



Università degli Studi di Salerno
Dipartimento di Informatica

Corso di Laurea Magistrale in Informatica

Analisi e contromisure per bitcoin nell'era del quantum computing

Docente

Prof.ssa **Genoveffa Tortora**

Candidato

Tecchia Dario

0522500736

Anno Accademico 2021-2022

Ringraziamenti

Non vorrei ringraziare chi c'è o chi ci sarà, ma chi ci è stato.

Abstract

Indice

Introduzione	2
1 Quantum Computing	3
1.1 Il Quantum Bit	5
1.1.1 Definizione di quantum bit	5
1.1.2 Rappresentazione geometrica di un qubit	9
1.2 Porte logiche quantistiche	13
1.2.1 Porte Logiche a singolo qubit	13
1.2.2 Porte logiche a qubit multipli	17
1.3 Misurazione di un sistema di qubit	19
1.4 Registri quantistici	20
1.5 Entanglement	21
1.6 Realizzazione di un computer quantistico	23
1.6.1 Classi di complessità	23
1.6.2 Macchina di Turing Quantistica	25
1.6.3 Condizioni per la realizzazione	26
2 Blockchain	31

INDICE	Cap.0
2.1 La storia della Blockchain	31
2.1.1 Introduzione ai Sistemi di Chaum	32
2.1.2 Timestamp	32
2.1.3 Alberi di Merkle	36
2.1.4 Bit gold	37
2.1.5 Bitcoin	39
2.2 Cos’è la blockchain	43
2.2.1 Architettura della blockchain	44
2.2.2 Funzionamento della Blockchain	46
2.3 Algoritmi di consenso	52
2.3.1 Proof of Work	52
2.3.2 Proof of Stake	54
2.4 Blockchain Pubbliche e Private	56
2.4.1 Blockchain Pubbliche	56
2.4.2 Blockchain Private	58
2.5 Generazioni di Blockchain	59
3 Blockchain nell’era quantistica e Bitcoin QR	62
4 Bitcoin QR Implementato	63
5 Conclusioni	64

Elenco delle figure

1.1	Rappresentazione di una Sfera di Bloch	10
1.2	Visualizzazione dei qubit	12
1.3	Visualizzazione dell'applicazione della Porta X	15
1.4	Visualizzazione dell'applicazione della Porta Y	15
1.5	Visualizzazione dell'applicazione della Porta Z	16
1.6	Visualizzazione dell'applicazione della Porta di Hadamard	17
1.7	Visualizzazione degli effetti dell'entanglement	22
1.8	Le classi di complessità per $P = NP$ e $P \neq NP$	24
1.9	La classe di complessità BQP rispetto a quelle classiche	25
1.10	Approcci per la realizzazione di qubit	28
2.1	Sequenza di blocchi	35
2.2	Esempio di albero di Merkle	36
2.3	Messaggio di Satoshi Nakamoto incorporato nella coin-base del primo blocco	40
2.4	Problema del double spending	41
2.5	Andamento del bitcoin	42
2.6	Architettura della blockchain di bitcoin	45

2.7	Principali componenti della transazione	48
2.8	Uso della firma digitale nelle transazioni	49
2.9	Struttura di un blocco contenente diverse transazioni .	50
2.10	Hash Chain	50
2.11	Confronto delle tipologie di rete	51
2.12	Meccanismo Proof-of-Work di bitcoin	52
2.13	Esempio di funzionamento di Hashcash	54

Elenco delle tabelle

1.1	Insieme delle possibili operazioni del gate CNOT	18
2.1	Evoluzione della Blockchain .	32

Introduzione

Introduzione

Capitolo 1

Quantum Computing

L’informatica quantistica combina l’informatica tradizionale con la meccanica quantistica ed è un campo di ricerca in rapida crescita. Questo interesse verso il calcolo quantistico inizia negli anni settanta con lo sviluppo di una serie di tecniche per ottenere il controllo completo di singoli sistemi quantistici.

Fino a quel momento, la teoria classica dell’informatica era stata fondata sulla tesi, ampiamente accettata, di *Church-Turing*, secondo la quale era possibile teorizzare una macchina ideale, nota come *Macchina di Turing*, capace di simulare in modo efficiente qualsiasi modello di calcolo esistente.

Tuttavia, l’emergente paradigma di calcolo basato sulle proprietà meccaniche quantistiche della natura portò molti scienziati a realizzare che, mentre un computer ordinario poteva essere usato per simulare un computer quantistico, era impossibile eseguire questa simulazione in modo efficiente: ogni tentativo di simulare l’evoluzione di un ge-

nerico sistema fisico-quantistico su una macchina di Turing sembrava richiedere un overhead esponenziale di risorse.

R. P. Feynman fu tra i primi fisici ad occuparsi della questione, dando le linee guida sul possibile utilizzo di sistemi quantistici come costituenti di un nuovo tipo di calcolatore; sottolineò, inoltre, come un calcolatore di questo tipo sarebbe allo stesso tempo un "simulatore" ideale per i sistemi quantistici. A partire dalle osservazioni sviluppate in quel periodo, si iniziò a costruire una nuova teoria dell'informazione, che tenesse conto delle possibilità, ancora teoriche, offerte dal calcolatore quantistico. In particolare, una nuova classificazione della complessità computazionale si rese necessaria, grazie alle peculiarità ed ai vantaggi offerti dal nuovo paradigma computazionale.

Contributi fondamentali sono stati dati da David Deutsch che, nel 1985, si chiese se le leggi della fisica quantistica potessero essere usate per derivare una versione ancora più forte della tesi di Church-Turing e tentò di definire un dispositivo computazionale che fosse capace di simulare in modo efficiente un sistema fisico arbitrario [8]. Questo dispositivo sarebbe diventato la moderna concezione di un computer quantistico e che questi dispositivi potessero avere poteri di calcolo ben superiori a quelli dei computer tradizionali, indipendentemente dai loro progressi ottenibili nel calcolo classico.

Negli anni seguenti, lo studio degli algoritmi quantistici si è evoluto come un sotto-campo dell'informatica quantistica con applicazioni di diverso tipo: ricerca e ottimizzazione, machine learning, simulazione di sistemi quantistici e crittografia.

Quest'ultimo campo è quello che più ci interessa, infatti, nel 1994 Peter Shor pubblica l'algoritmo che porta il suo nome per la fattorizzazione degli interi in tempo polinomiale [10]. Questo è stato una svolta epocale nella materia, perché un importante metodo di crittografia asimmetrica noto come RSA si basa sulla supposizione che la fattorizzazione degli interi sia difficile dal punto di vista computazionale. L'esistenza dell'algoritmo quantistico in tempo polinomiale può dimostrare che uno dei protocolli crittografici più usati al mondo sarebbe vulnerabile a un computer quantistico.

1.1 Il Quantum Bit

1.1.1 Definizione di quantum bit

L'informazione non può essere considerata separatamente dalla sua natura fisica: non si può, cioè, mantenere, modificare o trasmettere informazione senza un adeguato supporto fisico. Nei computer tradizionali viene utilizzato come modello fondamentale il *bit*, che rappresenta un sistema a due stati, 0 e 1. La scelta della rappresentazione binaria è dettata dalla semplicità e comodità di realizzazione nei sistemi elettronici. Il bit classico, quindi, mantiene correttamente l'informazione relativa ad una scelta esclusiva tra i due stati possibili in cui si può trovare (con n bit possiamo rappresentare 2^n stati).

La computazione quantistica introduce una nuova unità fondamentale che prende il nome di *quantum bit*, chiamato anche *qubit*. Un

qubit usa i fenomeni meccanici quantistici della sovrapposizione per ottenere una combinazione lineare di due stati. Un bit binario classico può rappresentare solo un singolo valore binario, ad esempio 0 o 1, ovvero può trovarsi solo in uno di due stati possibili. Un qubit tuttavia può rappresentare uno 0, un 1 o qualsiasi proporzione di 0 e 1 nella sovrapposizione di entrambi gli stati, con una determinata probabilità che si tratti di uno 0 e una determinata probabilità che si tratti di un 1.

Fisicamente viene rappresentato con un sistema microscopico a due livelli come lo spin di una particella¹, la polarizzazione di un singolo fotone o due stati di un atomo ottenibili cambiando il livello energetico di un suo elettrone.

Se volessimo descriverlo matematicamente potremmo definirlo come un vettore unitario descritto in uno spazio vettoriale di Hilbert complesso bidimensionale (\mathbb{C}^2).

Per rappresentare gli elementi di uno spazio vettoriale complesso è conveniente utilizzare **la notazione di Dirac** (notazione standard della meccanica quantistica). Tale scelta è motivata dal fatto che quando si opera su un computer quantistico reale si utilizzano numerosi qubit, la cui rappresentazione sotto forma di vettore diventerebbe estremamente difficoltosa.

L'algebra di Dirac comprende due tipi di vettori: **bra** e il suo vettore duale **ket**.

¹Lo spin è una forma di momento angolare, avendo di tale entità fisica le dimensioni e, pur non esistendo una grandezza corrispondente in meccanica classica, per analogia richiama la rotazione della particella intorno al proprio asse (viene anche definito come momento angolare intrinseco).

Un ket rappresenta un vettore colonna e viene utilizzato solitamente per descrivere lo stato di un sistema:

$$|a\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

Mentre un bra rappresenta la coniugata trasposta del vettore colonna ket:

$$\langle a| = \begin{pmatrix} \alpha & \beta \end{pmatrix}$$

Il prodotto scalare tra i due vettori si indica con $\langle\alpha|\beta\rangle$ in modo che il prodotto formi un **braket**.

Definendo due vettori:

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

e associandoli rispettivamente agli stati $|0\rangle$ e $|1\rangle$, essi formano una base *ortonormale*, cioè una base *ortogonale* di vettori aventi *norma 1*, nota come **base computazionale standard**.

Possiamo inoltre dare una definizione degli stati attraverso la forma matriciale (vettori colonne), ottenendo la seguente rappresentazione:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

I due vettori appena introdotti corrispondono esattamente agli stati classici 0 e 1. A questo punto è bene specificare la principale differenza

con il bit classico: un qubit, oltre a potersi trovare in uno degli stati fondamentali, potrà trovarsi contemporaneamente anche in un'altra qualsiasi combinazione di entrambi gli stati base.

Se definiamo $|\psi\rangle$ la seguente combinazione lineare:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

dove α e β rappresentano numeri complessi tali che valga:

$$|\alpha|^2 + |\beta|^2 = 1$$

allora $|\psi\rangle$ è un possibile stato del qubit la cui notazione algebrica equivalente sarà:

$$|\psi\rangle = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

Il che equivale a dire che $|\psi\rangle$ si trova in una sovrapposizione di stati. Quando abbiamo a che fare con un bit classico possiamo sempre stabilire con assoluta certezza in quale dei due stati esso si trovi, nel caso di un qubit non possiamo determinare con altrettanta precisione il suo stato quantistico, ossia i valori esatti di α e β .

La meccanica quantistica ci dice che soltanto attraverso l'effettiva misurazione del sistema otterremo un valore discreto del qubit, più precisamente si dice che lo stato collasserà nello stato $|0\rangle$ con probabilità $|\alpha|^2$ o in $|1\rangle$ con probabilità $|\beta|^2$. Proprio per questa ragione, i due valori α e β prendono il nome di **ampiezze di probabilità** (amplitudes).

Una prima semplice sovrapposizione è definita dallo stato:

$$\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

il quale ci tornerà utile in seguito.

Dunque per ora possiamo immaginare che fino al momento della sua effettiva misurazione, un qubit avrà una probabilità del 50% di trovarsi nello stato $|0\rangle$ e un altro 50% di trovarsi in $|1\rangle$; come se lanciando una moneta essa continuasse a girare su sé stessa fino al momento in cui la guardiamo e ne osserviamo il valore.

1.1.2 Rappresentazione geometrica di un qubit

Per ottenere una visualizzazione geometrica utile per comprendere meglio i diversi stati in cui un qubit può trovarsi, utilizziamo una sfera di raggio unitario la cosiddetta **Sfera di Bloch**, introdotta dal fisico Felix Bloch [6]. Gli stati del qubit verranno collocati in punti precisi della superficie della sfera, associando quindi ad ogni stato un punto. Lo stato $|1\rangle$ verrà collocato nel polo sud, lo stato $|0\rangle$ nel polo nord. I punti che giacciono sull'equatore avranno una probabilità del 50% di essere nello stato $|0\rangle$ e 50% di essere nello stato $|1\rangle$ così le altre locazioni indicheranno gli altri stati di sovrapposizioni quantistiche di $|0\rangle$ e $|1\rangle$.

