**Introduction**

This introductory lesson provides a short introduction to this chapter.

Before we dive deep into blockchain design it’s important to understand cryptography concepts on which blockchain is based. In this chapter we will cover those concepts. If you already have a clear understanding of these please feel free to skip this chapter and just jump to its recap section.

So lets start by understanding **hash functions**.

In the next lesson, we will discuss hash functions.

# Hash Functions

A hash function maps data of arbitrary length to a unique fixed length string, called digest.

**We'll cover the following**

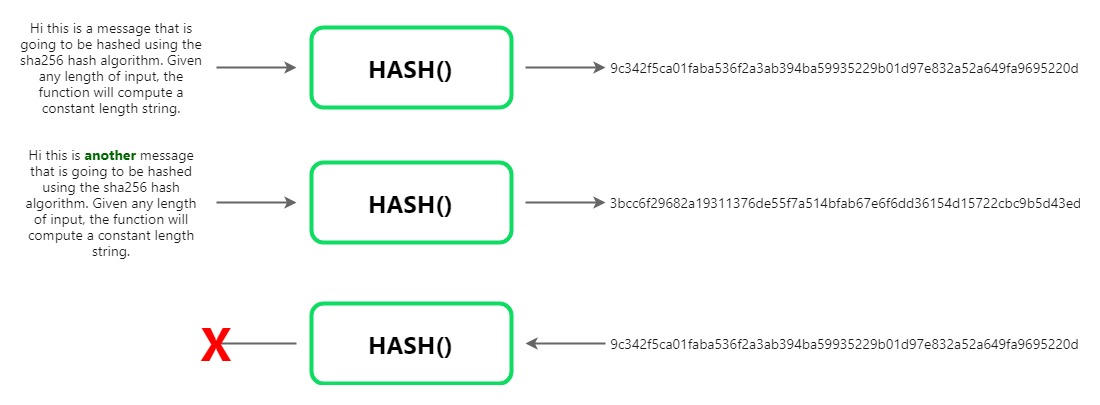
* + - [Hash Functions for Checking Integrity](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/JYzvKPXyA5P#Hash-Functions-for-Checking-Integrity-)
    - [SHA256 Hash Function](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/JYzvKPXyA5P#SHA256-Hash-Function-)

A hash value for a data is X is a function:

HASH(X) = Y

Such that:

* No other X’ can have HASH(X’) equal to Y. Its one to one mapping.
* The size of Y is fixed and the size of X can be arbitrary.
* Given Y you can not calculate X. Its a one-way function!



### Hash Functions for Checking Integrity [**#**](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/JYzvKPXyA5P#Hash-Functions-for-Checking-Integrity-)

This means that I can take a huge text file and compute its unique digest using a hash function. If I send that file and its computed hash along with it to a receiver, Bob, Bob can then recompute the hash to ensure that the content of that file were not corrupted in the transmission. When we download a file from the internet, it uses the same hash functions to verify its integrity.

### SHA256 Hash Function [**#**](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/JYzvKPXyA5P#SHA256-Hash-Function-)

There are multiple standardized hash function implementations that are used, such as SHA256, which we will be using in our course.

You can find libraries that implement SHA256 hash in all technologies, so you never have to write your code for SHA256 implementation.

Here is a small Javascript widget that will calculate sha256 of any data you enter:

# Public Key Cryptography

In this lesson, we will explain what public key cryptography is and it’s key usage.

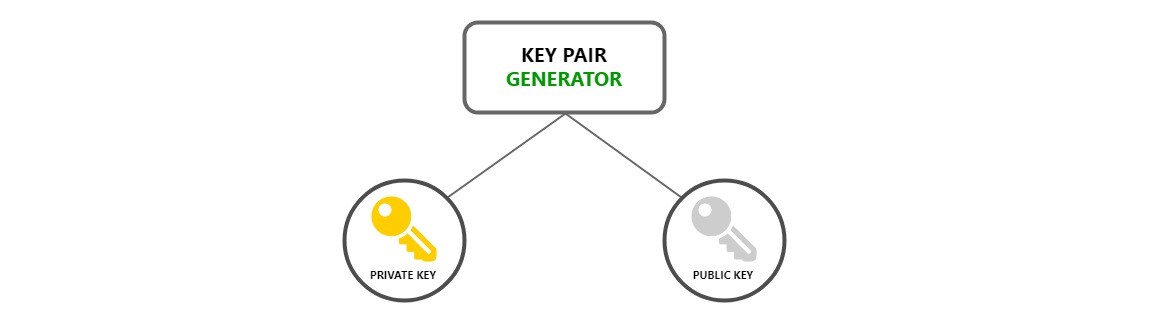
**We'll cover the following**

* + - [What is public key cryptography?](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/B8rpM8E16LQ#What-is-public-key-cryptography?-)
    - [Key usage](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/B8rpM8E16LQ#Key-usage-)
    - [Key generation playground](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/B8rpM8E16LQ#Key-generation-playground-)

### What is public key cryptography? [**#**](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/B8rpM8E16LQ#What-is-public-key-cryptography?-)

Public key cryptography is a cryptographic system used for the encryption/decryption of data. It is not one way like a hash function, meaning that the data, once encrypted, can be decrypted if you have the required key.

You start by generating a special, related pair of keys. These keys can be generated in a pair only. They are output from single execution of a key generation algorithm.



