dall'azione dell'hacker (cioè è presente sempre, con o senza hacker), ci si aspetta che, se l'hacker ha interagito con i qubit, l'effetto di questa interazione sia dato dalla **somma dell'effetto teorico** (ovvero il 25% delle misure, corrispondenti a risultati diversi dai relativi bit codificati) **con il rumore di fondo**. Cioè, se:

Effetto teorico = 25%
Rumore = R

allora, la percentuale di misure i cui risultati sono diversi dai bit codificati dovrebbe essere:

Effetto totale = 25% + R

Questo significa che, dall'analisi dati relativa allo step 2 dell'esperimento, ci si deve aspettare che la percentuale di risultati differenti **non sia 25%**, ma **25% + R**.

In ogni caso, l'obiettivo dell'analisi dati *non* è verificare che la perturbazione introdotta dall'hacker sia uguale a 25% + R. Al contrario, ai fini della sicurezza della trasmissione, l'importante è solamente verificare che, in presenza dell'hacker, la percentuale relativa ai risultati diversi sia *maggiore* di quella in assenza dell'hacker.
Il raggiungimento del valore aspettato, pertanto, è solamente un controllo in più, che permette di verificare l'esattezza delle previsioni.

4.1.1 I dispositivi IBM: descrizione e proprietà

I computer quantistici messi a disposizione da IBM sono diversi, ed ognuno ha proprie caratteristiche (ad esempio il numero di qubit ed il quantum volume). Ad ogni dispositivo corrisponde un *backend*, ovvero un'interfaccia tramite cui è possibile comunicare con i dispositivi ed eseguire gli esperimenti. A ciascun backend è assegnato il nome di una città (che non corrisponde al luogo in cui si trova il dispositivo, ma semplicemente al nome di una città in cui si trova una sede di IBM). Ad esempio, in questo lavoro di tesi sono stati presi in considerazione i sistemi quantistici *ibmq_ santiago*, *ibmq_ vigo* e *ibmq_ 5_ yorktown*.

Di seguito sono brevemente illustrate alcune caratteristiche dei dispositivi.

Numero di qubit di un dispositivo: è il numero massimo di qubit fisici realizzabili per un singolo circuito quantistico.

Quantum volume: è una misura della capacità e del rate di errore di un computer quantistico.
Per quantificare tale parametro, vengono riportate tre definizioni di quantum volume; le prime due sono state proposte da Nikolaj Moll *et al.* [11]:

$$V_Q = \min[N, d(N)]^2 \tag{4.1}$$

$$V_Q = \max_{n<N}\{\min[n, d(n)]^2\} \text{ con } d \simeq \frac{1}{n\epsilon_{eff}(n)} \tag{4.2}$$

dove N è il numero di qubit del dispositivo, d è la profondità del circuito (ovvero il numero di step eseguibili in un circuito), ϵ_{eff} è il rate di errore medio per una gate a due qubit, n è il numero di qubit che massimizza il quantum volume ($n < N$ in generale).
La formula (4.1) è relativa al caso in cui si considerano tutti gli N qubit del dispositivo, mentre la (4.2) si riferisce a quello in cui vengono utilizzati meno qubit di quanti ne sono disponibili (cioè n qubit al posto di N).

La terza definizione è una modifica proposta da ricercatori IBM (nell'articolo di Cross *et al.* [8]):

$$\log_2 V_Q = \arg\max_n \{\min[n, d(n)]\} \tag{4.3}$$

Il quantum volume può inoltre essere determinato attraverso un protocollo che sfrutta tecniche di quantum computing.

Definizioni a parte, la cosa importante è che il quantum volume cresce quanto più diminuisce il rate di errore medio, dunque *ad un quantum volume maggiore corrisponde una minore probabilità di errore nelle misure sui qubit.*

Calibrazione: almeno una volta al giorno i dispositivi vengono ricalibrati; è possibile accedere ai parametri della calibrazione, come per esempio i rate di errore (*readout error rate* e *CNOT error rate*).

Topologia: ad ogni backend è associato un diagramma, detto topologia o connettività, che schematizza la connessione tra i diversi qubit: di fatto, indica quali coppie di qubit supportano operazioni con gate a due qubit (come la $CNOT$ gate). Grazie all'uso di diverse gradazioni di colore, associate ciascuna ai valori dei parametri della calibrazione, è possibile visualizzare facilmente quali qubit presentano un rate di errore minore, oppure quali hanno una frequenza più alta e così via.

Gate di base: non tutte le gate possono essere inserite direttamente in un circuito quantistico eseguito su di un dispositivo reale; tuttavia, sono utilizzabili delle gate di base che, se opportunamente combinate, assolvono le stesse funzioni di tutte le altre gate. Le gate di base dei dispositivi considerati sono la $CNOT$, l'identità ($\mathbb{1}$ o Id), la RZ (un altro modo di indicare la R_ϕ, di cui la gate Z è un caso particolare), la X e la SX (ovvero l'equivalente per X della porta S: mentre la S gate opera una rotazione di un angolo $\frac{\pi}{2}$ attorno all'asse Z, la SX opera la stessa rotazione attorno all'asse X).

4.1.2 Parametri dei dispositivi considerati

I tre dispositivi considerati sono tutti dispositivi a **5 qubit**. Tuttavia, essi si distinguono per differenti quantum volume, per diverse topologie e naturalmente per diversi parametri di calibrazione.
Di seguito vengono riportati i dati relativi a ciascun dispositivo. Si noti che i parametri di calibrazione sono soltanto indicativi: infatti i valori precisi cambiano ad ogni calibrazione, e dunque non avrebbe senso parlare, per esempio, di un rate di errore precisamente costante nel tempo (subirà, al contrario, piccole variazioni).

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

Tabella 4.1: Parametri

Figura 4.1: Topologia

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

Tabella 4.2: Parametri

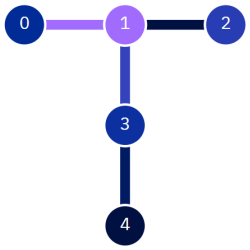


Figura 4.2: Topologia

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

Tabella 4.3: Parametri

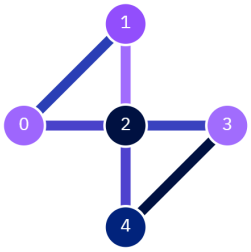


Figura 4.3: Topologia

4.1.3 Variazioni nella scrittura del codice

Nei programmi utilizzati per l'esperimento è stato necessario operare alcuni cambiamenti rispetto al codice usato in precedenza per la simulazione. In particolare, i dispositivi reali utilizzabili (i cosiddetti *open device*, ovvero quelli disponibili per un account base) non permettono di operare più di una misura sullo stesso qubit; inoltre, richiedono che ogni circuito venga terminato con una misurazione. Questo implica che *nessuna operazione possa più essere eseguita dopo che un qubit è stato misurato*. Un recente upgrade (gennaio 2021) ha introdotto la possibilità di effettuare diverse misurazioni sullo stesso qubit e anche all'interno del circuito, non solo alla fine; questa funzionalità, che apre nuovi orizzonti per gli esperimenti realizzabili, è tuttavia disponibile attualmente solo per i *premium device*.

Il cambiamento principale si trova nel contesto dello **Scenario 2**: la misura di Eve a metà circuito non può più essere resa in modo banale con un'operazione di misura. La soluzione trovata per questo problema è quella di realizzare lo stesso effetto che avrebbe una misura di Eve a metà circuito, inserendo al suo posto, di volta in volta, delle opportune gate.
Per fare questo, è stato necessario confrontare le stringhe associate alla scelta della base di autostati di Alice e di Eve. Se la scelta della base è **la stessa**, il programma inserisce la gate che rappresenta l'operatore associato alla base di autostati estratta. Ad esempio, se la scelta di Alice e quella di Eve corrisponde alla base $\{|0\rangle, |1\rangle\}$, Eve inserisce la **gate** Z; se la scelta è quella della base $\{|+\rangle, |-\rangle\}$, viene inserita la **gate** X. In questi due casi, lo stato in cui si trova il qubit rimane *imperturbato*. Al contrario, se la scelta della base di autostati di Eve è **diversa** da quella di Alice, viene inserita la **gate** H, che *cambia* lo stato in cui si trova il qubit.
Un altro piccolo cambiamento è dovuto al fatto che, nei dispositivi reali, la gate H non è direttamente applicabile. Quando il circuito quantistico viene processato sul sito di IBM, tuttavia, avviene automaticamente la sostituzione della gate H (così come è inserita nel programma) con la corretta combinazione di gate SX e RZ. Pertanto, nei programmi non è stato necessario sostituire manualmente la gate H con la combinazione di SX e RZ.

Si veda nell'**Appendice B** il programma definitivo dell'esperimento.

4.2 Esperimento: Scenario 1-2

4.2.1 Test preliminari

L'obiettivo di questi test è determinare quale sia la migliore configurazione di lavoro con cui progettare i circuiti quantistici, nonché la miglior scelta dei dispositivi. In particolare, sono stati cercati il dispositivo e la configurazione che minimizzino il rumore e che permettano di lavorare con il maggior numero possibile di qubit in parallelo.
È stato quindi scritto un semplice codice che implementa la preparazione di un qubit nello stato $|0\rangle$ e la misurazione sullo stesso qubit dell'operatore Z (perciò il test verrà in seguito indicato come *test di tipo* $|0\rangle$). Questa misurazione, in un caso ideale, dovrebbe dare sempre lo *stesso risultato*: un bit di valore 0. Se il risultato nel dispositivo reale, al contrario, restituisce il valore 1, significa che c'è stato un *errore*.

