ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

**SCUOLA DI SCIENZE**
**Corso di Laurea in Informatica per il Management**

# Blockchain tra anonimato e fama

**Relatore:**
**Chiar.mo Prof.**
**Davide Sangiorgi**

**Presentata da:**
**Giuseppe Pedullà**

**II Sessione**
**2016/2017**

*Un ringraziamento speciale va ai miei genitori senza i quali non avrei mai iniziato questo corso di laurea, e al mio amico Vins senza il quale non l'avrei mai finita.*

Giuseppe Pedullà

# Indice

# Elenco delle figure

*"Ecco che cosa ha fatto per me il Signore, nei giorni in cui si è degnato di togliere la mia vergogna tra gli uomini".(Lc 1,25)*

# Introduzione

In questa tesi è stato portato avanti uno studio su diversi modi di concepire la sicurezza: quello dell'anonimato e quello della segretezza anche con informazioni di pubblico dominio con il metodo della crittografia.
Il raffronto tra le due soluioni, quella dell'anonimato e quello della crittografia, appare quanto mai strano poichè due soluzioni diametralmente opposte. Entrambe le soluzioni offrono spunti interessanti.
Per quanto riguarda l'anonimato, si tratta di un meccanismo per cui si può diventare quasi completamente invisibili, saranno introdotte alcune delle tecniche utilizzate dagli hacker.
Sulla crittografia c'è da dire che è il modo con il quale vengono cifrati i messaggi che si vogliono spedire e si può anche decidere di renderli visibili a destinatari ben specificati.
Si è scelto di approfondire questi argomenti perchè ritenuti i più significativi nonchè i più utilizzati.
Nello specifico, nel **capitolo 1** sarà trattato il concetto di sicurezza come risultato dell'anonimato, saranno presentati alcuni meccanismi e saranno introdotti alcuni dei problemi legati proprio alla ricerca dell'anonimato; infine verrà spiegato il funzionamento del protocollo *onion routing*.
Nel **capitolo 2** sarà presentato un altro sistema di sicurezza utile soprattutto per mantenere la privacy nello scambio di messaggi tra diverse parti: la crittografia. Sarà trattato come argomento generale e saranno portati alcuni esempi tra i più noti.
Il **capitolo 3** darà un esempio in cui la crittografia funziona in maniera eccelsa: la blockchain. Verrà spiegato come è applicata la crittografia nella blockchain secondo il sistema delle chiavi asimmetriche.
Il **capitolo 4** spiegherà che cos'è la blockchain prendendo in considerazione moltepici punti di vista. Saranno introdotti alcuni concetti molto importanti per la blockchain mentre altri saranno approfonditi il tutto per permettere di comprendere meglio possibile questo sistema molto complesso ma molto affascinante.
Si è scelto di approfondire questo argomento trattandolo come esempio pratico nel quale la crittografia viene utilizzata e funziona molto bene.

# Capitolo 1

# Come essere sicuri online

Sempre più spesso vengono violati sistemi di sicurezza ma anche dispositivi personali.
Gli hacker più famosi dichiarano che tutti sono tracciati, ogni azione che viene eseguita su Internet, ogni *login* effettuato, ogni acquisto con carta di credito viene memorizzata e, a volte, diffusa. Ogni tipo di dispositivo può essere violato, manomesso e utilizzato dagli esperti nell'arte dell'hacking.
Non basta credere di essere sicuri perchè non si usa la propria carta di credito su internet o perchè non si confida la propria password per essere davvero al sicuro, per essere certi che le propria privacy sia al sicuro.
Non basta nemmeno pensare di essere al sicuro perchè non si è un obiettivo sufficientemente interessante.
Questo capitolo tratterà di come si possa essere sicuri online.

## 1.1 La necessità di essere al sicuro

Ogni giorno vengono violati sistemi di sicurezza in tutto il mondo e ogni tipo di dispositivo. Basti pensare a quanti computer o smartphone delle celebrità vengano violati per diffondere le informazioni che ci sono contenute.
Un altro esempio molto famoso è il cosiddetto *wikileaks*[1] con il quale sono stati diffusi segreti di stato degli USA.
Ma perchè tutti devono essere al sicuro? É proprio necessario che chiunque si mobiliti per difendere la propria privacy?
La risposta a queste domande sta nel fatto che anche se apparentemente non si è un obiettivo appetibile, l'attacco potrebbe essere rivolto a un obiettivo estraneo passando e servendosi, quindi, di una scarsa protezione.
Il caso a cui si fa riferimento riguarda un fatto realmente accaduto a *KrebsOnSecurity*, un ente per le investigazioni private e leader nel settore della sicurezza, inoltre è fornitore di alcuni sistemi di sicurezza come le telecamere di videosorveglianza.
L'attacco in questione è stato portato avanti da alcuni hacker che hanno manomesso alcune delle telecamere di sorveglianza che si connettevano ai server di KrebsOnSecurity. Hanno lanciato dei botnet dopo aver manomesso le

[1] Si fa riferimento al caso di Edward Snowden.

telecamere e hanno iniziato un attacco DDoS[2] contro KrebsOnSecurity, producendo un traffico di dati peri a 665 Gigabit per secondo: una quantità di richieste esorbitante, basti pensare che prima di questo attacco, il record registrato era di 363Gbps.
In sostanza i possessori di queste telecamere non erano adeguatamente protetti o forse avevano usato poca diligenza ed hanno permesso che gli hacker violassero le telecamere e le gestissero per mettere fuori uso il server di KrebsOnSecurity.
Con questo esempio si vuole introdurre il concetto di sicurezza personale come un qualcosa di particolarmente importante: tutti dovrebbero fare attenzione a quello che fanno tramite i dispositivi che utilizzano.
Ma quali sono le tecniche o le misure di sicurezza da adottare per potersi ritenere al sicuro?
Per prima cosa su internet bisognerebbe mantenere la propria identità più nascosta possibile. Quante più operazioni si possono fare senza che siano legate al proprio nome, meglio è. Il problema principale sta nell'architettura stessa di Internet, la quale si compone di due protocolli: **TCP** e **IP**. In estrema sintesi possiamo definire il protocollo IP come colui che amministra il traffico di dati e smista i pacchetti da un nodo ad un altro; mentre il TCP come il protocollo che, a partire da un messaggio di una certa lunghezza, lo riduce in piccoli pacchetti e, una volta giunti a destinazione, li ricompone nell'ordine corretto.
Il punto è che questi pacchetti viaggiano per il mondo spostandosi per nodi sparsi ovunque: è un problema perchè molti dei nodi sono situati negli USA i quali servendosi di una legge creata appositamente per combattere il terrorismo, *patriot act*, consentono agli organi di polizia tra i quali la *National Security Association*, NSA, di verificare tutto ciò che transita nel territorio nazionale.
Questo naturalmente implica che se viene spedito un messaggio e molti dei pacchetti passano per i nodi situati negli USA, la NSA sarà a conoscenza di gran parte del messaggio.
Un altro esempio è quello delle e-mail. Tutti i *provider* di servizi email, come gmail, yahoo, microsoft etc., si riservano la possibilità di interpretare un messaggio che viene spedito per posta elettronica per fini commerciali o per sicurezza. Riguardo a questo, in America, è stata aperta una *class action* che ancora non si è risolta[3].

## 1.2 La sicurezza sta nell'anonimato

Ci si chiederà a questo punto cosa sia possibile fare per proteggere la propria privacy.
Per alcuni degli hacker più famosi la soluzione starebbe nel diventare *anonimi[5]*,

[2] *Distributed Denial Of Service* è un attacco che consiste nel sovraccarico di richieste per un server che rende inaccessibili le informazioni.

[3] Ci si riferisce alla class action portata avanti contro yahoo da Stuart Diamond e alcuni suoi amici e colleghi.

praticamente invisibili. Le operazioni per rendersi anonimi sono diverse e alcune di esse potranno sembrare un po' esagerate, tuttavia è necessario tenere presente tutto ciò che è stato detto finora, ponendo particolare attenzione al fatto che la la privacy di tutti è seriamente a rischio.
Per prima cosa bisogna che la propria vita online sia separata da quella vera. Account di social network, blog etc. sono dannosi nella misura in cui tutte le informazioni affidate a questi siti fanno riferimento a una parte della persona che vi è rappresentata e quindi scovare un account e leggerne tutte le informazioni equivale a scoprire molto di una persona.
Inoltre per effettuare operazioni bancarie il consiglio è quello di avere uno strumetto adibito solo a tale scopo, un chromebook, o un iPad o un qualsiasi strumento che venga usato, con criterio, solo per quelle operazioni.
Bisogna far notare che accedere al proprio Wi-Fi equivale mostrare dove si è in un preciso momento ed è pericoloso soprattutto se ci si collega ad un sito, per esempio al proprio account di posta elettronica, con le proprie credenziali il che equivale a collegare il proprio indirizzo ad un account specifico.
Molti degli hacker professionisti fanno abitualmente uso di macchine virtuali. In questa maniera si può installare su un Mac un sistema Windows, o su un sistema Windows un sistema Linux e così via. Così facendo le richieste ai server non sono facilmente collegabili. Bisogna però tenere conto che ciascun dispositivo ha una scheda di rete cui è associato un indirizzo MAC univoco. Questo vuol dire che se ci si collega al proprio modem, il provider dei servizi di rete sarà a conoscenza che un dato indirizzo MAC si è connesso in un momento specifico in una posizione specifica. Per ovviare a questo problema si utilizza un *hotspot* mobile, per esempio il cellulare.
Questa soluzione è ottimale per rispondere ai problemi finora elencati. Ma cosa succede se si utilizza il proprio cellulare come hotspot e si resta in un luogo specifico per molto tempo? Questo è il caso di *Kevin Mitnick* il quale per sfuggire all'FBI ha adottato molte misure di sicurezza ma ha ingenuamente dimenticato questa; infatti rimanendo in una posizione per molto tempo il cellulare che fornisce la connessione internet sarà rintracciabile perchè connesso a una stazione precisa.

### 1.2.1 Onion routing

Uno strumento molto potente è sicuramente **tor**. Si tratta di un browser che utilizza il protocollo onion per gestire il traffico di rete. Utilizzando questo protocollo tor è in grado di nascondere l'*IP address*[4] di un dispositivo.
Che cos'è questo protocollo onion?
L'**onion routing** o **onion protocol** è un protocollo di sicurezza per la navigazione in rete. Utilizza un meccanismo di proxy per nascondere l'indirizzo IP di un dispositivo. La sua fondamentale caratteristica è quella di far rimbalzare il segnale da un punto ad un altro $n$ volte cambiando, passo dopo passo, indirizzo IP rendendo quindi impossibile identificare chi ha effettuato una data richiesta. Tor, inoltre, è il punto di accesso al dark web.

[4] É un numero collegato a un dispositivo che ha effettuato l'accesso a una rete. Questo dispositivo è detto anche *host*.

Si chiama onion routing perchè una richiesta, quando viene creata, è nascosta da diversi strati, ciascuno dei quali è codificato e cifrato. Ogni volta che questa richiesta atterra su un nodo specifico viene letta e decodificata, viene eliminato uno strato. Ogni nodo ha la capacità di eliminare un unico strato per volta e rispedire il pacchetto al prossimo nodo.
Ogni nodo contiene informazioni solo di sè e della posizione del nodo successivo. Quando il pacchetto atterra sull'ultimo nodo, viene decifrato e quindi la richiesta è in chiaro, ovvero non criptata.
In sostanza con questo sistema si può navigare in Internet senza che si sappia chi ha effettuato l'accesso a quale sito e da dove la richiesta sia partita; oppure si possono inviare messaggi rendendo molto complicato, se non impossibile, rintracciarli e scoprirne il mittente.
Tutti questi accorgimenti senza il buonsenso servono a poco. Un caso molto recente è quello di Gurtej Randhawa il quale aveva acquistato degli esplosivi utilizzando tor ma la NSA ha tracciato la spedizione e gli ha fatto recapitare una bomba falsa.
Questo è un esempio utile a dimostrare che anche gli strumenti più potenti se non sono usati con criterio possono rivelarsi controproducenti.
Tra i sistemi di sicurezza per mantenere segrete le proprie informazioni e i messaggi che vengono inviati certamente è doveroso citare la crittografia. Questo è un argomento particolarmente importante e pertanto verrà trattato nel capitolo seguente.

# Capitolo 2

# La crittografia come strumento di sicurezza nelle comunicazioni

In questo capitolo verrà trattato un altro sistema di sicurezza[6]: la cifratura dei messaggi. Se tutti i messaggi che inviamo sono cifrati adeguatamente è molto difficile che vengano letti e interpretati da chi non dovrebbe.
Verrà introdotto il concetto di sicurezza informatica e verranno trattati alcuni algoritmi di sicurezza delle comunicazioni, nella fattispecie i protocolli di sicurezza chiamati *a chiave privata* e *a chiave pubblica*.

## 2.1 Introduzione alla crittografia

Come abbiamo visto nel primo capitolo, essere anonimi è fondamentale per assicurarsi la protezione della privacy e proteggere sè stessi. Tuttavia non è l'unico modo per tenere al sicuro le proprie comunicazioni. Un modo molto più comune è quello di *cifrare* i propri messaggi.
Storicamente sono sempre stati usati metodi di cifratura per proteggere i messaggi che erano spediti per assicurarsi di non essere intercettati.
Un esempio è quello del *cifrario di Cesare* il quale, dato un alfabeto e un numero di posizioni $k$, consiste nel sostituire tutte le lettere del messaggio con una lettera che si trova $k$ posizioni dopo nell'alfabeto dato.

