SESSION 5

Blockchain - Bringing All the Elements Together

Learning Objectives

- Describe a block of a blockchain
- Discuss cryptocurrency
- Demonstrate how to use blockchain
- Discuss blockchain explorer

Introduction

Blockchain is a decentralized and distributed ledger which has become one of the fastest growing technologies today. It plays an important role in the safe and secure cryptocurrency transactions. Due to the popularity of the technology, you are required to know how data is organized and stored in a blockchain.

A blockchain is structured as a linked list of blocks of transactions that can be stored as a flat file or in a simple database. Blocks of a blockchain are linked to the previous block of the blockchain. Every block is identified with a hash generated by the SHA-256 hashing algorithm. Each block stores the hash value of the previous block, known as the parent block.

The most visible application of blockchain technology is cryptocurrency, which is a purely digital decentralized currency. Highly secured nature of transactions conducted through cryptocurrency is attributed to cryptography.

Blocks of a blockchain are browsed with the help of a blockchain explorer. It provides a wide range of information, such as recently mined blocks, transaction history, and the total number of unconfirmed transactions. All such functionalities provided by a blockchain explorer prove to be of great help to the users of cryptocurrencies such as Bitcoin and Altcoin.

In this session, you will gain insights into the concept of a block in blockchain along with its underlying structure and constituent elements. You will also be made conversant with cryptocurrency and its key properties. After this, you will learn about the working of blockchain using an online tool. Towards the end of the session, you will learn about blockchain explorer.

Overview of a Block

[LO - Describe a block of a blockchain]

10 min

Every blockchain is created by linking blocks together in order to form an immutable ledger. The most basic unit of a blockchain is a block. A block can be defined as a set of transaction entries that are stored in the nodes of a blockchain network. Every block contains batches of hashed or encoded transactions.

A blockchain can be either stored in a flat file or in a database. In Bitcoin, metadata about all the blocks is stored in Google's LevelDB database. Every block has a pointer that identifies a block. It is the hashed value of the header data of a block. It is considered as a unique identifier of the block. Each block has the hash of the previous block that helps in linking all the blocks to the initial block known as the genesis block. All the blocks are linked together by the hash value.

A block can also be referenced by the height of a blockchain. It refers to the distance of the block, or the block count, from the genesis block.

In the following sections, you will learn about the block structure, genesis block, block header, Merkle tree, transaction pool and candidate blocks.

Structure of a Block

Blocks of a blockchain are data structures that contain a collection of transactions and metadata about the block. A block is made up of a header and a body. The header contains the metadata which includes information about the data stored in the header. An average block can contain more than 500 transactions. Figure 1 shows the structure of a block:

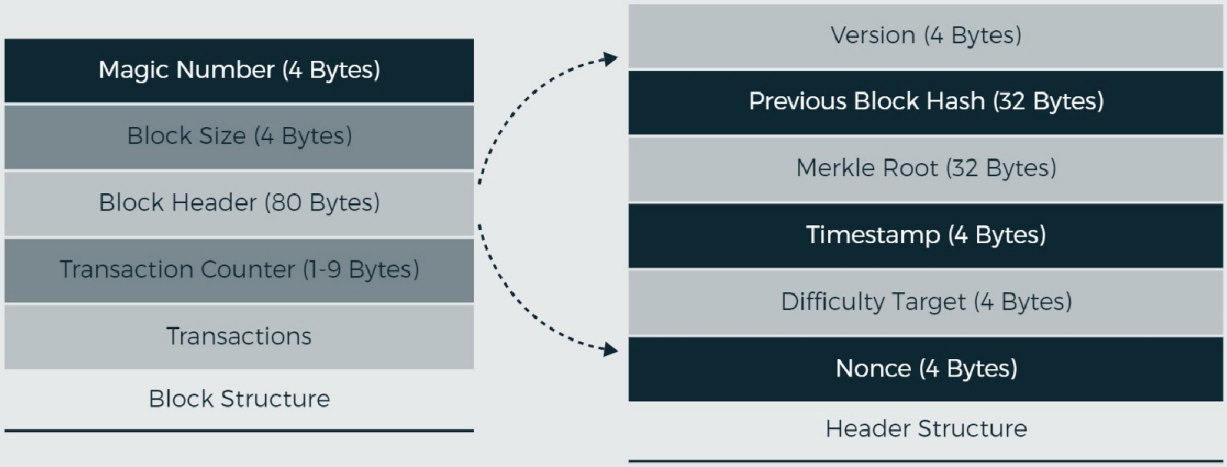

Figure 1: **Structure of a Block**

Table 1 describes the structure of a block.

