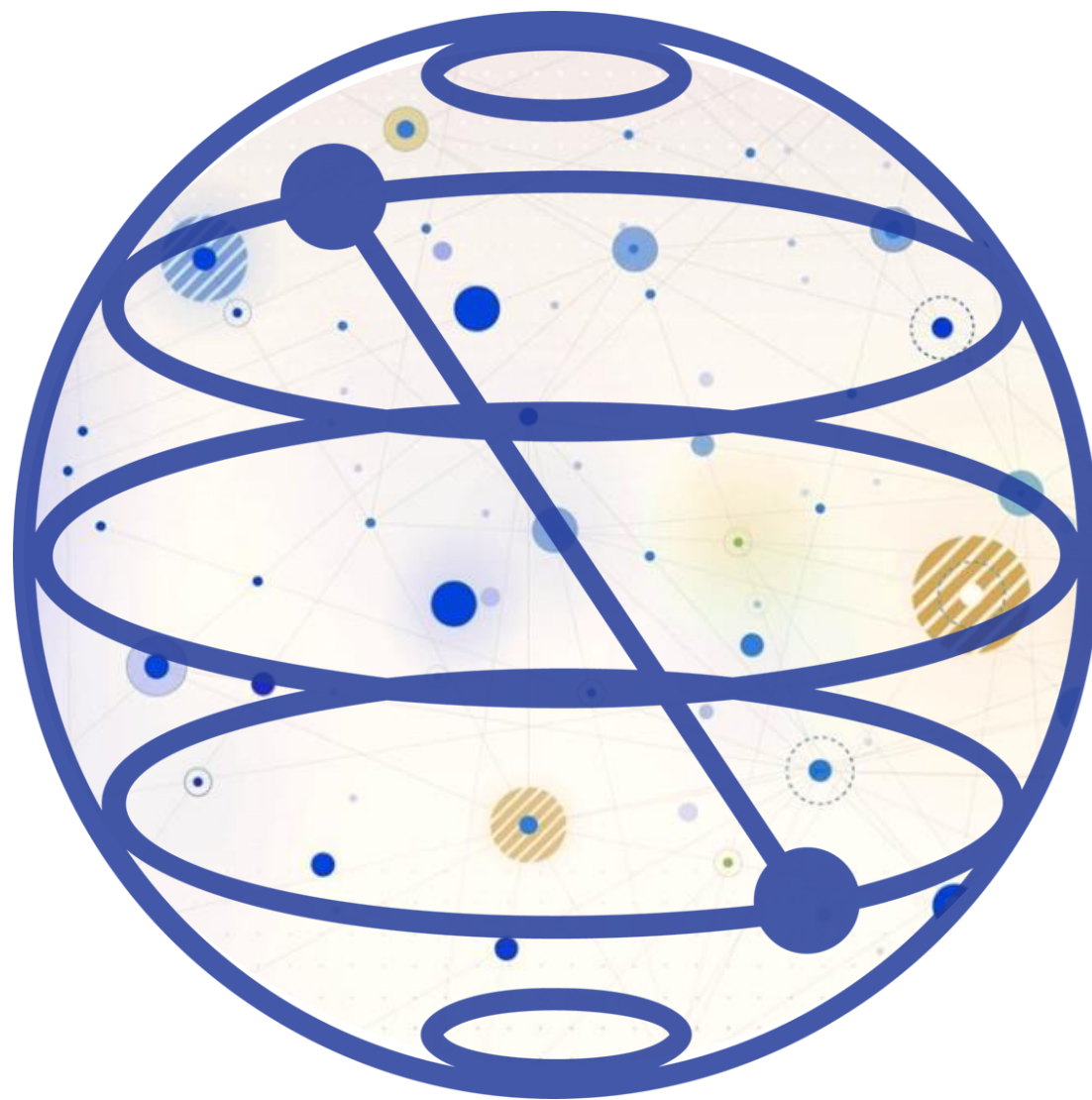


Corso di Quantum Computing - Giorno 4

Corso per Epigenesys s.r.l.

Docenti: Sara Galatro e Lorenzo Gasparini

Supervisore: Prof. Marco Pedicini



$$\frac{22}{48(1)} \left(\frac{1}{R} + \frac{1}{92} \right)$$

C) 148 / .00 P = 55 (C)

$$\left(\frac{1}{87} \right) \left\{ 1 \sqrt{85/125} \left(\frac{65}{227} \right) \frac{68}{713} = \right.$$

$$\pi \quad \phi \sqrt{319} \pi \quad 45000 / \pi$$

axis

$$\left(\frac{C}{2} \right) \quad \text{♀} \quad (95) \text{ LTT}$$

$$x \text{ hill}^4 \quad \frac{9}{10} (1 \div) \frac{1}{101} \geq \left(\frac{\text{eldu}}{\text{mork}} \right)$$