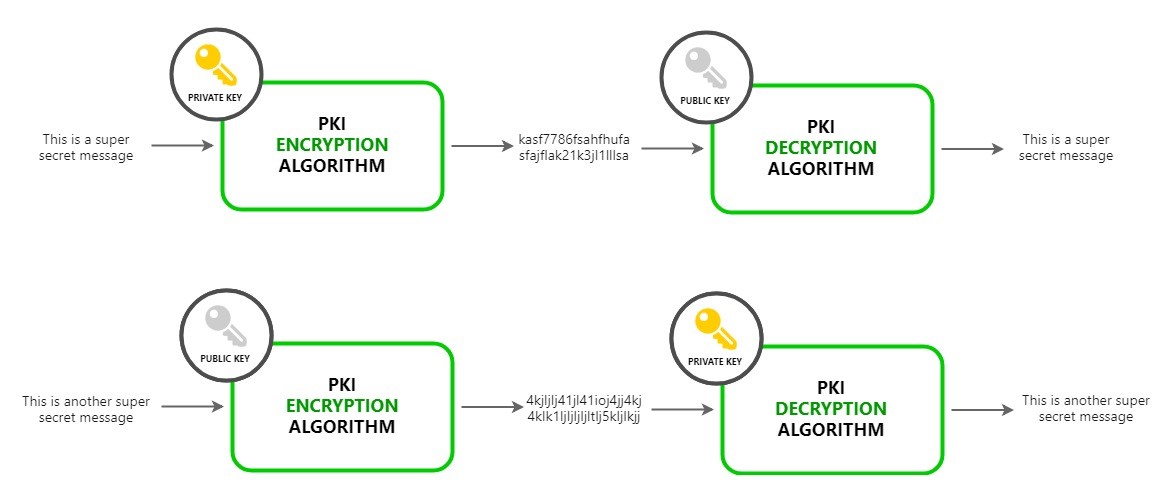
**Color coding in this course**:

* Golden key: private key
* Silver key : public key

### Key usage [**#**](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/B8rpM8E16LQ#Key-usage-)

Any key can lock or encrypt the data. To unlock or decrypt we need the other key. The only way to decrypt and make the data readable is by having the corresponding key.

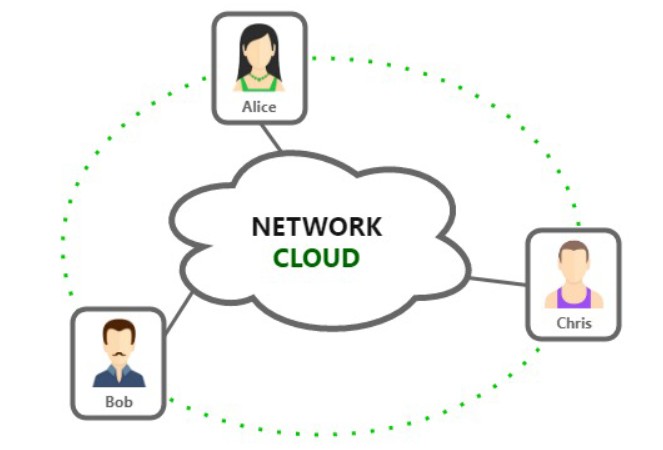
In the next lesson, we will show how this cryptographic technique is used to ensure communication security.



# Secure Communication using Public Key Cryptography

In this lesson, we will see how two parties can communicate securely via public key cryptography.

**We'll cover the following**



* + [How can we transmit a secret message using PKC?](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/B8nMkqBWONo#How-can-we-transmit-a-secret-message-using-PKC?-)

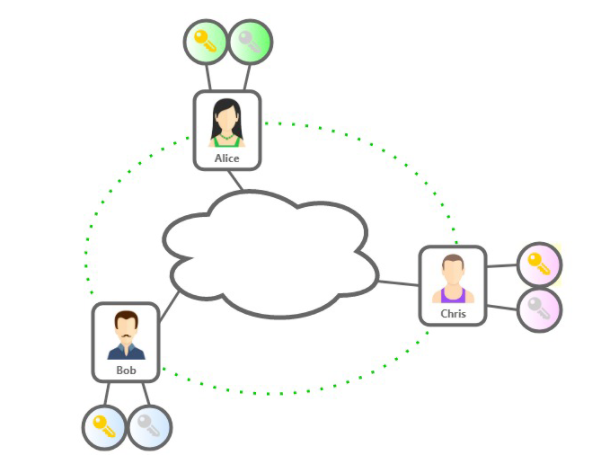
Now that you understand the basic features of asymmetric key pairs, let’s see how this is used to ensure that two or more parties can communicate private messages on an open and insecure network.

Lets say Alice, Bob and Chris are connected through a network on which everyone can read data being transferred. If Alice was to send a secret message to Bob only, it would not be possible as Chris gets to read everything transmitted through the network.