Per ognuno dei tre dispositivi considerati, sono stati eseguiti 100 circuiti di questo tipo. Poi sono stati realizzati 50 circuiti con due qubit in parallelo, sempre nello stesso modo (i qubit sono stati preparati nello stato $|0\rangle$ e misurati con l'operatore Z). Quindi sono stati eseguiti 33 circuiti con 3 qubit in parallelo, 25 circuiti con 4 qubit in parallelo e 20 circuiti con 5 qubit in parallelo, mantenendo sempre il numero totale di qubit pari a 100 (99 nel caso dei 3 qubit in parallelo).
Una procedura analoga è stata ripetuta inizializzando i qubit nello stato $|-\rangle$ e misurandoli con l'operatore X (*test di tipo* $|-\rangle$). In questo caso, il risultato dovrebbe essere idealmente sempre uguale a 1; se invece viene ottenuto 0, c'è stato un errore.

Infine sono stati realizzati dei grafici che mettono in relazione il numero di qubit in parallelo con la relativa percentuale di errore, calcolata dividendo il numero di risultati sbagliati per il numero totale di misure (cioè 100 oppure 99, a seconda del caso).

Dai grafici (Figura 4.4 e Figura 4.5 a pagina 46) risulta che **i backend ibmq_santiago e ibmq_vigo sono molto meno rumorosi del backend ibmq_5_yorktown**: le maggiori probabilità di errore riscontrate sono 5,05% nel test di tipo $|0\rangle$ per ibmq_santiago e 12% nel test di tipo $|-\rangle$ per ibmq_vigo, contro il 29,29% nel test di tipo $|-\rangle$ di ibmq_5_yorktown. Calcolando la media delle percentuali di errore relative ad ogni dispositivo (sono stati considerati entrambi i tipi di test, cioè il tipo $|0\rangle$ ed il tipo $|-\rangle$, insieme), si ottiene:

Device	**Errore medio**
ibmq_santiago	$2\% \pm 10\%$
ibmq_vigo	$4\% \pm 10\%$
ibmq_5_yorktown	$11\% \pm 10\%$

dove, per stimare l'incertezza sull'errore, è stata usata la formula relativa alla statistica di conteggio:

$$\sigma = \frac{1}{\sqrt{N}} \tag{4.4}$$

In questo caso, N è il numero di prove ($N = 100$ oppure $N = 99$; comunque $\sqrt{N} \simeq 10$). Dunque $\sigma \simeq \frac{1}{10} = 10\%$. L'errore è molto grande rispetto ai risultati; per avere una statistica significativa, il numero di prove N sarebbe dovuto essere maggiore. Nonostante questo, si può comunque notare che, mentre per i dispositivi ibmq_santiago e ibmq_vigo il rumore è compatibile con il valore ideale (cioè 0%), per ibmq_5_yorktown il rumore non è compatibile con il valore 0%.

In ogni caso, lo scopo di questa analisi approssimativa era principalmente quello di evidenziare una eventuale dipendenza del rumore dal numero di qubit in parallelo.
Osservando i grafici, tuttavia, *sembra non esserci una particolare correlazione tra il rumore ed il numero di qubit usati in parallelo*: infatti, se così fosse, ci si potrebbe aspettare un andamento (per esempio) crescente della percentuale di errore, cosa che invece non sembra essersi verificata.

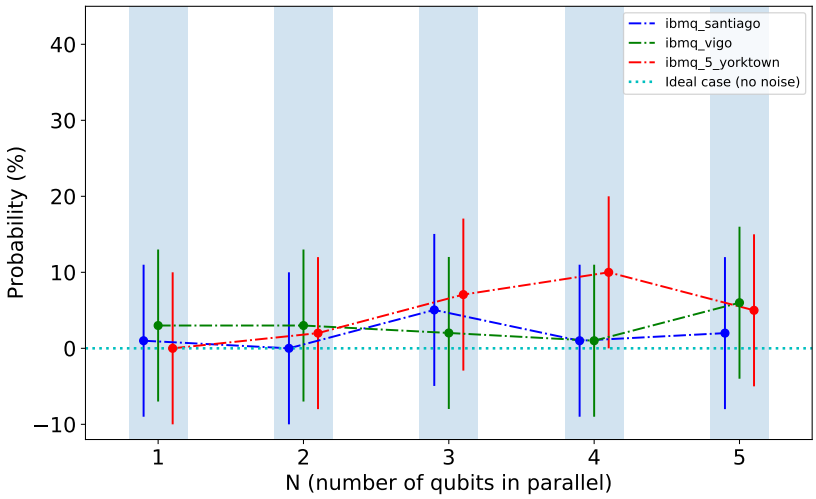


Figura 4.4: Test di tipo $|0\rangle$.
In questo grafico e in quello seguente sono riportati i valori di rumore (stimati come percentuali di errore) al variare di N qubit in parallelo, per i tre dispositivi utilizzati. Il caso ideale (0%) è stato evidenziato con una linea azzurra.

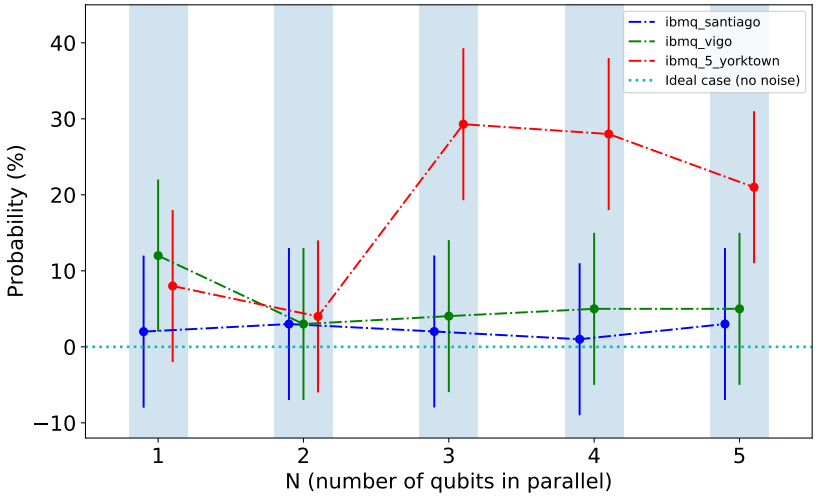


Figura 4.5: Test di tipo $|-\rangle$

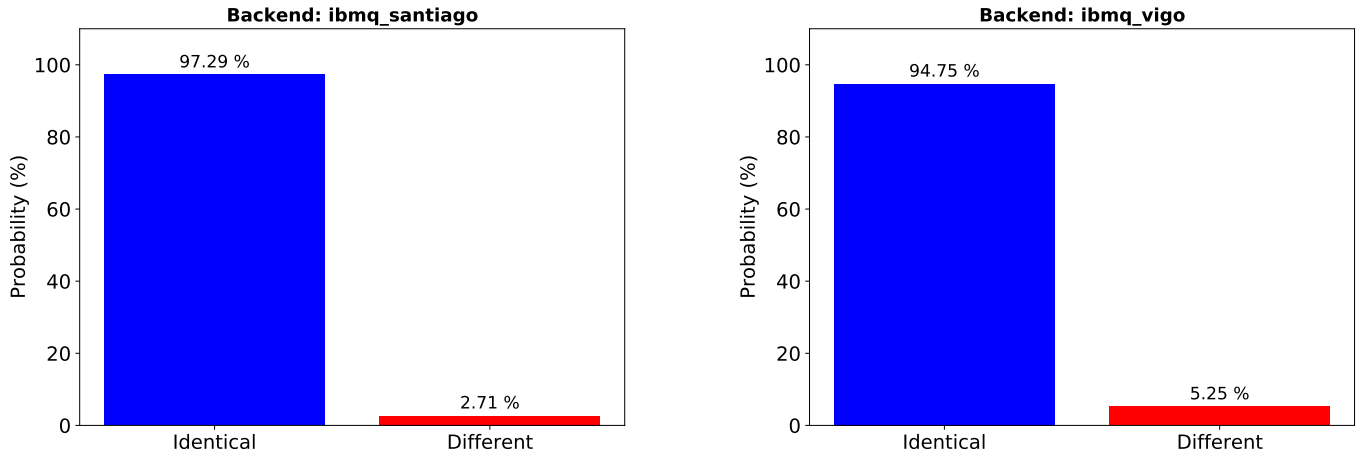
Pur con grosse incertezze sulle misure, le percentuali di rumore rilevate per ogni dispositivo sono in linea con la stima del quantum volume dei dispositivi: quello con maggior quantum volume, ovvero ibmq_santiago, è il più preciso dei tre, mentre quello con quantum volume minore, cioè ibmq_5_yorktown, è il più rumoroso.

Infine, dai test è stato possibile fare un'altra considerazione: il tempo di esecuzione impiegato dai tre backend è decisamente diverso per ognuno di essi. **Il più veloce è ibmq_5_yorktown**, mentre **ibmq_santiago è molto più lento**. In una posizione intermedia, ma decisamente più vantaggiosa, si pone ibmq_vigo: esso è sufficientemente veloce per portare a termine l'esperimento in tempi ragionevoli, pur commettendo qualche errore, ma in quantità modeste.

Per l'esperimento negli Scenari 1 e 2, dunque, è stato scelto di usare soltanto **ibmq_santiago** e **ibmq_vigo**; ibmq_5_yorktown è stato utilizzato invece nel contesto dello Scenario 3, come verrà mostrato in seguito. La configurazione ottimale del circuito quantistico è stata individuata nell'**utilizzo di tutti i 5 qubit in parallelo**.

4.2.2 Scenario 1

Per la definitiva stima del rumore, è stato lanciato l'esperimento relativo allo **Scenario 1** (assenza dell'hacker). Sia da ibmq_santiago che da ibmq_vigo sono stati eseguiti **1000 circuiti quantistici**, ognuno composto da **5 qubit in parallelo**, per un totale di **5000 qubit** ciascuno. I risultati ottenuti dalle misurazioni sono stati analizzati con lo stesso programma di analisi dati utilizzato per la simulazione.



