
**Università degli Studi di Verona**

Dipartimento di Informatica

# Dallo Spin al Bit: La Meccanica Quantistica come Fondamento della Nuova Cybersicurezza

**Studenti:**

Lorenzo Tessari (Matr. VR503115) Luis Qyli (Matr. VR501547)

## Obbiettivo

Il presente elaborato esplora la transizione dal formalismo della meccanica quantistica ai protocolli di sicurezza dell'informazione. Partendo dall'analisi storica e matematica dei sistemi a due stati (esperimento di Stern-Gerlach) e dal principio di sovrapposizione, si deriva il concetto di Qubit. Successivamente, si analizza la minaccia crittanalitica portata dall'algoritmo di Shor, fornendo un esempio numerico dettagliato della rottura di RSA. Infine, si contrappone a questa minaccia la sicurezza incondizionata del protocollo BB84 e la sicurezza computazionale della Post-Quantum Cryptography. Il lavoro include una simulazione software modulare del protocollo di scambio chiavi.

# Indice

# Capitolo 1

# Formalismo della Meccanica Quantistica e Informazione

## 1.1 Introduzione Storica: La Crisi della Fisica Classica

Prima di definire il Qubit, è necessario comprendere il contesto in cui la Meccanica Quantistica è nata. Alla fine del XIX secolo, la fisica classica sembrava completa. Tuttavia, fenomeni come la radiazione di corpo nero e l'effetto fotoelettrico sfuggivano alle spiegazioni di Maxwell e Newton.

Nel 1900, Max Planck introdusse l'idea che l'energia fosse quantizzata in pacchetti discreti ($E = h\nu$). Questa idea fu ripresa da Einstein nel 1905 per descrivere la luce non come un'onda continua, ma come un flusso di particelle (fotoni). Questo dualismo onda-particella è il fondamento su cui si basa l'intera teoria dell'informazione quantistica moderna e, di conseguenza, la crittografia che discuteremo in questo elaborato.

## 1.2 Sistemi a due stati: L'eredità di Stern-Gerlach

L'esperimento fondamentale per la comprensione del bit quantistico è quello condotto da Otto Stern e Walther Gerlach nel 1922. Essi inviarono un fascio di atomi di argento attraverso un campo magnetico disomogeneo. Classicamente, ci si aspettava una distribuzione continua degli atomi sullo schermo rilevatore, dipendente dall'orientamento casuale del loro momento magnetico. Sorprendentemente, il fascio si divise in due tracce distinte e separate.

Questo risultato dimostrò la **quantizzazione dello spin**. Per un elettrone (spin-1/2), lo spin può assumere solo due valori lungo un asse misurato: "Up" ($+\hbar/2$) o "Down" ($-\hbar/2$). Matematicamente, questo definisce uno spazio di Hilbert bidimensionale $\mathcal{H} \cong \mathbb{C}^2$, che è l'equivalente fisico di un bit di informazione.

## 1.3 Formalismo Matematico del Qubit

Definiamo i vettori della base computazionale corrispondenti agli stati di spin lungo l'asse Z:

$$|0\rangle = |\uparrow_z\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = |\downarrow_z\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{1.1}$$

A differenza di un bit classico, un qubit può esistere in uno stato di **sovrapposizione** coerente:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{1.2}$$

Dove $\alpha, \beta \in \mathbb{C}$ e soddisfano la condizione di normalizzazione $|\alpha|^2 + |\beta|^2 = 1$.