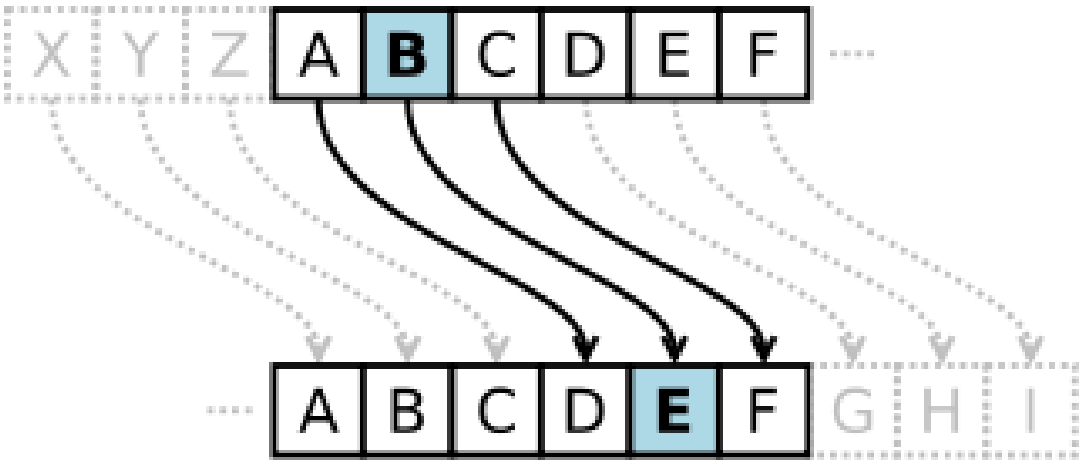


FIGURA 2.1: Il cifrario di cesare; fonte: https://commons.wikimedia.org/wiki/

Questo sistema, tuttavia, non è molto sicuro: basta infatti conoscere l'alfabeto e il numero $k$ che si utilizzano e il messaggio viene facilmente interpretato.
Ci si è posti quindi sin da subito il problema di un algoritmo più efficiente possibile per proteggere le proprie informazioni.

## 2.2 La crittografia

Dall'avvento dell'informatizzazione la crittografia si è complicata sempre di più e ad oggi sono molte le tecniche di cifratura che rendono quasi impossibile la decifratura di un messaggio.
Ma a volte non basta utilizzare un buon sistema crittografico, bisognerebbe far uso di uno perfetto. Si è soliti distinguere due tipi di sistema crittografico perfetto:

- *perfect secrecy*
- *computational secrecy*

**Definizione 2.1** (Definizione di sistema crittografico *perfect secrecy*)**.** Un sistema crittografico si dice *perfect secrecy* se garantisce una protezione matematicamente inattaccabile.

**Definizione 2.2** (Definizione di sistema crittografico *computational secrecy*)**.** Un sistema crittografico si dice *computational secrecy* se è inviolabile tenendo conto delle risorse di un attaccante[1].

I sistemi crittografici perfect e computational secrecy sono modelli verso cui un algoritmo crittografico tende.
Prima di trattare i concetti fondamentali, è necessario dare le definizioni di *cifratura* e *decifratura*:

**Definizione 2.3** (Funzione di cifratura)**.**
$C_k(m) = c$

La funzione $C$ cripta il messaggio $m$ con chiave $k$.

**Definizione 2.4** (Funzione di decifratura)**.**
$D_k(m) = d$

La funzione $D$ decripta il messaggio $m$ con chiave $k$. Matematicamente la funzione $D$ è l'inversa di $C$ tale che

**Definizione 2.5.**
$D_k(C_k(m)) = m$

Esistono due famiglie per le funzioni di cifratura:

- cifrari per uso ristretto;

[1] Attaccante inteso come colui che vuole infrangere il sistema ideato.

- cifrari per uso generale.

**Il cifrario per uso ristretto** si basa sul fatto che le funzioni di cifratura e decifratura sono nascoste, non sono conosciute e quindi la sua forza sta nella segretezza dell'algoritmo.

**Il cifrario per uso generale** si basa su un sistema di chiavi, poichè le funzioni di cifratura e decifratura sono note a tutti.
Queste chiavi si distinguono in *chiave pubblica* e *chiave privata* appunto perchè la chiave privata è tenuta da un singolo e non è conosciuta da nessun altro mentre la chiave pubblica può essere conosciuta da tutti. La sicurezza di questo algoritmo sta nell'uso delle chiavi e nella loro complessità.

### 2.2.1 Proprietà della crittografia

Come abbiamo già accennato prima, un sistema crittografico per essere considerato tale deve avere determinate caratteristiche, delle proprietà. Esse sono indispensabili per garantire la sicurezza e l'autenticità dei mittenti per i destinatari e la consistenza del messaggio inviato.
Enumeriamo quindi le **proprietà della crittografia**:

- **confidentiality**: la confidenzialità è quella proprietà secondo cui un sistema deve garantire la segretezza del messaggio cifrato.
  Una soluzione per proteggere la riservatezza delle informazioni è senza dubbio la crittografia la quale garantisce che solo il destinatario, colui che conosce il modo di decifrare il messaggio, possa riuscire ad interpretare le informazioni del messaggio. Per questo motivo funzioni crittografiche sono utilizzate in diversi protocolli di rete tra i quali è necessario menzionare il protocollo SSL/TLS: un protocollo per garantire la sicurezza nelle comunicazioni;

- **integrity**: l'integrità delle informazioni è quella proprietà secondo cui un sistema deve garantire la protezione dei messaggi dalla modifica di dispositivi non autorizzati. Dal momento in cui un messaggio ha valore se solo è consistente, cioè se non è stato modificato, l'integrità è un concetto fondamentale.
  Come per la riservatezza dei dati, la crittografia gioca un ruolo decisivo per far si che l'integrità dei dati sia assicurata. Di solito per prevenire la manomissione dei messaggio e assicurare, quindi, l'integrità delle informazioni si utilizzano sistemi di hashing dei dati e confronti con codici hash di messaggi precedenti.
  Va considerato, tuttavia, che la chiave per l'hash va condivisa in sicurezza quindi a sua volta ci sarà il pericolo che venga letta da malintenzionati. Per questo sono stati introdotti altri metodi come la firma digitale;

- **availability**: la disponibilità delle informazioni è quella proprietà secondo cui un sistema deve garantire che il messaggio arrivi al destinatario corretto. Le informazioni hanno valore solo se recapitate al destinatario giusto al momento giusto. Questo concetto è particolarmente

importante anche perchè praticamente ogni giorno vengono effettuati attacchi informatici verso siti o archivi per non consentire la disponibilità delle informazioni.
Di seguito un grafico che mostra le statistiche degli attacchi DDoS nel secondo trimestre del 2017.

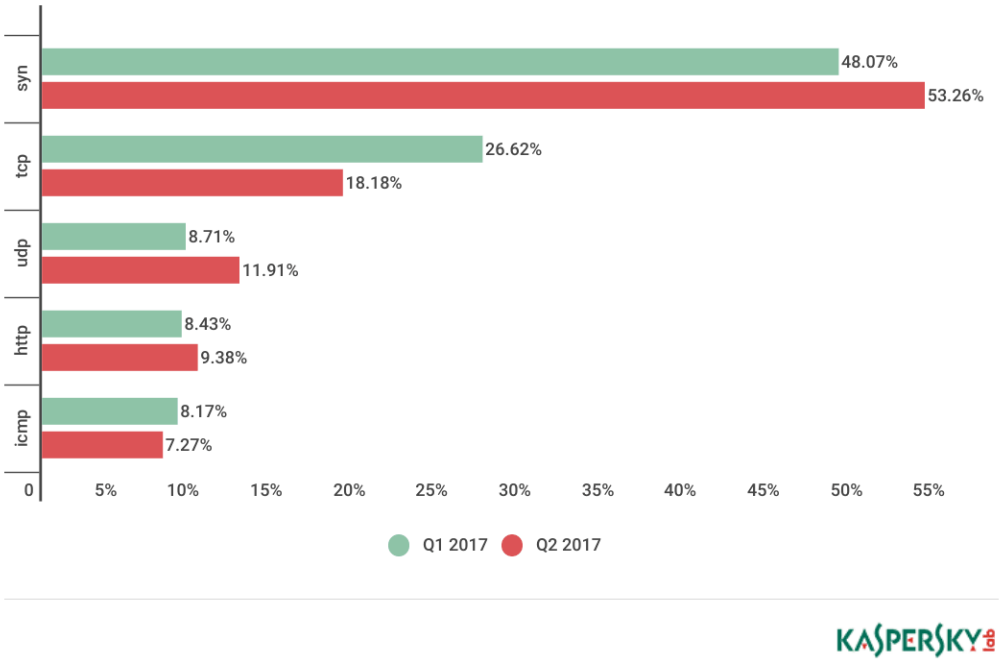


FIGURA 2.2: Immagine che mostra gli attacchi DdOS nel secondo trimestre del 2017; fonte: https://securelist.it/ddos-attacks-in-q2-2017/62709/

Nell'immagine 2.2 possiamo apprezzare[2], come i dati statistici raccolti riguardo al secondo trimestre del 2017 ci mostrano inequivocabilmente, come siano tornati alla ribalta gli attacchi DDoS caratterizzati da una notevole durata: lo 0,07% degli attacchi si è protratto per oltre 100 ore. L'attacco record, poi, si è esteso, in termini temporali, per ben 277 ore, ovvero 157 ore in più rispetto al record fatto segnare nel trimestre precedente.
È inoltre significativamente aumentata la quota riguardante gli attacchi di durata relativamente breve, non superiore alle 4 ore: si è in effetti passati dall'82,21% del trimestre precedente all'85,93% del trimestre qui preso in esame.
Al contempo, risultano complessivamente diminuiti i valori percentuali riconducibili agli attacchi con durata compresa tra le 5 e le 49 ore[2].
Il problema della disponibilità dei dati non riguarda solo problemi derivanti da attacchi informatici ma anche da effetti naturali: un terremoto, un incendio, un blackout possono causare indisponibilità di dati o la loro corruzione.

[2]L'immagine mostra gli attacchi che sono stati effettuati contro protocolli di rete.

Come si può garantire che i dati siano disponibili per le persone giuste al momento giusto? Tutto sta in un gestione intelligente e diligente dei backup.
Infatti il backup offline può indubbiamente circoscrivere i danni causati da eventi naturali;

- **non-repudiation**: la proprietà di non ripudio è quella proprietà secondo cui un sistema deve garantire che non sia possibile al mittente il negare di aver spedito un messaggio. Questo è un concetto molto legato alle firme elettroniche di cui la firma digitale è sottoinsieme.
  La crittografia, quindi, rafforza anche questo concetto perchè assicura che siano rispettati i criteri di *integrity* e *availability*.

## 2.3 Crittografia a chiave privata

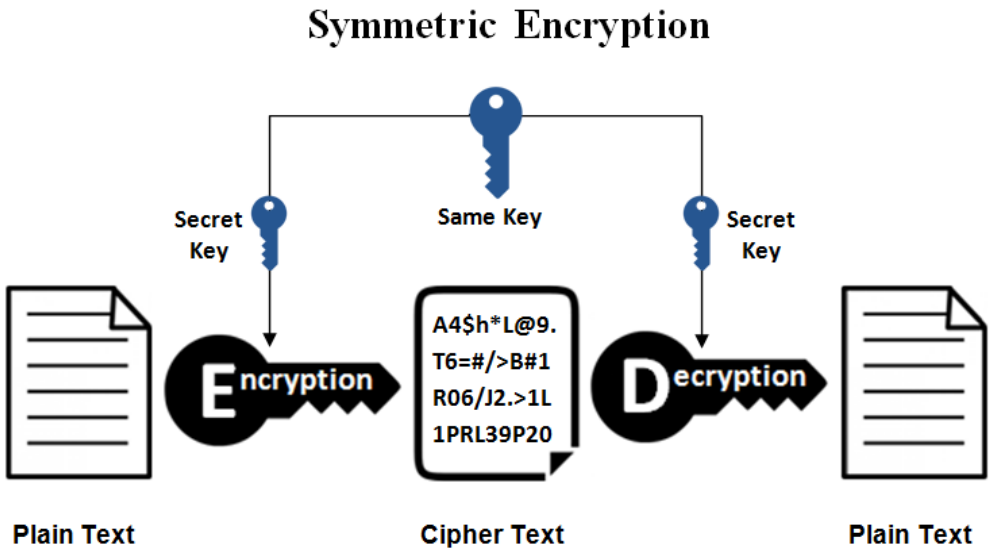


FIGURA 2.3: Funzioanmento della crittografia a chiave privata; fonte: https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences

Per capire come funzioni la crittografia a chiave privata verrà introdotto brevemente un protocollo e, successivamente, come si possano gestire le chiavi che servono per la condivisione dei messaggi tra mittente e destinatario.

### 2.3.1 DES

Il **DES**, o *Data Encryption Standard*, è un algoritmo molto famoso a chiave simmetrica. É stato considerato uno standard per molto tempo; ad oggi, invece, lo standard per la cifratura è **AES**, *Advanced Encryption Standard*, costituito come tale dal *NIST*.

**Caratteristiche del DES**: lavora in blocchi da 64 bit cioè utilizza chiavi da 64 bit di cui solo 56 bit sono usati mentre gli altri 8 servono per il "parity check[3]".
Per rinforzare ulteriormente l'algorimo sono possibili alcune operazioni, a volte vengono usate ripetutamente:

- **Permutazione**: viene permutato 1 bit sull'input e 1 bit sull'output;
- **Sostituzione**: viene sostituito un blocco di *input bits* da un unico blocco di *output bits*;
- **Espansione**: alcuni bit dell'input sono ripetuti più volte nell'output;
- **Scelta o Contrazione**: alcuni bit sono ignorati e non appaiono nell'output;
- **Shift**: vengono spostati alcuni bit a destra o sinistra.

**Management delle chiavi private**: è interessante anche lo studio di come si facciano a condividere le chiavi.
Basti pensare che per $n$ conversazioni in cui si utilizza la crittografia simmetrica è necessario utilizzare $n^2$ chiavi. In ogni caso per tutte le conversazioni sarà necessario comunicare in maniera privata. Assumiamo quindi che una *KDS (key distribution server)* condivida a priori con ogni utente una chiave diversa.
Alice e Bob vogliono stabilire una comunicazione sicura; KDS genera e distribuisce ad Alice e a Bob una *one-time session key* per essere usata durante la loro comunicazione: le successive comunicazioni fra Alice e Bob genereranno e utilizzeranno diffrenti *session keys*. Ci sono diversi altri sistemi questo, sopra descritto, è chiamato *basic*.