$$+ \frac{\pi}{148} \quad \text{Sture H}$$

$b_x = p$

$$\frac{15}{100} \left(\frac{4}{3} \right) + \sqrt{95619} \quad (\ddot{v})$$

Aniden Lindgrist

$\varphi = \lambda$

$T_F = \frac{9}{5}$

ΔE_{int}

Gina Eden V

W =

2/2

$$v_x \quad \text{Anscom}$$

$$|\psi(x,y,z,t)|^2$$

$$= \int |\psi|^2 dv \quad \frac{a1 - g}{|a1|}$$

Recap Matematico

Campo Complesso

Un **numero complesso** è un numero della forma

$$z = a + i b$$

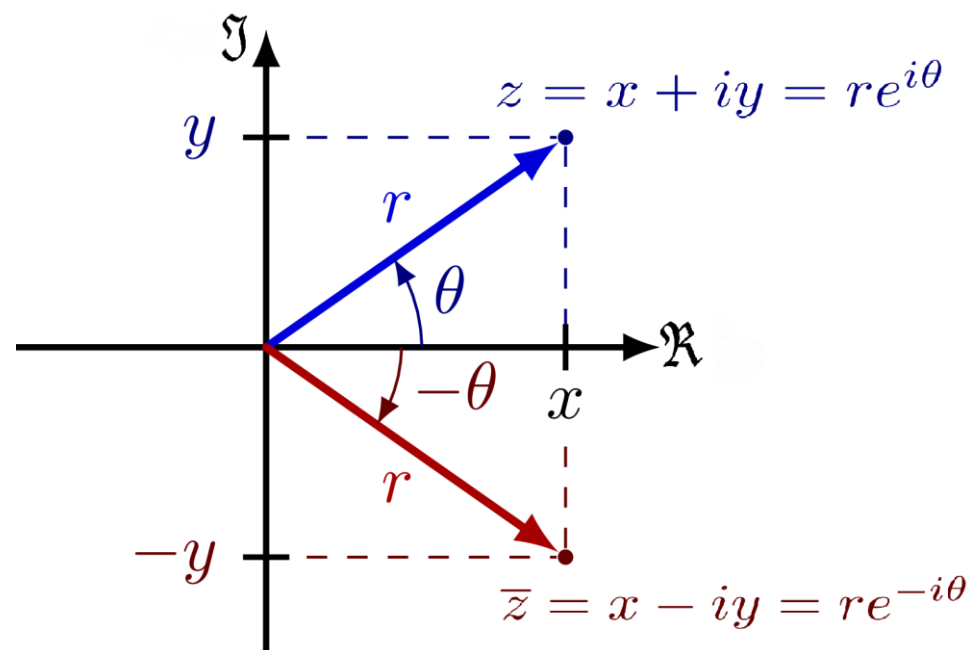
con $a, b \in \mathbb{R}$ e dove i , detta **unità immaginaria**, è soluzione dell'equazione $x^2 = -1$.

La scrittura $a + ib$ è detta **forma algebrica** di un numero complesso.

Una scrittura equivalente è la **forma esponenziale**, definita come

$$z = r e^{i\theta} = r(\cos(\theta) + i \sin \theta)$$

dove $\theta \in [0, 2\pi]$ è l'**angolo** che z forma con il semi-asse positivo dei reali e $r := |z| = \sqrt{a^2 + b^2}$ è detto **modulo** di z .



Matrici Unitarie e Hermitiane

- Una matrice \mathbf{A} è detta **unitaria** se la sua inversa coincide con la sua trasposta coniugata, ossia se $\mathbf{A}^{-1} = \mathbf{A}^\dagger$.
- **Condizioni equivalenti** a tale definizione sono:
 1. \mathbf{A} è unitaria;
 2. \mathbf{A} preserva il prodotto interno, i.e. $\langle \mathbf{A}\mathbf{v} | \mathbf{A}\mathbf{w} \rangle = \langle \mathbf{v} | \mathbf{w} \rangle$ per ogni \mathbf{v}, \mathbf{w} ;
 3. \mathbf{A} preserva la norma, i.e. $\|\mathbf{A}\mathbf{v}\| = \|\mathbf{v}\|$ per ogni \mathbf{v} ;
 4. $\|\mathbf{A}\mathbf{v}\| = 1$ se $\|\mathbf{v}\| = 1$;
 5. \mathbf{A} ha per colonne vettori ortonormali.
- Una matrice è detta **Hermitiana** se $\mathbf{H} = \mathbf{H}^\dagger$ e ricordiamo che identificano gli osservabili quantistici (primo postulato).
- Le matrici unitarie ed Hermitiane (quadrate) sono esempi di **matrici normali**, i.e. matrici che commutano con la loro coniugata trasposta:

$$\mathbf{H}\mathbf{H}^\dagger = \mathbf{H}^\dagger\mathbf{H}$$

Autovalori e Autovettori

- Un numero complesso λ è detto **autovalore** di una matrice quadrata \mathbf{A} se esiste un vettore non nullo \mathbf{v} , detto **autovettore**, tale che $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$.
- **Teorema Spettrale**: sia \mathbf{M} una matrice normale di dimensioni $N \times N$ a entrate complesse. Allora esistono una base ortonormale di vettori N -dimensionali complessi $\{|\psi_1\rangle, \dots, |\psi_N\rangle\}$ e N numeri complessi $\lambda_1, \dots, \lambda_N$ tali che

$$\mathbf{M} = \sum_{k=1}^N \lambda_k |\psi_k\rangle\langle\psi_k| \rightarrow \text{Decomposizione spettrale}$$

- Ad esempio, se definiamo $|\psi_\theta\rangle = \cos \theta |0\rangle + \sin \theta |1\rangle$, possiamo decomporre il gate Hadamard come

$$\mathbf{H} = |\psi_{\pi/8}\rangle\langle\psi_{\pi/8}| - |\psi_{5\pi/8}\rangle\langle\psi_{5\pi/8}|$$

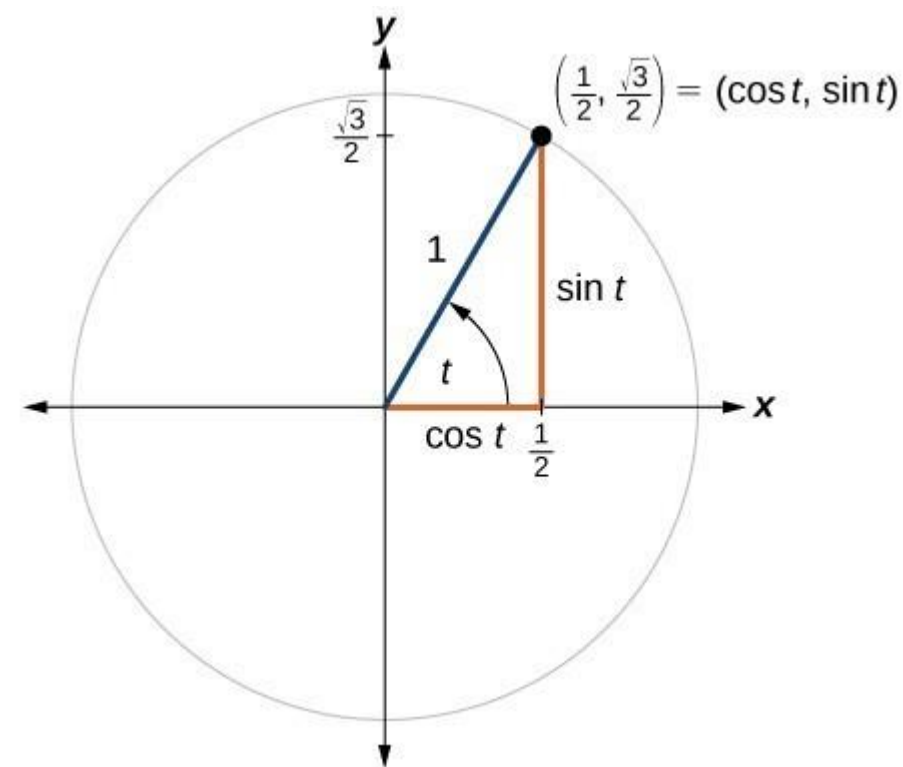
Autovalori e Autovettori

- Analizziamo lo spettro di una matrice unitaria: dato che una matrice unitaria mantiene la norma, si ha che