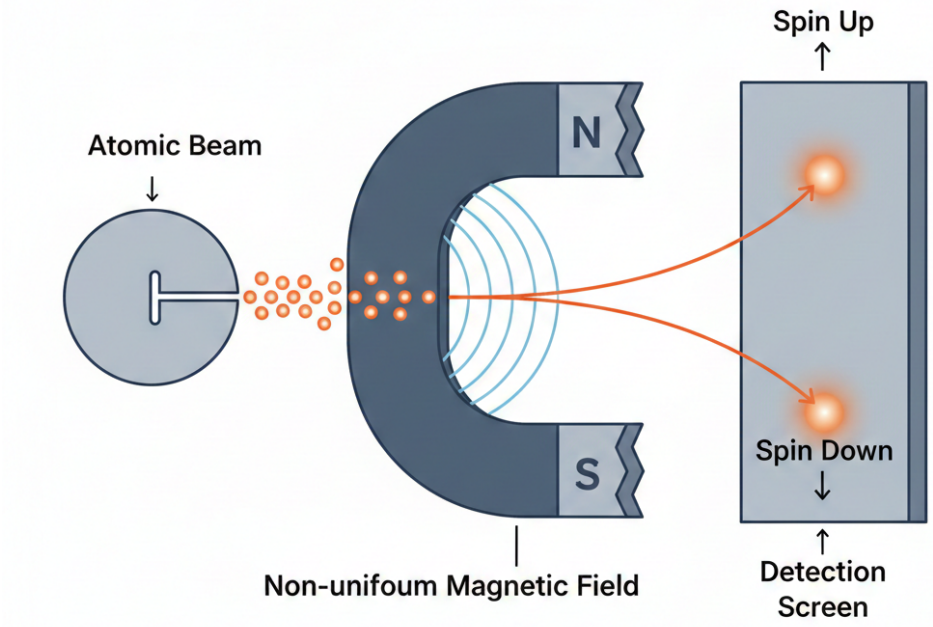


Figura 1.1: L'esperimento di Stern-Gerlach. Il magnete separa gli spin Up dagli spin Down, creando la base per il bit quantistico (0 e 1).

### 1.3.1 Le Matrici di Pauli

Gli operatori che agiscono su questi stati sono le Matrici di Pauli. Queste sono fondamentali per il protocollo BB84 perché definiscono le basi di misura.

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{1.3}$$

Gli autostati di $\sigma_x$ definiscono la base diagonale ($\times$), spesso usata da Alice e

Bob:

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{1.4}$$

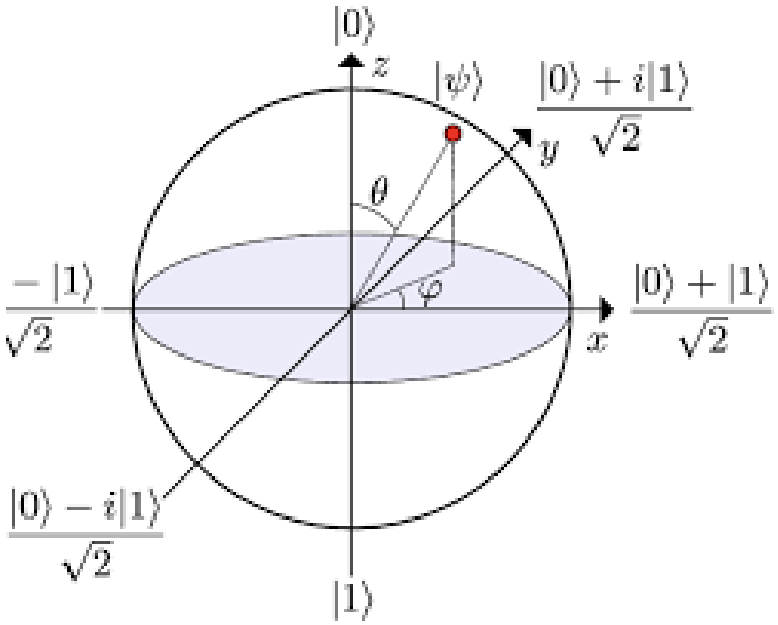


Figura 1.2: Sfera di Bloch: ogni punto sulla superficie rappresenta un possibile stato puro del Qubit. Le operazioni logiche sono rotazioni su questa sfera.

## 1.4 Realizzazione Fisica: Come costruire un Qubit?

La teoria descritta sopra è astratta. Per realizzare un computer quantistico o un sistema di crittografia QKD, dobbiamo implementare questi stati su hardware fisico.

### 1.4.1 Qubit Superconduttori (Transmon)

Utilizzati da IBM e Google, si basano su circuiti LC con giunzioni Josephson mantenuti a temperature prossime allo zero assoluto (20 mK). Sono veloci (nanosecondi) ma soffrono di decoerenza rapida.