Table 1: Structure of a Block

Field name	Size	Description
Magic Number	4 bytes	This value is an identifier for the Blockchain network. It always has a value of 0xD9B4BEF9. It indicates the start of the block and data is from a production network.
Block Size	4 bytes	This value indicates the size of a block. The original size of a Bitcoin block is 1 MB and the size of the newer version of Bitcoin known as Bitcoin Cash is 2 MB.
Block Header	80 bytes	This field consists of details such as the hash value of the previous block, nonce, Merkle root, and others.
Transaction Counter	1-9 bytes	This field indicates the total number of transactions included within a block. It is not necessary that every transaction is of the same size and there are chances that not every block will contain the same number of transactions.
Transactions	Variable	This field consists of a list of transactions that are taking place in a block.

Genesis Block

The genesis block is the first block in the blockchain. It is also known as block-0. It is the foundation on which additional blocks have been added to create the blockchain. It was created in 2009 by Satoshi Nakamoto. This block is a common ancestor to all the blocks. A node in a blockchain network can only start a blockchain having at least one block as the genesis block is statically encoded in the Bitcoin core node. Every node always has the information of genesis block's hash and structure, the time it was created and the single transaction in it. It provides a secure root to build a trusted blockchain.

Using a command line reference to Bitcoin Core, you can find the following information about the genesis block:

$ bitcoin-cli getblock 000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f

{

"hash": "000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f",

"confirmations": 521239,

"strippedsize": 285,

"size": 285,

"weight": 1140,

"height": 0,

"version": 1,

"versionHex": "00000001",

"merkleroot": "4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b",

"tx": [

"4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b"

],

"time": 1231006505,

"mediantime": 1231006505,

"nonce": 2083236893,

"bits": "1d00ffff",

"difficulty": 1,

"chainwork": "0000000000000000000000000000000000000000000000000000000100010001",

"nextblockhash": "00000000839a8e6886ab5951d76f411475428afc90947ee320161bbf18eb6048"

}

Now, if the value of Unix time stamp is converted, then this information can be found: Saturday 3rd January 2009 23:45:05. It is a reminder to the time when this block was created.

Block Header

The header of a block stores mainly metadata related to data stored in the block. Its size is 80 bytes. It contains the hash value of previous block, the difficulty target, timestamp, nonce and the value of the Merkle tree root. Table 2 describes the structure of a block header.

Table 2: The Block Header

Field name	Size	Description
Version	4 bytes	This field indicates the version number which is mainly used to track software updates.
Previous Hash Block	32 bytes	This field contains the hash of the block header of the previous block. When the fields in the previous block header are hashed with SHS-256 algorithm, then it produces a hash of 256 bits which is 32 bytes.

Field name	Size	Description
Merkle Root	32 bytes	This field contains the hash of all the transactions in the block. Merkle root is defined as the hash root of the Merkle tree. It helps in ensuring that no modifications are made in blocks. Merkle roots also help in determining if a transaction is a part of a block and it can be identified in O(n) time.
Timestamp	4 bytes	This field denotes the time taken to create a block in the Unix time format.
Difficulty Target	4 bytes	This field denotes the proof-of-work (PoW) difficulty level set for this block. Difficulty is a measure that defines how difficult it is to find a hash below a given target. For example, the difficulty of 7,879,456 means that at a given hash rate, it will take, on average, approximately 7.8 million times to find a valid block as it would at a difficulty of 1. In other words, it will take, on average, approximately 7.8 million times as many hashes to find a valid block.
Nonce	4 bytes	This field denotes a random number which is produced after solving the PoW mathematical problem. It is used to capture the transaction and to verify if the PoW is being implemented correctly.

Merkle Tree

A Merkle tree is a binary tree containing cryptographic hash pointers. It is named after its inventor Ralph Merkle. In a Merkle tree, each leaf node represents the hash of the data block. Every parent node has the hash values of its children nodes. Hashing is usually continued until the root node has been reached.