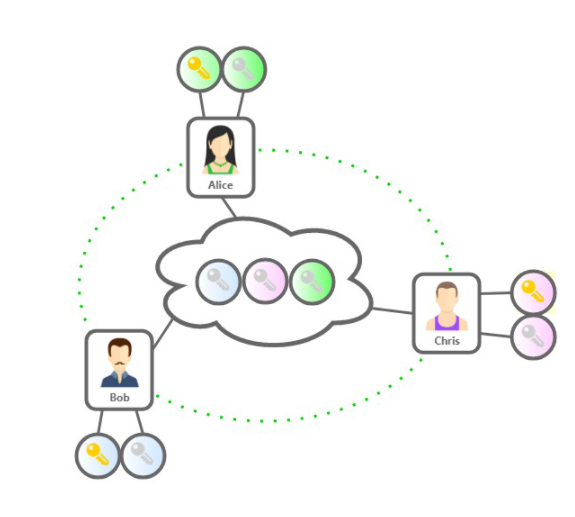
Insecure network

## How can we transmit a secret message using PKC? [**#**](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/B8nMkqBWONo#How-can-we-transmit-a-secret-message-using-PKC?-)

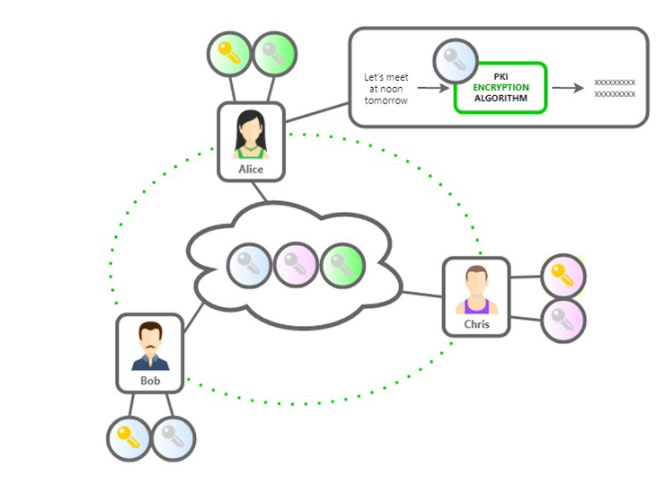
1. Each participant generates their own key pairs.



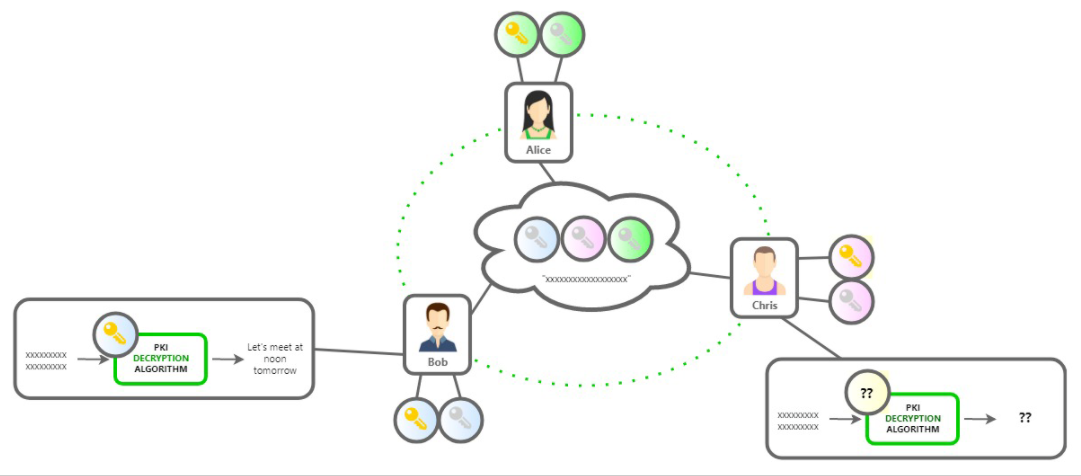
1. Each party **guards their private key** and **broadcasts the public key** on the network.



1. Now let’s say Alice wants to send a message to Bob saying “Let’s meet at noon tomorrow” but she does not want Chris to know about it. Alice will encrypt the message with Bob’s public key and share through the network.



1. Since the corresponding private key is only known to Bob, only Bob can decrypt it. Chris cannot make any sense of the encrypted message.



# Digital Signatures using Public Key Cryptography

In this lesson, we will see how to ensure the recipient of one's identity using public key cryptography.

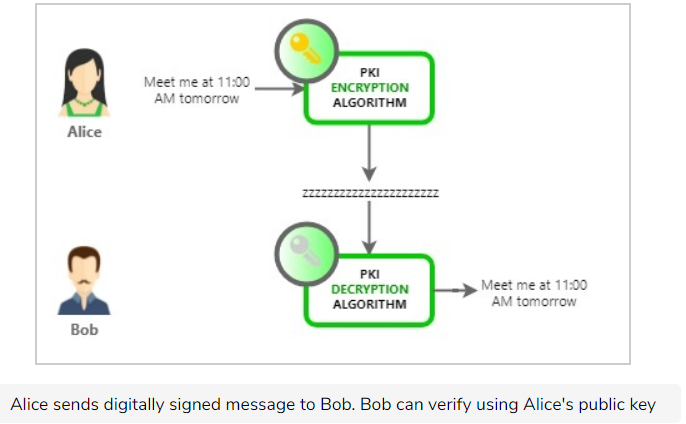
**We'll cover the following**

* + - [Adding a Digital Signature](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/gxQy88PwQEj#Adding-a-Digital-Signature-)
    - [Verifying a Digital Signature](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/gxQy88PwQEj#Verifying-a-Digital-Signature-)
    - [So far…](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/gxQy88PwQEj#So-far%E2%80%A6-)

Let’s continue with the previous example. Where Bob receives a message, which when he decrypts says “Meet me at 11:00 tomorrow”. How can Bob know for sure that the message was sent by Alice and not Chris?

This can be achieved if the message is “signed” by its sender (Alice). The receiver of the message can then verify the signatures.

We need a way for Alice to “digitally sign” the message so the receiver of the message (Bob) can trust that the message was sent by Alice. Alice can add a digital signature to the message by using her private key to encrypt it. When the message is decrypted by her public key only, Bob can know for sure it came from Alice.