## 2.4 Crittografia a chiave pubblica

Anche per capire il funzionamento della crittografia a chiave pubblica verrà introdotto un protocollo e, successivamente, come si possano gestire e condividere le chiavi. Per quanto riguarda il protocollo sarà trattato il tema della firma digitale in quanto è meccanismo usato anche nelle transazioni della blockchain.

### 2.4.1 La firma digitale

La firma digirale è un meccanismo complesso in quanto risponde ad un'esigenza particolare: permette a due persone, che non si conoscono (e non possono conoscersi) e che non hanno fiducia della controparte, di identificare con certezza l'altro interlocutore.
La firma digitale gode di diverse proprietà:

[3]Processo con cui viene assicurato l'accuratezza delle trasmissioni. Funziona aggiungendo il bit 1 a sinistra della stringa in maniera da rendere il messaggio illeggibile in caso di manomissioni.

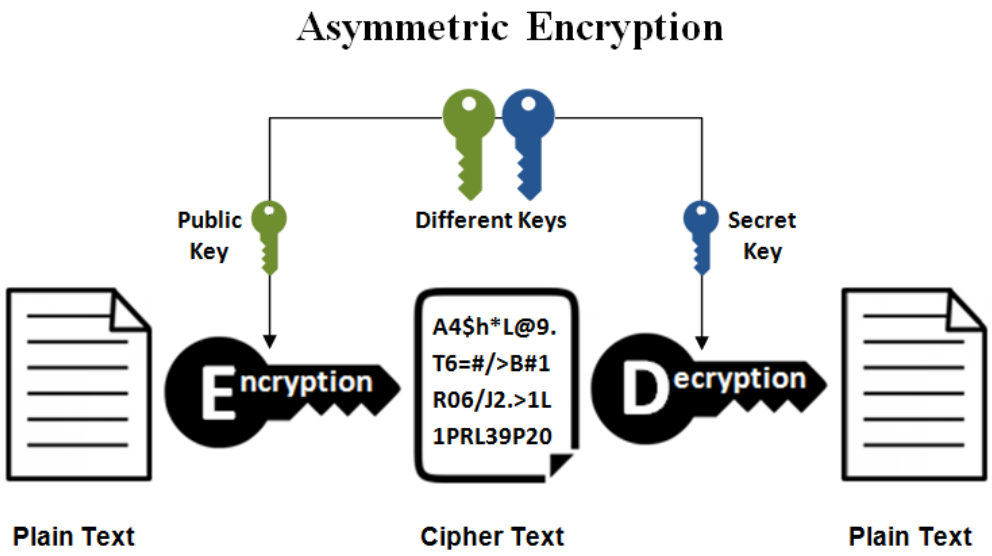


FIGURA 2.4: Funzionamento della crittografia a chiave pubblica; fonte: https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences

- **unica**: una stessa firma digitale è posseduta da una sola persona
- **non falsificabile**: non è possibile falsificarla
- **non riutilizzabile**: è legata al documento su cui è apposta
- **inalterabilità del documento**: rende il documento su cui è apposta non modificabile
- **non ripudiabile**: non può essere ripudiata da chi la utilizza

Vengono di seguito riportati i passaggi che vengono eseguiti per l'utilizzo della firma digitale.

**Data:** $m$ che è il messaggio di partenza
**Result:** il messaggio cifrato
Alice *firma* (cioè cifra) il messaggio e lo manda a Bob ;
Sign s = $D(m, k_{Alice}[priv])$ ;
Send(Alice, m, s);

**Data:** $m$ che è il messaggio cifrato
Bob verifica e salva il messaggio ;
Verify m* = $C(s, k_{Alice}[pub])$;
**if** $m$*=$m$ **then**
    yes
**else**
    no
**end**

Osservando il funzionamento di questo algoritmo sembra che occorra un cifrario commutativo: $D_k(C(m)) = C_k(D(m)) = m$ inoltre il documento firmato non è indirizzato ad uno specifico destinatario e tutti possono effettuare

la verifica.
Verifichiamo se è possibile che non sia rispettata una proprietà del protocollo:

- è falsificabile? La può generare solo una persona: Alice poiché deve conoscere $k_{Alice}[priv]$;
- è riutilizzabile? No in quanto è funzione del documento su cui è apposta;
- è alterabile? Il documento non è modificabile in quanto anche la firma andrebbe nuovamente generata;
- è ripudiabile? Non può essere ripudiata perché nessun altro potrebbe conoscere $k_{Alice}[priv]$ e quindi generarla se non Alice stessa.

Cosa succede se voglio mandare il file più volte? Nasce quindi la necessità di utilizzare una chiave che sia inalterabile; in questo caso si utilizza il *timestamp*. Che cos'è il timestamp?

**Definizione 2.6** (Definizione di timestamp)**.** Un timestamp (o marca temporale) è una sequenza di caratteri rappresentanti una data o un orario.

Il timestamp viene utilizzato di solito come strumento di verifica di avvenuta esecuzione di un certo evento dal momento in cui è possibile identificare l'evento stesso in un istante temporale preciso.

Un altro problema che Alice e Bob devono risolvere è il come fare a scambiarsi a vicenda le loro chiavi pubbliche.
Ecco che arriva il **MAC**, *Message Authentication Code*, cioè una rappresentazione breve di lunghezza fissa del messaggio che può essere generata da un solo mittente conosciuto dal destinatario.
Può essere usato per autenticare il mittente e verificare l'integrtà del messaggio. E' ottenuto attraverso una funzione hash e una chiave segreta condivisa tra il mittente ed il destinatario.
Otteniamo un MAC applicando una funzione hash alla concatenazione del messaggio $m$ con una chiave segreta $K$: $K \Rightarrow MAC(m) = f(m|k)$.
In questa maniera la funzione può solo essere ricalcolata ma non invertita.

**Management delle chiavi pubbliche**: Ci sono diversi modi per condividere la propria chiave pubblica per esempio distribuendola. A sua volta ci sono diversi tipi di distribuzione:

- **Distribuzione mediante annuncio pubblico**: l'utente rende di pubblico dominio la propria chiave pubblica ad esempio tramite mail o su una web page propria.
  - **vantaggi**: semplice, veloce e non necessita di intermediari.

  - **svantaggi**: nessuna garanzia (l'annuncio può essere facilmente alterato), inoltre l'intruso può pubblicare la propria chiave pubblica a nome di un altro utente il codissetto *man-in-the-middle attack*[4].

- **Elenco pubblico**: una directory è mantenuta da un'authority che mantiene le copie in formato <user,key>.
  Come avviene la pubblicazione delle chiavi? Ogni partecipante può presentare la propria chiave presso l'authority di persona o in modo sicuro e può aggiornarla nel medesimo modo in ogni momento.
  Per quanto riguarda l'accesso alle chiavi, viene effettuata una pubblicazione periodica della directory per esempio su un periodico o per accesso diretto alla directory tramite comunicazione elettronica per la quale occore una comunicazione autenticata e sicura.

  - **vantaggi**: la directory non può essere violata e necessita di protocolli di comunicazione sicuri per la pubblicazione e l'accesso alle chiavi;
  - **svantaggi**: necessità di un authority fidata e super partes;

- **Gestione dei certificati**: l'autenticità delle chiavi è certificata da un'autorità aggiungendo la sua firma.

  - **vantaggi**: garantisce l'identità dei partecipanti e l'attualità delle chiavi. Elimina un attacco *man-in-the-middle*: un intruso non può sostituire la chiave pubblica perché non può firmare e certificare con la chiave privata dell'autorità.
  - **svantaggi**: necessita di un'entità fidata super-partes che possa certificare in maniera sicura.

[4] É un attacco attivo e si verifica in caso di pubblicazione della chiave pubblica. Chiamiamo l'attaccante *X*, i partecipanti della conversazione *Alice* e *Bob*. Questo attacco avviene in questa maniera, *X* si intromette tra *Alice* e *Bob*. *X* si comporta in maniere differenti: come *Alice* per *Bob*, e come *Bob* per *Alice*; inoltre devia la comunicazione tra *Alice* e *B* facendola passare per se stesso. Funziona soprattutto perché Alice e Bob non hanno modo di verificare se stanno comunicando o meno con il vero interlocutore.

# Capitolo 3

# La crittografia nella blockchain

In questo capitolo verrà trattato un esempio di applicazione della crittografia: la crittografia viene applicata in ogni ambito, quello di cui ci si occupa in questa sede è quello della blockchain con alcuni riferimenti ai bitcoin[7].

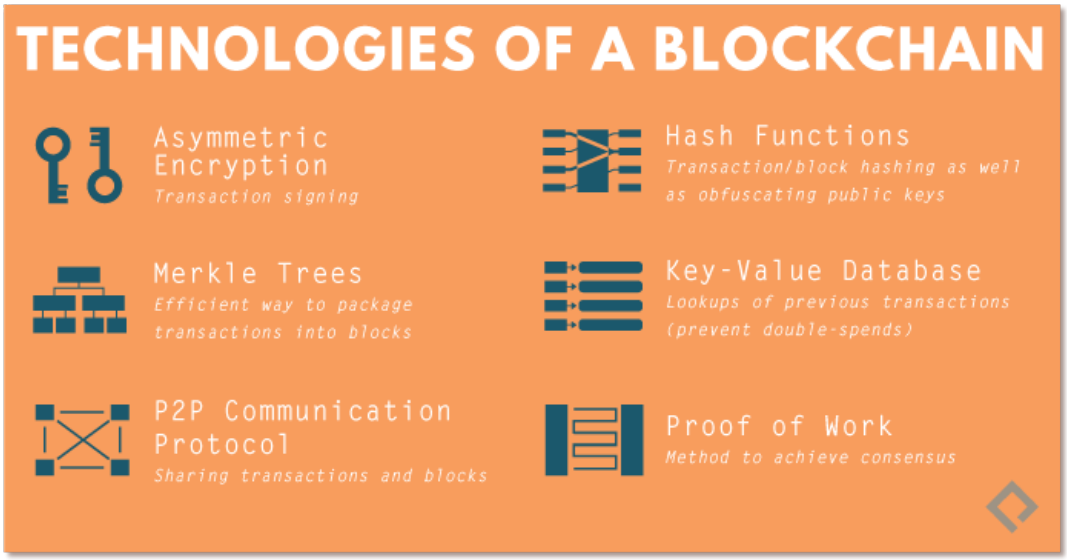


FIGURA 3.1: Caratteristiche crittografiche della blockchain; fonte: https://eng.paxos.com/the-blockchain-is-evolutionary-not-revolutionary

Come mostra la figura 3.1, le tecnologie che utilizza la blockchain per essere efficiente e sicura sono diverse:

- Crittografica asimmetrica
- Alberi di Merkle
- Protocollo di comunicazione P2P
- Funzioni hash
- Database *chiave-valore*
- Proof-of-work o consenso

## 3.1 Gli alberi di Merkle

La necessità di contenere così tante informazioni in un blocco solo e di gestire così tante operazioni fa sì che sia necessario un albero binario di ricerca.
*Un albero di Merkle*[1] è una struttura di dati usata, appunto, per la sua efficienza nel sintetizzare e verificare l'integrità di un set di dati molto elevato; si tratta di alberi binari di ricerca che contengono codici hash crittografici.
Gli alberidi Merkle sono costituiti da coppie di nodi codificate ricorsivamente finchè non ne rimanga una solo: la radice o *merkle root*. Nei nodi di questo albero non sono le transazioni ad essere memorizzate ma il loro valore cifrato: le transazioni vengono, infatti, prima cifrate e le poi inserite nelle foglie.
La funzione hash utilizzata è lo **SHA256** applicata due volte e per questo chiamata *double-SHA256*.
In un albero costituito da $N$ elementi, la funzione di ricerca costa al più $2log_2(N)$ computazioni, che si semplifica nella forma $O(log_2(N))$.
Come si costruisce un albero di Merkle? La costruzione degli alberi di Merkle è *bottom-up*. Un nodo dell'albero che è padre si compone di una sintesi dei suoi due nodi figli concatenando i due codici hash e cifrando ulteriormente la concatenazione.
Dal momento in cui l'albero di Merkle è un albero binario di ricerca necessita di avere un numero pari di nodi se quindi si verifica il caso in cui i nodi sono di un numero dispari, l'ultima foglia viene duplicata creando il cosiddetto *balanced tree*.

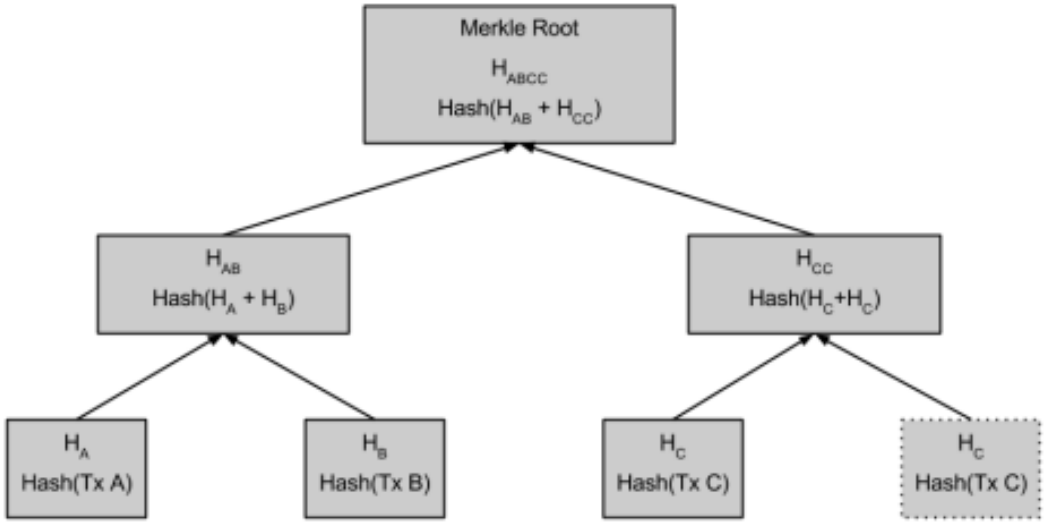


FIGURA 3.2: Albero Merkle bilanciato

Come si vede in figura, l'ultimo nodo a sinistra è la duplicazione di suo fratello.