In a Merkel tree, the leaf nodes are the hashes of transactions and the root is the Merkle root. Merkle trees recursively hash the nodes until only one hash is left which is saved in the Merkle root.

If there are N transaction in a block, then Merkle tree can find if a transaction is included in the block in at most 2*log2(N) calculations. This proves that Merkle trees provide a very efficient way to check transactions and their integrity.

Merkle trees are binary and therefore require an even number of leaf nodes. If the number of transactions is odd, the last hash will be duplicated once to create an even number of leaf nodes. It is used to summarize large sets of data. Merkle trees are used in Bitcoin, Ethereum, and other blockchain applications. Figure 2 shows an example of a Merkle tree:

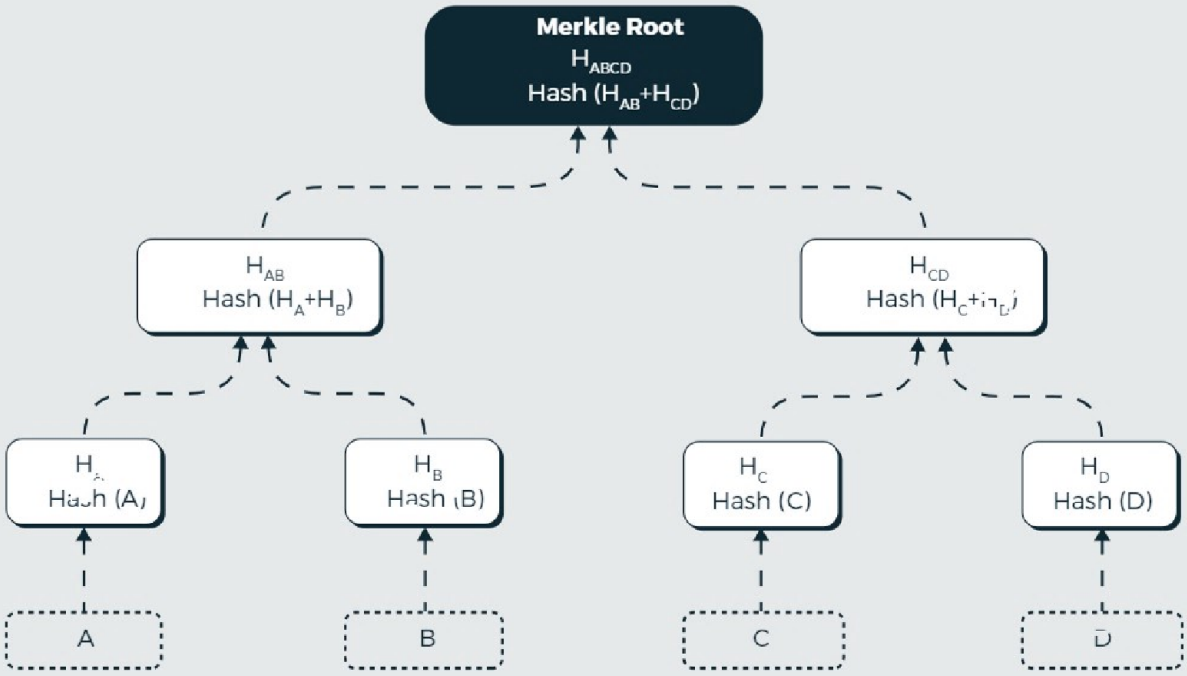

Figure 2: **Example of Merkle Tree**

A Merkle tree is constructed bottom-up. As shown in Figure 8, leaf nodes contain hash values of data blocks A, B, C, and D represented by HA, HB, HC, and HD respectively. Each parent node will construct its hash by concatenating the hash values of the children nodes and hashing them again:

$$H_{AB} = \text{Hash}\ (H_A + H_B)$$

The Merkle root contains the summary of the data in the transactions and is stored in the block header. It is used to maintain the integrity of the data. If any change occurs in the transactions or their order, the Merkle root is changed.

Some benefits of Merkle trees are as follows:

- They help in reducing the amount of data maintained by a trusted authority for verification.
- They help in proving the integrity and validity of data.
- They need less amount of disk space as the proofs are easy to compute in a fast manner.
- Only small amount of information for proofs and management is needed to be transferred across networks.
- They help in verifying the completeness and consistency of a log.