$$\| |\psi_k\rangle \| = \| \mathbf{U} |\psi_k\rangle \| = |\lambda_k| \cdot \| |\psi_k\rangle \|$$

dunque, dato che è solito supporre $|\psi_k\rangle \neq \vec{0}$, otteniamo che $\text{Sp}(\mathbf{U}) \equiv S^1 = \{z \in \mathbb{C} : |z| = 1\}$.

- Tramite forma esponenziale siamo dunque in grado di identificare univocamente un autovalore solo dall'angolo θ , chiamato anche **fase**.



The background is a light cream color with a complex network of thin, light green lines. These lines form a grid-like structure with various geometric shapes, including squares and circles, some of which are filled with a solid green color. The overall aesthetic is technical and modern, reminiscent of a quantum circuit diagram or a digital network.

Quantum Phase Estimation

Definizione del problema

- Supponiamo di avere uno stato quantistico $|\psi\rangle$ di n qubit e un circuito unitario che agisce su questi qubit. Ci viene **promesso** che lo stato $|\psi\rangle$ è un autovettore della matrice unitaria \mathbf{U} associata al circuito e vogliamo approssimare l'autovalore λ corrispondente a $|\psi\rangle$.
- Nello specifico, dato che λ si trova sul cerchio unitario, possiamo scrivere

$$\lambda = e^{2\pi i \theta}$$

con θ unico numero reale compreso in $0 \leq \theta \leq 1$ per cui sia vera tale relazione. Il nostro algoritmo dovrà riuscire ad **approssimare tale θ** e restituirlo come output.

- Alcune osservazioni:
 - a) Questo problema ha come **input** uno stato quantistico;
 - b) Al momento non stiamo facendo ipotesi sul **livello di approssimazione**;
 - c) Ci stiamo muovendo su un **cerchio**, dunque avere $\theta \approx 1$ è equivalente ad avere $\theta \approx 0$.

The background of the slide is a dark, moody image. On the left side, there is a close-up of a combination lock with several dials visible, showing numbers like 4, 5, and 6. To the right and slightly below the lock, there is a blurred image of a circuit board with various electronic components and traces. The overall lighting is low, creating a high-tech, mysterious atmosphere.

Algoritmo di Shor

Introduzione



- Ci dedicheremo sull'algoritmo fulcro del quantum computing, **l'algoritmo di Shor**, ideato da Peter Shor nel 1994.
- L'algoritmo ha riscosso successo poiché è stato il primo algoritmo a garantire uno speed-up esponenziale in relazione ad un problema alla base di tante applicazioni crittografiche.
- Vedremo prima il problema di **order-finding** e come esso si lega al problema della **fattorizzazione**, da sempre creduto un problema intrattabile classicamente.
- In seguito passiamo alla soluzione quantistica del problema di order-finding, che fonda le sue basi sull'algoritmo della **Quantum Phase Estimation**.
- Infine faremo un panoramica sull'**impatto crittografico** che un tale algoritmo ha sugli attuali sistemi di sicurezza informatici.

Aritmetica modulare



- Ricordiamo brevemente alcuni concetti sulle proprietà dei numeri nell'ottica dell'aritmetica modulare.

- Sia N un intero, definiamo il **gruppo delle classi resto modulo N** come

$$\mathbb{Z}_N = \{0, 1, \dots, N - 1\}$$

- Ricordiamo anche la definizione di **gruppo moltiplicativo**

$$\mathbb{Z}_N^* = \{x \in \mathbb{Z}_N \mid \text{MCD}(x, N) = 1\}$$

- Per ogni $a \in \mathbb{Z}_N^*$ definiamo **l'ordine di a modulo N** , indicato con **$\text{ord}_N(a)$** , come il più piccolo intero positivo r tale per cui $a^r \equiv_N 1$.
- Si definisce il periodo solo per elementi coprimi con N perché, se a ed N avessero fattori in comune, non ci sarebbe una potenza x per cui $a^x \equiv_N 1$, dato che ogni potenza di a avrebbe fattori in comune con N .

Esempio \mathbb{Z}_N

- Prendiamo $N = 15 = 3 \cdot 5$ allora le classi resto modulo 15 sono $\mathbb{Z}_{15} = \{0, \dots, 14\}$.
- Lavorando in \mathbb{Z}_N , si possono fare somme, moltiplicazioni. L'importante è rimanere in \mathbb{Z}_N riducendo sempre modulo N .
- **Somma:** $7 + 10 = 17 \equiv_{15} 2$.
- L'inverso di un elemento a rispetto alla somma esiste sempre ed è $-a$, dato che $a - a \equiv_N 0$ per qualsiasi modulo N . \mathbb{Z}_N è un gruppo rispetto alla somma.
- **Moltiplicazione:** $3 \cdot 9 = 27 \equiv_{15} 12$.
- L'inverso di un elemento a rispetto alla moltiplicazione esiste solo se a è coprimo con N , cioè non hanno fattori in comune. Se a è coprimo con N allora $a \cdot a^{-1} \equiv_N 1$ ha senso.
- **Inverso:** $7^{-1} \equiv_{15} 13$.

Esempio \mathbb{Z}_N^*

- Consideriamo sempre $N = 15$. Il gruppo moltiplicativo \mathbb{Z}_N^* è definito come tutti gli elementi in \mathbb{Z}_N coprimi con 15. Si verifica facilmente che

$$\mathbb{Z}_{15}^* = \{1, 2, 4, 7, 8, 11, 13, 14\}.$$

- \mathbb{Z}_N^* **è un gruppo rispetto alla moltiplicazione** e, come già detto, ogni elemento ha un ordine modulo N .
- Calcoliamo l'ordine di 7 modulo 15. Dobbiamo calcolare tutte le potenze 7^k fino a che non arriviamo ad 1:
 - $7^1 \equiv_{15} 7$
 - $7^2 \equiv_{15} 4$
 - $7^3 \equiv_{15} 13$
 - $7^4 \equiv_{15} 1$
- Abbiamo verificato che $\text{ord}_{15}(7) = 4$.