### 1.4.2 Trappole Ioniche

Utilizzati da IonQ, sfruttano singoli atomi ionizzati sospesi nel vuoto da campi elettromagnetici. Gli stati 0 e 1 sono livelli energetici dell'elettrone. Hanno una coerenza eccezionale ma sono più lenti nelle operazioni.

## 1.5 Teorema di No-Cloning

Il pilastro della sicurezza quantistica è il Teorema di No-Cloning (1982). Esso dimostra l'impossibilità di creare una copia perfetta di uno stato quantico arbitrario sconosciuto. Se un hacker (Eve) intercetta un fotone $|\psi\rangle$ inviato da Alice, non può copiarlo per analizzarlo dopo. Deve misurarlo subito. Ma la misura, per il postulato del collasso, altera irreversibilmente lo stato se la base è sbagliata. Questa alterazione è ciò che Alice e Bob rilevano come "errore" nella chiave.

# Capitolo 2

# Criptanalisi Quantistica: La Rottura di RSA

La sicurezza di protocolli come HTTPS, VPN e firme digitali si basa su RSA. RSA si basa sul fatto che moltiplicare due numeri primi è facile, ma fattorizzare il risultato è difficile.

## 2.1 RSA e la Complessità Computazionale

Dato un numero $N = p \cdot q$, il miglior algoritmo classico (General Number Field Sieve) impiega un tempo sub-esponenziale per trovare $p$ e $q$. Per un numero a 2048 bit, questo tempo è stimato in miliardi di anni con i supercomputer attuali.

## 2.2 L'Algoritmo di Shor: Esempio Numerico Passo-Passo

L'algoritmo di Shor (1994) permette di fattorizzare $N$ in tempo polinomiale $O((\log N)^3)$. Per comprendere la sua potenza, eseguiamo una simulazione manuale dell'algoritmo per fattorizzare un numero piccolo.

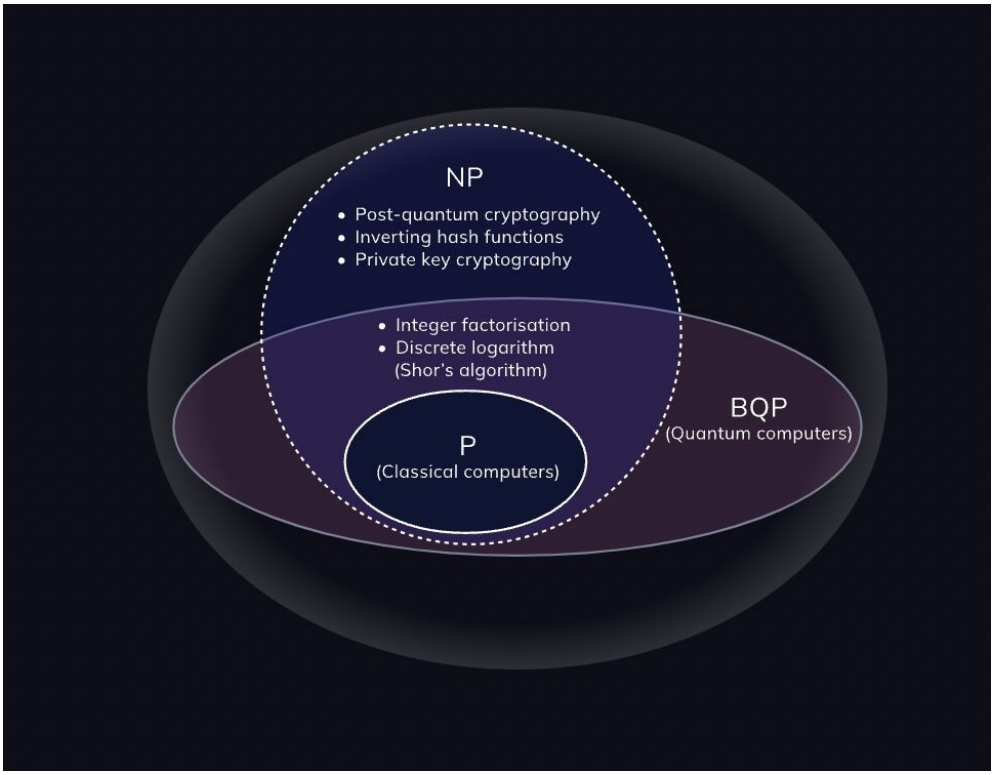


Figura 2.1: Confronto tra classi di complessità: L'algoritmo di Shor sposta la fattorizzazione da un problema intrattabile (NP) a uno risolvibile (BQP).