Come possiamo vedere nella figura 1.1, possiamo stabilire una corrispondenza biunivoca fra la rappresentazione generica dello stato di

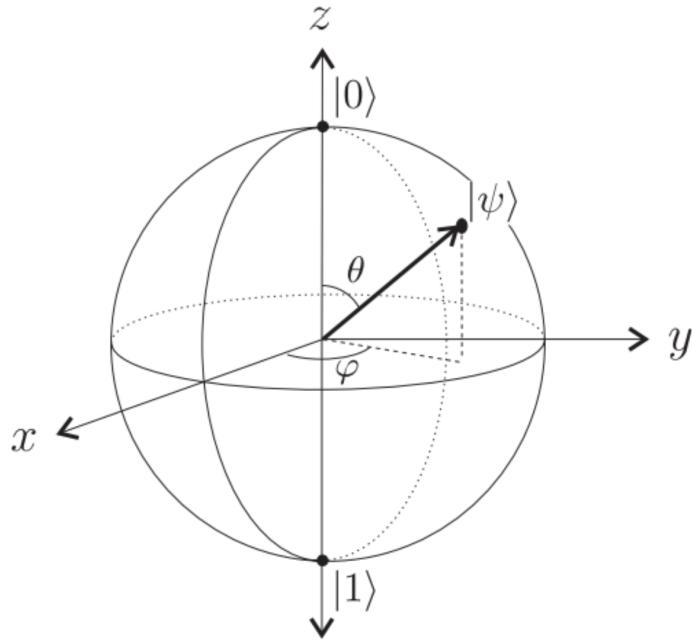


Figura 1.1: Rappresentazione di una Sfera di Bloch

un qubit:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

E la sua rappresentazione sulla sfera unitaria in \mathbb{R}^3 :

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle$$

Dove θ e ϕ sono le coordinate sferiche del punto. Si può quindi scrivere

$$|\psi\rangle = \alpha|0\rangle + e^{i\phi}\beta|1\rangle$$

Dato che il vettore di stato ha norma 1:

$$\sqrt{|\alpha|^2 + |\beta|^2} = 1$$

si usa l'identità trigonometrica:

$$\sqrt{\sin^2 x + \cos^2 x} = 1$$

Per descrivere α e β reali in termini della variabile θ :

$$\alpha = \cos\left(\frac{\theta}{2}\right), \beta = \sin\left(\frac{\theta}{2}\right)$$

da questo lo stato di ogni qubit si può descrivere usando le due variabili θ e ϕ :

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle$$

Interpretando θ e ϕ come coordinate sferiche, si può tracciare qualsiasi stato del qubit sulla superficie della sfera di Bloch. In figura 1.2 vengono visualizzati i seguenti vettori di stato del qubit:

- $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ con $\theta = 0$ e $\phi = 0$ cioè lo stato $|0\rangle$
- $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ con $\theta = 180$ e $\phi = 0$ cioè lo stato $|1\rangle$
- $\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} \end{bmatrix}$ con $\theta = \frac{\pi}{2}$ e $\phi = \frac{\pi}{2}$

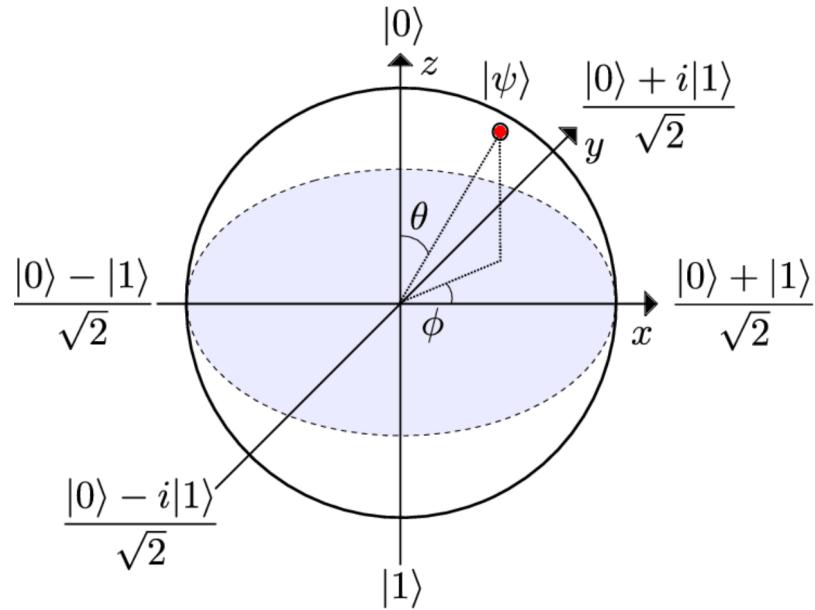


Figura 1.2: Visualizzazione dei qubit

- $\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$ con $\theta = \frac{\pi}{2}$ e $\phi = 0$ chiamato anche stato $|+\rangle$
- $\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{-i}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$ con $\theta = \frac{\pi}{2}$ e $\phi = \frac{3\pi}{2}$
- $\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$ con $\theta = \frac{3\pi}{2}$ e $\phi = 0$ chiamato anche stato $|-\rangle$

Dato che in input inizialmente i qubit hanno sempre stato $|0\rangle$, per poter operare sui qubit e ottenere degli stati differenti bisogna ruotare gli assi cardinali con le apposite *porte logiche quantistiche*.

1.2 Porte logiche quantistiche

Esattamente come nel modello di computazione classica utilizziamo delle porte logiche come l'*AND*, *OR* o il *NOT* per effettuare delle operazioni tra bit, nel modello quantistico avremo delle porte che si occuperanno di manipolare i qubit per ottenere un risultato. In particolare ogni gate quantistico deve rispettare due criteri fondamentali:

- **Reversibilità:** Un qubit a cui è stato applicato un cambiamento dello stato tramite l'utilizzo di una porta deve poter ritornare nello stato iniziale tramite l'applicazione della stessa porta all'output della prima.
- **Conservazione del vincolo di normalizzazione:** In questo modello, le porte logiche sono rappresentate da matrici unitarie. Una matrice quadrata U viene definita **unitaria** se vale $UU^* = I$, dove U^* è la matrice **trasposta** e I è la **matrice identità**.

Proprio come nel modello classico, abbiamo sia porte logiche che agiscono su un singolo qubit, che porte che agiscono su più qubit.

1.2.1 Porte Logiche a singolo qubit

Contrariamente a quanto accade per le porte classiche, in ambito quantistico le porte a singolo bit non si limitano al **NOT**. Infatti abbiamo in totale quattro porte: **porta X**, **porta Y**, **porta Z** e **porta di Hadamard**.

Le porte X, Y e Z prendono il nome di *Porte di Pauli* e corrispondono a delle rotazioni rispettivamente sull'asse x, y e z della sfera di Bloch.

Porta X

Analoga alla porta NOT classica, la porta X svolge la stessa operazione del NOT classico invertendo lo stato del qubit nel caso sia uno degli stati base. La differenza con la porta classica sta nel fatto che il NOT nel modello quantistico si dovrà occupare anche di gestire degli stati sovrapposti che sono caratterizzati dai coefficienti α e β del qubit. Immaginando di rappresentare in forma vettoriale il qubit, e definendo la matrice corrispondente al NOT quantistico come:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

è facilmente verificabile che applicando tale porta a un qubit nella forma $\alpha|0\rangle + \beta|1\rangle$ otterremo, seguendo la notazione vettoriale:

$$X \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}$$

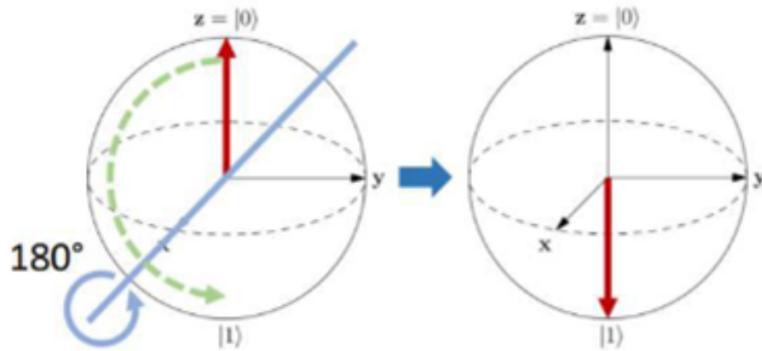


Figura 1.3: Visualizzazione dell'applicazione della Porta X

Porta Y

La porta Y è rappresentata dalla seguente matrice:

$$Y = \begin{bmatrix} 0 & -i \\ -i & 0 \end{bmatrix}$$

che mappa la componente $|0\rangle$ in $i|1\rangle$ e la componente $|1\rangle$ in $-i|0\rangle$.

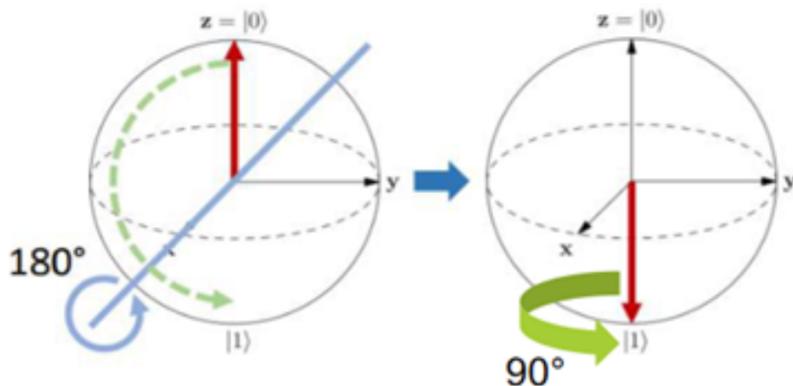


Figura 1.4: Visualizzazione dell'applicazione della Porta Y

Porta Z

La porta Z è rappresentata dalla seguente matrice:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

che cambia il segno esclusivamente alla componente nello stato $|1\rangle$.

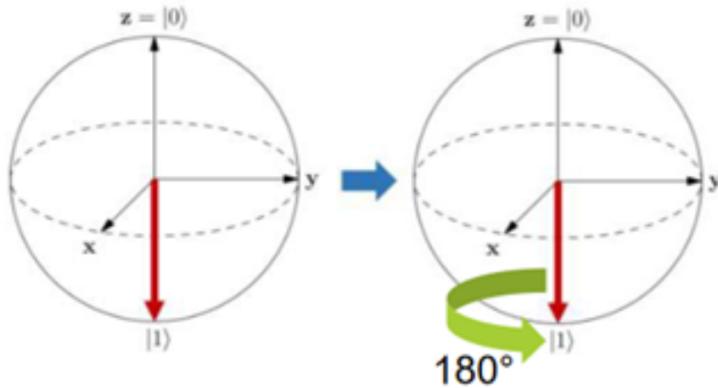


Figura 1.5: Visualizzazione dell'applicazione della Porta Z

Porta di Hadamard

La porta di Hadamard è rappresentata dalla seguente matrice:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

che si occupa di trasformare uno stato base in una sovrapposizione di tale stato che si trovi con il 50% di probabilità in uno dei due stati

fondamentali.

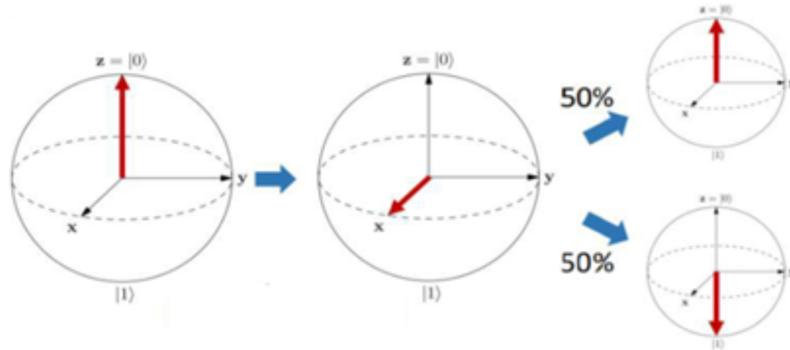


Figura 1.6: Visualizzazione dell'applicazione della Porta di Hadamard

1.2.2 Porte logiche a qubit multipli

Proprio come nel modello di computazione classico, anche in questo modello siamo interessati ad avere un insieme di gate capaci di realizzare tutte le operazioni del modello classico. Nel caso del modello quantistico, per ottenere tale risultato, si affiancano le porte a singolo qubit con un operatore chiamato **CNOT** o **NOT Controllato**.