L'efficienza degli alberi di Merkle cresce al crescere dei nodi che lo compongono. Come si può osservare nella figura 3.3 nonostante la grandezza del

[1]Il termine *albero* è usato in informatica per descrivere una struttura di dati che implementa ruoli gerarchici tra i nodi che compongono la struttura stessa. Al contrario degli alberi comuni la radice o *root* è in alto e i rami si estendono verso il basso.

| Number of Transactions | Approx. Size of Block | Path Size (Hashes) | Path Size (Bytes) |
|---|---|---|---|
| 16 transactions | 4 kilobytes | 4 hashes | 128 bytes |
| 512 transactions | 128 kilobytes | 9 hashes | 288 bytes |
| 2048 transactions | 512 kilobytes | 11 hashes | 352 bytes |
| 65,535 transactions | 16 megabytes | 16 hashes | 512 bytes |

FIGURA 3.3: Efficienza degli alberi di Merkle

blocco cresca molto rapidamente il percorso richiesto per provare l'inclusione di una transazione cresce molto più lentamente.
Un'altra caratteristica importante degli alberi di Merkle è che se un partecipante della blockchain dovesse verificare una transazione per poter decidere se introdurla nel blocco, può recuperare dall'albero solo gli header dei blocchi che lo compongono ed essere in grado, allo stesso tempo, di identificare la transazione.
L'operazione di inserimento richiede che venga recuperato una parte del percorso dell'albero, perchè la transazione venga inserita nella posizione corretta. Questa operazione può essere effettuata senza memorizzare completamente la blockchain che a volte diventa piuttosto ingombrante.

## 3.2 Il P2P nella blockchain

Il sistema della blockchain è implementato con il protocollo *peer-to-peer*, un'architettura di rete che sta alla base di molte applicazioni di internet.
Con il termine *peer-to-peer* o *P2P* si indica un sistema secondo il quale ognuno dei partecipanti della rete è trattato alla pari degli altri. Non ci sono dei nodi speciali e tutti questi partecipanti condividono la responsabilità e il ruolo sia di *server* che di *client*.
Nella rete di internet i ruoli di server e client sono in genere chiaramente distinti. Il server fornisce dei servizi e risponde alle richieste del client.
Per esempio nel www il client è rappresentato dal browser e il server è una macchina a cui il client richiede dei dati e che risponde alle richieste con delle pagine HTML contenenti i dati che il client aveva richiesto.
Il protocollo sul quale si basa questa comunicazione si chiama **HTTP**: *Hyper-Text Transfer Protocol*.

La caratteristica del protocollo P2P è quella di non avere dei server come negli altri protocolli, non ci sono servizi centralizzati, e non ci sono gerarchie nella rete.
Le reti P2P sono decentralizzate per definizioni e aperte. Decentralizzate per il fatto stesso che ciascun nodo della rete è sia server che client e aperte perchè chiunque può entrare a far parte della rete diventando a sua volta sia server che client.
Lo stesso consenso che permette alla blockchain di funzionare e di espandersi è implementato attraverso il protocollo P2P perchè è decentralizzato e aperto.
É importante dire che per quanto riguarda l'utilizzo del protocollo P2P nella

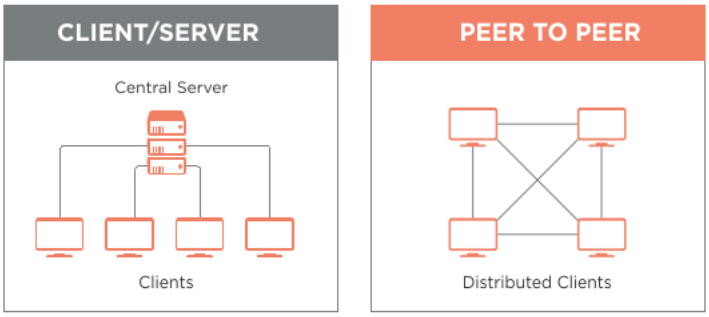


FIGURA 3.4: L'immagine mostra la differenza tra le architetture: client-server e P2P; fonte: https://bitsonblocks.net/2015/09/09/a-gentle-introduction-to-blockchain-technology/

bolckchain gli esempi concreti sono da ricercarsi negli usi che si fanno della blockchain stessa.
Per quanto riguarda i bitcoin, ad esempio, si tratta di una criptovaluta che ha ha basato la sua intera esistenza sul meccanismo del P2P tanto che la decentralizzazione del controllo, che è il principio fondante dell'intera architettura, può essere mantenuto solo basando l'intero sistema sul protocollo P2P.
Oltre al protocollo P2P sono utilizzati anche altri protocolli come *Stratum* il quale viene usato per il mining e per una gestione del proprio portafoglio virtuale. Questi protocolli insieme al P2P formano il cosiddetto *extended bitcoin network* ovvero l'insieme dei protocolli che bitcoin utilizza per la gestione delle transazioni.

## 3.3 Le funzioni hash

Una funzione hash è una funzione che mappa dei dati di lunghezza arbitraria in una stringa binaria di lunghezza fissa chiamata valore hash o codice, ma spesso viene utilizzato anche il termine inglese *message digest* o anche solo *digest*.
Le funzioni hash sono pensate per essere **non iniettive**, ovvero non invertibili: l'unico modo per ricreare i dati di input dall'output è quello di tentare una *brute force search*[2] tra i possibili input per vedere se c'è una qualche corrispondenza.

**Definizione 3.1.** Una funzione hash $f$ si dice fortemente libera da collisioni se è computazionalmente impossibile trovare un valore $x = x_1$ e $f(x) = f(x_1)$

Questa definizione risponde anche alla definizione di funzione non iniettiva infatti una funzione per non essere iniettiva deve associare a un elemento del dominio più elementi del codominio, tale che:
$f : X \to Y$ allora $a_1 \neq a_2$ implica che $f(a_1) = f(a_2)$.

[2]La *brute force search* è una metodologia per cui si cerca il un dato valore tra tutti i possibili casi.

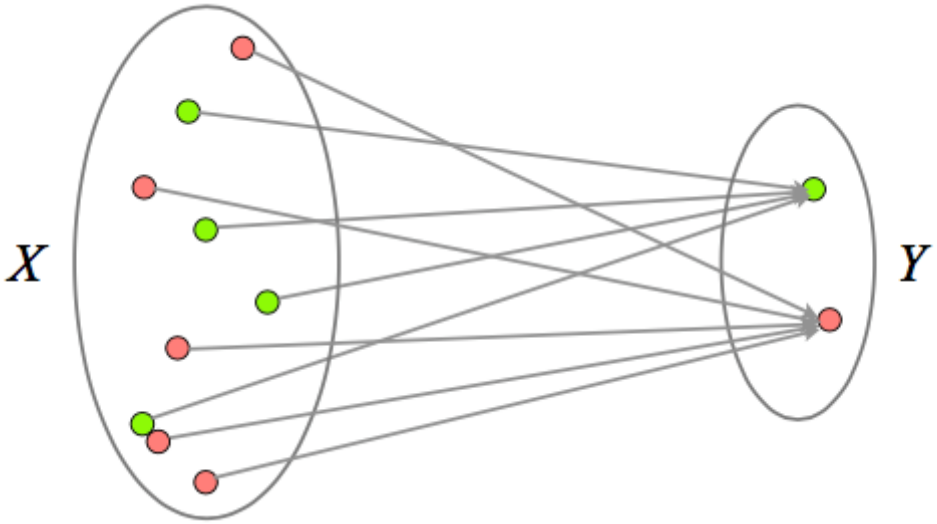


FIGURA 3.5: Immagine di una funzione non iniettiva

Naturalmente la funzione hash perfetta non esiste dal momento in cui, se nell'insieme di riferimento ci sono molti valori, la non iniettività della funzione decresce. Per ovviare a ciò bisognerebbe rendere computazionalmente perfetta la funzione scegliendo il dominio $X$ molto vasto.
Un altro esempio di funzioni hash sono le funzioni crittografiche, per esempio nei database relazionali si utilizza **MD5** che, data una stringa qualsiasi in input, ne fornisce una crittografata a 64 bit.
La funzione hash perfetta deve avere diverse proprietà:

- deve identificare univocamente il contenuto: non è possibile che due messaggi differenti, abbiano lo stesso codice hash;
- deve essere deterministico, in modo che una stessa stringa si traduca sempre nello stesso codice hash;
- deve essere semplice e veloce calcolare un codice hash da un qualunque tipo di dato;

Queste caratteristiche rendono le funzioni hash particolarmente interessanti e permettono loro di essere utilizzate largamente nei protocolli di sicurezza informatica. Ad esempio i loro utilizzi più classici sono:

- nella firma digitale
- nei codici di autenticazione dei messaggi
- nelle chiavi crittografiche dei database relazionali SQL
- rilevazione di impronte digitali
- rilevazione di dati duplicati
- identificazione di file univocamente e checksum
- rilevazione di corruzione a dabase o, in generale, di dati

## 3.4 Un database chiave-valore

Per quanto riguarda il database utilizzato dalla blockchain come sistema di memorizzazione è utilizzato il meccanismo chiave-valore.
Per spiegare questo meccanismo può essere utile questo schema:

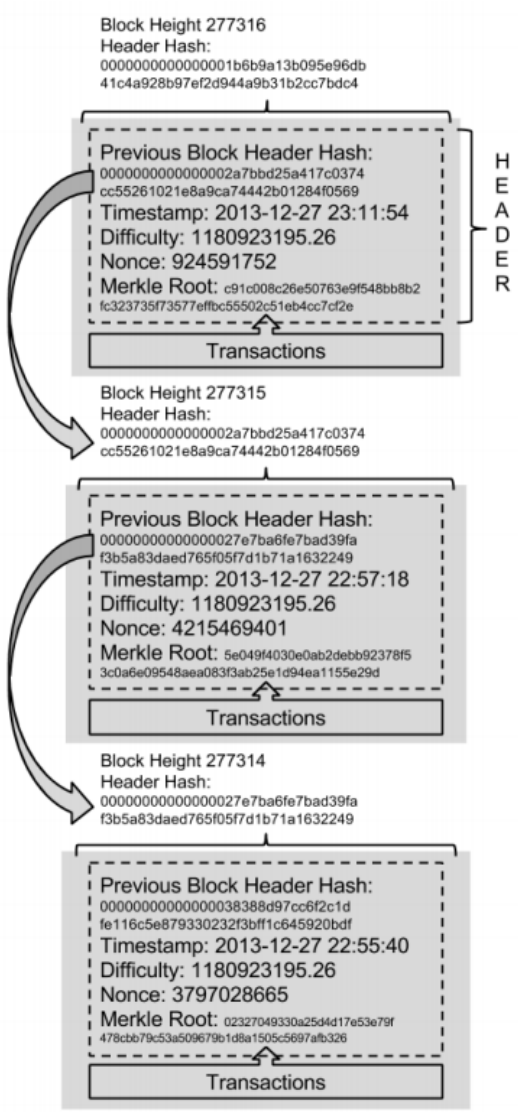


FIGURA 3.6

In questa figura è chiaro il collegamento tra i blocchi: il codice hash. Ciascun blocco, infatti, è collegato con il precedente tramite il codice hash del padre.
Pertanto secondo il principio del database chiave-valore, ogni blocco nel database è composto da codice hash del padre (la chiave), e il blocco stesso (il valore).

## 3.5 Crittografia asimmetrica nella blockchain

Come abbiamo visto la crittografia asimmetrica si compone di un sistema a due chiavi: una pubblica e una privata. Per trattare questo argomento applicato alla blockchain, verrà spiegato come vengono cifrate le chiavi private e le chiavi pubbliche nel caso specifico del bitcoin.
Il sistema dei bitcoin gestisce diverse problematiche con il sistema della crittografia a chiave asimmetrica: ad esempio il sistema del portafoglio elettronico.
Sostanzialmente le chiavi sono dei numeri: la chiave privata $k$ viene trovata praticamente random mentre quella pubblica $K$ viene trovata attraverso una funzione crittografica ellittica.

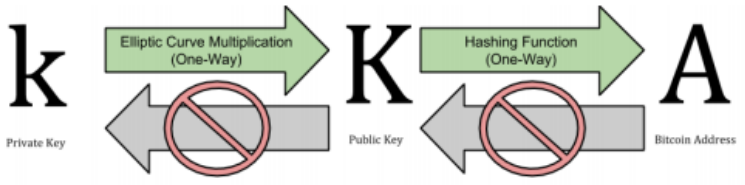


FIGURA 3.7: Sistema delle chiavi usate nella blockchain: privata e pubblica; fonte: *Mastering bitcoin*

Come detto la chiave privata è trovata tramite una funzione che restituisce un numero random. La sua funzione è quella di creare firme che sono richieste per autorizzare le transazioni.
Deve rimanere segreta sempre anche perchè condividerla equivale a condividere il controllo del proprio portafoglio; inoltre è importante che sia memorizzata in maniera sicura perchè sia protetta e non si perda facilmente.
In estrema sintesi si può dire che la chiave privata è un numero random compreso tra 1 e $2^{256}$.

### 3.5.1 Come si calcola una chiave privata

Per essere più precisi la chiave può essere qualsiasi numero tra 1 ed $n-1$ dove $n$ è una costante: $n = 1,158 * 1077$ definito nell'ordine delle curve ellittiche.
Viene quindi presa una stringa molto lunga e viene data come input a una funzione che la cifra e ne restituisce un numero a 256 bit: SHA256. Se il risultato è un valore minore di $n-1$ viene memorizzato, altrimenti viene ripetuto l'algoritmo.

**La crittografia delle curve ellittiche** è un tipo di crittografia asimmetrica basata sul problema dei logaritmi discreti.
Bitcoin usa una funzione specifica e diverse costanti matematiche secondo lo standard **secp256k1** stabilito dal *National Institute of Standard and Technology (NIST)*.
Per quanto riguarda le chiavi pubbliche sono generate partendo dalla chiave privata criptata usando il sistema della moltiplicazione per la curva ellittica sopra menzonata.