### 2.2.1 Obiettivo: Fattorizzare N = 15

Vogliamo trovare i fattori di 15 usando un computer quantistico.

1. **Scelta di un numero casuale:** Scegliamo un numero $a < 15$ coprimo con 15. Scegliamo $a = 7$.

2. **Oracolo Quantistico (Funzione Modulare):** Il computer quantistico valuta la funzione $f(x) = 7^x \pmod{15}$ per una sovrapposizione di tutti gli $x$. Vediamo i valori della funzione:

   - $x = 0 \to 7^0 \equiv 1$
   - $x = 1 \to 7^1 \equiv 7$
   - $x = 2 \to 7^2 = 49 \equiv 4 \pmod{15}$
   - $x = 3 \to 7^3 = 343 \equiv 13 \pmod{15}$

- $x = 4 \rightarrow 7^4 \equiv 1 \pmod{15}$ **(Il ciclo si ripete!)**

3. **Trovare il Periodo (QFT):** La Quantum Fourier Transform analizza questa sequenza in parallelo e restituisce il periodo $r$. In questo caso, la sequenza è $\{1, 7, 4, 13, 1...\}$, quindi il periodo è $r = 4$.

4. **Calcolo dei Fattori (Parte Classica):** Ora che abbiamo il periodo $r = 4$ (che è pari), usiamo la formula di Euclide:

$$\text{Fattori} = \text{MCD}(a^{r/2} \pm 1, N)$$

Calcoliamo i termini:

$$a^{r/2} = 7^{4/2} = 7^2 = 49$$

Primo fattore: $\text{MCD}(49 - 1, 15) = \text{MCD}(48, 15) = 3$. Secondo fattore: $\text{MCD}(49 + 1, 15) = \text{MCD}(50, 15) = 5$.

5. **Risultato:** Abbiamo trovato 3 e 5. Infatti $3 \times 5 = 15$. RSA è rotto.

Mentre per 15 questo calcolo è banale, per un numero di 600 cifre il passaggio 3 (trovare il periodo) è impossibile per un computer classico, ma richiede pochi secondi a un computer quantistico grazie alla sovrapposizione.

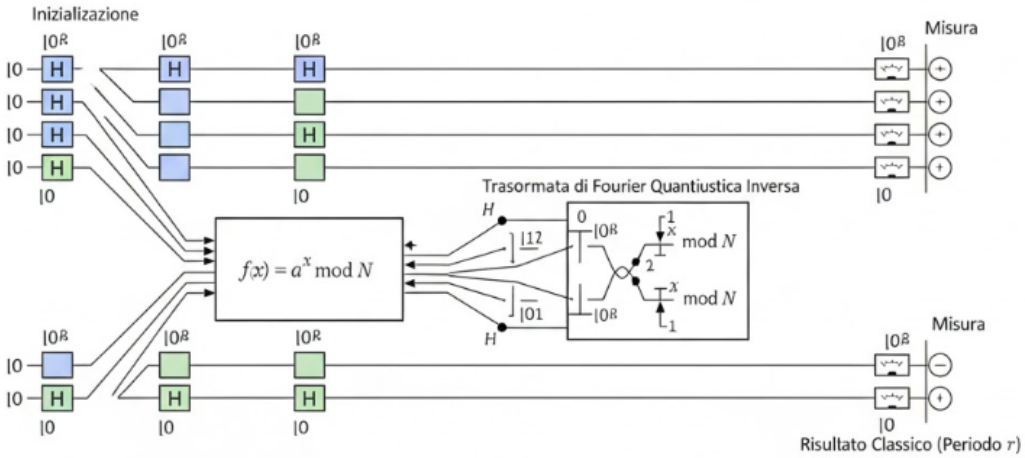


Figura: Circuito Quantistico per l'Algortimo di Shor

Figura 2.2: Circuito quantistico per l'algoritmo di Shor. Si noti l'uso della QFT inversa nella parte finale per estrarre il periodo.