Il CNOT, che corrisponde allo XOR del modello classico, è dotato di due qubit in ingresso, rispettivamente definiti *controllo* e *bersaglio* (o *target*). Dunque nel caso il qubit controllo si trovi nello stato zero allora il target viene lasciato inalterato, al contrario, se il qubit controllo è nello stato uno, allora il target viene invertito. Tale trasformazione può essere scritta come:

$$|A, B\rangle \mapsto |A, B \oplus A\rangle$$

Il gate è rappresentato dalla seguente matrice:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Dove effettivamente possiamo notare come gli ultimi due stati vengano rispettivamente invertiti e prendendo in esempio un sistema composto da due qubit, il CNOT eseguirà operazioni mostrate in tabella 1.1:

Input		Output	
Controllo	Target	Controllo	Target
$ 0\rangle$	$ 0\rangle$	$ 0\rangle$	$ 0\rangle$
$ 0\rangle$	$ 1\rangle$	$ 0\rangle$	$ 1\rangle$
$ 1\rangle$	$ 0\rangle$	$ 1\rangle$	$ 1\rangle$
$ 1\rangle$	$ 1\rangle$	$ 1\rangle$	$ 0\rangle$

Tabella 1.1: Insieme delle possibili operazioni del gate CNOT

Una delle proprietà fondamentali delle porte quantistiche, in particolare del CNOT e di tutte le porte viste a singolo qubit, è quella di essere invertibili, infatti a differenza delle porte classiche XOR e NAND generalmente irreversibili, permettono di ottenere l'input avendo a disposizione il valore di output. Combinando opportunamente CNOT e porte a singolo qubit, otteniamo l'insieme dei gate necessari per definire un insieme universale, capace dunque di inglobare le operazioni sufficienti alla rappresentazione di tutte le porte logiche quantistiche e quindi l'universalità delle operazioni quantistiche.

1.3 Misurazione di un sistema di qubit

Fin'ora abbiamo parlato di come vengono effettuate le operazioni sui qubit, tralasciando il modo in cui alla fine della computazione le informazioni sono raccolte. Immaginiamo che una particella sia dotata di un numero finito possibile di stati base e che tale particella li possieda tutti contemporaneamente fin quando non avviene l'evento della misurazione che farà ottenere uno degli stati base con probabilità uguale al quadrato del coefficiente associato a tale stato.

Nel nostro caso dato un qubit $|\phi\rangle$ generico, il risultato di questa misurazione ci restituisce 0 con probabilità $|\alpha|^2$ e 1 con probabilità $|\beta|^2$.

Il problema in questo caso è che la misurazione disturba il qubit, lasciandolo nello stato $|0\rangle$ se il risultato della misurazione è 0, e nello stato $|1\rangle$ se il risultato della misurazione è 1.

In un circuito quantistico, a differenza della controparte classica, dopo la misurazione di un qubit esso viene scartato in quanto il suo stato essendo collassato, non è più valido.

Altra differenza con la controparte classica è la predicitività, ovvero che se l'esperimento effettuato venisse ripetuto rispettando le condizioni, ci aspettiamo esattamente lo stesso risultato cosa che in ambito quantistico risulta incerta perché coadiuvata dal coefficiente associato allo stato.

1.4 Registri quantistici

Fin'ora abbiamo visto come rappresentare un solo qubit, per rappresentare un sistema a più qubit si utilizza un **registro quantistico**, che di fatto indica in che modo i qubit sono collegati tra loro. Per rappresentare questi registri si usa il **prodotto tensore** \otimes , un operatore che combina spazi vettoriali di una certa dimensione per generarne dei più grandi, infatti: $\otimes : \mathbb{C}^k \times \mathbb{C}^m \rightarrow \mathbb{C}^{km}$ quindi lo spazio totale di un registro quantistico sarà $\mathbb{C}^{2 \cdot \dots \cdot 2} = \mathbb{C}^{2^n}$.

Formalmente si definisce un registro quantistico, secondo il quarto postulato della meccanica quantistica², come:

$$|i_1\rangle \otimes |i_2\rangle \otimes \dots |i_n\rangle$$

dove $i = 0, 1$ e n è il numero di qubit e per convenienze possiamo rappresentare questo vettore semplicemente come $|i_1 i_2 \dots i_n\rangle$. Consideriamo un semplice sistema a due qubit, dove il primo è $|\psi\rangle = \alpha_0|0\rangle + \beta_0|1\rangle$ mentre il secondo $|\theta\rangle = \alpha_1|0\rangle + \beta_1|1\rangle$. Lo stato totale sarà una sovrapposizione dalla forma:

$$|\psi\rangle \otimes |\phi\rangle = \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle$$

Analogamente al singolo qubit dove il risultato della misurazione ci restituisce 0 con probabilità $|\alpha|^2$ e 1 con probabilità $|\beta|^2$. In un sistema

²Lo spazio degli stati di un sistema fisico composto è il prodotto tensore degli spazi degli stati dei sistemi fisici componenti. Se il sistema è composto da n sottosistemi e il componente i -esimo si trova nello stato $|\phi_i\rangle$ allora lo stato del sistema totale è $|\phi_1\rangle \otimes |\phi_2\rangle \otimes \dots |\phi_n\rangle$

di n qubit possiamo anche misurare solo un sottoinsieme degli n qubit. Ad esempio lo stato risulterà in $|00\rangle$ con probabilità $|\alpha_{01}|^2$, in $|01\rangle$ con probabilità $|\alpha_0\beta_1|^2$ e così via. Inoltre se volessimo sapere la probabilità di ottenere 0 al primo bit basta sommare le probabilità di $|00\rangle$ e $|01\rangle$ cioè $|\alpha_{01}|^2 + |\alpha_0\beta_1|^2$.

1.5 Entanglement

Dopo aver visto i registri quantistici una ulteriore proprietà legata ai possibili stati in cui può trovarsi il sistema è l'*entanglement*, proprietà che non possiamo ritrovare in nessun oggetto della fisica classica. Questi stati chiamati entangled rappresentano quelle possibili configurazioni di n qubit componenti che non hanno un proprio stato ben definito ma solamente la loro combinazione ne rappresenta uno concreto. Più semplicemente uno stato entangled non può essere descritto come prodotto tensore degli stati dei singoli componenti. Gli stati entangled si comportano come se fossero strettamente connessi l'uno all'altro indipendentemente dalla distanza fisica che li separa, in modo che una misurazione o un'operazione di uno dei due stati di una coppia entangled fornisce simultaneamente informazioni sulla coppia. Un esempio per spiegare questa proprietà è dato dallo stato $|00\rangle + |11\rangle$ che non può essere fattorizzato nel prodotto tensore di due qubit indipendenti, in quanto non esistono dei coefficienti $\alpha_1\alpha_2\beta_1\beta_2$ tali per cui valga:

$$|00\rangle + |11\rangle = (\alpha_1|0\rangle + \beta_1|1\rangle) \otimes (\alpha_2|0\rangle + \beta_2|1\rangle)$$

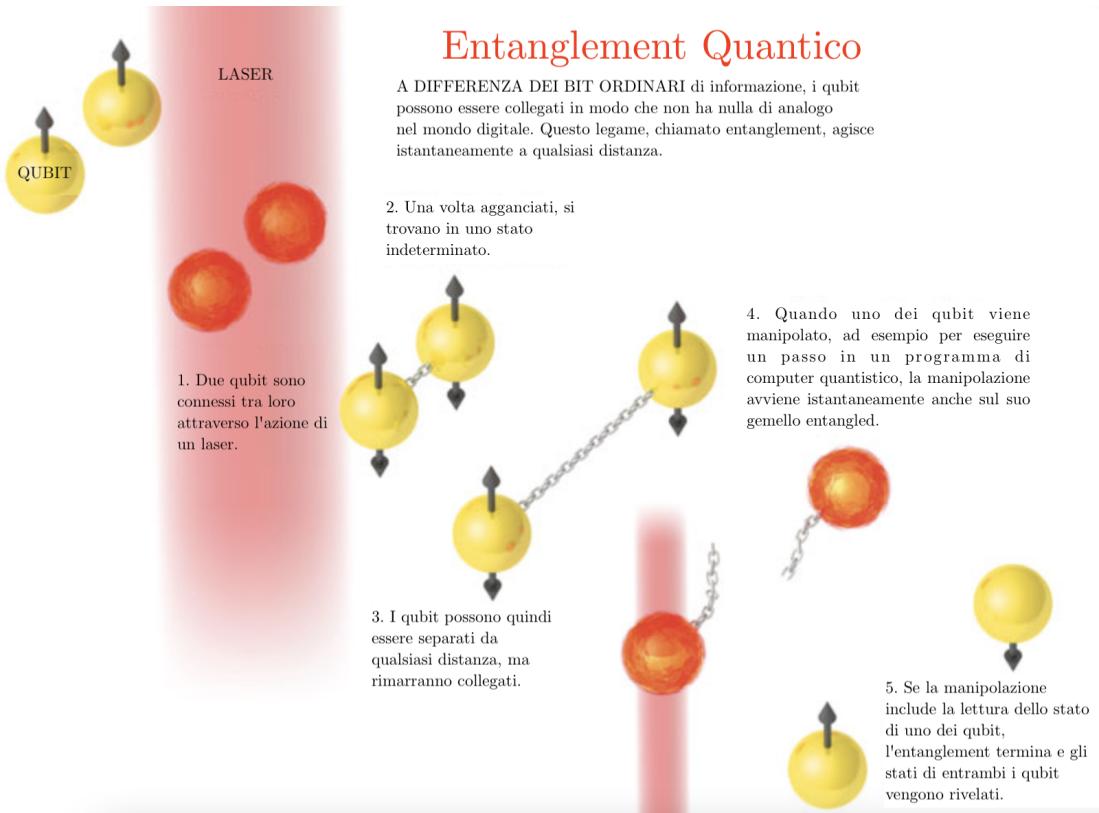


Figura 1.7: Visualizzazione degli effetti dell'entanglement

L'entanglement è alla base della risoluzione di alcuni di quei problemi informatici non riproducibili tramite informatica classica, grazie alla sua intrinseca proprietà, non esistente nella fisica classica, che da possibilità di ottenere un aumento esponenziale nella capacità di calcolo.

1.6 Realizzazione di un computer quantistico

Definiamo un computer quantistico come un calcolatore che segue il modello di computazione quantistico, sfruttando i dettami della fisica quantistica per eseguire dei calcoli che in alcuni casi risultano essere impossibili da realizzare in un calcolatore classico.

1.6.1 Classi di complessità

Prima di proseguire con l'introduzione delle componenti di un computer quantistico è bene tenere a mente le classi di complessità che non sono altro un insieme di problemi di una determinata complessità. I problemi vengono eseguiti su una macchina di Turing per individuarne la particolare classe di complessità in cui rientrano. Le due classi più importanti sono **P** e **NP**.

- La classe **P** è l'insieme dei problemi di decisione che possono essere risolti da una macchina di Turing deterministica in tempo polinomiale.
- La classe **NP** è l'insieme dei problemi di decisione che possono essere risolti da una macchina di Turing non deterministica in tempo polinomiale. Inoltre nella classe NP è composta anche dalle classi NP-Complete e NP-Hard.