This works, but to make it more efficient, instead of signing the entire lengthy message the sender computes a hash (digest) of message and signs that with his public key instead. The receiver can then re-compute the message hash and compare it with the signed hash to ensure that message was not tampered with.

### Adding a Digital Signature [**#**](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/gxQy88PwQEj#Adding-a-Digital-Signature-)

1. Alice computes a message digest by hashing the message she is about to send. SHA256(“Meet me at 11:00 tomorrow”).
2. Alice calculates the message digest = HASH(message);
3. Alice signs the message digest by encrypting it with her private key.
4. Alice appends the signed digest with message and encrypts with Bob’s public key Alice sends it over to Bob

### Verifying a Digital Signature [**#**](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/gxQy88PwQEj#Verifying-a-Digital-Signature-)

1. Bob decrypts the message using his private key
2. Bob decrypts the digest using Alice’s’ public key
3. Bob computes digest of the message. If it matches the digest he received as signatures it confirms him that:

* Message is not tampered with
* Message has been sent by Alice only - Since only Alice’s public key could decrypt the hash

### So far… [**#**](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/gxQy88PwQEj#So-far%E2%80%A6-)

We have seen how awesome public key cryptography is and how, when combined with hash functions, it provides us with the capability to digitally sign any data.

* It helps us communicate securely on insecure channel (open internet), ensuring only the intended receiver of a message can read the message
* Ensuring the receiver can trust that the message is coming from a specific sender
* Ensuring the message is not tampered with during transmission

**For all of this to work Bob needs Alice’s public key.** However, can he trust the public key he received indeed belongs to Alice? He received the key over the same untrusted network.

# Public Key Infrastructure

In this lesson, we will understand the concept of public key infrastructure and see how certificate authority works.

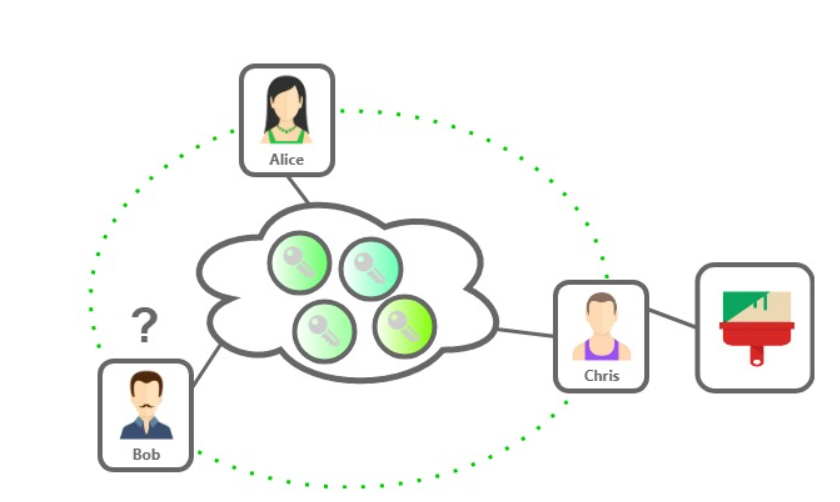
**We'll cover the following**

* + [Here is how it works with a CA:](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/YQVkjp548NM#Here-is-how-it-works-with-a-CA:-)
  + [Chain of Trust](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/YQVkjp548NM#Chain-of-Trust-)

We have seen how Hash functions and asymmetric encryption can help us digitally sign any message.

On an open or insecure internet, in order for a receiver(Bob) to verify a signature done by a sender(Alice), he needs Alice’s public key beforehand.

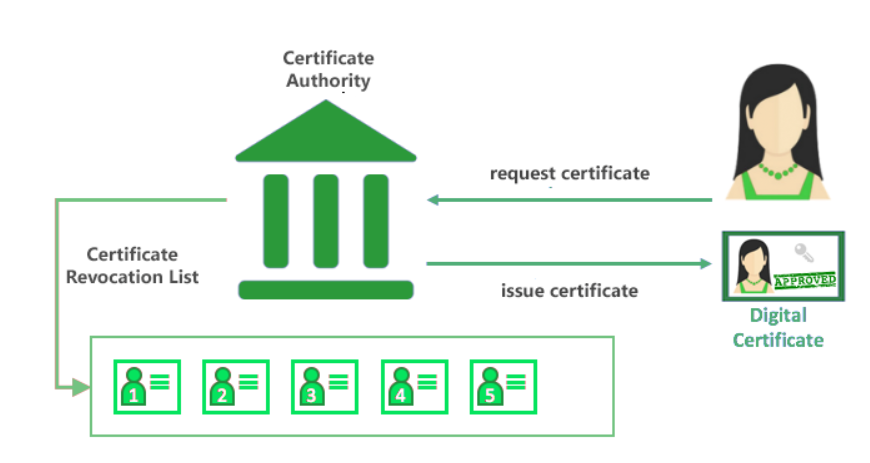
If Alice was to send her public key to Bob on the internet, how can Bob be sure that it is really Alice who has sent him her key and not Chris pretending to be Alice?



For the entire secure communication system to work, both parties(Bob and Alice) need to establish trust in each other’s public key while using the same open unsafe internet.

How to share the key and prove its ownership to ensure future secure communication?

To help solve this issue we have a concept of a trusted authority in the middle that all network participants trust. This trusted authority is called a CA or certificate authority.