(a) ibmq_santiago

(b) ibmq_vigo

Figura 4.6: Scenario 1 - misure dei due dispositivi. Come nella simulazione, sono considerate le probabilità complessive di bit correlati identici (*Identical*) e di bit diversi (*Different*).

Considerando separatamente i risultati dovuti ai due dispositivi, si può notare che in entrambi i casi essi corrispondono alle aspettative: i risultati identici sono la quasi totalità delle misure (il 97,29% nel caso di ibmq_santiago, e il 94,75% nel caso di ibmq_vigo). Al contrario, i risultati differenti corrispondono al 2,71% del totale per ibmq_santiago, e al 5,25% per ibmq_vigo.
Assumendo dunque che 2,71% sia la percentuale prevista di rumore per ibmq_santiago e che 5,25% sia quella prevista per ibmq_vigo, ci si può aspettare che, dall'esperimento nello Scenario 2, risulti che la percentuale relativa ai risultati diversi dai bit codificati sia pari a **25% + 2,71%** per ibmq_santiago (ovvero **27,71%**), e che sia pari a **25% + 5,25%** per ibmq_vigo (cioè **30,25%**).

Dunque la previsione *Theory + noise* per i due dispositivi è:

Device	Theory + noise
ibmq_santiago	$27,7\% \pm 1,4\%$
ibmq_vigo	$30,3\% \pm 1,4\%$

L'errore è stato calcolato dalla formula (4.4), in cui ora $N = 5000$, da cui $\sigma = \frac{1}{\sqrt{5000}} \simeq 0,014 = 1,4\%$.

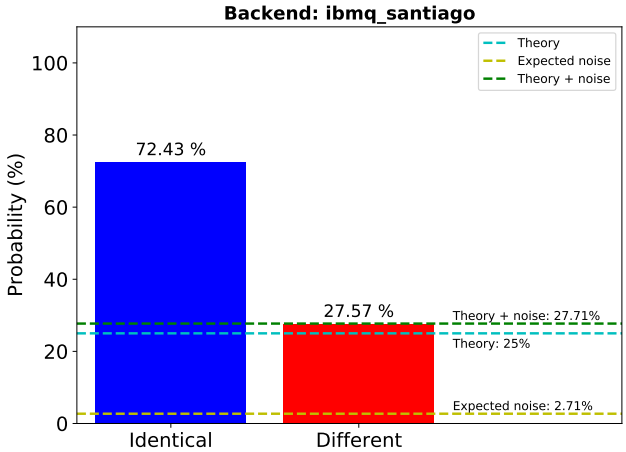
4.2.3 Scenario 2

Nello Scenario 2 entra in gioco l'hacker, che effettua una misura diretta su ogni qubit inizializzato. L'esecuzione dell'esperimento è la stessa di quello nello Scenario 1: sia ibmq_santiago che ibmq_vigo realizzano 1000 circuiti quantistici costituiti da 5 qubit in parallelo, in modo da ottenere in totale 5000 risultati da analizzare per ognuno. I risultati ottenuti sono i seguenti:

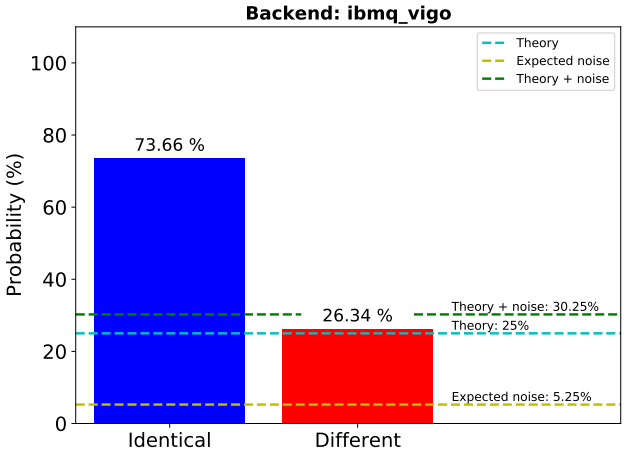
Device	Theory + noise	Experimental
ibmq_santiago	$27,7\% \pm 1,4\%$	$27,6\% \pm 1,4\%$
ibmq_vigo	$30,3\% \pm 1,4\%$	$26,3\% \pm 1,4\%$

Come previsto, l'intromissione dell'hacker ha generato un aumento della percentuale di risultati diversi ottenuti. In particolare, per entrambi i dispositivi la soglia del rumore è stata ampiamente superata (la differenza tra la percentuale della soglia di rumore e quella ottenuta è maggiore del 20%).

In Figura 4.7 sono riportati gli istogrammi che illustrano i risultati sperimentali, a cui sono stati sovrapposti i valori delle previsioni tramite delle linee tratteggiate. Tali valori sono indicati con **Theory + noise: 27,71%** per ibmq_santiago e **Theory + noise: 30,25%** per ibmq_vigo (linea verde). Sono indicate inoltre le rispettive soglie di rumore (**Expected noise: 2,71%** e **Expected noise: 5,25%**, linea gialla) e il valore ideale della perturbazione introdotta da Eve (**Theory: 25%**, linea azzurra).



(a) ibmq_santiago



(b) ibmq_vigo

Figura 4.7: Scenario 2 - risultato sperimentale e linee di predizione, relative ai dispositivi separati

Per avere un maggior riscontro dalle attese teoriche, è possibile effettuare un test di compatibilità. Ad esempio, si può calcolare la discrepanza tra il risultato atteso e quello ottenuto sperimentalmente, e poi compararla con l'errore sulle misure, secondo la formula:

$$t = \frac{|x_{atteso} - x_{ottenuto}|}{\sigma} \tag{4.5}$$

dove t è il rapporto tra discrepanza ed errore, mentre x_{atteso} è il valore aspettato e $x_{ottenuto}$ è il risultato sperimentale; σ è l'errore sulla differenza tra i due valori, calcolato tramite la somma in quadratura degli errori sui singoli risultati: $\sigma = \sqrt{(1,4\%)^2 + (1,4\%)^2} = 2,2\%$.
Ad esempio, se $t = 1$, si dice che *il risultato ottenuto si discosta dal valore atteso di una* σ.

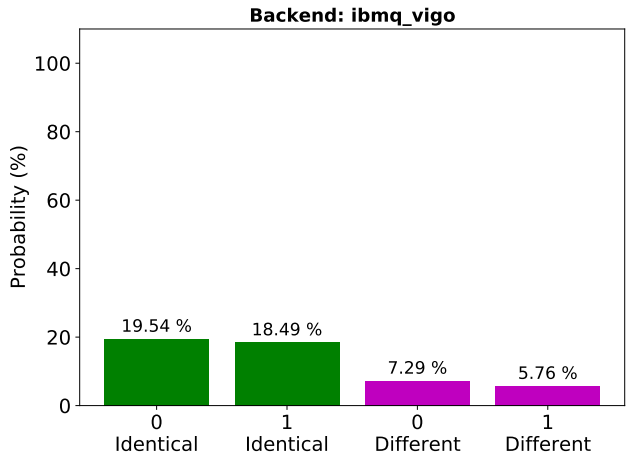
Nel caso di ibmq_vigo, ad esempio, si ha che:

$$t = \frac{|30,3\% - 26,3\%|}{2,2\%} = 1,8 \tag{4.6}$$

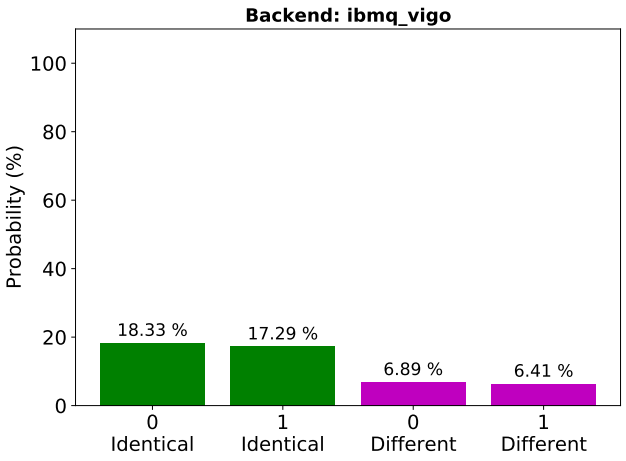
Mentre, nel caso di ibmq_santiago, si ha:

$$t = \frac{|27,7\% - 27,6\%|}{2,2\%} = 0,045 \tag{4.7}$$

Naturalmente, minore è il valore di t, e più il risultato è compatibile con la percentuale attesa. Dunque il dispositivo ibmq_santiago ha restituito un risultato più vicino alle aspettative, rispetto a quello ottenuto da ibmq_vigo. Pertanto, si può affermare che il risultato di ibmq_santiago concorda con la previsione, mentre quello di ibmq_vigo non è sufficientemente vicino ad essa, pur comunque rimanendone discostato meno di 2σ.
Un comportamento di questo tipo è comunque verosimile: ibmq_santiago presenta un quantum volume maggiore di ibmq_vigo, e dunque ci si può aspettare che quest'ultimo restituisca un risultato leggermente meno preciso.
Inoltre, considerando i grafici che raffigurano i risultati relativi alle due diverse scelte di base di autostati, si può osservare che, come mostrato nella simulazione, le probabilità relative a risultati diversi sono tutte simili tra loro, con un valore attorno al 6%. Ciò è indice di un attacco hacker basato sulla *misura diretta* dei qubit.
In Figura 4.8, sono riportati solamente i risultati di ibmq_vigo; quelli di ibmq_santiago sono simili, pur oscillando leggermente di più attorno al valore di 6% (vanno da 8% a 5,5%). È vero che ibmq_santiago dovrebbe essere il più preciso tra i due dispositivi, tuttavia in questo caso sembrerebbe il contrario. In realtà, questa fluttuazione è dovuta al fatto che, considerando i diversi stati $|0\rangle$, $|1\rangle$, $|+\rangle$ e $|-\rangle$, la statistica disponibile per ciascuno stato è molto minore (circa $\frac{1}{4}$ di quella disponibile per i risultati complessivi, che non tengono conto della distinzione tra gli stati). Pertanto, questi risultati possono essere molto più imprecisi e fluttuanti.