1.6. REALIZZAZIONE DI UN COMPUTER QUANTISTICO Cap.1

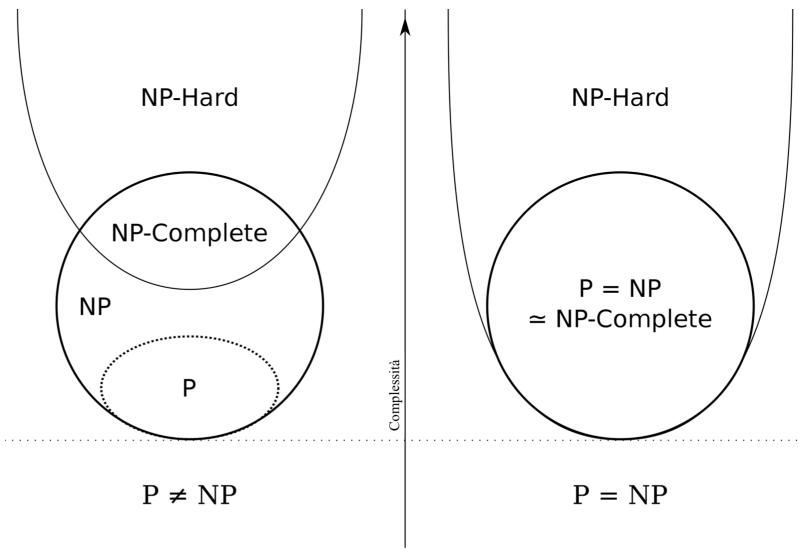


Figura 1.8: Le classi di complessità per $P = NP$ e $P \neq NP$

- La classe **NP-Complete** è l'insieme dei problemi più difficili nella classe NP nel senso che, se si trovasse un algoritmo in grado di risolvere "velocemente" (in tempo polinomiale) un qualsiasi problema NP-completo, allora si potrebbe usarlo per risolvere "velocemente" ogni problema in NP.
- In teoria della complessità, i problemi NP-difficili o NP-ardui sono una classe di problemi che può essere definita informalmente come la classe dei problemi almeno difficili come i più difficili problemi delle classi di complessità P e NP.

Gli informatici *Bernstein* e *Vazirani* nel 1997 definirono una nuova classe di complessità chiamata **BQP** [5] (*Bounded-error Quantum Polynomial time*) che è la classe di complessità dei problemi decisionali

1.6. REALIZZAZIONE DI UN COMPUTER QUANTISTICO Cap.1

che possono essere risolti con un errore bilaterale su una macchina di Turing quantistica in tempo polinomiale. In breve, tutti i problemi decisionali che i computer quantistici possono risolvere in maniera veloce. Inoltre, è stato dimostrato che $P \in BPQ$ e quindi è semplice dedurre che i computer quantistici possono risolvere tutti i problemi che i computer classici possono risolvere.

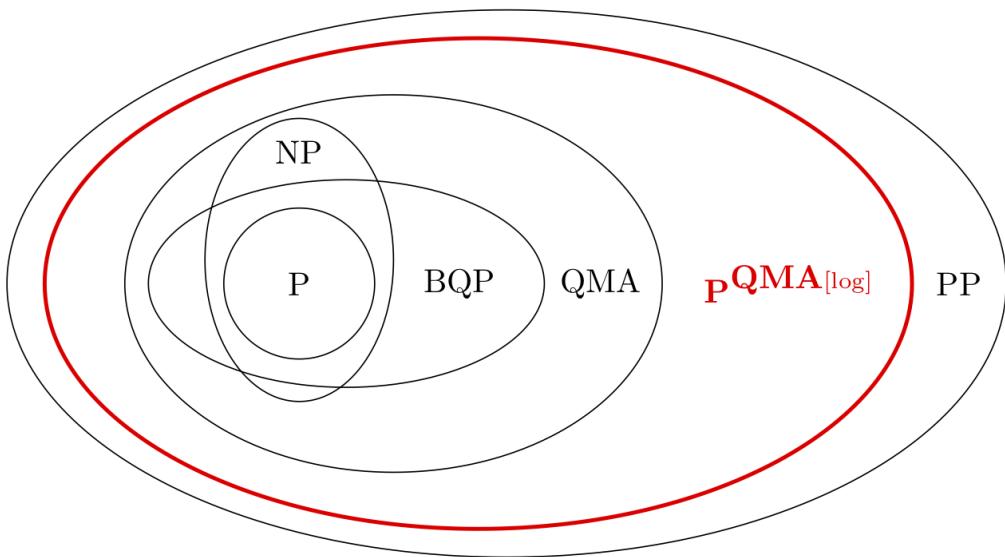


Figura 1.9: La classe di complessità BQP rispetto a quelle classiche

1.6.2 Macchina di Turing Quantistica

La **Macchina di Turing Quantistica (QTM)** è stata descritta per la prima volta da Deutsch [8]. L'idea di base è abbastanza semplice, un QTM è più o meno una Macchina di Turing probabilistica (PTM) con ampiezze di transizione complesse anzichè probabilità reali. A

1.6. REALIZZAZIONE DI UN COMPUTER QUANTISTICO Cap.1

sua volta una Macchina di Turing Probabilistica (PTM) è identica a una normale Macchina di Turing tranne per il fatto che ad ogni configurazione della macchina ($q_i S_j$) c'è un insieme finito di regole di transizione (ognuna con una probabilità associata) che si applicano e che una scelta casuale determina quale regola applicare. Fissiamo una soglia di probabilità maggiore delle quote pari (diciamo, 75%) e diciamo che una PTM specifica calcola $f(x)$ sull'input x se e solo se si ferma con $f(x)$ come output con probabilità maggiore del 75%.

1.6.3 Condizioni per la realizzazione

Per la realizzazione di un computer quantistico nel 2000 sono stati stilati dal fisico teorico Di Vincenzo i **criteri di DiVincenzo** [1] che consistono in sette condizioni necessarie per costruire un computer seguendo il modello quantistico, le prime cinque sono necessarie per il calcolo quantistico e sono:

1. Il sistema deve essere *scalabile*, con qubit ben caratterizzati;
2. Deve essere possibile preparare uno *stato iniziale generico*, ad esempio $|0000\rangle$. Diversamente, sarà impossibile introdurre dati nel computer;
3. I *tempi di de-coerenza* devono essere abbastanza lunghi, per poter realizzare un numero sufficiente di operazioni sfruttando la correlazione quantistica;

1.6. REALIZZAZIONE DI UN COMPUTER QUANTISTICO Cap.1

4. Occorre un *insieme universale di porte quantistiche*, ovverosia si deve poter costruire una varietà sufficiente di porte quantistiche per permettere qualsiasi operazione logica;
5. Si deve disporre di un modo per misurare lo stato dei qubit, senza il quale sarebbe impossibile estrarre l'informazione processata dal computer;

Le restanti due servono per la comunicazione quantistica e sono:

6. Deve esserci un sistema per *convertire i qubit immagazzinati* in qubit messaggeri, ovvero deve esistere un sistema per trasmettere informazioni;
7. La capacità di *trasmettere fedelmente* qubit tra le varie locazioni specificate, per la medesima ragione del punto precedente.

Negli ultimi anni si stanno sperimentando vari modi per la realizzazione dei computer quantistici come:

- Superconduttori
- Ioni intrappolati di un atomo o di una molecola
- Risonanza magnetica nucleare
- Quantum annealing o ricottura quantistica
- Silicium quantum dot

1.6. REALIZZAZIONE DI UN COMPUTER QUANTISTICO Cap.1

Atomi		Elettroni				Fotoni	
Tecnologie Qubit	Ioni intrappolati	Atomi freddi	Superconduttori	Silicio (+quantum dot)	Impurità del diamante (Centro NV)	Fermione di Majorana (topologico)	
Dominio Applicativo	metrologia, informatica, comunicazione (ripetitore, accoppiamento dei fotoni)	metrologia, informatica, comunicazione (accoppiamento dei fotoni)	metrologia, informatica	metrologia, informatica	metrologia, comunicazione, informatica	informatica	metrologia, comunicazione, informatica
Natura dei qubit	ioni intrappolati elettromagneticamente	atomi intrappolati da pinzette laser	anello/circuito superconduttore	elettroni intrappolati in un semiconduttore	elettroni da una cavità di diamante vicino a un atomo di azoto atomo di azoto	quasi-particelle, coppie di anyon, in nanofili superconduttori	fotoni circolanti in guide d'onda
Stati quantistici di qubit	livello energetico della trappola intrappolata	livello energetico dello ione intrappolato	3 tipi: qubit di fase, qubit di carica carico alias transmon (livello attuale) e flusso (direzione della corrente)	spin dell'elettrone	livello energetico degli elettroni dell'N	senso di anyon	1-proprietà del fotone (polarità o altro)

Figura 1.10: Approcci per la realizzazione di qubit

Tra tutti i più utilizzati dai produttori come IBM, Google, Rigetti sono:

Ioni intrappolati di un atomo In questa tipologia di approccio, vengono costruite delle cosiddette *ion trap* o trappole di ioni. Il loro scopo è trattenere all'interno degli ioni, come ad esempio un atomo di calcio che tramite l'utilizzo di un raggio laser è stato privato di uno dei due elettroni più esterni. Un chip costruito con questo approccio dei qubit è molto simile ai chip di cui sono composte le CPU classiche: si tratta infatti di un chip composto di oro su cui sono presenti gli ioni di calcio e al di sopra di essi, circa ad una distanza pari al diametro di un capelli, è presente un sottile strato di oro che alternando appositamente il suo campo

1.6. REALIZZAZIONE DI UN COMPUTER QUANTISTICO Cap.1

magnetico, riesce a tenere gli ioni nella loro posizione ed evitare che fuoriescano (da qui si capisce il termine ion trap).

Come è possibile trattare questi ioni come qubit? Innanzitutto, gli ioni naturalmente seguono i principi della meccanica quantistica, ed è possibile ottenere i due stati base di un qubit tramite l'utilizzo dello spin, presente in ogni atomo, che rappresenta una intrinseca forma di momento angolare di una particella elementare. Possiamo immaginare lo spin dell'elettrone del nostro atomo di calcio come un magnete: il nord può puntare verso l'alto, ottenendo un qubit in uno stato $|1\rangle$ che in questo caso corrisponde a $|\uparrow\rangle$, oppure verso il basso, ottenendo uno stato $|0\rangle$ corrispondente a $|\downarrow\rangle$. Per passare fra lo stato $|\uparrow\rangle$ e $|\downarrow\rangle$, basta utilizzare delle microonde che hanno l'effetto di ruotare lo spin dell'elettrone. È possibile quindi ruotare e fermarsi in un qualsiasi stato compreso fra i due spin, ottenendo quella che abbiamo in precedenza chiamato superposizione.

Superconduttori L'approccio che utilizza i superconduttori per costruire i qubit viene utilizzato ad esempio nei computer quantistici di Google o IBM. Proprio con questo metodo di costruzione, nel 2016 Google ha annunciato di aver raggiunto la Quantum Supremacy [15], cioè è stato risolto un problema che nessun calcolatore classico potrebbe risolvere in un ragionevole lasso di tempo, utilizzando un computer quantistico a 56 qubit, prodotti con questo approccio. Un superconduttore è un particolare materiale che raffreddato ad una temperatura molto vicina allo zero

1.6. REALIZZAZIONE DI UN COMPUTER QUANTISTICO Cap.1

assoluto (0K oppure -273.15C) annulla la sua resistività elettrica completamente e grazie a queste particolari caratteristiche risulta adatto per essere utilizzato come qubit.

Capitolo 2

Blockchain

Alla base della più moderna forma di commercio, incentrata sulle criptovalute, troviamo una delle forme di commercio più antica mai messa agli atti. Infatti, il viaggio all'interno della Blockchain e le criptovalute ha inizio nel 1400 d.C. in una piccola isola della Micronesia, l'isola di Yap.

2.1 La storia della Blockchain

L'evoluzione della blockchain può essere riassunta nei seguenti passaggi principali mostrati nella tabella temporale 2.1.

Nel 1982, il crittografo David Chaum ha proposto per la prima volta un protocollo simile alla blockchain nella sua tesi del 1982 *"Computer e sistemi creati, mantenuti e resi attendibili da gruppi di individui reciprocamente sospettosi"* [13], da qui in poi li definiamo **Sistemi di Chaum**. Siamo così di fronte alla prima idea di tecnologia blockchain.

1982	• Sistemi di Chaum
1991	• Timestamp
1992	• Alberi di Merkle
2005	• Bitgold
2008	• Bitcoin

Tabella 2.1: Evoluzione della Blockchain

2.1.1 Introduzione ai Sistemi di Chaum

Probabilmente, molti degli elementi delle blockchain odierne sono contenuti nel sistema di caveau di David Chaum del 1979, descritto nella sua tesi di laurea del 1982 a Berkeley. Chaum descrive la progettazione di un sistema informatico distribuito che può essere creato, mantenuto e reso attendibile da gruppi di individui reciprocamente sospettosi.