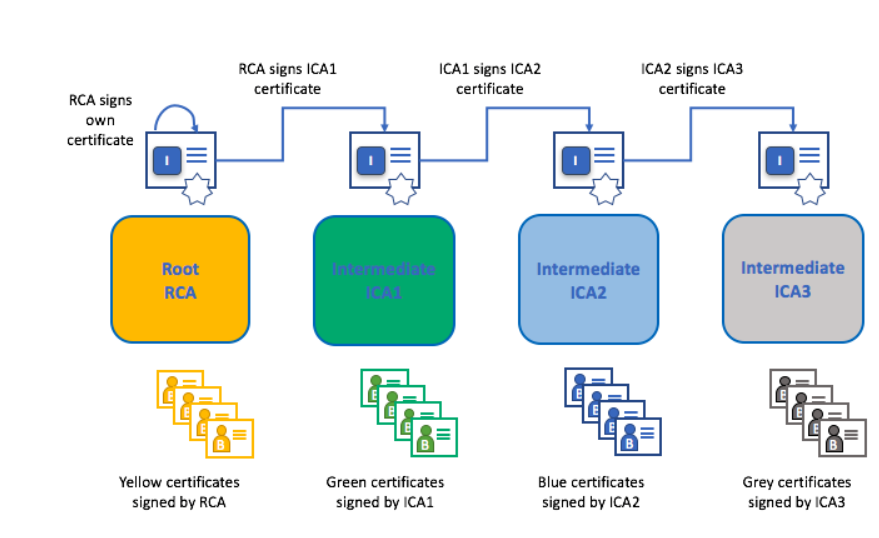


## Here is how it works with a CA: [**#**](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/YQVkjp548NM#Here-is-how-it-works-with-a-CA:-)

1. Alice must contact the CA to get a certificate(digitally signed public key) to prove her public key to others. All other participants on the network will be able to trust the public key is Alice’s if they trust the CA.
2. The CA would take Alice through an approval/on-boarding/manual-verification process and issue a certificate. The certificate itself is a list of certified attributes of the entity its issued to Alice. It has attributes like the public key, name of the holder etc. All this data is digitally signed by the CA using its own private key.
3. Alice then shares this certificate as a proof of her public key to Bob.
4. Since Bob has the CA in his trusted CA list(CA’s public key), he can verify and trust the certificate shared by Alice and hence trust her public key.

## Chain of Trust [**#**](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/YQVkjp548NM#Chain-of-Trust-)

A chain of trust is established between a Root CA and a set of Intermediate CAs as long as the issuing CA for the certificate of each of these Intermediate CAs is either the Root CA itself or has a chain of trust to the Root CA.



# Blockchain Data Structure

In this lesson, we well discuss the data structure of blockchain

**We'll cover the following**

* + - [What is blockchain?](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/7nA1nGn3V6A#What-is-blockchain?-)
    - [Distributed Consensus](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/7nA1nGn3V6A#Distributed-Consensus-)
    - [Data Storage in Blocks](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/7nA1nGn3V6A#Data-Storage-in-Blocks-)
    - [Activity:](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/7nA1nGn3V6A#Activity:-)

### What is blockchain? [**#**](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/7nA1nGn3V6A#What-is-blockchain?-)

In a blockchain system, data is stored in blocks of transactions. The most common definition is as follows:

A timestamped log of transactions that is replicated on multiple peers.

### Distributed Consensus [**#**](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/7nA1nGn3V6A#Distributed-Consensus-)

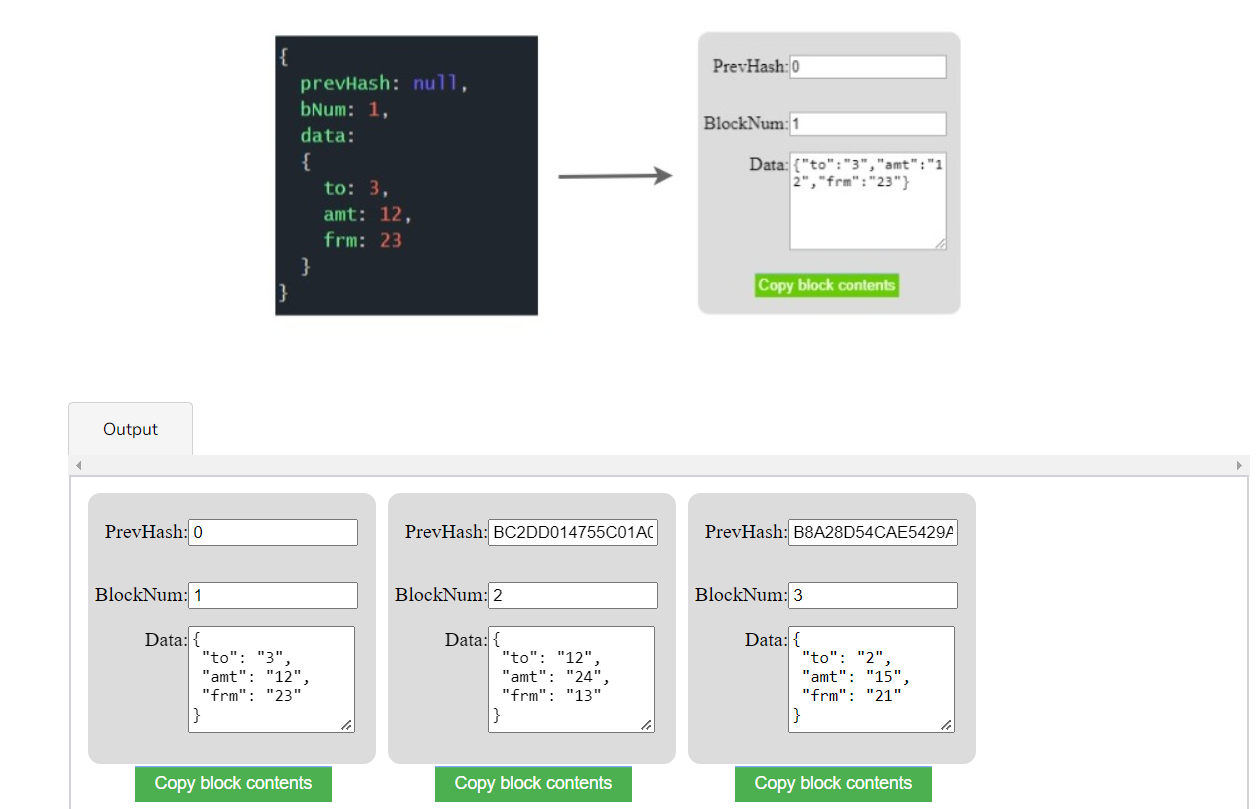
Now, some of these peers might be evil and intentionally report a tampered version of blockchain data. The entire network uses democracy to come to a consensus on the current state of data and any non-conformant or outlier peers are ignored/blocked.