Order-Finding



- Con le basi di aritmetica modulare, passiamo a definire il problema centrale.
- Siano dati in input due interi N e a entrambi ad n bit, con $a \in \mathbb{Z}_N^*$. Il **problema di order-finding** consiste nel trovare il periodo $r = \text{ord}_N(a)$.
- La parte quantistica dell'algoritmo di Shor, in effetti, non è specifica per la risoluzione del problema della fattorizzazione bensì del problema di order-finding.
- Ma vedremo in che modo, tramite una semplice riduzione classica, una soluzione del problema di order-finding porta alla soluzione del problema della fattorizzazione di un numero intero.
- Il problema di order-finding infatti è un problema decisamente più generale rispetto a quello della fattorizzazione.

Order Finding e Fattorizzazione



- Passiamo a vedere in che modo fattorizzare un numero intero si riduce al trovare l'ordine di un elemento nel gruppo moltiplicativo.
- Consideriamo per semplicità che l'intero N da fattorizzare sia prodotto di due primi distinti $N = p \cdot q$. Il caso in questione è il caso direttamente coinvolto nelle applicazioni crittografiche, in particolare nello schema RSA.
- Selezioniamo un elemento a coprimo con N e supponiamo di avere a disposizione $r = \text{ord}_N(a)$ tale che r sia un numero pari.
- Se r non è pari, si sceglie un diverso a . Sono sufficienti pochi guess di a per avere una buona probabilità di avere un periodo r che sia pari.
- Se r è pari, a^r è un quadrato modulo N e per ipotesi $a^r - 1 \equiv_N 0$. Dunque $a^r - 1$ deve essere per forza un multiplo di N .

Order Finding e Fattorizzazione

- Scriviamo $a^r - 1$ tramite la classica forma della differenza di quadrati:

$$a^r - 1 = \left(a^{\frac{r}{2}} + 1\right) \cdot \left(a^{\frac{r}{2}} - 1\right).$$

- N divide $a^r - 1$, inoltre N è prodotto di due primi distinti p e q . Abbiamo quindi tre possibilità per i fattori di N :
 - p divide $\left(a^{\frac{r}{2}} + 1\right)$ e q divide $\left(a^{\frac{r}{2}} - 1\right)$.
 - q divide $\left(a^{\frac{r}{2}} + 1\right)$ e p divide $\left(a^{\frac{r}{2}} - 1\right)$.
 - p e q dividono entrambi $\left(a^{\frac{r}{2}} + 1\right)$.
- Il caso in cui sia p che q dividono $\left(a^{\frac{r}{2}} - 1\right)$ contraddice la minimalità di r .
- Date queste tre possibilità, diciamo che a è una buona scelta se $\text{MCD}\left(a^{\frac{r}{2}} + 1, N\right) \in \{p, q\}$.

Order Finding e Fattorizzazione

- Nel caso in cui abbiamo dunque una buona scelta di a , conoscendo r , possiamo trovare uno dei fattori di N semplicemente calcolando il massimo comun divisore tra $\left(a^{\frac{r}{2}} + 1\right)$ e il modulo N .
- Otteniamo così un fattore di N ed otteniamo l'altro tramite una semplice divisione.
- Abbiamo visto quindi che è possibile fattorizzare un intero $N = pq$ a partire da una soluzione del problema di order-finding.
- Tale **riduzione è efficiente** poiché
 - il calcolo della potenza $a^{\frac{r}{2}}$ può essere portato a termine con il famoso trucco del repeated squaring per le esponenziazioni modulari,
 - Il calcolo del massimo comun divisore, tramite l'utilizzo dell'algoritmo di Euclide, è estremamente facile da portare a termine.

QPE e algoritmo di Shor



- La parte fondamentale per portare a termine la fattorizzazione di un numero intero è **trovare l'ordine r di un certo elemento a coprimo con N .**
- A questo scopo viene utilizzata la Quantum Phase Estimation, come subroutine quantistica, adattata ovviamente al problema di order finding.
- Parlando della QPE, **l'operatore U** e **l'autovettore $|\psi\rangle$** sono dati in input. Ma nell'ottica del problema di order-finding è necessario definire come sono fatti questi due oggetti.
- Il punto sarà quindi definire l'operatore e l'autovettore sui quali viene eseguita la QPE per trovare il periodo r .
- Essendo la QPE la chiave per la soluzione al problema di order-finding, è chiaro che, per come funziona la QPE, essa provvede a fornire un'approssimazione della fase associata all'autovettore di U . Andiamo per passi e procediamo definendo U e $|\psi\rangle$.

L'operatore



- Consideriamo la seguente operazione definita sugli stati della base standard $|0\rangle, \dots |N-1\rangle$ a seconda del valore di $a \in \mathbb{Z}_N^*$:

$$U_a |x\rangle = |ax \bmod N\rangle$$

- Osserviamo che l'operatore è una matrice di permutazione, dato che mappa uno stato della base in un altro stato della base.
- Notiamo una cosa fondamentale riguardo tale operatore U_a : prendiamo due elementi invertibili $a, b \in \mathbb{Z}_N^*$, allora il prodotto sequenziale tra U_a e U_b è dato da U_{ab} come si vede da tale semplice relazione:

$$U_a U_b |x\rangle = U_a |bx\rangle = |abx\rangle$$

Proprietà di U_a

- Da quest'ultima osservazione concludiamo due importanti proprietà sulla matrice U_a , soprattutto nell'ottica di operatore per la Quantum Phase Estimation.