Si tratta di un sistema contenente record in grado di maneggiare la sicurezza e la privacy dei singoli individui tramite sicurezza fisica. Gli elementi costitutivi di questo sistema includono "caveau" fisici (sicuri), primitive crittografiche (crittografia simmetrica e asimmetrica, funzioni hash crittografiche e firme digitali), e una nuova primitiva introdotta da Chaum.

2.1.2 Timestamp

Un ulteriore lavoro su una catena di blocchi protetta da crittografia è stato descritto nel 1991 da Stuart Haber e W. Scott Stornetta [9]. Essi volevano implementare un sistema in cui i timestamp dei documenti

non potessero essere manomessi, oggi considerata la prima applicazione della blockchain.

L'utilizzo del timestamp richiede il superamento di due problematiche:

- I dati DEVONO essere contrassegnati con l'ora esatta
- Il calendario DEVE essere immutabile

I due, idearono una soluzione a queste problematiche, definita "naive", la quale consisteva nell'utilizzo di una *cassetta di sicurezza digitale*. Ogni volta che un cliente ha un documento da marcare temporalmente, lo trasmette a un servizio di marcatura temporale (TSS). Il servizio registra la data e l'ora di ricezione del documento e ne conserva una copia. Se l'integrità del documento del cliente viene messa in discussione, viene confrontata con la copia conservata dal TSS. Se le due copie sono identiche, è la prova che il documento non è stato manomesso dopo la data riportata nei registri del TSS.

Questa procedura soddisfa di fatto il requisito centrale per la marcatura temporale di un documento digitale. Tuttavia, questo approccio solleva diverse preoccupazioni:

Privacy Questo metodo compromette la privacy del documento in due modi: una terza parte potrebbe origliare mentre il documento viene trasmesso e, dopo la trasmissione, il documento è a disposizione del TSS stesso. Il cliente deve quindi preoccuparsi

non solo della sicurezza dei documenti che tiene sotto il suo direttorio controllo, ma anche della sicurezza dei suoi documenti presso il TSS.

Larghezza di banda e archiviazione Sia il tempo necessario per inviare un documento per la marcatura temporale che la quantità di memoria richiesta al TSS dipendono dalla lunghezza del documento da marcare. Pertanto, il tempo e la spesa necessari per la marcatura temporale di un documento di grandi dimensioni potrebbero essere proibitivi.

Incompetenza La copia del documento inviata al TSS potrebbe essere danneggiata durante la trasmissione al TSS, potrebbe essere marcata in modo errato quando arriva al TSS, oppure potrebbe essere danneggiata o persa del tutto in qualsiasi momento mentre è conservata presso il TSS. Ognuno di questi eventi invaliderebbe la richiesta di marcatura temporale del cliente.

Fiducia Il problema fondamentale rimane: nulla in questo schema impedisce al TSS di accordarsi con un cliente per affermare di aver apposto la data e l'ora su un documento diverso da quello reale.

Per risolvere queste criticità, Haber e Stornetta, formularono una soluzione: proposero di sottoporre il documento ad un algoritmo di hashing crittografico, ottenendo così un ID univoco ed immutabile del documento. Semplicemente, anzichè trasmettere al TSS il documento

x , viene trasmesso il suo valore $hash(x) = y$. Per quanto riguarda l'autenticazione, il timestamp di y sarà valido quanto il timestamp di x . Inoltre, questa soluzione riduce drasticamente il problema della larghezza di banda e dell'archiviazione e in più risolve anche il problema della privacy in quanto non viene trasmesso il documento in toto. A seconda degli obiettivi di progettazione, potrebbe essere una singola funzione di hash comune o una per ogni singola utenza.

A ciò si abbinava la firma digitale, utilizzata per identificare in modo univoco il firmatario. Controllando la firma, al client viene garantito che il TSS abbia elaborato la richiesta, che l'hash sia stato ricevuto correttamente e che l'ora inclusa sia corretta. Questo risolve il problema dell'incompetenza da parte del TSS.

Nella figura 2.1 è riportata una sequenza d'esempio in cui abbiamo una catena di blocchi connessi da un valore hash.



Figura 2.1: Sequenza di blocchi

In questa sequenza di blocchi, ogni documento digitale è modificato dai client in diversi istanti di tempo e la catena mantiene un elenco di valori di timestamp relativi agli eventi accaduti sequenzialmente. I valori di timestamp non sono modificabili e in caso di controversie ogni modifica apportata al documento può essere consultata.

2.1.3 Alberi di Merkle

Dave Bayer, contribuì ad integrare la struttura per la marcatura temporale di Haber e Stornetta, con la realizzazione dei Merkle Tree (Alberi di Merkle) [3], offrendo l'opportunità di raccogliere più documenti in un singolo blocco (Figura 2.2). Tali alberi ricevono il nome da Ralph Merkle e in essi i nodi foglia sono contrassegnati da un blocco dati, mentre i nodi non-foglia dall'hash crittografico delle etichette dei loro nodi figlio. Detti anche Alberi di hash, mostrano una versione più generica di liste e catene hash e consentono una verifica sicura ed efficace del contenuto di grandi strutture dati.

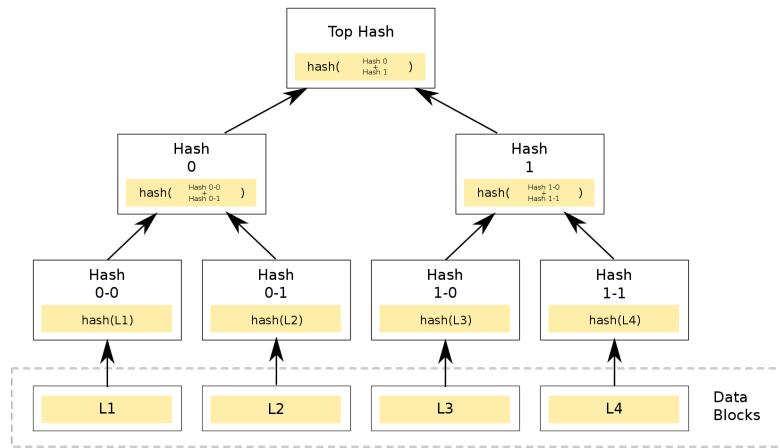


Figura 2.2: Esempio di albero di Merkle

Nella figura 2.2 possiamo vedere come i valori hash dei blocchi sono definiti "foglie", mentre i valori hash dei loro figli sono detti "nodi". Gli alberi di Merkle vengono utilizzati per rilevare incongruenze tra le repliche e per ridurre al minimo la quantità di dati.

2.1.4 Bit gold

Nel 2005, si ha avuto il primo tentativo di moneta decentralizzata grazie all'informatico Nick Szabo, il quale ha proposto una nuova valuta basata sulla blockchain: **Bit gold** [14]. Moneta che però non ha riscosso molto successo, ma nonostante ciò il 2005 rappresenta un anno cruciale nel contesto blockchain.

La proposta dell'informatico si basa sul calcolo di una stringa di bit a partire da una stringa di bit di sfida, utilizzando funzioni chiamate in vario modo "client puzzle function", "proof of work function" o "secure benchmark function". La stringa di bit risultante è la proof of work.

Ecco le fasi principali del sistema bit gold che Szabo ha definito:

1. Viene creata una stringa pubblica di bit, la "stringa di sfida" (vedi passo 5).
2. Alice sul suo computer genera la stringa di proof of work dai bit di sfida utilizzando una funzione di benchmark.
3. La proof of work viene registrata in modo sicuro con un timestamp. Questo dovrebbe funzionare in modo distribuito, con diversi servizi di timestamp in modo che non sia necessario affidarsi a un particolare servizio di timestamp.
4. Alice aggiunge la stringa di sfida e la stringa di proof of work con timestamp a un registro di proprietà distribuito per il bit gold. Anche in questo caso, non si fa affidamento su un singolo server per il corretto funzionamento del registro.

5. L'ultima stringa creata di bit gold fornisce i bit di sfida per la stringa creata successivamente.
6. Per verificare che Alice sia la proprietaria di una particolare stringa di bit gold, Bob controlla la catena di titoli non falsificabile nel registro dei titoli di bit gold.
7. Per verificare il valore di una stringa di bit gold, Bob controlla e verifica i bit di sfida, la stringa di proof of work e il timestamp.

Si noti che il controllo di Alice sul suo bit gold non dipende dal suo solo possesso dei bit, ma piuttosto dalla sua posizione di leader nella catena di titoli non falsificabile (catena di firme digitali) nel registro dei titoli.

Tutto questo può essere automatizzato da un software. I limiti principali alla sicurezza dello schema sono la capacità di distribuire la fiducia nelle fasi (3) e (4) e il problema dell'architettura della macchina, che verrà discusso di seguito.

Hal Finney ha implementato una variante di bit gold chiamata **RPOW (Reusable Proofs of Work)**. Si basa sulla pubblicazione del codice informatico della "zecca", che viene eseguito su un computer remoto a prova di manomissione. L'acquirente di bit gold può quindi utilizzare l'attestazione remota, che Finney chiama tecnica del server trasparente, per verificare che un determinato numero di cicli sia stato effettivamente eseguito.

Il problema principale di tutti questi schemi è che gli schemi di proof of work dipendono dall'architettura del computer, non solo da

una matematica astratta basata su un "ciclo di calcolo" astratto. (Pertanto, potrebbe essere possibile essere un produttore a bassissimo costo (di diversi ordini di grandezza) e inondare il mercato di bit gold. Tuttavia, dal momento che il bit gold è marcato a tempo, il tempo creato e la difficoltà matematica del lavoro possono essere dimostrati automaticamente. Da ciò si può solitamente dedurre il costo di produzione in quel periodo.

A differenza degli atomi d'oro fungibili, ma come nel caso degli oggetti da collezione, una grande disponibilità in un determinato periodo di tempo farà scendere il valore di questi particolari oggetti. Da questo punto di vista, il "bit gold" si comporta più come gli oggetti da collezione che come l'oro. Tuttavia, la corrispondenza tra questo mercato ex post e l'asta che determina il valore iniziale potrebbe creare un profitto molto consistente per il "minatore di bit gold" che inventa e distribuisce un'architettura informatica ottimizzata.

Pertanto, il bit gold non sarà fungibile in base a una semplice funzione, ad esempio, della lunghezza della stringa. Per creare unità fungibili, i commercianti dovranno invece combinare unità di valore diverso.

2.1.5 Bitcoin

Bitcoin nasce ufficialmente agli inizi del 2009 con la creazione del "block genesi", ma se ne inizia a parlare nel 2008 a seguito della pubblicazione di un paper scientifico intitolato "*Bitcoin: A Peer-to-Peer Electronic Cash System*" [11].

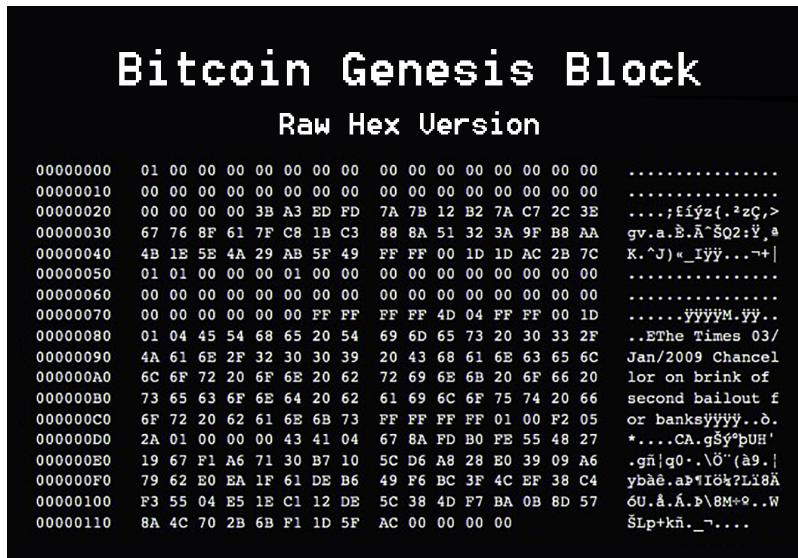


Figura 2.3: Messaggio di Satoshi Nakamoto incorporato nella coinbase del primo blocco

Il libro bianco o ”whitepaper” di Bitcoin fu pubblicato in un articolo scientifico tramite Cryptography Mailing List nel mese di ottobre del 2018. Essendo pubblicato in modo anonimo sotto lo pseudonimo Satoshi Nakamoto genera ancora più mistero e confusione. Così tanto che ancora oggi si cerca un vero nome dietro quel soprannome.