This means that in order for a blockchain network to be fair and valid, most of the nodes have to be fair. If 51% of the nodes are compromised the network is hacked. Since the networks are globally distributed, this is not a possibility.

### Data Storage in Blocks [**#**](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/7nA1nGn3V6A#Data-Storage-in-Blocks-)

Let’s see how blockchain stores data in blocks. Each block stores a data blob(which is usually a list of transactions), its block number, and hash of the previous block:

Each block is represented by a json object here for better understanding:



Let’s look at the previous block hash, referenced as “prevHash” in the above diagram.

When creating a new block, the hash of the previous block is calculated and added to the next block. Now if the previous block is altered later in time, the next block will be invalid as the prevHash stored in it will not match the actual hash of previous block.

Using one-way hash functions, the data in the blockchain is safeguarded from tampering.

### Activity: [**#**](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/7nA1nGn3V6A#Activity:-)

Imagine that there are three more transactions to be recorded and a new block needs to be added to the block data structure shown above. What should be the value of PrevHash in new BlockNum 4.

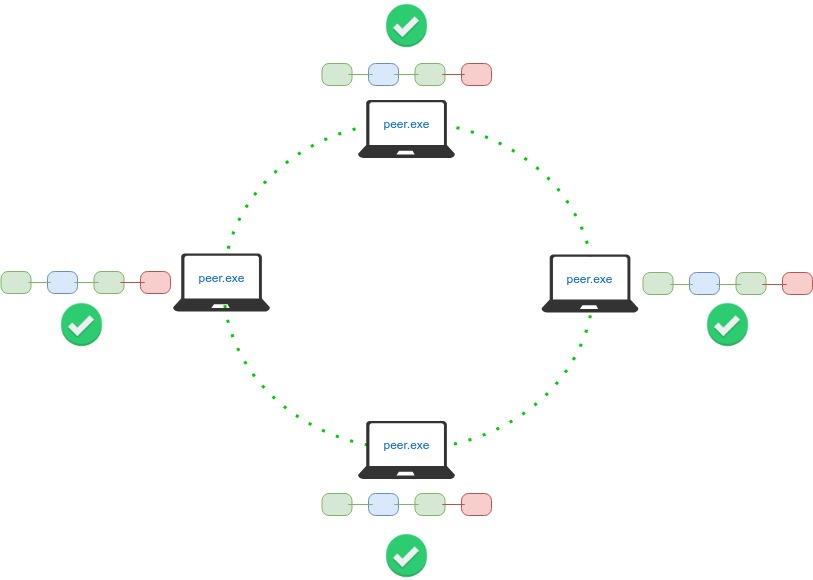
Use the hashing widget below to help calculate your answer.

# Blockchain Big Picture

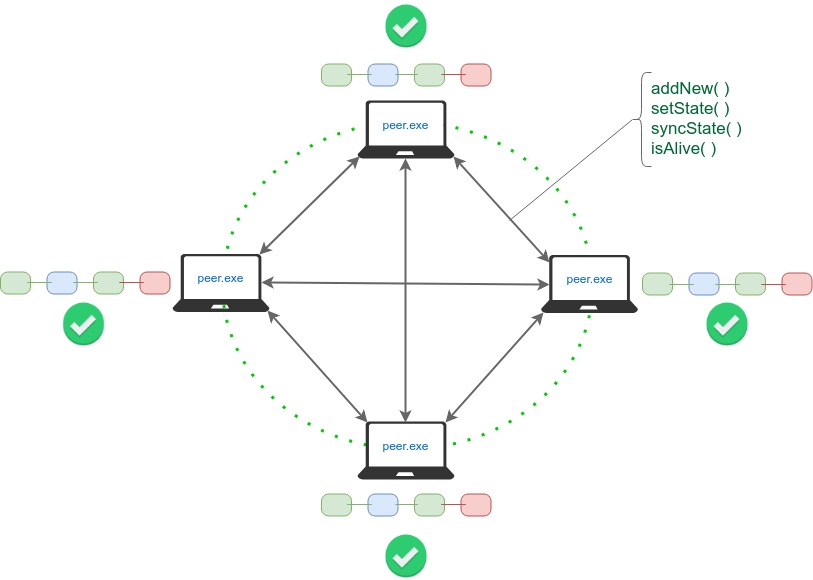
In this lesson, we will see the bigger picture of Blockchain.