- 1) L'operatore inverso di U_a è dato da $U_{a^{-1}}$, il quale è univocamente determinato dall'inverso di a modulo N (esiste sicuramente supponendo a coprimo con N):

$$U_a U_{a^{-1}} |x\rangle = U_{a \cdot a^{-1}} |x\rangle = U_1 |x\rangle = \mathbb{I} |x\rangle = |x\rangle.$$

- 2) Le potenze U_a^k dell'operatore sono date da U_{a^k} e quindi univocamente determinate dalla potenza a^k dell'elemento a . In particolare le potenze del tipo $U_a^{2^k}$ sono determinate dalle potenze a^{2^k} :

$$U_a U_a \dots U_a U_a |x\rangle = U_{a^k} |x\rangle = |a^k x\rangle.$$

Implementazione U_a

- Per implementare l'operatore dobbiamo prima chiarire lo spazio in cui ci troviamo.
- A priori non è detto che N sia una potenza di 2. Quindi, per descrivere l'azione di U_a sugli stati $|0\rangle, \dots, |N-1\rangle$, **è necessario espandere lo spazio degli stati** alla potenza di 2 più vicina ad N .
- Lavoreremo con un numero n di bit, necessari per rappresentare tutti gli stati di sopra. Ci basta prendere **$n = \lceil \log N \rceil$** , così da espandere lo spazio degli stati al minimo indispensabile rimanendo coerenti con uno spazio quantistico.
- Operando in uno spazio degli stati $|0\rangle, \dots, |2^n - 1\rangle$, l'operatore U_a deve agire su ognuno di essi. Sapendo già come agisce U_a sugli stati $|0\rangle, \dots, |N-1\rangle$, dobbiamo definire in che modo esso agisca sui rimanenti $|N\rangle, \dots, |2^n - 1\rangle$.
- L'idea è semplice: vogliamo **che U_a si comporti come l'operatore identità sugli stati $|N\rangle, \dots, |2^n - 1\rangle$** .

Implementazione U_a

- Passiamo quindi all'implementazione dell'operatore, facendo uso dei metodi visti per la simulazione quantistica di circuiti classici.

1) Sia data la funzione classica f_a definita come segue:

$$f_a(x) = \begin{cases} ax \bmod N & 0 \leq x \leq N-1 \\ x & N \leq x \leq 2^n - 1 \end{cases}$$

2) Costruiamo, tramite simulazione gate-by-gate, il circuito quantistico che implementi l'operazione $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus f_a(x)\rangle$. Facciamo due semplici osservazioni sul costo:

- f_a come operazione classica è implementata tramite moltiplicazioni e divisioni con resto, entrambe operazioni con costo $O(n^2)$, da cui un costo globale quadratico di f_a .
- Possiamo simulare f_a con un numero di gate quantistici pari a $O(t)$, dove t è il numero di gate nella costruzione classica di f_a . Da ciò possiamo concludere che l'implementazione dell'operazione che mappa $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus f_a(x)\rangle$, **ha costo $O(n^2)$ e quindi risulta efficiente.**

Implementazione U_a

- 3) Scambiamo i due registri, facendo uso di n *SWAP* gate per scambiare individualmente i qubit, così da ottenere lo stato $|y \oplus f_a(x)\rangle |x\rangle$.
- 4) Procediamo analogamente al passo 1, costruendo il circuito quantistico che simula la funzione inversa $f_{a^{-1}}$, implementando l'operazione che mappa $|x\rangle |y\rangle \mapsto |x\rangle |y \oplus f_{a^{-1}}(x)\rangle$. Anche qui un'osservazione sul costo:
 - Ugualmente ad f_a , **la simulazione di $f_{a^{-1}}$ richiede un costo pari a $O(n^2)$** e il calcolo dell'inverso a^{-1} a partire da a è efficiente sfruttando l'algoritmo di Euclide. Il costo globale è dunque efficiente.
- 5) Applichiamo questo circuito inverso allo stato ottenuto al passo precedente:

$$|y \oplus f_a(x)\rangle |x\rangle \mapsto |y \oplus f_a(x)\rangle |x \oplus f_{a^{-1}}(y \oplus f_a(x))\rangle$$

- Seguendo il procedimento appena descritto, inizializzando il secondo registro a $|y\rangle = |0^n\rangle$, riusciamo a costruire l'operatore U_a che porta avanti la seguente operazione:

$$|x\rangle |0^n\rangle \mapsto |f_a(x)\rangle |0^n\rangle$$

Implementazione $U_a^{2^k}$

- Come ultima cosa sull'implementazione dell'operatore della QPE, rimane da stabilire come implementare le potenze $U_a^{2^k}$.
- Questo operatore U_a risulta essere molto speciale nell'ottica dell'efficienza della QPE. In generale l'implementazione di tutte le potenze U^{2^k} , di un generico operatore U , ha un costo che cresce esponenzialmente con il crescere di k .
- Il fatto che $U_a^{2^k} = U_{a^{2^k}}$, ci permette di costruire $U_a^{2^k}$ senza reiterare 2^k volte il circuito di U_a , bensì calcolando direttamente a^{2^k} e **costruire** $U_{a^{2^k}}$ **simulando** $f_{a^{2^k}}$, seguendo il procedimento per la simulazione di f_a .
- Per nostra fortuna il calcolo di a^{2^k} modulo N può essere portato a termine efficientemente da un computer classico con l'utilizzo del famoso algoritmo del **repeated squaring** con un costo pari a $O(n^3)$.
- In conclusione, l'implementazione di ogni potenza $U_a^{2^k}$ ha costo cubico $O(n^3)$.

L'autovettore



- Abbiamo descritto in maniera esaustiva l'operatore che giocherà il ruolo primario nella Quantum Phase Estimation. Va allora definito con quale autovettore inizializzeremo il nostro circuito.
- Definiamo i seguenti r autovettori di U_a :

$$|\psi_k\rangle = \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} \omega_r^{-jk} |a^j\rangle$$

- Non è difficile verificare che **$|\psi_k\rangle$ è autovettore di U_a con autovalore associato $\lambda_k = \omega_r^k$** per ogni $k \in \{0, 1, \dots, r-1\}$.