# Capitolo 3

# Quantum Security: Protocollo BB84

Il protocollo BB84 rappresenta la risposta della fisica alla minaccia matematica di Shor. Non si basa sulla difficoltà di calcolo, ma sull'impossibilità fisica di misurare un sistema senza perturbarlo.

## 3.1 Meccanica del Protocollo

Alice e Bob dispongono di due canali:

1. **Canale Quantistico:** Fibra ottica o spazio libero, per inviare fotoni polarizzati.
2. **Canale Classico:** Internet o telefono, per comunicare dati di servizio (non la chiave!).

Alice sceglie a caso tra due basi: Rettilinea (+) e Diagonale (×).

- Se Alice invia $|\uparrow\rangle$ (Base +) e Bob misura in Base +, ottiene $\uparrow$ (Corretto).
- Se Alice invia $|\uparrow\rangle$ (Base +) e Bob misura in Base ×, ottiene $|\nearrow\rangle$ o $|\searrow\rangle$ col 50% di probabilità (Errore/Random).

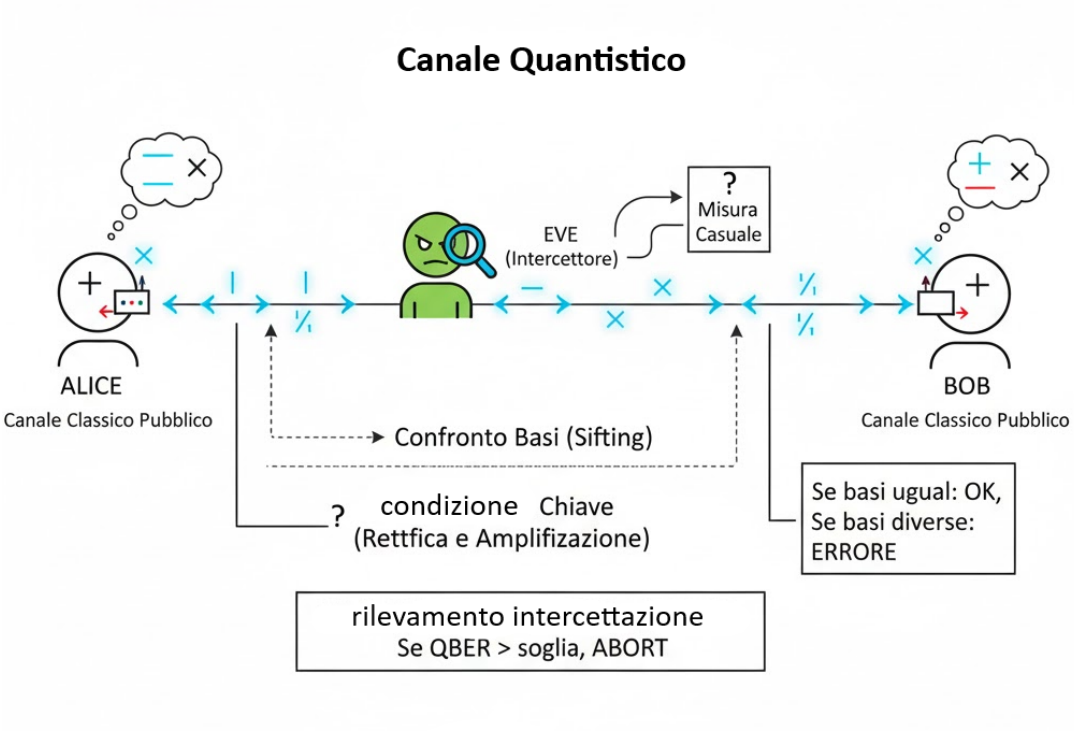


Figura 3.1: Schema completo dello scambio BB84. Si evidenzia la fase di Sifting dove le basi non coincidenti vengono scartate.