Prima di iniziare è bene fare una precisazione: bitcoin con la b minuscola è la moneta digitale, Bitcoin con la b maiuscola è il protocollo che la governa.

L’obiettivo di Satoshi era quello di creare un sistema di pagamento tramite una versione puramente peer-to-peer di denaro elettronico che permetterebbe di effettuare pagamenti online da un’entità ad un’altra

senza passare tramite un'istituzione finanziaria centrale. I nodi peer-to-peer, che costituiscono la rete, non formano gerarchie client-server ma agiscono al contempo sia da client che da server.

Le firme digitali offrono una soluzione parziale al problema, ma i benefici principali sono persi se una terza persona di fiducia è ancora richiesta per prevenire la doppia spesa. Ovvero, quando un utente fa una transazione ci deve essere la garanzia che i soldi appena spesi non possano essere utilizzati una seconda volta per compierne un'altra, problema illustrato in Figura 2.4.

La moneta fisica risolve alla radice questo problema non potendo esistere in due luoghi contemporaneamente. In merito ai pagamenti digitali, in un sistema di fiducia centralizzato il problema è gestito da una terza parte che fa controlli su ogni operazione effettuata dagli utenti.

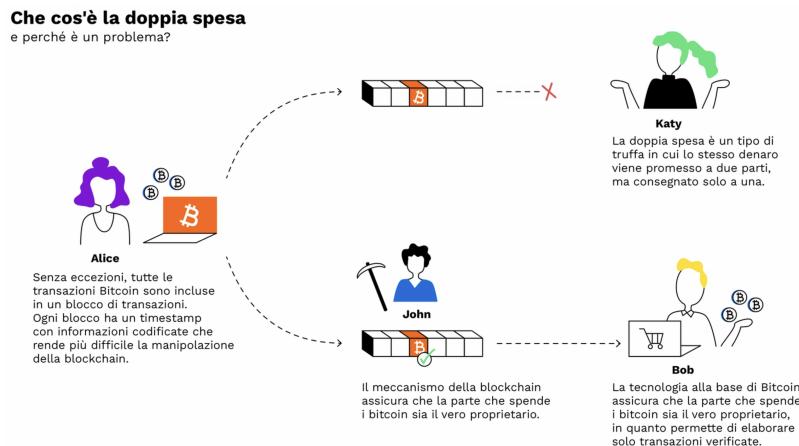


Figura 2.4: Problema del double spending

Satoshi propone una soluzione al problema della doppia spesa me-

diante l'utilizzo di una rete peer-to-peer, includendo elementi di crittografia.

La legge di mercato della domanda e offerta determinano il valore economico assunto dal bitcoin. Bitcoin è quotato su siti appositi chiamati Exchange. Tali siti permettono di scambiare Bitcoin con Euro, Dollaro Americano o altre monete emesse dai governi, dette anche **monete fiat**¹. Il primo Exchange è andato online nel marzo del 2010 e quotava bitcoin a soli 0,003\$. Il 22 maggio 2010 vengono acquistate due pizze in Florida per 10.000,00 bitcoin. Meno di un anno dopo la criptomoneta raggiunge il valore di 1,00\$. Nel 2013 la valutazione subisce alti e bassi arrivando a toccare un massimo di 1200,00\$. Con l'apertura di nuovi exchange e grazie alla speculazioni da parte di un numero sempre maggiore di utenti il prezzo sale fino a 10.000,00\$ a novembre 2017. Oggi (fine luglio 2022) ha un valore di 20.893,00\$ circa. L'andamento del bitcoin è illustrato in Figura 2.5.

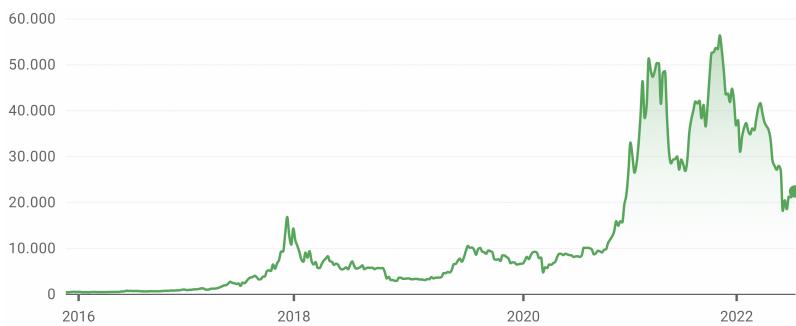


Figura 2.5: Andamento del bitcoin

¹Moneta legale (o moneta a corso legale o, ancora, moneta fiduciaria)

2.2 Cos'è la blockchain

La blockchain è una sottofamiglia di tecnologie in cui il registro è strutturato come una catena di blocchi contenenti le transazioni e la cui validazione è affidata a un meccanismo di consenso, distribuito su tutti i nodi della rete nel caso delle *blockchain permissionless o pubbliche* o su tutti i nodi i nodi che sono autorizzati a partecipare al processo di validazione delle transazioni da includere nel registro nel caso delle *blockchain permissioned o private*.

Per alcuni, la blockchain è la nuova generazione di Internet, o meglio ancora è la Nuova Internet. Si ritiene che possa rappresentare una sorta di Internet delle Transazioni arrivndo a creare e rappresentare la Internet del Valore sulla base di sette caratteristiche [4]:

1. **Decentralizzazione**
2. **Trasparenza**
3. **Sicurezza**
4. **Immutabilità**
5. **Consenso**
6. **Responsabilità**
7. **Programmabilità**

Partendo da questi principi, la blockchain introduce un nuovo concetto di fiducia al punto che alcuni ritengono che la blockchain possa

assumere anche un valore per certi aspetti di tipo “sociale e politico”: le operazioni avvengono in modo onesto e trasparente, senza la dipendenza da un supervisore.

Bisogna però distinguere la blockchain dalla Blockchain Bitcoin, che è la prima Blockchain. A questa identificazione si è sovrapposta anche quella con la criptocurrency bitcoin e ha portato un pò a ”confondere” la blockchain con altri ambiti di innovazione come le digital currency. Forse per quest’ultima ragione la blockchain è stata spesso associata ad un concetto di digital currency alternativa o complementare e di digital payment. In realtà, come vedremo, la blockchain è un fenomeno assai più ampio e articolato.

2.2.1 Architettura della blockchain

La tecnologia blockchain è un database pubblico decentralizzato che tiene testimonianza di chi possiede beni digitali e di chi effettua transazioni attraverso una rete peer-to-peer. Queste sono protette da crittografia e raccolte cronologicamente all’interno di blocchi di dati, a loro volta protetti e collegati. In tal modo viene creato un registro immutabile, che tiene traccia di tutte le transazioni effettuate, replicato su ogni computer che sfrutta la rete. La blockchain può essere considerata come un insieme di meccanismi interconnessi che forniscono funzionalità specifiche all’infrastruttura, come illustrato in Figura 2.6.

Alla base di questa infrastruttura ci sono le **transazioni**, firmate tra i peers. Queste indicano un accordo tra due partecipanti, che può comportare il trasferimento di risorse fisiche o digitali. Almeno un par-

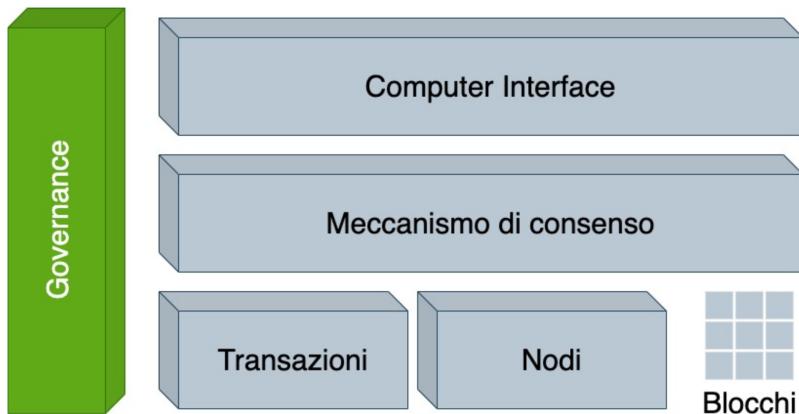


Figura 2.6: Architettura della blockchain di bitcoin

tecipante firma questa transazione, poi divulgata ai suoi vicini. L'entità connessa alla blockchain è chiamata **nodo** e i nodi che verificano tutte le regole blockchain sono chiamati nodi completi (**miner**). Questi raggruppano le transazioni in **blocchi** e determinano se le transazioni sono valide, quindi conservate nella blockchain, e quali no.

Al livello del **Meccanismo di consenso**, i nodi devono raggiungere un accordo su quali transazioni devono essere mantenute nella blockchain per garantire che non ci siano rami corrotti e divergenze. La **Computer Interface** permette alle blockchain di offrire più funzionalità: mentre una blockchain memorizza uno stato, ad esempio l'insieme di tutte le transazioni effettuate dagli utenti, la Computer Interface permette di memorizzare stati complessi che vengono aggiornati dinamicamente utilizzando il calcolo distribuito. Infine, il livello di **Governance** estende l'architettura blockchain coprendo le interazioni umane che avvengono nel mondo fisico. I protocolli blockchain sono influenzati da input di diversi gruppi di persone che integrano

nuovi metodi, migliorano i protocolli e applicano patch al sistema. Benchè queste parti siano necessarie per la crescita di ciascuna blockchain, costituiscono processi esterni alla catena. Per cui, la governance della blockchain si occupa di come questi diversi attori si uniscono per produrre, mantenere o modificare gli input che compongono una blockchain.

2.2.2 Funzionamento della Blockchain

Prendendo in considerazione la più famosa blockchain e cioè la blockchain di Bitcoin, verranno di seguito analizzate le caratteristiche ed il funzionamento generale della tecnologia blockchain. I suoi principali elementi sono:

- Nodi
- Transazioni
- Hash
- Blocchi
- Registro (Ledger)
- Nodi completi (Miner)

Hashing

Viene utilizzata una funzione hash h , per convertire un messaggio di una certa lunghezza in una stringa di dimensioni fisse che genera valori

hash (digest). La mappatura avviene in maniera tale da non consentire di risalire al messaggio originale partendo dalla stringa, la cui lunghezza è direttamente proporzionale al livello di sicurezza della funzione. Una funzione hash ideale dovrebbe:

1. Conteggiare facilmente i valori hash, per qualsiasi dato a disposizione;
2. Generare i medesimi digest per due o più input uguali;
3. Garantire difficoltà di previsione dei digest conoscendo i dati in ingresso;
4. Ostacolare di risalire alle informazioni iniziali a prescindere dalla tipologia di input;
5. Generare valori hash molto diversi tra loro anche per input simili.

A seconda dell'algoritmo adottato varia la lunghezza dei digest. Bitcoin usa l'algoritmo **SHA-256** che restituisce un output di 256 bit. Ecco un esempio:

```
$ hash("Ciao, Mondo!");
```

```
SDFSDF98S8J9238FJ130F2K3FM23F0MI23923F023F230MF203F
```

Nodi, transazioni e blocchi

La **transazione** è un'operazione di scambio di risorse fisiche o digitali tra utenti (**nodi**) connessi ad una rete peer-to-peer. Al suo interno

saranno contenuti gli indirizzi pubblici dei nodi coinvolti nello scambio, l'amount della transazione e la firma digitale del mittente (Figura 2.7).

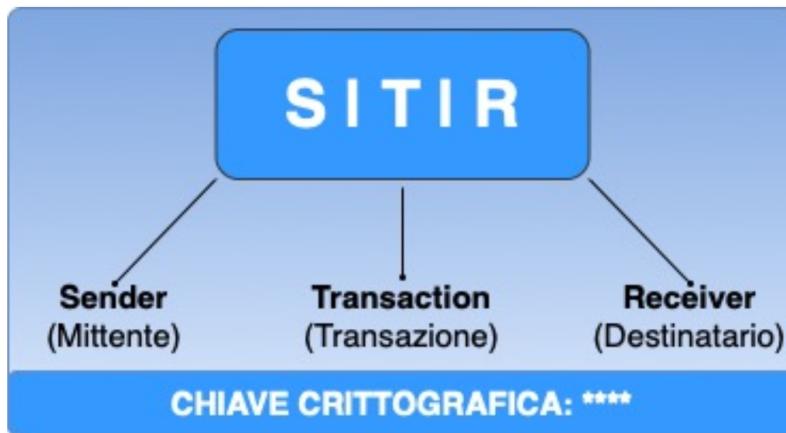


Figura 2.7: Principali componenti della transazione

Le firme digitali sono utili ai nodi per dimostrare la loro identità senza rivelare la propria chiave privata. La firma è il risultato della combinazione tra la chiave privata del mittente e la funzione hash (Figura 2.8).