Trovare il periodo con la QPE

- Iniziamo eseguendo la **Quantum Phase Estimation con operatore $U = U_a$ e autovettore $|\psi\rangle = |\psi_k\rangle$, facendo uso di m qubit di controllo.**

- Analizziamo prima il caso in cui inizializziamo il registro target con l'autovettore $|\psi_1\rangle$:

$$|\psi_1\rangle = \frac{1}{\sqrt{r}} \left(|1\rangle + \omega_r^{-1}|a\rangle + \omega_r^{-2}|a^2\rangle + \dots + \omega_r^{-(r-1)}|a^{r-1}\rangle \right).$$

- L'autovalore associato a $|\psi_1\rangle$ è $\omega_r = e^{2\pi i \cdot \frac{1}{r}}$.
- La QPE ci permette di ottenere un'approssimazione (ad m bit) della fase, che in questo caso specifico è data da $\theta = \frac{1}{r}$. Da ciò ci basta calcolare il reciproco per ottenere una stima del periodo cercato r .

Approssimare correttamente r

- Usando m bit di precisione, **otteniamo un'approssimazione del tipo** $\frac{x}{2^m}$ per un qualche $x \in \{0, 1, \dots, 2^m - 1\}$.
- Da tale approssimazione, possiamo **ricavare r arrotondando** $\frac{2^m}{x}$ all'intero più vicino.
- Quanto deve essere grande m affinché la stima di r sia esatta? Vogliamo evitare che $\frac{1}{r}$ venga confuso con la sua frazione più vicina $\frac{1}{r+1}$, la cui distanza da $\frac{1}{r}$ è data da:

$$\frac{1}{r} - \frac{1}{r+1} = \frac{1}{r(r+1)} = d_r$$

Approssimare correttamente r

- Richiediamo che l'approssimazione $\frac{x}{2^m}$ non più della metà di d_r dal valore preciso $\frac{1}{r}$.

$$\left| \frac{x}{2^m} - \frac{1}{r} \right| < \frac{1}{2r(r+1)}$$

- Tale stima deriva tutta da r . Possiamo però andare leggermente in eccesso del numero di qubit di controllo sapendo che $r < N$.
- Possiamo garantire una precisione sufficiente per trovare r senza ambiguità, impostando la seguente condizione più forte:

$$\left| \frac{x}{2^m} - \frac{1}{r} \right| < \frac{1}{2N^2} < \frac{1}{2r(r+1)}$$

- Scegliendo $m = 2\lceil \log N \rceil + 1$, la relazione è soddisfatta e otteniamo l'approssimazione $\frac{2^m}{x}$, il cui arrotondamento ci porta alla soluzione esatta r .

Caso generale $|\psi_k\rangle$

- Consideriamo il caso generale in cui l'autovettore da cui partiamo è un generico $|\psi_k\rangle$.
- L'autovalore associato è $\omega_r^k = e^{2\pi i \cdot \frac{k}{r}}$.
- Eseguendo la QPE su un tale autovettore, otteniamo una stima $\frac{x}{2^m}$ per la fase $\theta = \frac{k}{r}$. Come ricaviamo r a partire $\frac{x}{2^m}$?
- Qui entra in gioco un importante algoritmo classico, l'algoritmo delle **frazioni continue**.
- Dato un intero $N \geq 2$ ed un numero reale $\alpha \in (0, 1)$, esiste al massimo una scelta di $u, v \in \{0, \dots, N-1\}$ coprimi tra loro tale che $\left| \alpha - \frac{u}{v} \right| \leq \frac{1}{2N^2}$. Dati α ed N , l'algoritmo delle frazioni continue trova u e v con un costo computazionale pari a $O(n^3)$ se N è a n bit.
- Nel nostro caso, θ gioca il ruolo di α .

Caso generale $|\psi_k\rangle$

- Tramite l'algoritmo delle frazioni continue **otteniamo la frazione $\frac{u}{v} = \frac{k}{r}$ ridotta ai minimi termini.**
- A partire da $\frac{u}{v}$ non possiamo direttamente ricavare r .
- In generale k ed r possono avere fattori in comune, quindi la soluzione $\frac{u}{v}$ potrebbe corrispondere a più scelte di k ed r .
- Per ovviare a tale problema ci basterà semplicemente iterare il circuito ottenendo valori diversi del tipo:

$$\frac{k_j}{r_j} = \frac{k}{r}$$

Caso generale $|\psi_k\rangle$

- Avendo a disposizione varie coppie k_j, r_j , possiamo ottenere il periodo r calcolando

$$\text{mcm}(r_1, \dots, r_j, \dots, r_t) = r$$

- Bastano poche iterazioni del circuito per avere un'alta probabilità di ottenere r calcolando il minimo comune multiplo.
- L'idea generale è che è poco probabile di scegliere tanti k randomici tutti con fattori in comune con r .
- Dopo poche iterazioni ci si aspetta di ottenere un k coprimo con r .