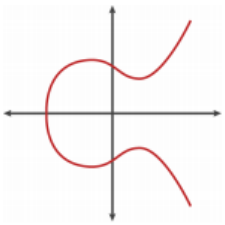

FIGURA 3.8: Grafico di una curva ellittica; fonte: *Mastering bitcoin*

Questo passaggio è fondamentale perchè di fatto si tratta di un'operazione irreversibile:

**Definizione 3.2** (Funzione della chiave pubblica)**.** $K = k * G$

Dove $K$ è la chiave pubblica, $k$ la chiave privata e $G$ la costante chiamata *generatore di punti*.
La funzione inversa, cioè trovare $k$ dato $K$, è molto difficile perchè bisognerebbe usare il sistema di brute force per ricercarla e, considerando che le chiavi private sono numeri dell'ordine di $10^{77}$, appare un calcolo impossibile[3].

### 3.5.2 Sistema di cifratura delle chiavi private

| | |
|---|---|
| Private Key (WIF) | 5J3mBbAH58CpQ3Y5RNJpUKPE62SQ5tfcvU2JpbnkeyhfsYB1Jcn |
| Passphrase | MyTestPassphrase |
| Encrypted Key (BIP0038) | 6PRTHL6mWa48xSopbU1cKrVjpKbBZxcLRRCdctLJ3z5yxE87MobKoXdTsJ |

FIGURA 3.9: Esempio di una chiave privata cifrata; fonte: *Mastering Bitcoin*

Mantenere segrete le chiavi private è una necessità importante ma anche molto difficile da attuare nella pratica. Inoltre tenere al sicuro le chiavi private è complicato anche per la possibilità che vengano perse.
Una chiave privata memorizzata in un portafoglio che è cifrato da una password dovrebbe essere sicuro ma è necessario anche che il portafoglio sia messo al sicuro con un backup.
Può darsi che si debba cambiare le chiavi per spostarle da un portafoglio a un altro, per esempio. Cosa succede se anche il backup viene perso o rubato? Ci sono quindi degli altri sistemi per rendere più sicure possibili le chivi private in maniera tale che, se rubate, comunque non possono essere usate.
Il meccanismo usato nel sistema dei bitcoin è il **BIP0038**: *Bitcoin Improvement Proposal 38*.
Questa proposta si pone come obiettivo di essere uno standard per la cifratura delle chiavi private con una *passphrase*, una password molto lunga in genere superiore ai 20 caratteri. Questa passphrase è cifrata insieme con la chiave privata con il meccanismo del *Base58Check* così che possano essere memorizzate e occupino poca memoria.

[3] La grandezza dell'universo visibile è stimato intorno ai $10^{80}$ atomi

### 3.5.3 Sistema di cifratura delle chiavi pubbliche.

Gli indirizzi bitcoin sono di solito cifrati con il sistema **P2SH**, *Pay-To-Script-Hash*. Si utilizza questo meccanismo per designare il destinatario di una transazione usando il codice hash invece della chiave pubblica.
Per questo genere di operazioni viene usata un'altra proposta inventata ad hoc: **BIP0016**, *Bitcoin Improvement Proposal 16*.
Diversamente le operazioni che si occupano di inviare fondi, le transazioni, usano il protocollo **P2PKH**, *Pay-To-Public-Key-Hash*, il quale richiede più che la presentazione della chiave pubblica e dell'autenticazione con la chiave privata per riconoscere il proprietario del portafoglio. Infatti le credenziali di cui ha bisogno sono, di solito, definite alla creazione dell'indirizzo. Un indirizzo P2PKH è creato dallo script che riporta la transazione e definisce chi può spendere il denaro derivato dalla transazione.
La particolarità di questo sistema è che per essere implementato necessita di ben due algoritmi di cifratura: uno a 256 bit e l'altro a 160 in modo da poter mantenere segretezza e occupare uno spazio ristetto: occupa infatti 20 byte. Gli algoritmi che vengono utilizzati per la cifratura dell'indirizzo pubblico sono *SHA256* e *RIPEMD160*.
A sua volta questo codice cifrato è utilizzato come input per l'algoritmo Base58Check. Questo sistema di cifratura "multipla" rende l'indirizzo pubblico praticamente impossibile da contraffare.

```
script hash = RIPEMD160(SHA256(script))
```

FIGURA 3.10: Funzione di cifratura delle chiavi pubbliche; fonte: Mastering bitcoin;

**Una falla nel sistema?** É stata tuttavia individuata una vulnerabilità nel sistema degli accessi degli utenti che utilizzano il bitcoin. Secondo quanto riporta il sito *https://latesthackingnews.com/*:
"Si nota che quando si effettua il login con sistemi come dropbox o google drive, ed è stato autorizzato il salvataggio automatico, la blockchain salverà automaticamente il file di backup nel proprio dropbox o google drive usando l'access token".
Questo è quanto è stato trovato da alcuni ricercatori e si riferisce al fatto che, si potrebbe costruire un accesso a google drive ed ottenere un URL di questo tipo: *https://blockchain.info/wallet/gdrive-update?code=YourGdriveToken*.
A questo punto se si volesse rubare il backup del portafoglio virtuale di qualcuno si potrebbe eseguire questi passaggi:

- autenticarsi con google drive su blockchain.info;
- prendere il token di google drive;
- far pervenire alla vittima il seguente URL su cui accedere: *https://blockchain.info/wallet/gdrive-update?code=GoogledriveToken*
- se la vittima accede il backup del portafoglio virtuale viene salvato sull'account di google drive dell'attaccante.

## 3.6 La fiducia nella community e il ruolo del consenso

Analizziamo quindi per ultimo il sistema di sicurezza del consenso o *proof-of-work*. Mentre in genere per quanto riguarda transazioni bancarie o operazioni molto importanti di questo genere la fiducia è riposta nell'autorità che lo governa e gestisce, nel sistema della blockchain il controllo sulle transazioni è in chiaro a tutti, è decentralizzato: le transazioni sono aperte a tutti.
Quindi in questo caso si dice che la blockchain è *permissionless*: non esiste alcuna autorità che possa negare a qualcuno di partecipare.
Esistono blockchain che invece sono di tipo *permissioned*: necessitano di un controllo e definiscono delle regole per attribuire a un dato soggetto (o a un gruppo) le operazioni, la gestione e l'autorità per quanto riguarda le operazioni:

- accessi
- controlli
- autorizzazioni
- aggiunta di transazioni

Grazie alle *permissioned blockchain*, le pubbliche amministrazioni, le banche, le imprese possono coniugare le necessità di controllare le modalità e le identitià di chi esegue le transazioni sfruttando congiuntamente la potenza della blockchain.
Si può venire a creare un sistema non solo distribuito a livello mondiale ma, com'è auspicabile, molti sistemi, distribuiti per le caratteristiche della blockchain nella forma della sua immutabilità, della sicurezza e della garanzia di stabilità a favore, quindi, di ciascuno secondo le sue esigenze.
Il meccanismo della blockchain fornisce un modo per creare transazioni **irrevocabili**. In questo modo è assicurato che le transazioni siano definitive: non si possono nè annullare nè modificare.

### 3.6.1 La *proof-of-work*

Il consenso, nella struttura intrinseca della blockchain, riveste un ruolo fondamentale dal momento in cui perfino tutta la blockchain viene aggiornata solo dopo aver ottenuto il consenso da ogni nodo: solo dopo il consenso vengono aggiornati i nodi uno ad uno con la versione appena validata.
Ma come si passa dal concetto di risoluzione di un problema al concetto di fiducia, di consenso? E cos'è il consenso distribuito?
Le risposte a queste domande sono in realtà semplici dal momento in cui il meccanismo è intrinseco alla blockcain.
Quando si vuole aggiungere una transazione al blocco e lo si chiude è necessario risolvere un problema matematico, un calcolo, che risulta essere complicato. Questi calcoli servono per stabilire con certezza l'autenticità delle

transazioni e inoltre servono per creare la fiducia nella community. I partecipanti alla validazione non si conoscono tra di loro cosicchè la *proof of work* rappresenta un modo concreto per costruire un rapporto di fiducia.
Questa fiducia ha due significati:

- fiducia verso il sistema;
- fiducia verso i nodi.

Dove, ovviamente la fiducia verso il sistema è quella che ciascun partecipante ripone nel sistema della blockchain; mentre la fiducia verso i nodi è quella fiducia condivisa tanto importante ai fini della validazione del blocco e, quindi, di tutta la catena che ciascun nodo ripone negli altri.

# Capitolo 4

# Blockchain

In questo capitolo verrà analizzata la blockchain[4]. Nonostante sia un argomento esteso e variegato si proverà a tracciarne un profilo sotto il punto di vista tecnologico; inoltre verranno introdotti alcuni dei suoi usi.

## 4.1 Un'introduzione alla blockchain

Per spiegare cosa sia la blockchain bisogna innanzitutto introdurre alcuni concetti che in genere non hanno molto in comune:

- libro mastro
- fiducia
- crittografia
- trasparenza
- condivisione

Tutti questi concetti e caratteristiche assieme compongono una struttura stabile, completa, ricca di potenziale: la blockchain. Una delle caratteristiche che appariranno evidenti in questo capitolo sarà che la blockchain può sfruttare appieno internet come rete di reti composte di nodi; ma anche come comunità unita e contemporaneamente in competizione per il conseguimento di un obiettivo comune.
Per capire cosa sia la blockchain un modo interessante sarebbe conoscere qual è il senso comune e cosa pensano gli addetti ai lavori: per alcuni è addirittura la nuova generazione di internet.
Alcuni ritengono che la blockchain possa assumere anche un valore quasi politico, come piattaforma che consente lo sviluppo e la concretizzazione di una nuova forma di democrazia, decentralizzata e in grado di garantire a tutti la possibilità di verificare e disporre di una totale trasparenza e di dare vita ad archivi condivisi da tutta la comunità e che, proprio per questo, risultano inalterabili, immodificabili e dunque immuni da corruzioni e manomissioni.
Alcuni di questi concetti sono stati già spiegati precedentemente, altri verranno spiegati in questo capitolo.
**Il libro giornale** è l'insieme delle attività di gestione dell'azienda raccolte attraverso la partita doppia secondo l'ordine cronologico in cui le operazioni,

uscite ed entrate, si sono verificate.
Il modo in cui sono collegate le operazioni avviene secondo il periodo amministrativo cui esse fanno riferimento. Bisogna sicuramente fare attenzione al fatto che parlando della raccolta di gestione secondo ordine cronologico, ogni singola operazione venga rappresentata secondo la sua personale natura.
**Il libro mastro** o **ledger** è invece la raccolta delle operazioni di gestione dell'azienda rappresentate in ordine sistematico. Il libro mastro quindi si compone di un insieme di rappresentazioni finanziarie ed economiche, della raccolta sistematica dei singoli conti della contabilità generale raggruppati secondo la loro natura.
Uno dei modi in cui la blockchain può essere considerata, ritenendola un'innovazione, è quella di rappresentare in una maniera nuova le transazioni, creando di fatto una nuova rete: una *internet delle transazioni*.

Riguardo a questo Domenico Gammaldi, direttore superiore del servizio supervisione sui mercati e sul sistema dei pagamenti alla banca d'Italia, si è espresso sottolineando che "Se l'innovazione porta benefici pubblici, il regolatore la deve facilitare". Inoltre al tema ha dedicato un gruppo interdisciplinare, focalizzato soprattutto sui pagamenti.
Le grandi banche internazionali prestano grande attenzione al tema blockchain e ne stanno sperimentando i vantaggi. Stesso interesse si manifesta in Italia, anche se ancora in poche sperimentano, secondo l'indagine dell'Osservatorio Digital Finance, realizzata con interviste al top management di oltre 40 attori in ambito finanziari fra scettici e titubanti si colloca ad esempio:

- cheBanca!, che aspetta siano le grandi banche a fare i primi passi;
- Banca Mediolanum guarda con attenzione al fenomeno che sta studiando da circa un anno;
- Banco Posta, si colloca a metà fra osservatore e sperimentatore;
- Intesa Sanpaolo invece si pone la sfida di passare dallo stadio proof of concept a prodotti capaci di offrire ritorni[1].

## 4.2 Definizione di blockchain

Talvolta la blockchain viene confusa con il bitcoin ovvero con un utilizzo specifico e, in particolare, con quella che sta alla base della digital currency o criptomoneta. L'errore è quello di confondere un elemento dell'insieme con l'insieme stesso.
Forse, proprio per questa ragione, l'intera blockchain viene associata al concetto di digital currency e/o di pagamento. In realtà, come vedremo, la blockchain può essere utilizzata come piattaforma di pagamento o come per creare delle criptomonete, ma non è necessariamente limitata a questo campo di applicazione, anzi le criptomonete sono solo uno degli utilizzi possibili che offre la blockchain.
Com'è evidente la blockchain si presta a essere interpretata; per questo, com'è naturale, ne esistono diverse interpretazioni e diverse definizioni.