Il destinatario riceve i dati crittografati insieme alla firma digitale, e sfruttando la chiave privata del mittente, può decifrare i dati. La chiave pubblica è stata precedentemente condivisa tra gli utenti coinvolti o consultabile all'interno del server. L'algoritmo usato per la creazione della firma digitale è l'*Elliptic Curve Digital Signature Algorithm (ECDSA)*[19]. I **blocchi**, unità fondamentali della catena, sono caratterizzati da un insieme di transazioni, un timestamp che li colloca temporalmente, un valore hash posto nell'header (Figura 2.9) e l'hash dei blocchi precedenti, in modo tale da poter monitorare lo stato at-

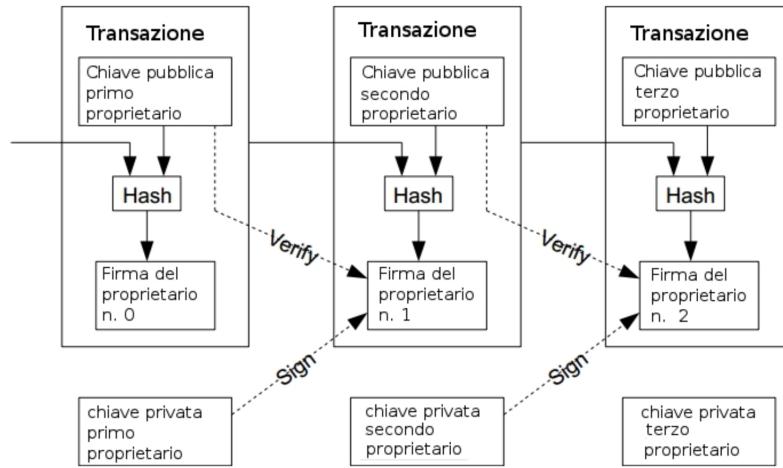


Figura 2.8: Uso della firma digitale nelle transazioni

tuale della catena anche a seguito dell'aggiunta di nuovi blocchi. Come possiamo vedere nella (Figura 2.10), ogni blocco contiene più transazioni, un valore hash proprio e quello del blocco precedente: forma una catena di hash o blockchain. L'ordine dei blocchi è deterministico. Ogni nodo conserva una copia dell'intera blockchain, in modo da poter verificare ogni transazione.

Il registro (Ledger)

Abbiamo tre tipi di network differenti:

Rete centralizzata (Centralized Network) I dati vengono raccolti all'interno di un'unica macchina, alla quale gli utenti si connettono per accedere alle informazioni desiderate. Si crea un *single-point-of-failure*. Se l'intermediario centrale non è attivo o

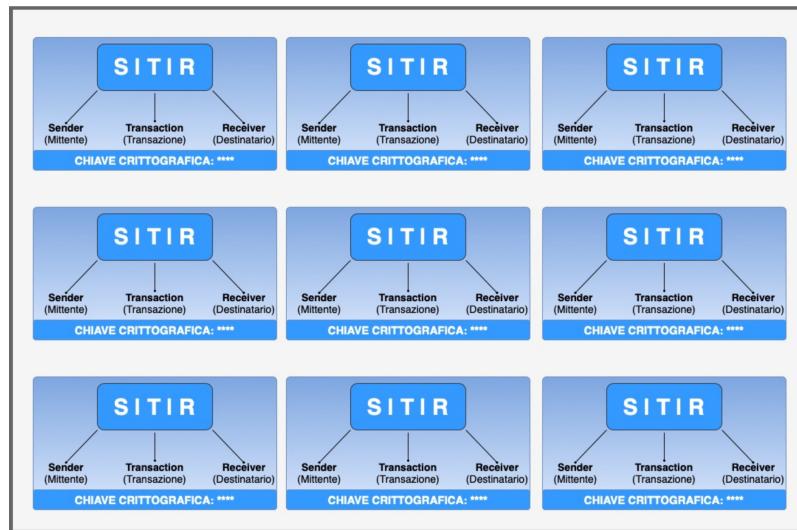


Figura 2.9: Struttura di un blocco contenente diverse transazioni



Figura 2.10: Hash Chain

viene attaccato, l'intera rete smette di funzionare (Figura 2.11 Sinistra).

Rete Decentralizzata (Decentralized Network) Non esiste un'unica macchina per l'archiviazione dei dati in quanto più server lavorano assieme per fornire agli utenti le informazioni desiderate. Non contiene *single-point-of-failure*. Se uno dei nodi, è inattivo o è attaccato, il resto della rete può ancora funzionare normalmente (Figura 2.11 Centro).

Rete Distribuita (Distributed Network) Non esiste alcuna macchina specializzata all'archiviazione dei dati, in quanto ogni singolo nodo della rete contiene le medesime informazioni. Tutti i nodi possono vedere tutto ed esiste un meccanismo di timestamp distribuito (Figura 2.11 Destra).

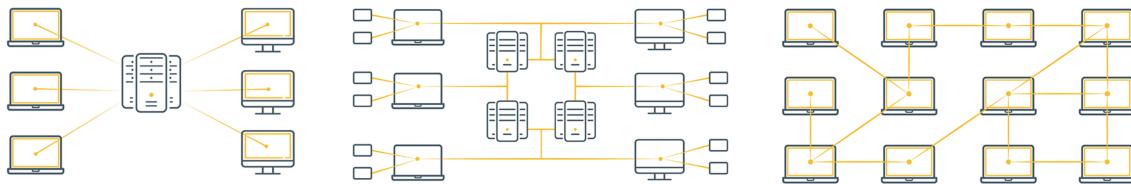


Figura 2.11: Confronto delle tipologie di rete

L'ultima configurazione è tipica della tecnologia blockchain nella quale il registro, definito anch'esso distribuito, tiene nota di tutte le transazioni avvenute all'interno della rete e offre informazioni aggiornate sullo stato attuale della catena. La configurazione distribuita è più sicura di quella centralizzata per due motivi: in primis non esiste

un punto vulnerabile centrale; in secondo luogo la potenza di calcolo richiesta ad un hacker per modificare una transazione all'interno di un blocco, e di conseguenza in tutti i blocchi della catena, non è raggiungibile con le attuali tecnologie.

2.3 Algoritmi di consenso

2.3.1 Proof of Work

L'algoritmo di consenso adottato dalla Bitcoin blockchain è il protocollo *Proof-of-Work (PoW)* illustrato in Figura 2.12.

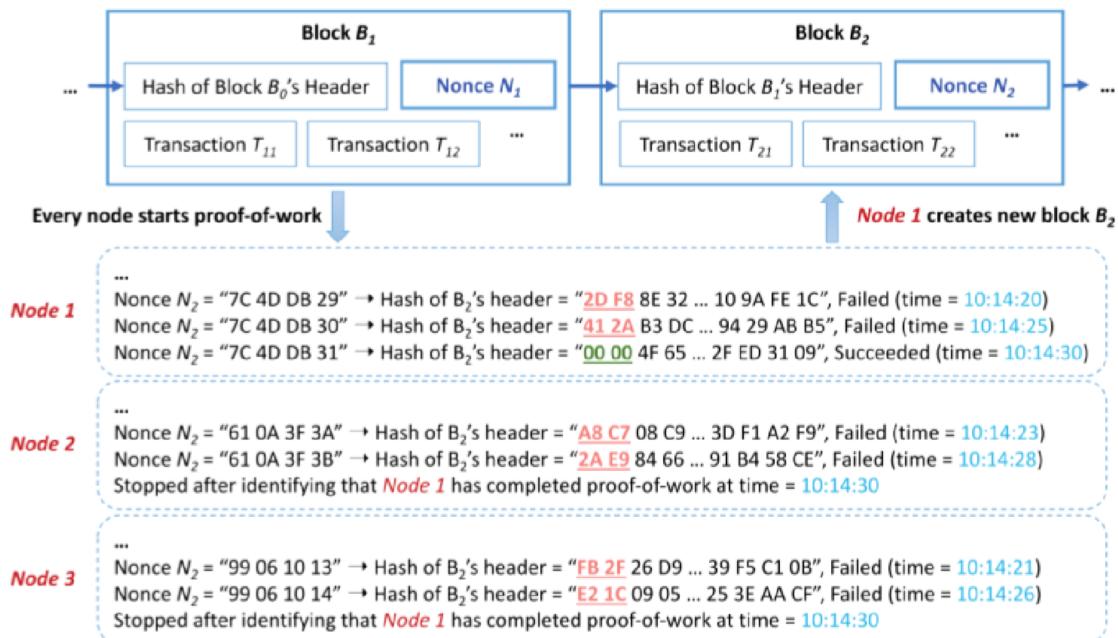


Figura 2.12: Meccanismo Proof-of-Work di bitcoin

Per provare (**proof**) il lavoro di hashing, il *nonce*² viene incrementato di 1 bit per ogni calcolo dell'hash (**work**) finchè il digest da 256 bit (SHA-256) non conterrà 16 bit zero iniziali (target). Le transazioni non confermate vengono raccolte in un pool di memoria per ciascun nodo, mentre al primo nodo che riesce a completare la Proof-of-Work, viene concessa la creazione di un nuovo blocco, la verifica delle transazioni, lo spostamento delle transazioni confermate in un nuovo blocco al fine di allungare la catena e una ricompensa (token). Questo processo è detto mining nella blockchain Bitcoin, e i nodi coinvolti sono i miner. Essi sono chiamati ad affrontare dei problemi computazionali proposti dall'algoritmo, al fine di convalidare nuovi blocchi della catena a seguito della loro risoluzione. Tra i problemi troviamo:

- Individuare un input partendo dal digest della funzione hash;
- Scomposizione in numeri primi;
- *Guided tour puzzle control*: ad alcuni nodi è richiesto il calcolo di una funzione di hash in caso di attacco *DoS (Denial of Service)*.

La difficoltà del problema è proporzionale al numero di miner, alla potenza di calcolo e al carico della rete e deve essere bilanciata: se troppo elevata rallenta la creazione dei blocchi, se troppo bassa rende la rete facilmente attaccabile. La creazione di un blocco richiede mediamente 10 minuti. Il problema in Bitcoin è definito Hashcash, e l'utente che riesce a risolverlo è ricompensato in Bitcoin Figura 2.13.

²Generalmente di 32 bit, è un contatore aggiunto al blocco che funge da input della funzione hash

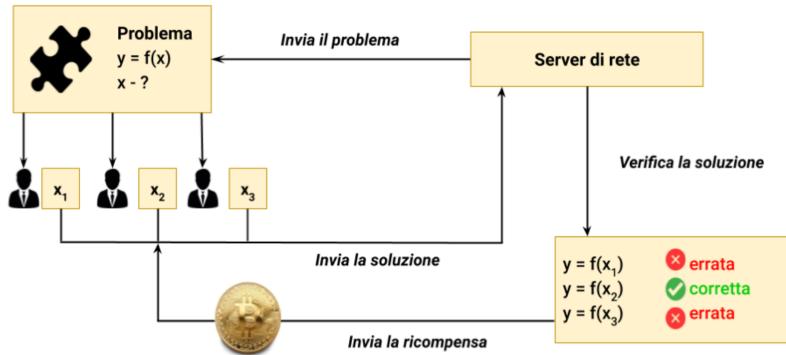


Figura 2.13: Esempio di funzionamento di Hashcash

Sulla base di quanto visto fin qui, la tecnologia blockchain può definirsi **sicura, trasparente, immutabile e trustless**.