Inizializzazione dell'autovettore

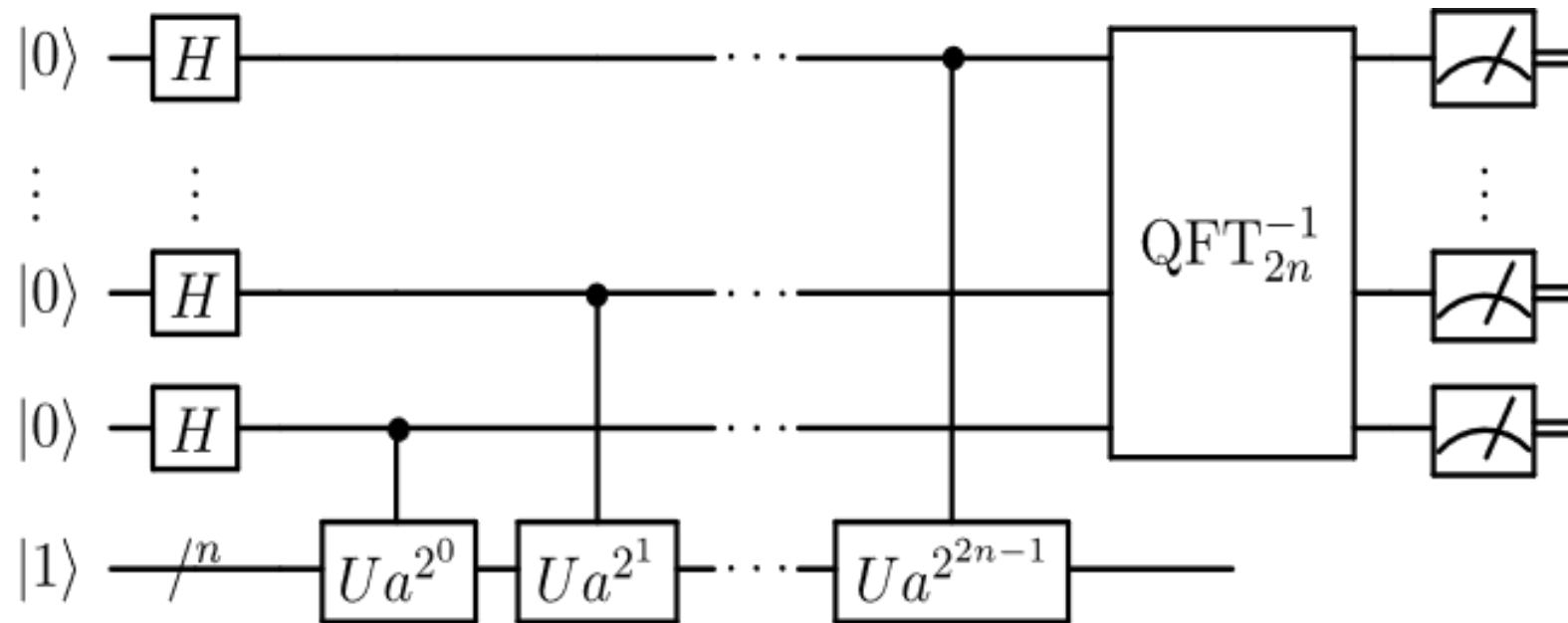


- Nella pratica non avremo a disposizione l'autovettore $|\psi_k\rangle$ su cui applicare la Quantum Phase Estimation.
- Serve inizializzare il secondo registro su uno stato facilmente riproducibile.
- Inizializzeremo lo stato in una **sovrapposizione di tutti gli autovettori** $|\psi_k\rangle$. Ciò è facilmente implementabile poiché vale la seguente relazione:

$$|1\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |\psi_k\rangle$$

- Applicando la QPE, inizializzando il secondo registro a $|1\rangle$, si ottiene in ogni caso un valore k randomicamente da $\{0, \dots, r-1\}$, da cui possiamo applicare gli stessi ragionamenti fatti precedentemente per ottenere r .

Il circuito quantistico



- Questo è il circuito quantistico completo della Quantum Phase Estimation con operatore U_a e autovettore inizializzato a $|1\rangle$. **Tale circuito, iterato un numero sufficiente di volte, permette di ricavare $r = \text{ord}_N(a)$.**


Analisi computazionale

- Analizziamo i costi computazionali del circuito presentato elencando di seguito i principali contributi:
 - m operatori controllati $CU_a^{2^k}$, il costo di ognuno dei quali è $O(n^2)$. Scegliendo $m = 2n$, abbiamo un costo totale di questi operatori pari a $O(n^3)$.
 - m gate di Hadamard che contribuiscono linearmente al costo globale.
 - Una singola Quantum Fourier Transform su m qubit, la quale necessita di $O(n^2)$ gate per l'implementazione.
 - I calcoli classici che devono essere effettuati sono il calcolo di a^{-1} e delle potenze a^{2^k} , entrambi efficientemente implementabili in tempo cubico $O(n^3)$.
- **Il costo globale del circuito è in conclusione $O(n^3)$** , un costo polinomiale che rende l'algoritmo di Shor un algoritmo efficiente e uno strumento computazionalmente potente per il problema della fattorizzazione.

L'algoritmo completo



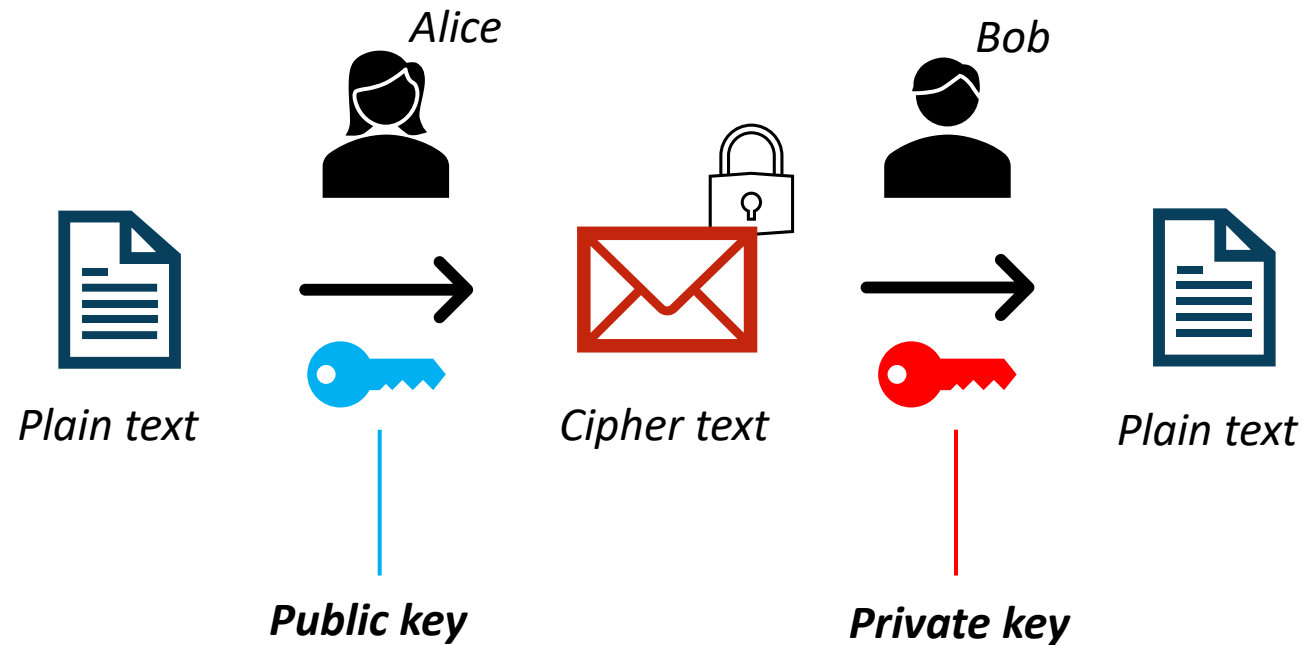
- Sia dato in input un intero N dispari non una potenza di un primo.
 - 1) Scegliere randomicamente un elemento $a \in \{2, \dots, N - 1\}$.
 - 2) Calcolare $d = \text{MCD}(a, N)$.
 - 3) Se $d > 1$, allora restituiamo in output $b = d$ e $c = \frac{N}{d}$. Altrimenti continuare con il prossimo step.
 - 4) Calcolare $r = \text{ord}_N(a)$ tramite la subroutine quantistica (QPE). Se r è dispari tornare al punto 1, altrimenti proseguire al prossimo step.
 - 5) Calcolare $x = a^{\frac{r}{2}} - 1$ (modulo N) e $d = \text{MCD}(x, N)$.
 - 6) Se $d > 1$, allora restituiamo in output $b = d$ e $c = \frac{N}{d}$. Altrimenti tornare al punto 1.