### 4.2.1 Un database per le transazioni

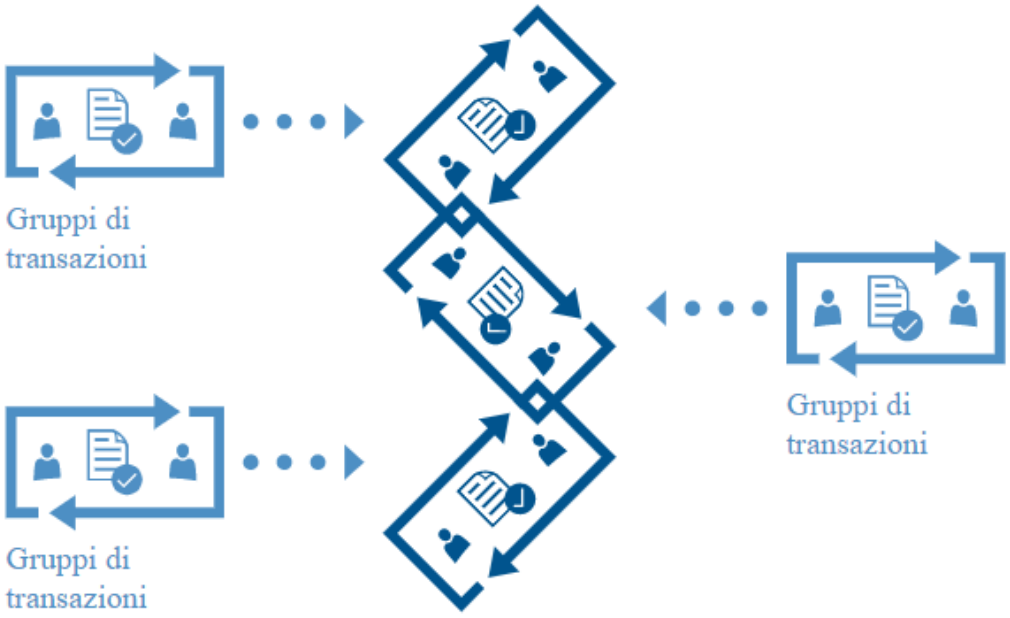


FIGURA 4.1: Un database per le transazioni;

La blockchain è una tecnologia che permette la creazione e il coordinamento di un complesso database distribuito per la sincronizzazione di transazioni condivisibili tra molti nodi di un grafo o di una rete.
Si tratta di un database strutturato in blocchi che sono tra loro collegati in modo che ciascuna transazione avviata sulla rete sia validata dalla rete stessa o meglio dai nodi stessi. In estrema sintesi la blockchain è rappresentabile con una catena di blocchi che contengono e coordinano diverse transazioni.
Ogni nodo ha diversi compiti da svolgere:

- vedere le transazioni degli altri nodi
- controllare che tutte le transazioni siano coerenti
- approvare le operazioni di ciascuna transazione

Tutto questo meccanismo crea una rete che permette la tracciabilità di tutte le transazioni.
Ciascun blocco a sua volta è anche un archivio per tutte le transazioni poichè tutto lo storico di ciascuna transazione viene scaricata da ogni nodo perchè riesca a lavorarla. Per essere approvate dalla rete e per essere presenti su tutti i nodi della rete devono essere assolutamente immodificabili se non attraverso la riproposizione degli stessi a tutta la rete e solo dopo aver ottenuto nuovamente l'approvazione: per questo sono immutabili.
Oltre alla immutabilità l'altra fondamentale caratteristica che conferisce sicurezza alla blockchain è l'unione profonda con la crittografia che garantisce la massima sicurezza di ogni transazione, come abbiamo già visto.

### 4.2.2 Il nuovo libro mastro?

La blockchain è una implementazione del *distributed ledger*. Il distributed ledger è un concetto che può essere interpretato come evoluzione dal *centralized ledger* passando per il *decentralized ledger*.
Vogliamo, ora, spiegare questi concetti a partire dal centralized ledger:

- **Centralized Ledger**: la logica è centralizzata e rappresenta il tradizionale centralized ledger con un rapporto centralizzato *one-to-many*, dove tutto deve essere gestito facendo riferimento a una struttura o autorità centralizzata.
  Nel centralized ledger la fiducia è nell'autorità, nell'autorevolezza del soggetto o sistema che rappresenta il fulcro dell'organizzazione.
  Sostanzialmente tutto deve passare dal centro, il quale contiene le informazioni necessarie e dirige i lavori per chiunque ne abbia necessità.

- **Decentralized Ledger**: Il decentralized ledger ripropone la logica della centralizzazione a livello locale con alcuni centri organizzati come satelliti sempre nella forma di *one-to-many*.
  Questa organizzazione supera il centralized ledger perchè non c'è più un unico soggetto centrale il quale viene sostituito da tanti soggetti locali.
  Dal punto di vista degli utilizzatori, la fiducia è riposta in un soggetto centrale, logicamente più vicino, ma comunque centralizzato.

- **Distributed Ledger**: il vero cambiamento è rappresentato dal distributed ledger, ovvero da una logica distribuita dove non esistono più centri, nè locali nè tantomeno centralizzati, e dove la logica di governance e fiducia è stabilita e costruita attorno al concetto di fiducia tra soggetti, cioè quei soggetti che sono si utilizzatori ma occupano un ruolo di governance.
  In questo modello, nessuno ha la possibilità di prevalere soprattutto perchè il processo decisionale passa rigorosamente attraverso un processo di costruzione del consenso che si esplica nel consenso generale di tutti gli utenti per la loro duplice funzione.

## 4.3 Cos'è la blockchain

La blockchain è, quindi, un database decentralizzato, crittografato, distribuito che gestisce le informazioni riguardanti i blocchi che compongono la catena. Inoltre crea i presupposti perchè ciascun nodo possa gestire le informazioni provenienti dagli altri nodi.
La cosa più interessante è che la sua sicurezza sta nel fatto che tutte queste informazioni siano accessibili a tutti i partecipanti: tutti coloro che fanno parte della catena avranno le informazioni di ciascun nodo, potranno verificarne l'identità e avranno lo storico di ciascuna transazione, che poi sono le unità di misura atomiche che costituiscono i blocchi.

Pertanto ciascuna informazione è alla portata di tutti: è pubblica e, nonostante questo, sicura.

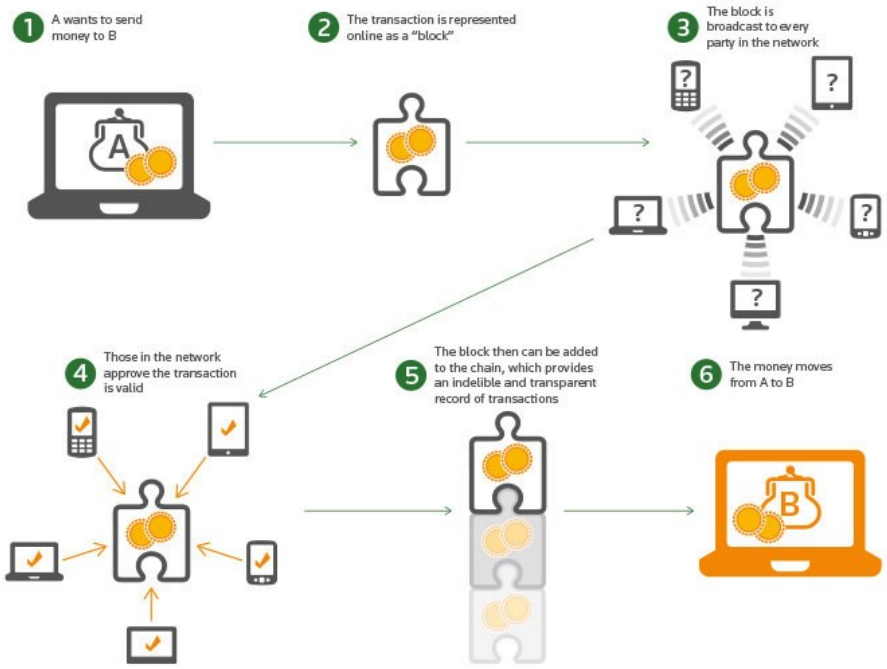


FIGURA 4.2: Una rappresentazione del funzionamento della blockchain; fonte: http://www.blockchain4innovation.it/esperti/blockchain-perche-e-cosi-importante/

La blockchain costituisce un nuovo protocollo di comunicazione utile alla creazione e alla gestione di una tecnologia che nasce sulla logica del database distribuito.

### 4.3.1 Cos'è un database distribuito?

La blockchain è organizzata per aggiornarsi automaticamente su ciascun client che partecipa alla catena. Ogni operazione effettuata deve essere confermata da tutti i singoli anelli della catena per esaminare un pacchetto di dati per volta, definiti a chiave privata, che viene utilizzato per firmare le transazioni garantendo l'identità di chi le ha autorizzate.
La blockchain si struttura su un database distribuito o, secondo alcuni, è un vero e proprio database distribuito. Quindi, per capire bene cosa sia è necessario capire meglio cos'è un database distribuito.
Un database distribuito, come accade per tutti i databse, è gestito da un *Database Management System* (DBMS) nel quale le strutture di dati non sono rese persistenti su una medesima macchina, ma su più calcolatori chiamati anche nodi. Questo vuol dire che il database può essere dislocato in più computer che si trovano nello stesso posto fisico, oppure distribuito in una rete di computer colleagati tra loro tramite un sistema distribuito.

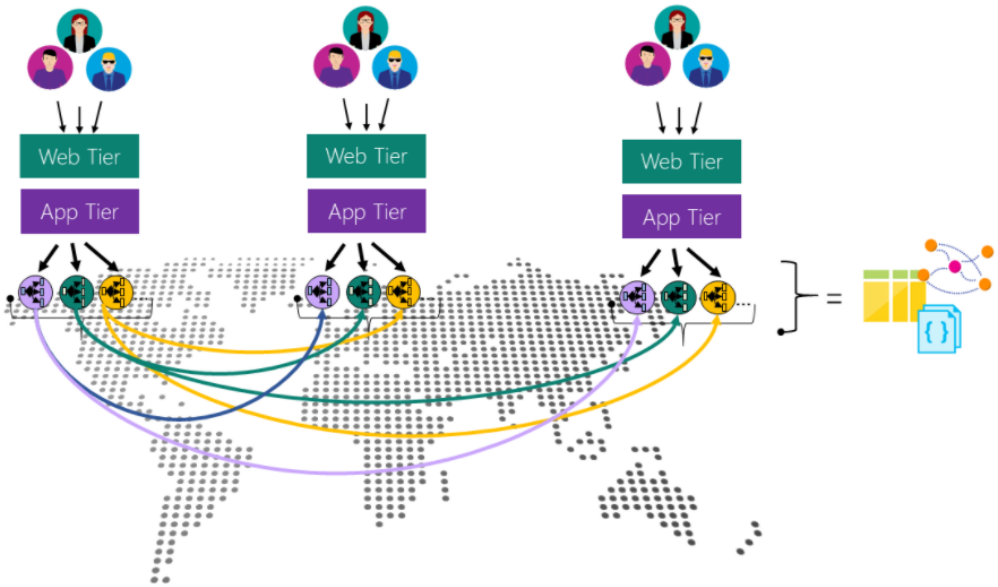


FIGURA 4.3: Azure come implementazione di database distribuito; fonte: https://docs.microsoft.com/it-it/azure/cosmos-db/distribute-data-globally

Si ottiene così uno schema logico comune a tutti i calcolatori e inoltre un insieme di schemi logici locali che dipendono da quello globale. Per passare dai singoli schemi locali a quello più generale, globale, si effettuano delle operazioni chiamate frammentazioni che sono delle vere e proprie trasformazioni a livello di dati.
Il DBMS serve ad assicurare che la struttura dei dati sia stata costituita in maniera intelligente, efficace ed efficiente. Inoltre provvede a garantire rapidità di accesso alle informazioni di ciascun nodo.
Naturalmente rispetto a un database situato localmente o su un singolo server disposto in un unico luogo fisico le considerazioni sull'ottimizzazione del funzionamento sono diverse e più complicate; proprio per questo il DBMS di un database distribuito si impegna a garantire:

- **affidabilità**: i dati devono essere conservati, per quanto possibile, anche in caso di malfunzionamento;
- **riservatezza**: i dati devono essere protetti, per impedire che il database venga danneggiato o corrotto da interventi non autorizzati.

La tecnica di suddivisione dei dati su differenti macchine può essere scelta tra due possibilità: *verticale* o *orizzontale*.
In entrambi i casi la tecnica di suddivisione, o distribuzione, deve necessariamente garantire due condizioni: *completezza* e *ricostruibilità*. Per definire i concetti di completezza e ricostruibilità prendiamo il caso in cui su un db distribuito, ci sia un'entrità (una tabella per esempio) e anche un suo frammento, allora avremo che:

- completezza: ogni elemento della tabella deve essere presente in almeno un frammento;

- ricostruibilità: presi tutti i frammenti deve essere sempre possibile ricostruire la tabella iniziale

**Frammentazione orizzontale** Nella frammentazione orizzontale ogni frammento consiste in una sequenza di tuple di una relazione, mantenendo l'intero schema della relazione in ogni frammento. Solitamente in questo caso i dati non vengono replicati, così è possibile ricostruire la base di dati semplicemente unendo i vari frammenti. I livelli di trasparenza sono parametri che definiscono quanto il programmatore che scrive query per una base di dati distribuita su differenti DBMS può astrarre il suo codice. Questo è dovuto alle differenti tecniche di approccio alla frammentazione, query, ecc. dei vari DBMS.

**Frammentazione verticale** Nella frammentazione verticale le relazioni vengono divise per insiemi di attributi. In questo caso è necessario replicare la chiave primaria in ogni frammento, così da permettere la ricostruzione attraverso una JOIN.

**Livelli di trasparenza:** Possiamo identificare quattro principali livelli di trasparenza:

- di frammentazione: il programmatore non conosce (nè gli interessa) come e quando la base di dati è frammentata e distribuita;
- di allocazione: il programmatore conosce la struttura dei frammenti, ma non l'allocazione fisica dei dati;
- di linguaggio: il programmatore conosce sia la struttura dei frammenti che la loro allocazione ed è costretto a indicarle manualmente, però è avvantaggiato dall'avere un solo linguaggio per tutti i DBMS. Si noti che se la maggior parte dei DBMS utilizzi SQL come linguaggio, ogni DBMS ha un proprio dialetto che complicherebbe lo sviluppo in caso di DBMS diversi;
- nessuna trasparenza

Si rende necessario adottare tecniche per garantire una verifica sulla concorrenza anche in ambito distribuito. Fortunatamente le proprietà CID rimangono valide anche in ambito distribuito:

- consistenza: se ogni sottotransazione preserva l'integrità locale, anche i dati globali saranno coerenti e consistenti;
- isolamento: se ogni sottotransazione è 2PL o TS allora la transazione è globalmente serializzabile;
- persistenza: se ogni sottotransazione utilizza il log in formato corretto, la persistenza dei dati globali è garantito.

Per garantire l'atomicità tutti i nodi devono decidere ugualmente se effettuare un commit o un abort, tenendo anche conto dei possibili errori di comunicazione che possono avvenire (caduta di un nodo, downlink, ack mancante, ecc.). Per questo si rendono necessari i cosiddetti protocolli di commit[3].