You have seen how we can store data in a structure called blockchain. In a distributed blockchain network:

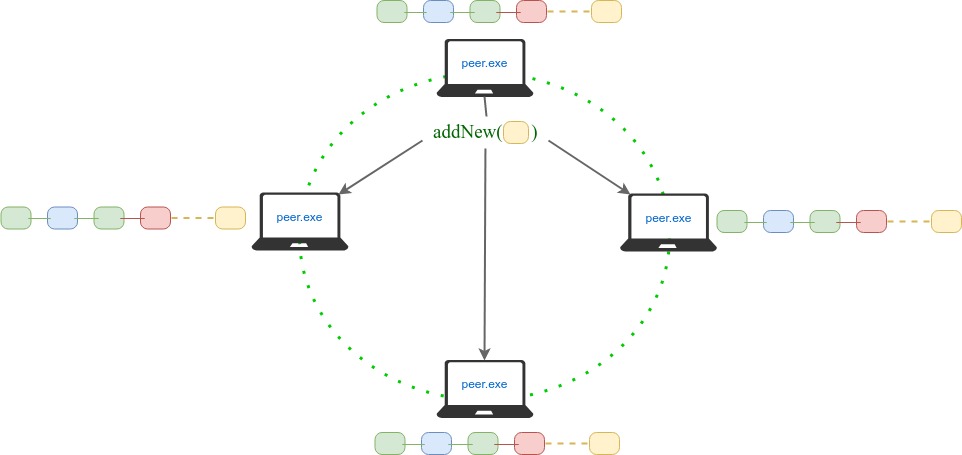
Multiple peers have a process(peer.exe) running that maintains a ledger on their local storage.



This process connects to other process instances running on other machines to receive updates on new blocks, transactions, health and validity checks etc to keep itself updated.



A transaction/block can be appended by any peer and is then broadcast-ed to all peers.



Since multiple peers are adding transactions/blocks simultaneously, the consensus protocol along with the ledger implementation ensures “validity” and ordering of transactions in blocks forming the blockchain.

We will dig in to the consensus and validity more in future lessons. Let’s start with the Bitcoin example to see how all this can create real value.

In the next chapter, we will understand the value of Blockchain through the example of Bitcoin.

# The Bitcoin Blockchain History

In this lesson, we will check out the bitcoin history and how transactions work in bitcoins.

**We'll cover the following**

* + - [Membership/Transaction Signing:](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/B8OO0L8gRkW#Membership/Transaction-Signing:-)
    - [Calculating Account Totals:](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/B8OO0L8gRkW#Calculating-Account-Totals:-)
    - [Coin Generation:](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/B8OO0L8gRkW#Coin-Generation:-)

The Bitcoin blockchain was born in 2009, by a pseudonym Satoshi Nakomoto.

Take a look [here](https://www.blockchain.com/btc/block/000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f) (<https://www.blockchain.com/btc/block/000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f>)

 at the first block(block 0) or the “genesis block” of Bitcoin blockchain.

There is one transaction in this block to Satoshi Nakomoto’s address. Interestingly Satoshi’s account address has never been used to transfer out money so we don’t know if it’s intentional, or if the private key lost?

Now look at [another](https://www.blockchain.com/btc/block/000000000000000004ec466ce4732fe6f1ed1cddc2ed4b328fff5224276e3f6f) block, this one has 1660 transactions.

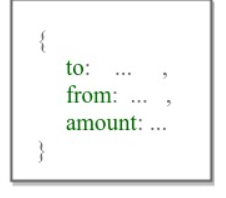
<https://www.blockchain.com/btc/block/000000000000000004ec466ce4732fe6f1ed1cddc2ed4b328fff5224276e3f6f>

A transaction is a record of transfer of bitcoins from one address to another.

### Membership/Transaction Signing: [**#**](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/B8OO0L8gRkW#Membership/Transaction-Signing:-)

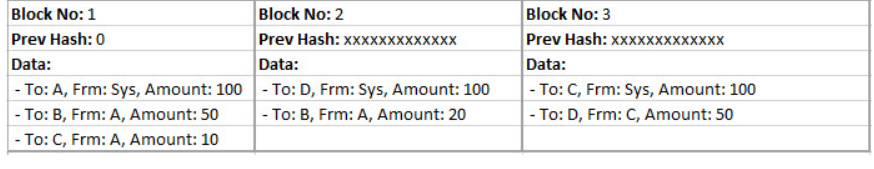
A transaction is always digitally signed by the ‘from’ an account holder to ensure that only the owner of account can perform a transaction using their balance.

In Bitcoin, anyone can generate an asymmetric keypair and start transacting on the Bitcoin blockchain.



### Calculating Account Totals: [**#**](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/B8OO0L8gRkW#Calculating-Account-Totals:-)

Consider three blocks with multiple transactions in each block. The timestamp has been omitted here to give a simplified view:



This would mean the accounts totals for each accounts are as follows, after executing all transactions:

* A: 20
* B: 70
* C: 60
* D: 150

### Coin Generation: [**#**](https://www.educative.io/courses/hands-on-blockchain-hyperledger-fabric/B8OO0L8gRkW#Coin-Generation:-)

In our example, each block has a transaction from “Sys” which awards some balance to an account. In Bitcoin, this is labelled as “newly generated coins” and these are the new coins the system awards to the miner of the block.