(a) Risultati relativi alla scelta della base $\{|0\rangle, |1\rangle\}$



(b) Risultati relativi alla scelta della base $\{|+\rangle, |-\rangle\}$

Figura 4.8: Scenario 2 - Risultati relativi alla diversa scelta di base con ibmq_vigo

L'hacker è stato individuato! Infatti la sua interazione con i qubit ha introdotto una perturbazione facilmente identificabile nei risultati delle misure. Inoltre, è stato possibile determinare il tipo di attacco effettuato: la *misurazione diretta* dei qubit spediti da Alice.

4.3 Esperimento: Scenario 1-3

In quest'ultima parte, è stata sperimentata la variante con l'Entanglement del protocollo BB84. A causa di alcuni problemi dei dispositivi, insorti durante la presa dati, non è stato possibile effettuare una statistica soddisfacente (invece che 5000 qubit, si è potuto usarne solamente 3973; in seguito ne verrà spiegato il motivo).
Dato che la versione originale del protocollo BB84 è, in realtà, quella implementata nello Scenario 2, l'esperimento dello Scenario 3 non è fondamentale per la trattazione; tuttavia, si è scelto di includerlo per completezza.

4.3.1 Test preliminari

Per questo esperimento è stato usato il dispositivo **ibmq_5_yorktown**, nonostante sia il dispositivo più rumoroso.
La scelta è dovuta al sopraggiungere di alcune difficoltà tecniche: prima del lancio di questo ultimo esperimento, IBM ha annunciato l'introduzione di tre nuovi sistemi quantistici a 5 qubit, al posto di tre dispositivi analoghi già presenti, di cui uno è ibmq_vigo. Nella fase di transizione tra i vecchi ed i nuovi dispositivi, ibmq_vigo e gli altri due dispositivi uscenti sono stati messi in manutenzione, e resi dunque inutilizzabili. Di contro, i tre sistemi nuovi non sono stati ancora messi in funzione. La conseguenza è stata un sovraccarico delle richieste di esecuzione ai pochi dispisitivi rimanenti, seguita da un complessivo rallentamento dell'esecuzione dei circuiti quantistici. In queste condizioni, l'utilizzo di ibmq_santiago sarebbe stato improponibile a causa dei lunghi tempi che avrebbe richiesto. Viceversa, ibmq_5_yorktown avrebbe ovviato in parte al problema, in quanto dotato di maggiore velocità di esecuzione. Inoltre, come si è potuto osservare nell'esperimento precedente, il fatto che ci sia rumore non è un problema così grave: nota l'entità del rumore e la sua fluttuazione, è comunque sempre possibile fare le considerazioni necessarie senza che il rumore comprometta l'esito dell'esperimento.

Per stimare l'entità del rumore dovuto all'errore sulle misure di ibmq_5_yorktown, è stato mantenuto il risultato ottenuto nel test effettuato precedentemente per gli Scenari 1 e 2. Dunque l'incertezza sul rumore di fondo è stata calcolata come la deviazione standard dei valori dell'errore sulle misure: essa è pari a **10,72%**.

Nello **Scenario 3**, per poter creare stati entangled, viene fatto uso di porte $CNOT$; può essere dunque utile operare un nuovo test che evidenzi l'andamento dell'errore nelle misure su coppie di qubit resi entangled, in modo da valutare quale sia la miglior configurazione del circuito quantistico da eseguire sul dispositivo.
Il test ideato consiste nel creare stati entangled tra due specifici qubit, e misurarli entrambi. In particolare, lo stato entangled realizzato è $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Come già descritto in precedenza, tale stato entangled viene costruito preparando il *control qubit* nello stato $|+\rangle$ e il *target qubit* nello stato $|0\rangle$, per poi applicare la $CNOT$ gate.
Misurando i due qubit, dunque, nel caso ideale si dovrebbe ottenere sempre lo *stesso risultato* (un bit di valore 0 per entrambi, oppure un bit di valore 1 per entrambi). Contando quante coppie di misure danno risultati diversi, si ottiene la percentuale di errore.

Il test è stato effettuato per ogni possibile coppia di qubit (per identificare le possibili coppie, è sufficiente osservare la topologia del dispositivo, in cui le linee che uniscono i qubit identificano quali coppie di qubit possono essere rese entangled). Per ogni circuito quantistico, è stata inizialmente resa entangled *una sola coppia* di qubit. Successivamente, è stato ripetuto il test considerando due coppie di qubit per circuito quantistico.
Per ogni coppia di qubit sono stati inizializzati ed eseguiti 100 circuiti quantistici, in modo da ottenere 100 coppie di risultati.

Dal grafico di Figura 4.9 a pagina 51, risulta chiaro che le coppie Q0-Q2 e Q1-Q2 sono le più rumorose; in particolare, le percentuali di errore sono rispettivamente 54% e 46%, elevatissime. Le **meno rumorose** sono invece le coppie **Q2-Q3** e **Q3-Q4**, con una percentuale di errore dell'**1%**.

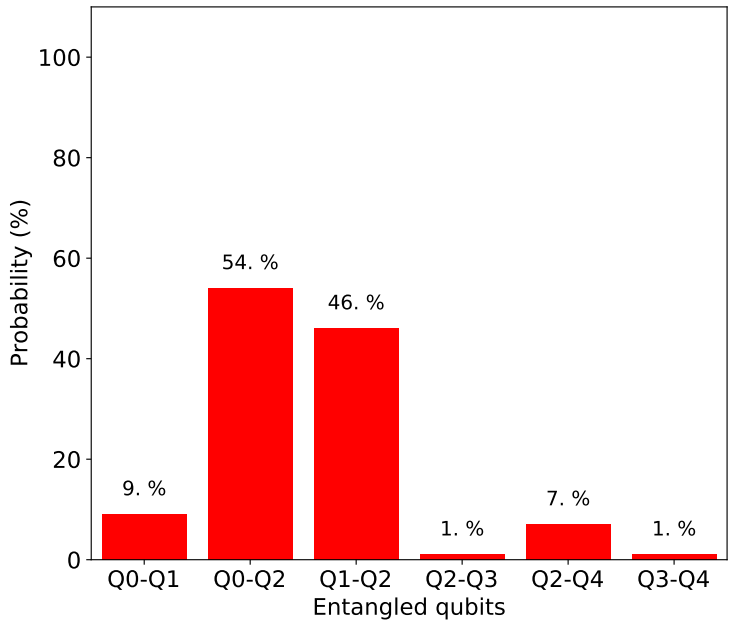
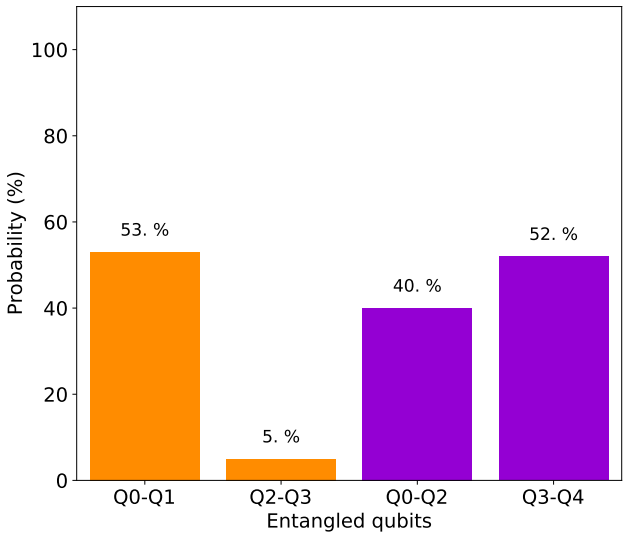
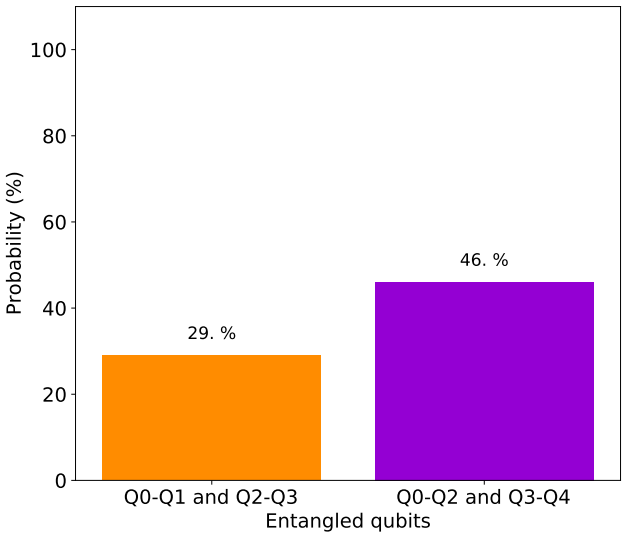


Figura 4.9: Test a **singola coppia** per circuito quantistico: percentuali di errore relative alle diverse coppie di qubit entangled.