2.3.2 Proof of Stake

La *Proof-of-Stake* è un metodo di verifica alternativo alla *Proof-of-Work*. Non viene difatti più usata l'idea di dover dimostrare di aver svolto dei calcoli fornendo un risultato computazionalmente complesso da calcolare, bensì viene introdotto un nuovo concetto, la *coin age*. Essa serve per dimostrare di avere interesse nella gestione della moneta e acquisire quindi la fiducia della comunità per la fase di mining. La *coin age* è facile da calcolare: ad esempio se Bob ha ricevuto 10 coin da Alice e li ha posseduti per 90 giorni, Bob ha 900 *coin-days* di *coin age*. Quando Bob spenderà quei 10 coin ricevuti da Alice, la *coin age* accumulata da Bob con quei coin sarà detta consumata (o distrutta).

Scelta del blocco successivo

Ogni qualvolta un nuovo blocco è aggiunto alla blockchain, deve essere scelto il creatore del blocco successivo. Poichè quest'ultimo non può essere l'account che possiede la maggiore quantità della criptovaluta (altrimenti potrebbe creare tutti i blocchi), sono stati ideati diversi metodi di selezione:

Selezione casuale Nei BlackCoin è utilizzata una funzione casuale per predire il generatore del blocco successivo, impiegando una formula che cerca il valore hash più basso rapportato alla dimensione della somma in gioco. Essendo quest'ultimo pubblico, è facile fare previsioni su quale account si aggiudicherà il diritto di minare il nuovo blocco.

Selezione basata sulla velocità Per evitare di far accumulare agli utenti grandi quantità di monete ed aumentare il proprio patrimonio di coin age, alcune monete come i Reddcoin scelgono il minatore successivo in base alla velocità di movimentazione, incoraggiando quindi lo scambio di moneta.

Selezione basata sul voto A differenza di altre monete che si basano su funzioni del tutto estranee ad interventi umani diretti, alcune monete come i BitShares hanno implementato un sistema che permette agli utenti di votare e scegliere chi far diventare il minatore successivo.

Selezione basata sull'anzianità Nei PPCoin è stato introdotto, affiancato ad una selezione casuale, il concetto di anzianità. L'anzianità è misurata in coin age, valore calcolato considerando solamente monete non spese per almeno 30 giorni e non più di 90, ciò permette di non avere utenti con un'anzianità tale da dominare gli altri. Ad ogni mining il minatore consuma la quantità di coin age: si ha così la garanzia che non sia sempre lo stesso a dominare la blockchain.

Al giorno d'oggi sono poche le monete che hanno implementato la Proof-of-Stake in maniera pulita, una di queste è *Nxt* (AGGIUNGERE CIT), che però ha numerosi problemi di sicurezza su cui sta ancora cercando una soluzione efficace.

2.4 Blockchain Pubbliche e Private

La tecnologia Blockchain è in continua evoluzione: dalla comparsa di Bitcoin la ricerca ha fatto venire alla luce diverse varianti ed evoluzioni, tanto che oggi è già possibile farne una prima classificazione suddividendo le blockchain in due famiglie principali, **blockchain pubbliche** e **blockchain private** (o **permissioned**).

2.4.1 Blockchain Pubbliche

Le Blockchain Pubbliche non prevedono alcun tipo di restrizione, chiunque vi può prendere parte senza che sia necessario alcun permesso; sono

tipicamente open-source.

In particolare, tutti possono:

- Scaricare l'intera blockchain e il programma per diventare un nodo della rete iniziando così a partecipare alla validazione delle transazioni da aggiungere alla blockchain secondo il protocollo di consenso adottato;
- Inviare transazioni alla rete che verranno inserite nella blockchain se valide;
- Visualizzare tutte le transazioni memorizzate all'interno della blockchain attraverso un *block explorer*. Le transazioni possono essere più o meno trasparenti garantendo un certo livello di privacy e anonimato a seconda del tipo di blockchain.

Questo tipo di blockchain adotta delle forme di incentivi economici per coloro che si occupano della validazione delle transazioni (i cosiddetti "miner") adottando sistemi come Proof of Work o Proof of Stake. Tali schemi permettono di conservare la blockchain al sicuro e di funzionare in un contesto totalmente trustless, ovvero in un ambiente dove i nodi della rete non si conoscono e non si fidano l'un con l'altro.

La fondamentale caratteristica delle blockchain pubbliche è dunque quella di offrire un mezzo tecnologico per poter realizzare la deintermediazione nella fornitura di molteplici servizi e consentono la creazione e l'esecuzione di applicazioni decentralizzate senza che sia necessario sostenere alcun costo per l'infrastruttura. Ai vantaggi appena indicati

si contrappongono alcuni svantaggi quali la possibilità di gestire un numero di transazioni al secondo significativamente più basso rispetto a quelle che sono in grado di gestire le blockchain permissioned e l'impossibilità di memorizzare all'interno della blockchain informazioni riservate dal momento l'intera catena di blocchi è pubblicamente consultabile [16]. Le blockchain pubbliche più note sono Bitcoin [12] e Ethereum [17].

2.4.2 Blockchain Private

Le blockchain private prevedono limitazioni sull'accesso contemplando la presenza di un sottosistema per l'identificazione e la gestione dei permessi associati ai vari nodi. Esclusivamente chi è stato autorizzato dall'amministratore a prendere parte alla rete può quindi, in base ai diritti che gli sono stati concessi, ricercare il contenuto della blockchain, inviare o verificare nuove transazioni.

Tale tipologia di blockchain trova impiego in tutti quei settori dove vi è un consorzio di attori che lavorano insieme e che desiderano scambiarsi informazioni in modo riservato ma che non si fidano completamente l'uno dell'altro. Una delle applicazioni più abituali che vengono citate quando si parla di blockchain permissioned, è quella della gestione della filiera logistico-produttiva, un contesto nel quale vi sono diverse parti (fornitori, trasportatori, intermediari finanziari, catene di distribuzione) che hanno bisogno di condividere fra loro informazioni senza che però siano pubbliche e visibili a tutto il mondo (si consideri il caso di segreti o strategie aziendali che rappresentano un

vantaggio rispetto alla concorrenza) o anche solamente ad altre parti coinvolte nella filiera stessa (si pensi ad esempio ad un prezzo speciale concordato con una certa catena di distribuzione che non vuole essere reso noto alle altre).

Le blockchain permissioned risolvono positivamente questo problema, permettendo la costituzione una base di dati distribuita e condivisa tra tutte le parti dove tutte le modifiche rimangono registrate e immutabili e dove tutti partecipano alla validazione delle transazioni evitando così di concentrare tutto il potere presso un unico ente in cui tutte le parti dovrebbero riporre la fiducia, esattamente come accadrebbe adottando un database condiviso tradizionale. La ricerca nel campo delle blockchain permissioned è iniziata solo negli ultimi anni e risulta essere ancora agli inizi. Fra le blockchain più promettenti di questo tipo oggi la più famosa è probabilmente *Hyperledger Fabric* [2].

2.5 Generazioni di Blockchain

La suddivisione in generazioni in base alle caratteristiche e funzionalità offerte dalla blockchain stessa è un ulteriore modo adottato per la classificazione delle blockchain. Ad oggi si possono conteggiare tre generazioni:

Prima generazione: Criptovalute La prima applicazione della blockchain è stata la realizzazione di criptovalute come Bitcoin e altre semplici alt-coin come *Litecoin*. Con il loro ingresso è stato reso possibile per la prima volta lo scambio diretto di denaro tra due

parti senza la necessità di nessun intermediario (peer-to-peer) in modo sicuro, veloce ed economico.

Seconda generazione: Digital Assets, Smart Contract e dApp

L'avvento della seconda generazione di blockchain si è avuto con la nascita di *Ethereum*, la prima blockchain ad introdurre il concetto di smart contract, ossia semplici programmi che possono essere scritti, distribuiti ed eseguiti all'interno di un sistema informatico decentralizzato, sicuro, immutabile e affidabile.

Le blockchain di seconda generazione permettono anche la definizione e lo scambio di un qualsiasi asset digitale e non solo si limita più quindi a permettere lo scambio di denaro. Gli sviluppatori possono creare su di esse nuovi token e applicazioni decentralizzate (*dApps*).

Terza generazione: Scalabilità, interoperabilità e IoT

La definizione esatta di blockchain di terza generazione è un tema ancora estremamente dibattuto, ma sono già diversi i progetti che si sono dati questa etichetta, il più noto fra questi è Cardano [7]. I problemi che le blockchain di terza generazione stanno cercando di risolvere sono legati all'interoperabilità tra blockchain diverse, allo sviluppo di tecnologie ad-hoc per la realizzazione di applicazioni blockchain *M2M* (*machine to machine*) in ottica Internet of Things e alla scalabilità, segnatamente attraverso la creazione di molteplici layer. Questo ha portato anche alla nascita di Lightning Network, layer di secondo di livello per Bitcoin che avremo

modo di affrontare più avanti.

Capitolo 3

Blockchain nell'era quantistica e Bitcoin QR

Capitolo 4

Bitcoin QR

Implementato

Capitolo 5

Conclusioni

Bibliografia

- [1] In: 48 (set. 2000). doi: [10.1002/1521-3978\(200009\)48:9<771::aid-prop771>3.0.co;2-e](https://doi.org/10.1002/1521-3978(200009)48:9<771::aid-prop771>3.0.co;2-e). URL: <https://doi.org/10.1002%2F1521-3978%28200009%2948%3A9%2F11%3C771%3A%3Aaid-prop771%3E3.0.co%3B2-e>.
- [2] Elli Androulaki et al. “Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains”. In: *Proceedings of the Thirteenth EuroSys Conference*. EuroSys ’18. Porto, Portugal: Association for Computing Machinery, 2018. ISBN: 9781450355841. DOI: [10.1145/3190508.3190538](https://doi.org/10.1145/3190508.3190538). URL: <https://doi.org/10.1145/3190508.3190538>.
- [3] Dave Bayer, Stuart Haber e W Scott Stornetta. “Improving the efficiency and reliability of digital time-stamping”. In: *Sequences II*. Springer, 1993, pp. 329–334.
- [4] Mauro Bellini. *Blockchain: Cos’è, come funziona e applicazioni oggi*. <https://www.blockchain4innovation.it/esperti/blockchain-perche-e-così-importante/>. Lug. 2021.

- [5] Ethan Bernstein e Umesh Vazirani. “Quantum complexity theory”. In: *SIAM Journal on computing* 26.5 (1997), pp. 1411–1473.
- [6] F. Bloch. “Nuclear Induction”. In: *Physical Review* 70.7-8 (ott. 1946), pp. 460–474. DOI: [10.1103/PhysRev.70.460](https://doi.org/10.1103/PhysRev.70.460).
- [7] Conor. *What's next for Blockchain? 3rd generation platforms*. Set. 2021. URL: <https://medium.com/web3labs/whats-next-for-blockchain-3rd-generation-platforms-a26f34da4d59>.
- [8] David Deutsch. “Quantum theory, the Church–Turing principle and the universal quantum computer”. In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 400.1818 (1985), pp. 97–117.
- [9] Stuart Haber e W Scott Stornetta. “How to time-stamp a digital document”. In: *Conference on the Theory and Application of Cryptography*. Springer. 1990, pp. 437–455.
- [10] David Mermin. *Breaking RSA Encryption with a Quantum Computer: Shor's Factoring Algorithm*. URL: <https://web.archive.org/web/20121115112940/http://people.ccmr.cornell.edu/~mermin/qcomp/chap3.pdf>.
- [11] Satoshi Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. URL: <https://bitcoin.org/bitcoin.pdf>.
- [12] Satoshi Nakamoto. “Bitcoin: A peer-to-peer electronic cash system”. In: *Decentralized Business Review* (2008), p. 21260.

- [13] Alan T. Sherman et al. “On the Origins and Variations of Blockchain Technologies”. In: *CoRR* abs/1810.06130 (2018). arXiv: 1810.06130. URL: <http://arxiv.org/abs/1810.06130>.
- [14] Nick Szabo. *Unenumerated*. <https://web.archive.org/web/20061213062310/https://unenumerated.blogspot.com/2005/12/bit-gold.html>. Dic. 2005.
- [15] Tavares. *Google and NASA achieve quantum supremacy*. <https://www.nasa.gov/feature/ames/quantum-supremacy/>. Ott. 2019.
- [16] Marko Vukolić. “The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication”. In: *International workshop on open problems in network security*. Springer. 2015, pp. 112–125.
- [17] Darya Yafimava. *Hyperledger Enterprise Solutions: Top 5 real use cases*. Mag. 2019. URL: <https://openledger.info/insights/hyperledger-enterprise-solutions-top-5-real-use-cases/>.