The background of the image is a close-up, high-angle shot of a copper-colored printed circuit board (PCB). The board is covered in a complex network of white and gold-colored conductive traces and pads. A silver-colored metal padlock is positioned diagonally across the upper left portion of the frame. The padlock has a rectangular body with rounded corners and a U-shaped shackle. The text "Conseguenze crittografiche" is overlaid in white, sans-serif font, centered horizontally and partially overlapping the padlock and the circuit board. A thin white horizontal line is positioned below the text.

Conseguenze crittografiche

Crittografia a chiave pubblica

- La crittografia a chiave pubblica prevede l'uso di una coppia di chiavi, una **chiave privata** e una **chiave pubblica**.



Crittografia a chiave pubblica



- La crittografia a chiave pubblica offre molti servizi:
 - Cifratura e decifratura per garantire la **confidenzialità** delle comunicazioni.
 - Firme digitali per l'**autenticità**, l'**integrità** e il **non-ripudio** del dato.
 - Protocollo di **scambio della chiave** su canali insicuri, per un utilizzo futuro di crittosistemi simmetrici.
- Le applicazioni sono moltissime, tra cui:
 - Internet communication
 - Autenticazione
 - E-mail encryption
 - Secure Shell
 - Virtual Private Network
 - Certificate Authorities

Crittografia a chiave pubblica



- I principali schemi crittografici a chiave pubblica alla base dell'attuale sicurezza informatica sono i seguenti:

RSA

Principale crittosistema a chiave pubblica, la cui sicurezza si basa sulla difficoltà del **problema della fattorizzazione**.

DH

Protocollo di scambio della chiave basato su crittografia a chiave pubblica, la cui sicurezza si basa sulla difficoltà del **problema del logaritmo discreto**.

ECDSA

Schema di firma digitale basato su curve ellittiche, la cui sicurezza è basata sul **problema del logaritmo discreto su curve ellittiche**.

RSA



- L'RSA è un crittosistema che opera sul gruppo \mathbb{Z}_N . Il modulo N è preso come prodotto di due primi distinti $N = p \cdot q$.
- Vengono selezionati due interi e, d tale che $e \cdot d \equiv 1 \text{ mod } \varphi(N)$. In particolare $\varphi(N) = (p - 1) \cdot (q - 1)$.
- La chiave pubblica è data da (N, e) .
- La chiave privata è data da (p, q, d) .
- **Cifratura:** $C \equiv_N M^e$, dove $M \in \mathbb{Z}_N$ è il messaggio da mandare.
- **Decifratura:** $C^d \equiv_N M^{ed} \equiv_N M$, dovuto al fatto che e e d sono inversi modulo $\varphi(N)$.

Breaking RSA



- Tutta **la sicurezza dell'RSA si fonda sul problema della fattorizzazione**, da sempre ritenuto intrattabile da un computer classico.
- Tuttavia, avendo visto l'algoritmo di Shor, sappiamo che esso è in grado di fattorizzare il modulo N in tempo polinomiale con l'ausilio di un computer quantistico.
- Sfruttando l'algoritmo di Shor, un eventuale attaccante Eve che vuole conoscere il messaggio segreto M può fattorizzare N e ricavare efficientemente i fattori p, q , entrambi parte della chiave privata.
- Eve calcola così $\varphi(N) = (p - 1) \cdot (q - 1)$, ricavando d tramite l'algoritmo di Euclide per trovare l'inverso di e modulo $\varphi(N)$.
- Avendo a disposizione il valore d , **Eve è in grado di decifrare C e rompere definitivamente il crittosistema RSA.**

Diffie-Hellman

- Lo schema di **Diffie-Hellman** è il **principale protocollo di scambio della chiave**. Così come per l'RSA, anche nel protocollo DH si lavora sul gruppo \mathbb{Z}_N . Riportiamo gli step principali:
 - 1) Vengono scelti due parametri pubblici (g, N) .
 - 2) Alice sceglie $a \in \mathbb{Z}_N$. Alice calcola $A \equiv_N g^a$ e lo invia a Bob.
 - 3) Bob sceglie $b \in \mathbb{Z}_N$. Bob calcola $B \equiv_N g^b$ e lo invia ad Alice.
 - 4) Alice calcola $B^a = (g^b)^a = g^{ab}$. Bob calcola $A^b = (g^a)^b = g^{ba}$.
 - 5) Alice e Bob hanno ottenuto entrambi il valore g^{ab} che possono usare come chiave condivisa per comunicazioni future.
- La sicurezza di DH si basa sulla difficoltà del **problema del logaritmo discreto (DLP)**: data la relazione $g^x \equiv_N y$, si chiede di trovare il valore $x = \log_g y$.

Breaking modern cryptography



- L'algoritmo di Shor, per come è stato descritto, fattorizza un numero N e permette di rompere l'RSA.
- Ma la sua potenza non si ferma qua. Tramite appropriati adattamenti può andare oltre e risolvere in maniera efficiente altri problemi da sempre considerati intrattabili.
- In particolare **l'algoritmo di Shor è in grado di risolvere il problema del logaritmo discreto**, sia su campi finiti che su curve ellittiche.
- Vengono messi a rischio sistemi alla base dell'attuale sicurezza informatica, come Diffie-Hellman, schemi di firma digitale e anche gli schemi su curve ellittiche basati sulla difficoltà del DLP come l'ECDH (**Elliptic Curve Diffie-Hellman**) e l'ECDSA (**Elliptic Curve Digital Signature Algorithm**).