(a) Percentuali di errore relative alle singole coppie



(b) Percentuali di errore relative alle coppie in parallelo

Figura 4.10: Test a **doppia coppia** per circuito quantistico

Il test con due coppie entangled *in parallelo*, ovvero nello stesso circuito, non ha dato buoni risultati. Sono state rese entangled prima le coppie Q0-Q1 e Q2-Q3, e poi le coppie Q0-Q2 e Q3-Q4; ma, come si può notare dai grafici di Figura 4.10, porre due coppie di qubit entangled nello stesso circuito quantistico porta ad aumentare notevolmente l'errore anche nelle misure sulle coppie poco rumorose (Q0-Q1 e Q3-Q4).
Pertanto, si è scelto di *escludere le configurazioni con due coppie di qubit entangled*, e al contrario di *usare solamente circuiti quantistici contenenti un'unica coppia di qubit entangled*. In particolare, sono state usate le coppie **Q2-Q3** e **Q3-Q4** (separatamente, cioè con una sola di esse per circuito quantistico).

4.3.2 Scenario 1

In precedenza, lo Scenario 1 non era stato sperimentato su ibmq_5_yorktown; è dunque necessario ripetere l'esperimento dello Scenario 1 per poter stimare il rumore di fondo presentato dal dispositivo. A causa dei rallentamenti nelle esecuzioni dei circuiti quantistici sopra descritte, si è scelto di dimezzare la statistica per diminuire il tempo di presa dati. Bisogna considerare infatti che, in questo caso, il dispositivo utilizzabile è uno solo; inoltre, nell'esperimento dello Scenario 3, per ogni circuito quantistico si ottiene **un solo risultato** (quello relativo ad uno dei qubit della coppia entangled), e non cinque, come in quello dello Scenario 1. Dunque, il numero di misurazioni totale è stato fissato a **5000**, sia per lo Scenario 1 che per lo Scenario 3.

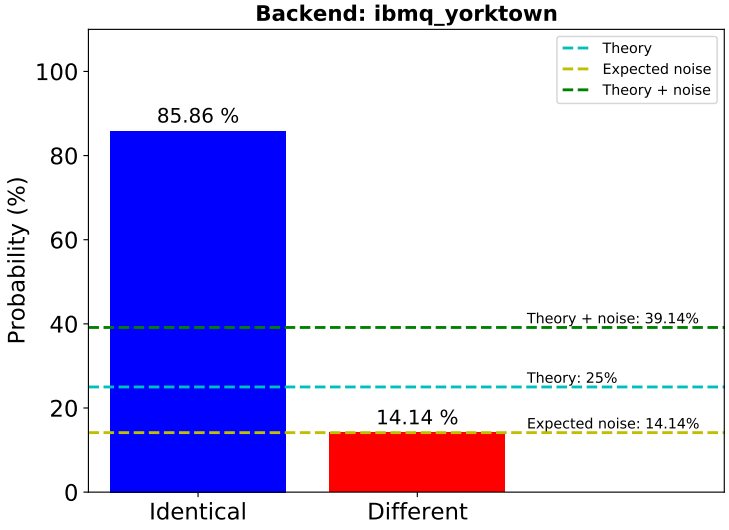
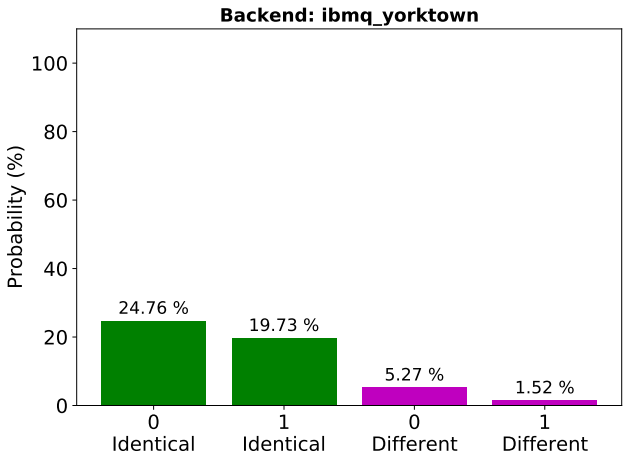


Figura 4.11: Scenario 1 - probabilità complessive con linee di predizione

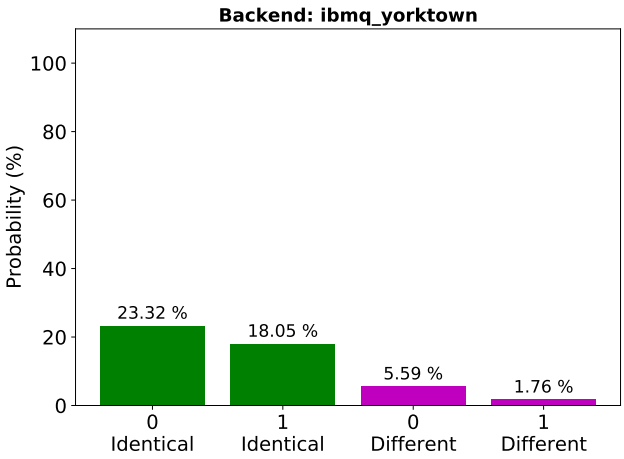
L'analisi dati mostra che il rumore sulle misure complessive è pari a $14,1\% \pm 1,4\%$. Pertanto, sommandoci la percentuale data dalla teoria nel caso ideale senza rumore, ovvero 25%, si ottiene un valore previsto per lo Scenario 3 pari a:

$$\text{Theory} + \text{noise} = 39,1\% \pm 1,4\%\% \tag{4.8}$$

Un'ultima considerazione sullo Scenario 1 può essere fatta osservando i grafici dei risultati parziali, cioè quelli relativi alle misurazioni nella base $\{|0\rangle, |1\rangle\}$ e quelli nella base $\{|+\rangle, |-\rangle\}$. Si può notare un leggero squilibrio tra i risultati 0 ed i risultati 1 in entrambi i casi. Tuttavia, questo scompenso non è rilevante (è soltanto rumore del dispositivo, e lo squilibrio potrebbe essere dovuto alla minore statistica di cui si è fatto uso). Più importante sarà invece confrontare questi grafici con i rispettivi istogrammi realizzati per l'esperimento nello Scenario 3. Infatti, quando l'hacker entra in azione usando l'Entanglement, ci si aspetta che il grafico relativo alla base $\{|0\rangle, |1\rangle\}$ rimanga circa lo stesso



(a) Risultati relativi alla scelta della base $\{|0\rangle, |1\rangle\}$



(b) Risultati relativi alla scelta della base $\{|+\rangle, |-\rangle\}$

Figura 4.12: Risultati differenziati a seconda della scelta della base

di quello dello Scenario 1; al contrario, ci si attende che il grafico relativo alla base $\{|+\rangle, |-\rangle\}$ presenti quattro valori di probabilità tutti simili tra loro, sia per i risultati identici che per quelli differenti.

4.3.3 Scenario 3

L'ultimo esperimento di questa trattazione mette alla prova lo Scenario 3.
Ogni circuito quantistico contiene 5 qubit, di cui **ad una sola coppia**, Q2-Q3 oppure Q3-Q4, è applicata una $CNOT$ gate. Sono stati lanciati due programmi diversi: uno che prendeva in considerazione la prima coppia, l'altro la seconda. Il primo programma è stato eseguito con due terminali che lavoravano in contemporanea, il secondo con tre terminali; in questo modo, l'esperimento è stato portato avanti da cinque esecuzioni simultanee.
Ogni esecuzione avrebbe dovuto creare e misurare 1000 circuiti quantistici, in modo da ottenere la stessa statistica dello Scenario 1, ovvero 5000 misurazioni in totale. Tuttavia, quando l'esperimento era quasi giunto al termine, il dispositivo ibmq_5_yorktown è stato messo a sua volta in manutenzione; in tal modo non è stato più possibile completare la presa dati. I dati raccolti, dunque, sono **3973** invece che 5000.

Il valore sperimentale ottenuto è $34,2\% \pm 1,6\%$ (l'errore ora è $\sigma = \frac{1}{\sqrt{3973}} \simeq 1,6\%$).

Device	**Theory + noise**	**Experimental**
ibmq_5_yorktown	$39,2\% \pm 1,4\%$	$34,2\% \pm 1,6\%$

Il test di compatibilità restituisce un valore:

$$t = \frac{|39,2\% - 34,2\%|}{2,3\%} = 2,2 \tag{4.9}$$

dove $\sigma = 2,3\%$ è stato calcolato come somma in quadratura degli errori $1,4\%$ e $1,6\%$.
Il risultato è quindi ancora più discostato dalla previsione, rispetto a quanto ottenuto nello Scenario 2. Era comunque prevedibile: il dispositivo ibmq_5_yorktown è il più impreciso dei tre dispositivi utilizzati, ma soprattutto è stata utilizzata una statistica molto inferiore.