### 4.3.2 Il blocco

Come abbiamo detto, il blocco è un elemento centrale della blockchain ma com'è fatto questo blocco? Non è altro che una serie di informazioni riguardanti transazioni. Quese stesse transazioni che il blocco contiene a loro volta lo costituzcono.

| Size | Field | Description |
|---|---|---|
| 4 bytes | Block Size | The size of the block, in bytes, following this field |
| 80 bytes | Block Header | Several fields form the block header (see below) |
| 1-9 bytes (VarInt) | Transaction Counter | How many transactions follow |
| Variable | Transactions | The transactions recorded in this block |

FIGURA 4.4: Composizione di un blocco

Come possiamo vedere nella figura 4.4 un blocco è costituito da una serie di attributi:

- informazioni sula dimensione
- header
- contatore di transazioni
- informazioni sulle transazioni

Queste informazioni possono rendere un blocco un oggetto anche molto grande, basti pensare che le informazioni sulle transazioni possono essere anche 1000 volte più ingombranti di quelle dell'header.

| Size | Field | Description |
|---|---|---|
| 4 bytes | Version | A version number to track software/protocol upgrades |
| 32 bytes | Previous Block Hash | A reference to the hash of the previous (parent) block in the chain |
| 32 bytes | Merkle Root | A hash of the root of the Merkle-Tree of this block's transactions |
| 4 bytes | Timestamp | The approximate creation time of this block (seconds from Unix Epoch) |
| 4 bytes | Difficulty Target | The proof-of-work algorithm difficulty target for this block |
| 4 bytes | Nonce | A counter used for the proof-of-work algorithm |

FIGURA 4.5: Composizione dell'header di un blocco

Come possiamo vedere l'header di un blocco è composto da differenti attributi esso stesso:

- versione
- informazioni sul blocco precedente
- informazioni sulla costituzione del blocco
- Timestamp della creazione del blocco
- Grado di difficoltà della *proof-of-work*
- Un contatore per la *proof-of-work*

Spiegando brevemente cos'è un *Nonce* introduciamo un concetto fondamentale, che verrà approfondito in seguito, per la blockchain: **la crittografia**. Infatti il *Nonce* non è altro che un numero casuale che si può usare solo una volta. L'utilità di questo numero casuale sta nel fatto che si usa per protocolli di autenticazione per assicurare che i dati che sono già stati scambiati non vengano, poi, riutilizzati in futuro.

### 4.3.3 Come si collegano i blocchi

I nodi della blockchain si collegano tra di loro come si collegano anelli di una catena, e il punto di collegamento è quasi come fosse una *chiave parentale* che ciascun blocco ha del blocco che lo precede: questa chiave è il codice hash.
Dal momento in cui ogni blocco ha una copia locale dell'intera catena è anche costantemente aggiornato ad ogni creazione di nuovi blocchi. Non appena un nodo riceve blocchi nuovi dalla rete li valida immediatamente e crea con loro un collegamento alla bolckchain esistente.
Per stabilire questo collegamento ciascun nodo dovrà necessariamente esaminare il blocco, nella fattispecie l'header del blocco, e quindi verificare il codice hash del blocco precedente.
Per esempio[1] se un nodo ha 277314 blocchi nella sua copia locale e ha un header con il codice hash di
0000000000000027*e*7*ba*6*fe*7*bad*39*faf*3*b*5*a*83*daed*765*f*05*f*7*d*1*b*71*a*1632249,
il nuovo node che riceve un blocco lo parsa in questa maniera:

Quando il nodo esamina questo nuovo blocco trova il campo *"previousblockhash"* che ha come valore il codice hash del blocco precedente, o *parent block*.
Questo codice hash è conosciuto dal nodo e dall'ultimo blocco della catena in posizione 277314. Perciò questo nuovo blocco è un figlio dell'ultimo blocco della catena il quale andrà ad allungare questa catena.
Infine il nodo aggiungendo questo nuovo blocco alla fine della catena la estende per renderla di 277315 blocchi totali.

[1] Questo esempio è tratto dal libro *Mastering bitcoin*

```
{
    "size": 43560,
    "version": 2,
    "previousblockhash":
        "000000000000000027e7ba6fe7bad39faf3b5a83daed765f05f7d1b71a1632249",
    "merkleroot":
        "5e049f4030e0ab2debb92378f53c0a6e09548aea083f3ab25e1d94ea1155e29d",
    "time": 1388185038,
    "difficulty": 1180923195.2580261,
    "nonce": 4215469401,
    "tx": [
        "257e7497fb8bc68421eb2c7b699dbab234831600e7352f0d9e6522c7cf3f6c77",
        #[... many more transactions omitted ...]
        "05cfd38f6ae6aa83674cc99e4d75a1458c165b7ab84725eda41d018a09176634"
    ]
}
```


FIGURA 4.6: Forma di un blocco; fonte: *Mastering bitcoin*

## 4.4 I libri mastro distribuiti

Anche nell'era dell'informatica, i libri mastri sono stati utilizzati in modo centralizzato. Per la gestione di questi libri mastri, qualcuno fisicamente gestisce il data entry, qualcun altro i sistemi, qualcun altro ancora gestisce l'estrazione dei dati e/o la loro elaborazione.
Di certo l'informatizzazione dei sistemi ha reso tutto più veloce efficiente ma si porta dietro una pesante catena di lavoro, una struttura ancora complessa e ingombrante.
Al di là dei motivi politici o di contratto per cui queste operazioni sono fatte in questa maniera, esisteva uno scoglio tecnico non banale da superare; ma con l'avvento della blockchain tutto questo può cambiare.

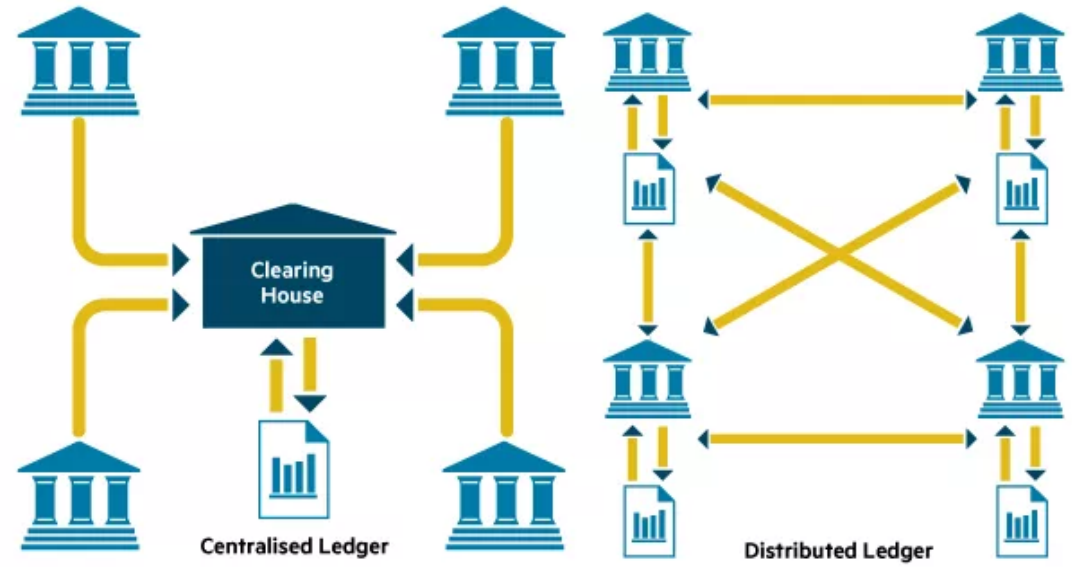


FIGURA 4.7: Centralised e distributed ledgers;
fonte: www.linkedin.com/pulse/block-chain-technology-what-holds-future-steffi-mathew/

Come si vede in figura 4.7, c'è una differenza sostanziale tra il libro mastro centralizzato e quello distribuito, dal momento in cui per il centralizzato è obbligatorio passare per un punto unico che simsta poi le operazioni; per quanto riguarda, invece, quello distribuito, ciascun nodo è responsabile delle operazioni e contemporanemante tutti sono coinvolti nella validazione di ogni singola operazione.

### 4.4.1 Innovazioni del "distributed ledger"

Imprese, banche e pubbliche amministrazioni hanno utilizzato i libri mastro per gestire la contabilità e le informazioni delle transazioni.
Per ogni cambiamento e ogni volta che avveniva una transazione si procedeva a una modifica del libro mastro tramite un'autorità centrale che ha proprio il compito di gestire il libro mastro centrale. Con questa organizzazione strutturata a passaggi, gli uffici di banche, pubbliche amministrazioni etc. avrebbero potuto sapere con certezza una serie di informazioni riguardanti gli attori di una transazione o i possessori di certi beni.
Questa verifica, per quanto alle volte lunga e macchinosa, comportava una certa rigidità sulla quale gli istituti autorizzati potevano basarsi per la verifica delle transazioni e in particolare sulla identità degli attori, appunto, e quindi sulla loro legittimità; inoltre questo sistema mette a disposizione degli istituti legittimati alla verifica, anche informazioni sullostato dei beni (immobili, etc.) assicurando casi anche banali riguardanti lo *stato* del bene: ad esempio se una casa è già stata venduta, se è all'asta, etc.
Il concetto che sta dietro al libro mastro centralizzato è il fatto che tutti ripongono fiducia nelle istrituzioni che amministrano ed esaminano proprio questo libro mastro, considerando anche l'autorevolezza che, notoriamente, hanno banche, pubbliche amministrazioni e, in generale, queste istituzioni. Bisogna considerare il fatto che è ormai nell'uso comune il riporre fiducia verso questi istituti tanto che spessp lo si da per scontato.
Dal momento in cui esiste questo rapporto di fiducia, le persone possono comprare e vendere anche senza essersi mai incontrate prima e, ormai sempre di più, senza conoscere la controparte e quindi senza nutrire verso di essa alcuna fiducia; ma tutto questo è possibile perchè i soggetti nutrono fiducia verso il soggetto terzo che gestisce questo libro mastro e garantisce lui per tutti.
Le istituzioni che gestiscono il libro mastro, gestiscono anche le informazioni che riguardano le controparti di uno scambio. Per esempio nel caso delle banche solo i correntisti possono accedere alle funzioni del proprio conto corrente e solo di quello; ma se devono acquistare un bene è la banca che garantisce per loro nel senso che si occupa di assicurare che hanno effettivamente la somma necessaria all'acquisizione o che siano persone che esistono veramente, tutto questo senza permetere ad altri che non siano i correntisti di accedere al conto: questo è un punto focale.
La digitalizzazione ha senz'altro cambiato il modo in cui si utilizza il libro mastro, infatti la digitalizzazione ha fatto sì che questo processo, o questa serie di processi, si evolvesse e diventasse più veloce; tuttavia qesti processi

sono ancora ancorati al vecchio concetto di central ledger in qualche modo.
Nonostante abbiano aumentato l'efficienza e l'efficacia delle operazioni, il vecchio concetto di libro mastro non è stato messo in discussione. Gli strumenti e le procedure sono rimaste aderenti a una struttura centralizzata, chiusa e riservata: non è cambiata la *governance*, le regole di accesso etc. ma, come abbiamo visto, questi concetti sono fondamentali per lo sviluppo dell'economia anche basilare.

### 4.4.2 Un nuovo concetto: il libro mastro è di tutti

Il primo passo tra la gestione tradizionale e quella innovativa, introdotta dalla blockchain, del libro mastro è data dal fatto che i libri mastro sono diversi e che sono accessibili a praticamente tutti. Il secondo riguarda il fatto che tutti possono effettuare una transazione o modificarne una già esistente: in entrambi i casi la richiesta specifica sarà effettuata se e solo se la maggior parte degli utenti accettano di attuarla.
Questa gestione della verifica sull'affidabilità delle transazioni rende la blockchain un sistema "democratico", almeno secondo chi la pensa così. C'è però da dire che in questo sistema non è la democrazia la base, ma un concetto di *sharing*, come vedremo.
La richiesta di ciascuna transazione, sarà accettata o meno in relazione del fatto che i partecipanti siano d'accordi sulla legittimità della stessa, ovvero se i partecipanti sono d'accordo sulla legittimità allora la richiesta verrà effettuata mentre l'autorizzazione dell'operazione singola, prescinde da questi controlli. Tutti i partecipanti, tutti gli utenti, sono invitati a controllare la legittimità delle richieste e, quindi, l'identità delle controparti.
Naturalmente la verifica è possibile se tutti i partecipanti possono controllare che la richiesta provenga da una persona che abbia un'autorizzazione per svolgerla, altrimenti la verifica si fa complessa e, nella maggior parte dei casi, non viene autorizzata.
Seguendo questo filo conduttore, il discorso ci porta a pensare a come sia possibile effettuare questi controlli, infatti non è banale riconciliare questi pezzi; basti pensare a come sia possibile che tutti i partecipanti possano in qualche maniera accedere a informazioni riguardanti le parti in causa per qualunque tipo di transazione. Al di là dello scoglio tecnico, si introduce un quesito giuridico non indifferente.
La risposta a questa domanda è, invece, piuttosto semplice; infatti questi controlli vengono fatto in maniera del tutto affidabile e automatizzato per ciascun utente; ogni operazione, ogni transazione contribuisce a creare un sistema rapido e sicuro.
Si viene a creare una rete, un grafo il quale per il fatto di basarsi su un database distribuito che è installato sulle macchine di tutti i partecipanti, poichè tutti i partecipanti hanno una copia di ogni operazione, è anche in grado di resistere a eventuali manomissioni e di mantenere le informazioni nonostante eventuali problemi di rete.