Now lets see what mining is…

**How Bitcoin Becomes Valuable**

In this lesson, we will discuss how Bitcoin became a recognized currency.

*Value* for anything is driven by the balance of its demand and supply.

Anything found in abundance is of no great value. So, in order for something to be valuable, it must be limited in supply. Bitcoin is limited due to the proof-of-work requirement that adds new Bitcoins into the system.

*Demand* is higher for things people find useful. It could be something useful for two reasons:

1. In terms of the actual utility it provides.

* For example, cars provide a transportation utility.
* Bitcoin provides instant value transfer facilities which is not found in traditional currencies.

1. In terms of the satisfaction it brings.

* For example, people like to have diamonds for physiological satisfaction. Crystals are just as comparable in terms of utility.
* Decoration pieces - aesthetic satisfaction.

A supply-demand balance makes Bitcoin a valuable and desirable currency.

It’s nothing physical. It’s just a digital, immutable, trustable log of transfer transactions. But all these properties give it great utility and therefore value!

It’s important to keep in mind that money doesn’t really exist. ‘Moneyness’ is a property. This property of moneyness is held by fiat currency (legal tender whose value is backed by the government that issued it)and bills due to the State Bank (trusted authority) that promises to hold its value. This ‘moneyness’ is in anything that we deem valuable and therefore can be used as a unit of account, a store of value, or a medium of exchange.

Bitcoin has these properties of ‘moneyness’ making it a valuable asset.

* It allows you to store value
* It allows you to consume value (better than fiat)
* It allows you to exchange value (better than fiat)

In the next chapter, we will discuss the concepts of HyperLedger Fabric.

##### **Hyperledger Fabric - Concepts**

# Introduction

In this lesson, we will give a quick introduction to Hyperledger Fabric.

Blockchain technology is also referred to as Distributed Ledger Technology, or DLT, as it works on a ledger that is distributed on multiple peers.

Now that we have a high-level understanding of what DLT is we are ready to jump into Hyperledger Fabric.

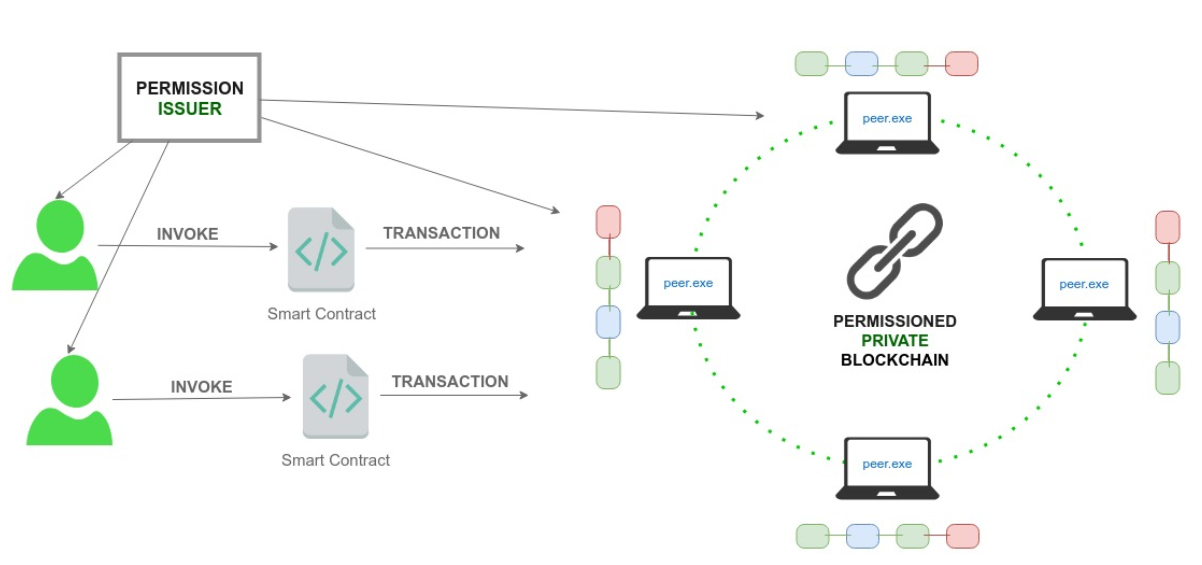
In this section, we will understand core Hyperledger fabric concepts and terminology at a high level and explore a “hello world” smart contract built on Hyperledger fabric.

Hyperledger Fabric is a “blockchain platform for the enterprise”. It is open source and modular - allowing different modules to be used, plug and play style. This enables a wide variety of enterprise requirements. It is designed to provide speed and scalability that is lacking in public chains due to proof of work requirement, which is essentially nonce mining.

Hyperledger fabric is ideal for building a permissioned, private blockchain business network. By private, it means that it should not be publicly open for everyone to run a peer or transact on the network. For enterprises, this a big requirement that Hyperledger fabric meets. Enterprises need more control on their data access policies. They also need a permissioned network so they can implement access control as per their own requirements.

At a high level, this is how a Hyperledger network works. The permission issuer issues or revokes permissions for all participants and infrastructure components of the network. This permission or access control in Fabric is based on X509 PKI infrastructure. Which means there is a trusted certificate authority that issues certificates to all participants.

Smart contracts hold logic that defines who can change what on the ledger. And participants write transactions on the ledger by invoking smart contracts.



In the next lesson we will understand hyperledger network concepts