È interessante, in questo esperimento, il fatto che sia comunque possibile verificare che l'hacker abbia tentato un attacco basato sull'Entanglement. Basta semplicemente confrontare i grafici dei risultati

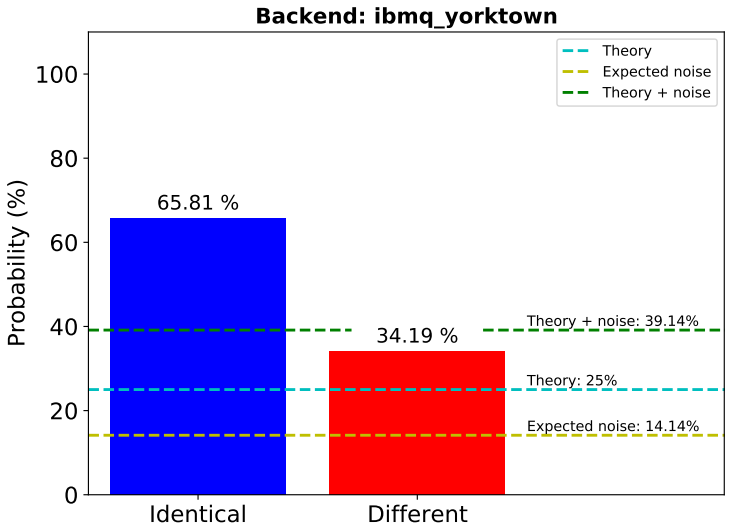
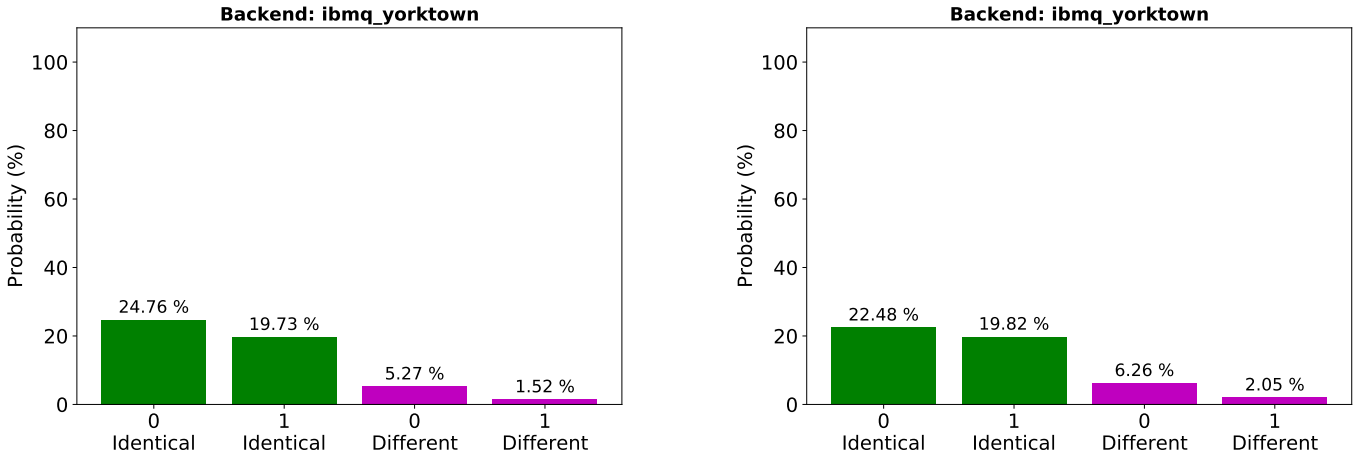


Figura 4.13: Scenario 3 - risultato sperimentale e valore previsto

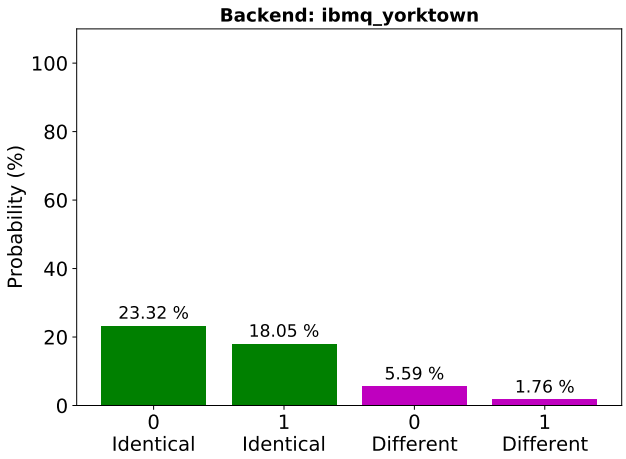
che corrispondono alle diverse basi di autostati. Infatti, nell'istogramma relativo alla base $\{|0\rangle, |1\rangle\}$ (Figura 4.14 (b)), i valori delle probabilità sono rimasti **circa gli stessi** di quelli del rispettivo grafico nello Scenario 1, in cui non vi era presenza dell'hacker. Al contrario, nell'istogramma relativo alla base $\{|+\rangle, |-\rangle\}$ (Figura 4.15 (b) a pagina 55), i valori delle probabilità **sono cambiati**: ora *tutti e quattro i valori delle probabilità sono pari circa a* **12%**.



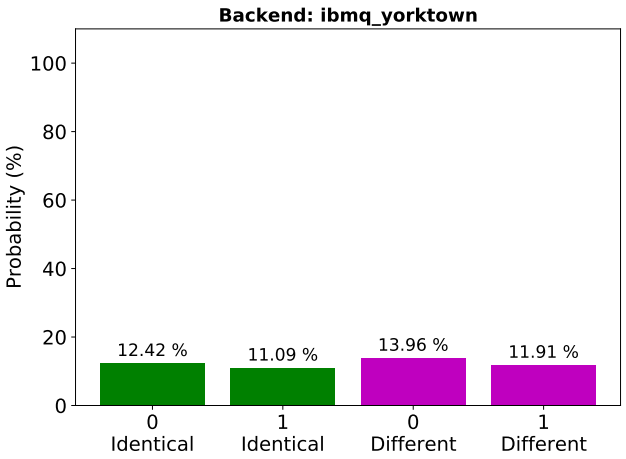
(a) Scenario 1

(b) Scenario 3

Figura 4.14: Confronto tra Scenario 1 e Scenario 3 - Scelta della base $\{|0\rangle, |1\rangle\}$



(a) Scenario 1



(b) Scenario 3

Figura 4.15: Confronto tra Scenario 1 e Scenario 3 - Scelta della base $\{|+\rangle, |-\rangle\}$

Non soltanto l'hacker è stato individuato: è stato anche possibile individuare il tipo di attacco su cui si è basato! Eve ha infatti cercato di rubare la chiave trasmessa sfruttando l'Entanglement.

Capitolo 5

Conclusioni

In questa tesi si è voluto mostrare come alcuni fenomeni quantistici possano essere utilizzati per un'applicazione pratica specifica, in particolare per lo scambio di una chiave crittografica privata attraverso il protocollo BB84. Questa, tra le applicazioni della Meccanica Quantistica, è forse quella che più si presta ad essere sfruttata su larga scala, in quanto la sicurezza dei dati informatici è un tema di grande importanza, a cui si stanno interessando non solo diversi grandi centri di ricerca e grosse aziende multinazionali, ma anche i governi degli Stati che vogliono migliorare la protezione dei propri dati sensibili.
In questa trattazione è stata analizzata la sicurezza del protocollo BB84. Ad oggi, il protocollo è considerato sicuro e infallibile: la sua forza sta nell'individuazione sistematica di qualunque tipo di attacco hacker. Inoltre, l'efficacia del protocollo BB84 non si basa sulla difficoltà comportata dalla decifrazione del messaggio (al contrario, per esempio, del protocollo RSA per la codifica dei messaggi con due chiavi diverse): anzi, per un hacker può essere molto semplice intercettare i qubit e effettuare su di essi delle misurazioni. Invece, il protocollo BB84 basa la sua infallibilità sulle leggi della Meccanica Quantistica. L'hacker, se agisce, viene sempre individuato: *non può essere altrimenti.* Un altro punto di forza del protocollo BB84 è il fatto che, per mezzo del suo utilizzo, viene trasmessa la chiave crittografica (a sua volta codificata grazie ai qubit), e non il messaggio crittografato. Di conseguenza, se anche l'hacker riesce ad intercettare la trasmissione, non entra in possesso del messaggio, ma soltanto di parte della chiave, la quale viene poi cambiata in quanto l'hacker viene sicuramente scoperto. Dunque nessuna informazione sensibile risulta mai essere esposta al pericolo di intercettazione.

Tutto ciò sembrerebbe dimostrare che, grazie al protocollo BB84, la continua sfida tra crittografia e crittoanalisi sia stata definitivamente vinta dalla prima. In effetti, ora la crittografia è in grado di fornire un metodo di codifica sicuro e pressoché inviolabile da qualsiasi tecnica di crittoanalisi. Tuttavia, raccogliendo gli insegnamenti della Storia, sarebbe forse meglio non dare per scontata questa vittoria. Non si può sapere a quali nuove scoperte potrebbe portare la Scienza, e non è escludibile che, un giorno, venga trovato un modo di violare persino alcune leggi che, oggi, sono considerate solide e del tutto corrette.

Al di là della crittografia quantistica, in ogni caso, le possibilità aperte dallo studio del Quantum Computing sono veramente molte. Possono essere sviluppate applicazioni a diverse tecnologie, così come possono essere implementate nuove tecniche di calcolo utilizzabili in Fisica teorica, nello studio dei sistemi complessi o in ambito informatico, per esempio nel Machine Learning.
L'aspetto che, forse, riguarda maggiormente la Fisica, sta nella possibilità di rapportarsi alla Natura attraverso degli strumenti che, in qualche modo, ne condividono le caratteristiche essenziali. È un modo di entrare più in profondità nel mondo reale: per farlo, bisogna veramente *comportarsi* come il mondo reale.

[...] because nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy. Così Feynman concludeva la sua conferenza del 1981.

Appendice A

Grafici delle simulazioni

In questa appendice sono riportati i grafici delle simulazioni effettuate nel Capitolo 3. In Figura A.1 vi è la probabilità che Alice e Bob effettuino (oppure no) la stessa scelta della base di autostati. Tale probabilità è uguale per tutti gli scenari. In Figura A.2 e in Figura A.3 vi sono le probabilità associate alla misura nelle diverse basi di autostati, rispettivamente per lo Scenario 1 e per lo Scenario 2. In Figura A.4 e Figura A.5, sono mostrati i grafici relativi alla variante del protocollo con l'Entanglement.