## 3.2 Simulazione Software Modulare

Per dimostrare il funzionamento del protocollo e il rilevamento di un intruso (Eve), abbiamo sviluppato una simulazione in Python. Il codice è diviso in moduli logici.

### 3.2.1 Modulo 1: Preparazione (Alice)

Alice genera i bit (il messaggio segreto) e sceglie le basi di trasmissione.

```
import numpy as np

def step1_alice_prepare(n_qubits):
    # Generazione casuale dei bit (0 o 1)
    alice_bits = np.random.randint(0, 2, n_qubits)

```

```
7     # Generazione casuale delle basi (0=Rett., 1=Diag.)
8     alice_bases = np.random.randint(0, 2, n_qubits)
9
10    return alice_bits, alice_bases
```

**Analisi del Modulo Alice**

In questo blocco, utilizziamo la libreria `numpy` per generare array di numeri casuali. Questo simula un generatore quantistico di numeri casuali (QRNG). L'array `alice_bits` rappresenta l'informazione pura. L'array `alice_bases` rappresenta l'impostazione fisica del polarizzatore (0 gradi o 45 gradi) che Alice userà per inviare i fotoni.

### 3.2.2 Modulo 2: Il Canale e l'Attaccante (Eve)

Qui simuliamo il transito dei fotoni. Se Eve è presente, tenta di misurare i fotoni ("Man-in-the-Middle").

```
def step2_channel_eve(alice_bits, alice_bases, eavesdropper=
    True):
    modified_bits = alice_bits.copy()
    n = len(alice_bits)

    if eavesdropper:
        # Eve sceglie basi casuali
        eve_bases = np.random.randint(0, 2, n)

        for i in range(n):
            # Se Eve misura nella base sbagliata...
            if eve_bases[i] != alice_bases[i]:
                # ...il fotone collassa.
                # 50% probabilita' che il bit si inverta.
                if np.random.random() < 0.5:
                    modified_bits[i] = 1 - modified_bits[i]

    return modified_bits
```

**Analisi del Modulo Eve**

Questo è il cuore fisico della simulazione. La condizione `if eve_bases[i] != alice_bases[i]` verifica se Eve ha sbagliato base. Secondo la meccanica quantistica, misurare $|\uparrow\rangle$ nella base diagonale proietta lo stato su $|\nearrow\rangle$ o $|\searrow\rangle$. Questo fenomeno è irreversibile. Quando Eve reinvia il fotone a Bob, non sta inviando l'originale, ma uno stato perturbato. Nel codice, questo "rumore quantistico" è rappresentato dal bit flip casuale.

### 3.2.3 Modulo 3: Ricezione e Sifting (Bob)

Bob misura i fotoni e poi si confronta con Alice per scartare le misure inutili.

```
def step3_bob_measure_sift(bits_in, bases_a):
    n = len(bits_in)
    bob_bases = np.random.randint(0, 2, n)
    bob_results = []

    # 1. Fase di Misura
    for i in range(n):
        if bob_bases[i] == bases_a[i]:
            # Basi uguali: Bob legge il bit correttamente
            bob_results.append(bits_in[i])
        else:
            # Basi diverse: Risultato casuale
            bob_results.append(np.random.randint(0, 2))

    # 2. Fase di Sifting (Riconciliazione)
    match_mask = (bases_a == bob_bases)
    alice_key = bits_in[match_mask]
    bob_key = np.array(bob_results)[match_mask]

    # 3. Calcolo QBER
    errors = np.sum(alice_key != bob_key)
    qber = errors / len(alice_key) if len(alice_key)>0 else
   0

    return qber
```

**Analisi del Modulo Bob**

Il "Sifting" è la fase in cui il protocollo sacrifica efficienza per sicurezza. La maschera `match_mask` elimina circa il 50% dei bit trasmessi (quelli dove le basi erano diverse). Sui bit rimanenti, Alice e Bob dovrebbero avere valori identici. Se però il calcolo finale `errors` è alto (QBER $\approx 25\%$), significa che Eve ha toccato i fotoni nel Modulo 2. La simulazione dimostra che la presenza di Eve è rilevabile matematicamente.