### 4.4.3 Un nuovo, grande libro mastro

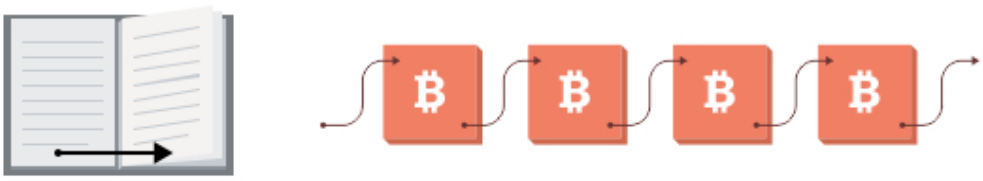

FIGURA 4.8: Raffigurazione del passaggio dal libro mastro alla blockchain; fonte: https://bitsonblocks.net/2015/09/09/a-gentle-introduction-to-blockchain-technology/

Quando c'è necessità che una nuova transazione venga approvata e quindi, successivamente, registrata nel database distribuito, questa transazione viene unita ad altre per andare a creare un *blocco*.
Questo blocco è quindi composto da diverse transazioni unite insieme dal momento in cui sono state eseguite in un certo lasso di tempo, e contiene informazioni sulle sue transazioni e informazioni sul blocco precedente a cui si lega.
L'insieme dei blocchi va a formare ovviamente la catena il cui storico è posseduto da tutti i partecipanti (idealmente tutti gli utenti).
Ovviamente un blocco deve essere validato prima di essere aggiunto alla catena, ma per far girare tutto il meccanismo è necessario che ogni volta che viene costituito un nuovo blocco venga contestualmente portato a termine un complesso calcolo crittografico.
Questi calcoli crittografici sono dei veri e propri problemi matematici che vengono risolti solo grazie ad un impegno non indifferente; questa operazione si chiama *mining*. Un *miner* è quindi colui che si impegna a svolgere operazioni di mining; il suo lavoro è fondamentale per la gestione delle blockchains.
Naturalmente per far si che questo meccanismo sia condiviso, portato avanti da più persone possibili è quindi esteso a tutti cioè tutti possono essere dei miners.
A questo punto è necessario chiarire come sia possibile giungere a una soluzione del problema matematico e se tutti cercano di risolvere uno stesso problema, come si decide chi lo ha risolto per prima. Infine bisogna capire se c'è un incentivo affinchè i miners sfruttino la potenza di calcolo dei loro computer.
Il meccanismo che si viene a creare tra i minsers è una vera e propria competizione; inoltre dal momento in cui si tratta di un impegno non indifferente che implica un costo questo impegno è ricompensato.
Esistono diversi modi in cui si può creare una blockchain, una differenzazione può essere quella tra *blockchain pubblica* e *blockchain privata*. Naturalmente per questi due tipi di blockchain esistono due diversi modi di interpretare il ruolo dei miners.
Nel caso della blockchian privata il ruolo del miner è affidato all'autorità che

attiva la blockchain; in quello della blockchain pubblica, chiunque partecipi alla blockchain può espletare le funzioni di miner.
Per le blockchain pubbliche le ricompense sono definite da regole stabilite al momento della costituzione della catena. In genere il primo che si aggiudica il titolo di creatore di blocco viene ricompensato con una specie di rimborso spese per le commissioni delle transazioni.
A quali commissioni si riferscono? Le commissioni sopra accennate si riferiscono a quelle del blocco ma come valori unitari per ogni singola transazione del blocco preso in esame. I blocchi però contengono diverse transazioni a volte moltissime perchè le transazioni sono aggiunte regolarmente al blocco il quale viene chiuso, appunto, quando viene risolto il problema matematico. In conseguenza di ciò è probabile che la remunerazione in seguito alla risoluzione del problema sia anch'essa non indifferente.
C'è da dire che i modi in cui vengono ricompensati i miners sono diversi: quello sicuramente più interessante (per lo meno per questa tesi) è quello della creazione di nuove valute.
L'idea di creare nuove valute non è proprio la cosa più originale del mondo ma sicuramente è una svolta importante in tutti i settori: tanto più si espanderà la blockchain tanto più queste valute prenderanno campo e quanto più la blockchain entrerà a far parte della vita di tutti i giorni, tanto più si utilizzeranno queste valute.
Di questi argomenti parleremo più in avanti, torniamo perciò al concetto di cui trattavamo prima.

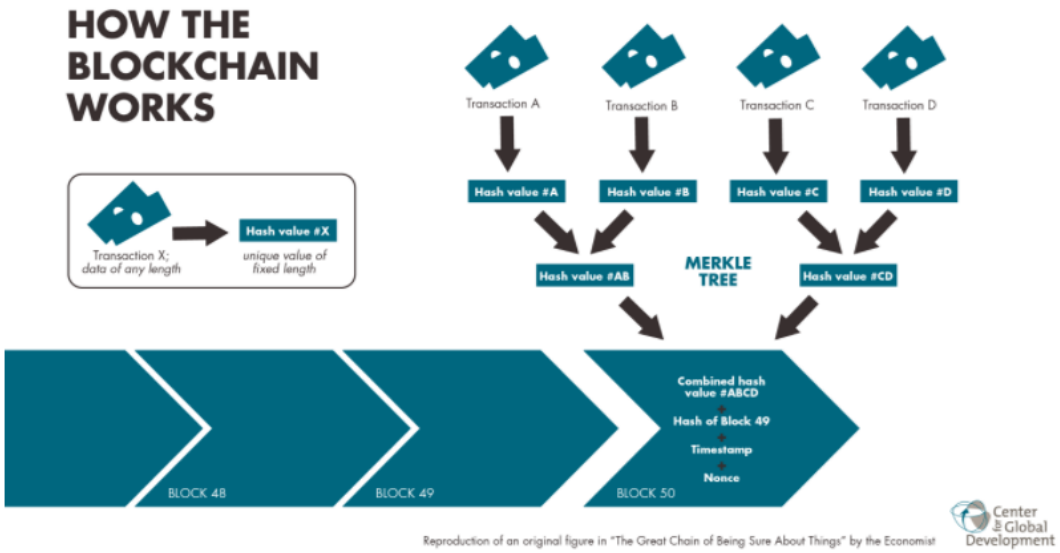


FIGURA 4.9: Come funziona la blockchain; fonte: Center for global development

Quando viene aggiunto un nuovo blocco alla catena, viene aggiornato il libro mastro che è distribuito e posseduto da tutti i partecipanti alla catena. I partecipanti accettano il blocco appena creato quando viene verificato che tutte le transazioni che compongono il blocco sono valide.
Se si verifica un'anomalia, un errore o un qualsiasi problema la creazione

del blocco viene rifiutata e tutti i partecipanti hanno evidenza del fatto che quest'operazione non è stata autorizzata, anche perchè se per caso qualcuna delle transazioni che vanno a comporre il blocco non è valida, tutto il blocco viene rifiutato. Ovviamente se tutte le transazioni del blocco sono valide (e quindi ritenute tali da tutti i partecipanti) allora il blocco viene creato e aggiunto alla catena.
Non è da dare per scontato che un blocco venga creato, quindi, ma cosa possiamo dire della sua permanenza nella catena? Per quanto riguarda la permanenza di un blocco alla sua catena di riferimento, questo è un elemento che possiamo dare per certo.
Il meccanismo che regola il legame tra blocco e catena è uno dei meccanismi fondanti della blockchain ovvero la sua **immutabilità**. Questa immutabilità riguarda di conseguenza anche la sicurezza dei dati che compongono un blocco e, quindi, l'intera catena.
Eccoci quindi arrivati ad un altro punto focale per il quale la blockchain è un meccanismo, un sistema, un mondo preferibile in quanto nuovo meccanismo da sostituire al vecchio che è quello appunto del libro mastro.
Si può obiettare a questa considerazione portando avanti l'argomento che per violare un libro mastro è necessario violarne l'autorità centrale che ne garantisce il funzionamento e lo gestisce.
Sebbene come argomento sia molto valido, si scontra con un più solido concetto che è quello dell'impossibilità di violare il meccanismo della blockchain. Sebbene questo possa sembrare un argomento un po' utopico, tuttavia c'è da considerare il fatto che per violare il sistema della blockchain bisognerebbe violare tutti i partecipanti contemporanemente anche perchè tra i meccanismi di certificazione, come abbiamo visto, c'è quello del timestamp.
É pur vero che bisogna considerare le dimensioni della blockchain ovvero il numero di partecipanti e valutare anche i sistemi di sicurezza dei partecipanti.
Questo è un altro dei motivi per cui viene incoraggiato il ruolo di miner: più sono gli utenti più la blockchain risulta un meccanismo sicuro.
Tornando al concetto di sicurezza della blockchain, questo sistema riesce a garantire con verosimiglianza che è impossibile ottenere o produrre un libro mastro finto o in qualche maniera corrotto dal momento in cui tutti i partecipanti possiedono una copia dello stesso. Di conseguenza se anche uno dei sistemi che possiedono una copia del libro mastro venisse violato, tutta la comunità si accorgerebbe della discrepanza dei suoi dati e, come succede in alcuni casi, il sistema violato può venire escluso dal processo.

## 4.5 Blockchain private e pubbliche

A questo punto diventa necessario approfondire il concetto di *permissioned ledger* e *unpermissioned ledger* per poter capire meglio gli ambiti di utilizzo della blockchain.

### 4.5.1 Blockchain pubbliche

Le *unpermissioned ledgers* sono istanze di blockchain pubbliche, aperte a tutti che non sono sottoposte a un'autorità o a un sistema di riferimento e sono fatte apposta per non essere controllate da nessuna autorità centrale.
Nascono con lo scopo di lasciare la possibilità a chi vuole parteciparvi, di contribuire all'aggiornamento dei dati sul libro mastro di riferimento e quindi di ottenere la copia di ogni operazione effettuata sullo stesso.
Il loro campo di utilizzo è vastissimo e possono quindi essere utilizzate per moltissime situazioni, da quelle più comuni a quelle più specifiche e singolari. Per poterne sfruttare appieno la potenza, un ambito di applicazione interessante potrebbe essere quello di utilizzarlo come databse globale per tutti quei documenti che richiedono l'immutabilità nel tempo, a meno di aggiornamenti importanti, e che richiedano la massima sicurezza.
Un esempio utile potrebbe essere un contratto di proprietà di un bene (per esempio un immobile) o un testamento.

### 4.5.2 Blockchain private

Le blockchain private sono dette anche *persmissioned ledgers* e hanno delle specificità che riguardano il modo in cui gestiscono le informazioni sulle transazioni e le transazioni stesse, che permettono loro di essere gestite da un'autorità centrale.
Il sistema di validazione delle transazioni cambia in quanto non si verifica attraverso il consenso condiviso della maggioranza degli utilizzatori del sistema, ma da un certo numero ben specificato e definito di soggetti che costituiscono quell'autorità centrale assente nella blockchain pubblica.
Il modo in cui viene costituita la catena però è lo stesso, i blocchi continuano a essere gestiti secondo un preciso ordine, le transazioni vengono validate comunque da più calcolatori i quali possono comunque essere dislocati a distanza perchè la blockchain privata continua ad essere un'istanza di un database distribuito.
In conseguenza di ciò le caratteristiche che rendono la blockchain preferibile al libro mastro centralizzato persistono e, anzi, il fatto che esista la possibilità di creare delle catene gestibili da un'autorità centrale sembra garantire una diffusione maggiore per quegli utenti che non ripongono fiducia nel sistema della condivisione.
In questo senso però non si raggiunge del tutto il concetto di consenso condiviso anche perchè di fatto è il punto focale di divergenza tra le due tipologie di blockchain.
Secondo un altro modo di concepire questo concetto di autorità, tuttavia, la blockchain privata può svilupparsi sulla base non di un'autorità centrale ma di un insieme di soggetti che l'autorità centrale garantisce come *trusted*. In questa maniera si ha un concetto di consenso condiviso che si trova in mezzo tra la blockchain pubblia e l'implementazione sopra menzionata di blockchain privata.
Le altre caratteristiche che contraddistinguono la blockchain privata da quella pubblica consistono nel fatto che possono essere gestiti dei meccanismi di

protezione dei dati nella misura in cui si costruiscono delle autorizzazioni e queste ultime vengono gestite e stabilite utente per utente; in questa maniera l'accesso alle informazioni sarà diversificato a seconda delle abilitazioni che ciascun utente avrà.
Questa caratteristica va a infoltire le potenzialità della blockchain da un lato, poichè la rende molto configurabile, mentre mantiene un contatto con i concetti di *governance* applicati nei vecchi libri mastro.
Un altro aspetto positivo delle permissioned ledgers sono le performance: dal momento in cui viene garantita la sicurezza e l'identità dei partecipanti allo sviluppo dei blocchi, non è necessario un livello di complessità dei problemi da risolvere come in quella pubblica.
Alla luce di queste differenze il profilo della blockchain appare molto più completo in maniera tale che ciascuno può usufruirne in maniera differente a seconda delle necessità.

# Capitolo 5

# Conclusioni

Il lavoro presentato è stato un contributo per lo studio sulla crittografia e la blockchain. Da sempre il problema della sicurezza sulle comunicazioni affligge tutti.
Oggi con le moderne tecniche di cifratura è possibile tenere nascosti i propri dati, le proprie informazioni e i messaggi che ci si vuole scambiare.
In questa sede l'intento era quello di mostrare le due facce della sicurezza, quella basata sull'anonimato e quello basato sulla crittografia.
Leggendo questo lavoro potrebbe sembrare strano che uno strumento molto diffuso e pubblicamente conosciuto come la blockchain sia molto sicuro[1].
L'intento è quindi quello di fare interrogare su come ci si protegge e su come si utilizzano le proprie informazioni.
Nonostante ognuno di noi abbia la responsabilità sulle proprie informazioni e sulla propria privacy oggi più che mai sembra che non sia necessario curarsene; con questo lavoro si vuole portare avanti l'idea che oggi più che mai è necessario proteggere le proprie informazioni perchè tutto quello che facciamo ormai passa da internet, e tutto ciò che passa da internet è possibile che sia tracciato da qualcun altro.

[1] Per esempio il codice sorgente dei bitcoin è pubblico e lo si può trovare a questo indirizzo: https://github.com/bitcoin/bitcoin.

# Bibliografia

[1] Internet delle transazioni; fonte: https://www.zerounoweb.it/cio-innovation/pa-digitale/blockchain-banking-potrebbe-diventare-linternet-delle-transazioni/

[2] Attacchi DDoS nel secondo trimestre del 2017; fonte: https://securelist.it/ddos-attacks-in-q2-2017/62709/

[3] Database distribuito; fonte: https://it.wikiversity.org/wiki/

[4] La blockchain; fonte: http://www.blockchain4innovation.it/esperti/blockchain-perche-e-cosi-importante/

[5] Kevin Mitnick, The Art Of Invisibility

[6] Ozalp Babaoglu, slides: http://www.cs.unibo.it/babaoglu/courses/security/lucidi/pdf/

[7] Andreas M. Antonopoulos, Mastering Bitcoin