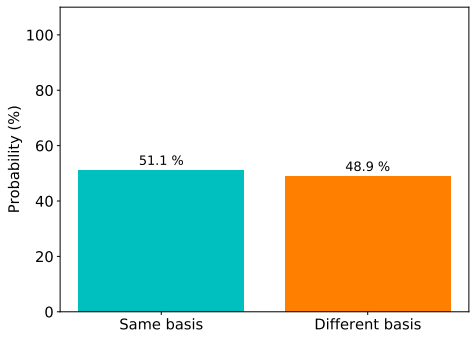
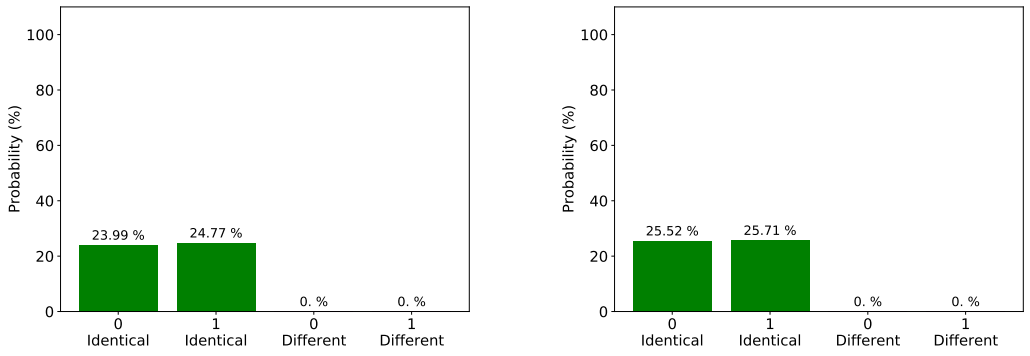
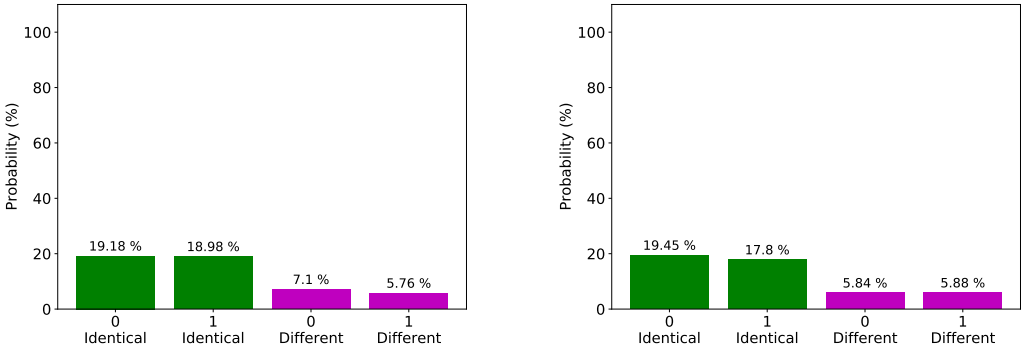


Figura A.1: Probabilità di effettuare la stessa scelta della base di stati o di scegliere una base diversa (entrambe pari circa a 50%; in tutti gli scenari queste due probabilità rimangono invariate)



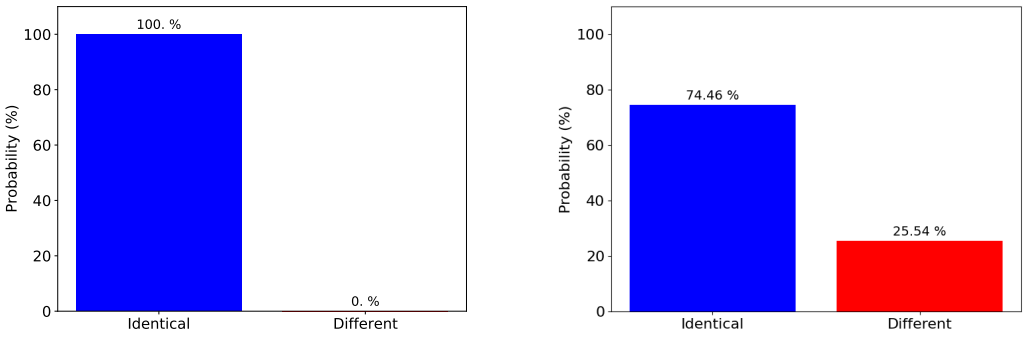
(a) $P_0(j)$ (b) $P_1(j)$

Figura A.2: Grafici dello **Scenario 1** - probabilità che i bit correlati siano identici o diversi, relativamente alle diverse basi di autostati



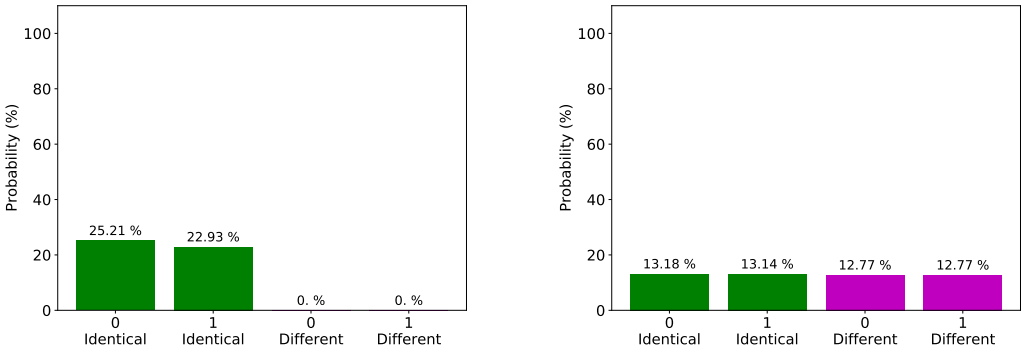
(a) $P_0(j)$ (b) $P_1(j)$

Figura A.3: Grafici dello **Scenario 2** - probabilità relative alle diverse basi di autostati



(a) Scenario 1 (b) Scenario 2

Figura A.4: Confronto tra le probabilità complessive - **Scenario 1** e **Scenario 3**



(a) $P_0(j)$ (b) $P_1(j)$

Figura A.5: Grafici dello **Scenario 3** - probabilità relative alle diverse basi di autostati