# Capitolo 4

# Difesa Algoritmica: Post-Quantum Cryptography

Mentre la QKD richiede hardware dedicato (laser, fibre, rivelatori), il mondo ha bisogno di una soluzione software immediata per proteggere i dati che viaggiano sull'internet classico. Questa soluzione è la PQC (Post-Quantum Cryptography).

## 4.1 I Reticoli Matematici (Lattices)

La PQC si basa su problemi matematici che sono "difficili" anche per i computer quantistici. Il candidato principale è la teoria dei Reticoli. Un reticolo è una griglia di punti multidimensionale. Immaginiamo di dover trovare il punto del reticolo più vicino a un punto arbitrario nello spazio (Shortest Vector Problem - SVP). In 2 dimensioni è facile. In 500 dimensioni, diventa un problema mostruoso.

### 4.1.1 Learning With Errors (LWE)

L'algoritmo standardizzato dal NIST (Kyber) usa il problema *Learning With Errors*. Si tratta di risolvere un sistema lineare:

$$\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \pmod{q} \tag{4.1}$$

Il termine $\mathbf{e}$ (errore) rende il sistema "sfocato". Un computer classico o quantistico dovrebbe provare tutte le combinazioni possibili di $\mathbf{e}$ per risalire a $\mathbf{s}$, rendendo l'attacco impraticabile.

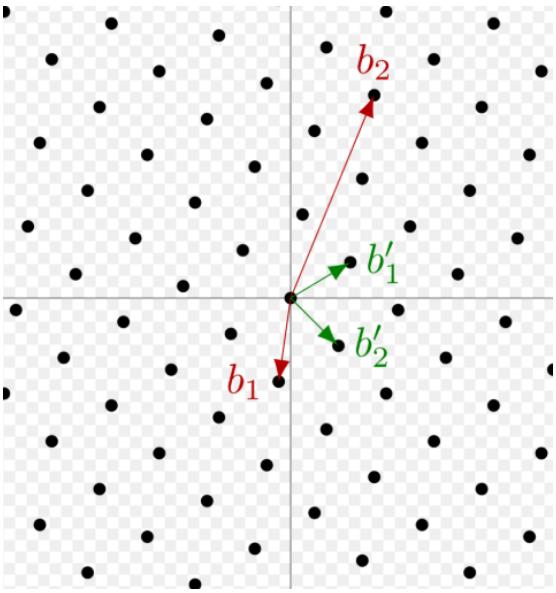


Figura 4.1: Rappresentazione 2D di un reticolo. La PQC sfrutta la difficoltà di orientarsi in reticoli a centinaia di dimensioni.

# Capitolo 5

# Conclusioni

In questo elaborato abbiamo percorso un viaggio che inizia dalla fisica fondamentale del 1920 e arriva alle sfide ingegneristiche del 2024. Abbiamo dimostrato che:

1. Il **Qubit** non è solo un concetto astratto, ma una realtà fisica che sfrutta la sovrapposizione e l'entanglement.

2. L'algoritmo di **Shor** rappresenta una minaccia esistenziale per RSA, poiché trasforma la fattorizzazione da problema intrattabile a problema polinomiale.

3. Il protocollo **BB84** offre una via d'uscita elegante: usando il Teorema di No-Cloning, garantisce che ogni tentativo di intercettazione venga rivelato come un errore statistico (QBER).

La sicurezza del futuro sarà ibrida: useremo la PQC per autenticarci su Google e la QKD per trasferire i segreti di stato. La fisica quantistica, dunque, non ha distrutto la sicurezza; l'ha semplicemente costretta ad evolversi.

# Bibliografia


[1] Shor, P. W. (1994). *Algorithms for quantum computation.*

[2] Bennett, C. H., & Brassard, G. (1984). *Quantum cryptography.*

[3] Feynman, R. P. (1982). *Simulating physics with computers.*

[4] Gerlach, W., & Stern, O. (1922). *Der experimentelle Nachweis.*

[5] NIST. (2022). *Post-Quantum Cryptography Standardization.*

[6] Regev, O. (2005). *On lattices, learning with errors.*

[7] Nielsen, M. A., & Chuang, I. L. (2010). *Quantum Computation and Quantum Information.* Cambridge University Press.