Appendice B

Codice dell'esperimento

Viene qui riportato il codice definitivo usato per condurre l'esperimento su computer quantistico.

```
# Implementation of the QKD BB84 protocol
# Experiment on a real device: ibmq_santiago

# Importing standard Qiskit libraries and configuring account
from qiskit import QuantumCircuit, QuantumRegister, \
    ClassicalRegister, execute, Aer, IBMQ
from qiskit.compiler import transpile, assemble
from qiskit.tools.jupyter import *
from qiskit.visualization import plot_bloch_multivector, \
    plot_histogram
from qiskit.extensions import Initialize
from math import sqrt, pi
import numpy as np
import matplotlib.pyplot as plt
from random import *
# Loading your IBM Q account(s)
provider = IBMQ.load_account()

# DEFINITIONS

# Implementation of the measurement in the {|+>, |->} basis:

def x_measure (quantumcircuit, qubit, cbit):
    quantumcircuit.h(qubit)
    quantumcircuit.measure(qubit, cbit)
    return quantumcircuit

# Implementation of a function which builds up
# a single qubit circuit based on Alice's strings

def encoding_circuit_builder (alice_bits, alice_basis, nth_circuit):
    i = nth_circuit    # the n-th circuit which contains 5 qubits
    encoding_circuit = QuantumCircuit(5,5)
    for k in range(5):
        if alice_bits[5*i+k] == 0 and alice_basis[5*i+k] == 0:
        # Alice chooses {|0>, |1>} basis
            pass # Apply I (nothing happens)
        if alice_bits[5*i+k] == 1 and alice_basis[5*i+k] == 0:
        # Alice chooses {|0>, |1>} basis
            encoding_circuit.x(k) # Apply X-Gate (flip |0> to |1>)
        if alice_bits[5*i+k] == 0 and alice_basis[5*i+k] == 1:
        # Alice chooses {|+>, |->} basis
            encoding_circuit.h(k) # Apply H-Gate (change |0> to |+>)
        if alice_bits[5*i+k] == 1 and alice_basis[5*i+k] == 1:
        # Alice chooses {|+>, |->} basis
            encoding_circuit.x(k)
            encoding_circuit.h(k) # Apply X-Gate and H-Gate (so |0> goes in |->)
     encoding_circuit.barrier()
     return encoding_circuit
```

```
# Implementation of the function with which Bob measures Alice's qubit

def circuit_measure (backend_name, encoding_circuit, bob_basis, nth_circuit):

    i = nth_circuit
    list_of_results = []
    inverted_list = []
    definitive_results = []

    for k in range(5):
        if bob_basis[5*i + k] == 0: # Bob chooses {|0>, |1>} basis
        # Measurement with the default {|0>, |1>} basis
            encoding_circuit.measure(k,k)
        if bob_basis[5*i + k] == 1: # Bob chooses {|+>, |->} basis
        # Measurement with the {|+>, |->} basis
            x_measure(encoding_circuit, k, k)

    backend = provider.get_backend(backend_name)
    job = execute(encoding_circuit, backend, shots=1, memory=True)
    result = job.result()
    list_of_results = result.get_memory()

    list_of_results = list(map(int, str(list_of_results[0])))
    # But these results are ordered backwards!
    # Their order must be inverted!

    for k in range(5):
        inverted_list.append(list_of_results[4-k])


    # Scenario 1-2:
    definitive_results = inverted_list
    """
    # Scenario 3:
    # We have to consider ONLY qubits q3, which has been entangled with q4
    definitive_results.append(inverted_list[3])
    """

    return definitive_results

def eve_hacking_measure (hacker_activated, encoding_circuit, \
    alice_basis, nth_circuit):

    if hacker_activated == True:
    # Eve measures each qubit sent by Alice. After that, Eve sends it to Bob:
    i = nth_circuit
    for k in range(5):
        if eve_basis[5*i+k] == 0: # Eve chooses {|0>, |1>} basis
        # Measurement with the default {|0>, |1>} basis
        if alice_basis[5*i+k] == 0:
            encoding_circuit.z(k) # --> ''right'' choice
                                  # --> the state remains the same
        if alice_basis[5*i+k] == 1:
            encoding_circuit.h(k) # --> ''wrong'' choice --> change basis
        if eve_basis[5*i+k] == 1: # Eve chooses {|+>, |->} basis
        # Measurement with the {|+>, |->} basis
        if alice_basis[5*i+k] == 1:
            encoding_circuit.x(k) # --> ''right'' choice
                                  # --> the state remains the same
        if alice_basis[5*i+k] == 0:
            encoding_circuit.h(k) # --> ''wrong'' choice --> change basis
    encoding_circuit.barrier()
    return encoding_circuit
    else:
        pass

def eve_hacking_entangle (hacker_activated, encoding_circuit):

    if hacker_activated == True:
    # Eve ENTANGLES qubits q0 and q3 sent by Alice
    # with |0> state qubits (q2 and q4).
    # After that, Eve sends the entangled qubit to Bob:
```

```
    encoding_circuit.cx(3, 4)
    encoding_circuit.barrier()
    return encoding_circuit

# --------------------------- MAIN PROGRAM ---------------------------

# Number of qubits that Alice is going to use:
number_of_circuits = 1000
number_of_qubits = 5 * number_of_circuits

# Backend:
backend_name = "ibmq_santiago"

# Alice generates n random bits (some of these bits will form the key)
alice_bits = []
for i in range (number_of_qubits):
    alice_bits.append(randint(0,1))

print("\nAlice's bits (first 20 bits):\n", alice_bits[0:19])

# Alice randomly chooses the bases in which she is going to measure
alice_basis = []
for i in range (number_of_qubits):
    alice_basis.append(randint(0,1))
print("\nAlice's basis (first 20 bits):\n", alice_basis[0:19])

# Bob also randomly chooses the bases in which he is going to measure
bob_basis = []
    for i in range (number_of_qubits):
    bob_basis.append(randint(0,1))
print("\nBob's basis (first 20 bits):\n", bob_basis[0:19])

print("\nChoose an option [digit 1, 2 or 3]: \
\n\n1. Transmission without hacker's attack" \
"\n2. Transmission with a measurement-based hacker's attack" \
"\n3. Transmission with an Entanglement-based hacker's attack\n")
scelta = input()

if scelta == "1":
    hacker_activated1 = False
    hacker_activated2 = False
if scelta == "2":
    hacker_activated1 = True
    hacker_activated2 = False
if scelta == "3":
    hacker_activated1 = False
    hacker_activated2 = True
if scelta != "1" and scelta != "2" and scelta != "3":
    print("\nTry again (digit only 1, 2 or 3)")

# Eve randomly chooses the bases in which she is going to measure (like Bob)
if hacker_activated1 == True:
    eve_basis = []
    for i in range (number_of_qubits):
        eve_basis.append(randint(0,1))
    print("\nEve's basis (first 20 bits):\n", eve_basis[0:19])

print("------------------------------------------------------------")

print("\nThe experiment has been launched!\n")
print("Backend:", backend_name)
print("Number of circuits:", number_of_circuits)
print("Number of qubits per circuit:", 5)
print("Total number of qubits:", number_of_qubits)

print("\n\n------------------------------------------------------------")

# For each classical bit which Alice wants to encode and transmit to Bob,
# they proceed as it follows:

bob_measures = []
```

```
for n in range(number_of_circuits):

    # Alice codes the (5*n+k)-th bit of her initial string as a qubit.
    # Then she builds up the n-th circuit with 5 of these qubits,
    # and sends it to Bob
    circuit = encoding_circuit_builder(alice_bits, alice_basis, n)

    # Bob measures the qubit with his own basis:
    # but what if Eve is hacking the message?
    eve_hacking_measure (hacker_activated1, circuit, alice_basis, n)
    eve_hacking_entangle (hacker_activated2, circuit)
    new_results = circuit_measure (backend_name, circuit, bob_basis, n)
    bob_measures = bob_measures + new_results

    counter = 0

    # For each job, we immediately register the result in the data file:
    data_file = open("bb84_yorktown_scenario3_data.txt", "a")

# Scenario 1-2:

for k in range(5):
    data_file.write(str(alice_bits[5*n+k]))
    data_file.write("\t")
    data_file.write(str(alice_basis[5*n+k]))
    data_file.write("\t")
    data_file.write(str(bob_basis[5*n+k]))
    data_file.write("\t")
    data_file.write(str(bob_measures[5*n+k]))
    data_file.write("\n")
    counter = counter + 1
"""
# Scenario 3: again, we must consider only qubit q3

data_file.write(str(alice_bits[5*n+3]))
data_file.write("\t")
data_file.write(str(alice_basis[5*n+3]))
data_file.write("\t")
data_file.write(str(bob_basis[5*n+3]))
data_file.write("\t")
data_file.write(str(bob_measures[n]))    # Because we have
data_file.write("\n")                    # only 1 result per circuit
counter = counter + 1
"""
data_file.close()

print(n, "-th job")
print("Results:", new_results)
if scelta == 1 or scelta == 2:
    if counter == 5:
        print("Datas have been correctly saved")
    else:
        print("DATA WERE NOT CORRECTLY SAVED!!")
if scelta == 3:
    if counter == 1:
        print("Datas have been correctly saved")
    else:
        print("DATA WERE NOT CORRECTLY SAVED!!")

print("-------------------------------------------------------")


# Let's see the first 20 results of the measurements!

print("\nBob's measurements (first 20 measurements):\n")
print(bob_measures[0:19])

print("\nThe experiment ended with success!")

plt.show()
```

Bibliografia

[1] Forte, Stefano; Rottoli, Luca (2019), *Fisica Quantistica*, Zanichelli

[2] Griffiths, David J.; Schroeter, Darrell F. (2018), *Introduction to Quantum Mechanics - Third Edition*, Cambridge University Press

Articoli:

[3] Arute, F., Arya, K., Babbush, R. et al., *Quantum supremacy using a programmable superconducting processor*, Nature 574, 505–510 (2019). https://doi.org/10.1038/s41586-019-1666-5

[4] Benioff, P., *The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines*, J. Stat. Phys. 22, 563–591 (1980), https://doi.org/10.1007/BF01011339

[5] Bennett, C. H.; Brassard, G. (1984) *Quantum cryptography: Public key distribution and coin tossing*, in Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, volume 175, New York
http://researcher.watson.ibm.com/researcher/files/us-bennetc/BB84highest.pdf

[6] Chuang, I., Gershenfeld, N., & Kubinec, M. (1998), *Experimental Implementation of Fast Quantum Searching*, Physical Review Letters, 80, 3408-3411.

[7] C. E. Shannon, *Communication theory of secrecy systems*, in The Bell System Technical Journal, vol. 28, no. 4, pp. 656-715, Oct. 1949, doi: 10.1002/j.1538-7305.1949.tb00928.x.

[8] Cross, Andrew W.; Bishop, Lev S.; Sheldon, Sarah; Nation, Paul D.; Gambetta, Jay M. (2018), *Validating quantum computers using randomized model circuits*, Phys. Rev. A. 100 (3): 032328. arXiv:1811.12926

[9] Feynman, Richard P., *Simulating physics with computers*, Int. J. Theor. Phys. 21, 467–488 (1982). https://doi.org/10.1007/BF02650179

[10] Houck, A. A.; Koch, Jens; Devoret, M. H.; Girvin, S. M.; Schoelkopf, R. J. (11 February 2009), *Life after charge noise: recent results with transmon qubits*, Quantum Information Processing, 8 (2–3): 105–115. arXiv:0812.1865, doi:10.1007/s11128-009-0100-6

[11] Moll, Nikolaj et al. (2018), *Quantum optimization using variational algorithms on near-term quantum devices*, Quantum Sci. Technol. 3 030503

[12] Shor, P.W. (1994) *Algorithms for quantum computation: discrete logarithms and factoring*, Proceedings 35th Annual Symposium on Foundations of Computer Science, IEEE Comput. Soc. Press: 124–134, doi:10.1109/sfcs.1994.365700, ISBN 0818665807

[13] Nairz, O.; Arndt, M.; Zeilinger, A. (2003), *Quantum interference experiments with large molecules*, Am. J. of Phys. 71, 319; https://doi.org/10.1119/1.1531580

Ciclo di conferenze:

[14] *A practical introduction to quantum computing: from qubits to quantum machine learning and beyond*: https://indico.cern.ch/event/970903/

Siti web:

[15] *Achieving Quantum Volume 128 on the Honeywell Quantum Computer* (Settembre 2020) https://www.honeywell.com/us/en/news/2020/09/achieving-quantum-volume-128-on-the-honeywell-quantum-computer

[16] *D-Wave Previews Next-Generation Quantum Computing Platform* (27 febbraio 2019) https://www.dwavesys.com/press-releases/d-wave-previews-next-generation-quantum-computing-platform

[17] *IBM Quantum Experience*: https://quantum-computing.ibm.com

[18] *Qiskit Textbook*: https://qiskit.org/textbook/preface.html

Ringraziamenti

Grazie ai miei relatori:
Dr. Andrea Giachero e Dr. Elena Ferri dell'Università degli Studi di Milano - Bicocca,
Prof. Paolo Solinas dell'Università degli Studi di Genova.

Grazie a Gabriele, Luciana, Mirco, Elisa, Ada, Mirco, Giovanni.

Grazie a Greta, Serena, Emma, Riccardo, Dario, Luca, Carlo, Andrea,
Simone, Emanuele, Francesco, Federico.

Grazie a Save, Seba, Edo, Nick, Natan.

Grazie ad Anna, Nick, Ale, a Walter e Luca, a Federico e Tommaso.

Sono i nomi delle persone con cui, e grazie a cui, ho condiviso un'avventura. E che avventura!
Perché solo il nome? Perché stat rosa *aurea* nomine... nomina nuda tenemus.