Transaction Pool

In Bitcoin, a transaction pool, also known as a mempool or memory pool, is used to store unconfirmed transactions that are currently stuck in the Bitcoin network.

Suppose a user wants to send Bitcoins to someone. These bitcoins will be transferred via a transaction. The user will instruct his/her wallet to make a transaction of desired Bitcoins. Then the wallet will sign the transaction with the user's private keys and then broadcast the transaction onto the Bitcoin network for verification.

Now, the transaction will not be completed as soon as it is broadcast. It will be stored in the mempool. It is known as mempool because it is created using the RAM of nodes in the Bitcoin network. For completion of the transaction, it needs to be verified. For verification, at least one miner is required to pick the transaction and validate it. As a standard practice, a transaction is considered valid if it has got at least six confirmations. What does it mean to get six confirmations? This means that the transaction is part of at least 6 consequent blocks. Again, why 6 blocks? This is because it is usually considered enough to thwart a reversal attempt by an attacker with less than 10% of hash power of the network. However, to be 99% safe would require more than 60 blocks or 12 hours (since one block is generated every 10 minutes).

The main issue with mempool is its size. There is only a limited number of nodes available. Therefore, the amount of memory used to store unconfirmed transactions is limited.

Candidate Blocks

A candidate block is a block which a miner is trying to mine in order to receive the reward. It is a temporary block which can be either discarded or validated by the network. Generally, miners compete with each other to validate the next block. They create candidate blocks by collecting and organizing multiple unconfirmed transactions from the memory pool.

Exploring Cryptocurrency

[LO - Discuss cryptocurrency]

A cryptocurrency is defined as a tradeable digital form of money which exists only online and is built on blockchain technology. Cryptography is mainly used in cryptocurrencies to verify and secure transactions. There are over one thousand cryptocurrencies that exist today.

Every cryptocurrency works in a similar manner. The major difference in fiat currencies and cryptocurrencies is that cryptocurrencies are purely digital and there is no option to have a cryptocurrency in paper or coin form.

Cryptocurrencies work like fiat currencies that are being used today to pay for goods and services. Some of the properties of cryptocurrencies are:

- **Secure**: Cryptocurrencies are secured by cryptographic techniques used by organizations like NSA. Cryptocurrencies belonging to a user are stored as a decentralized ledger entry secured by an asymmetric key cryptography system. A user can only transact if he/she has the corresponding private key.

- **Anonymous:** It is not easily possible to relate transactions or accounts with real-world identities. Addresses used in Bitcoins are 30-character strings which cannot be easily connected with any real-world identity.
- **Irreversible:** Once a transaction is added to a block and enough blocks are added after it (confirmations) then it cannot be reversed under any circumstances.
- **Fast**: Transactions can be performed even in minutes or seconds depending on cryptocurrency being sent.
- **Globally accessible**: As long as a user has an Internet connection, he/she can transact cryptocurrency anywhere in the world.
- **Volatile:** It is very prone to price fluctuations.

Demonstrating Blockchain using Online Tools

[LO - Demonstrate how to use blockchain]

In this section, you will learn about blockchain using an online tool provided by Anders Brownworth. Open a web browser and type the following URL in the address bar:

https://anders.com/blockchain/block.html

Figure 3 shows a block of a blockchain:

Figure 3: **A Block of a Blockchain**

Blockchains use SHA-256 hash function on a combination of block's data and nonce. Whenever the block's data or nonce is changed, the hash value will be changed, as shown in Figure 4:

Figure 4: **Changed Hash Values**

As shown in Figure 4, the nonce value is changed to 54964, so the hash value is also changed. Now in order to be considered as a valid or mined block, the hash value and nonce should fulfill certain conditions. For example, the hash value should always start with four zeros. Mining can become more complex by making the conditions more complex.

The miners are required to find a nonce value that makes the hash value to fulfill the mining condition. Now, in the online tool, if you will type some data in the **Data** text box and click the **Mine** button, the app will start generating nonce values that make the leading four digits of the hash value equal to "0000". If the block is considered as mined, the background color will turn green from pink. Figure 5 shows a nonce value with some data in a mined block:

Figure 5: **A Mined Block**

Now, a block contains transactions and each block contains the hash of a previous block. Any change in the data of any block will affect the hash values of all the blocks after that block and they will become invalid.

Using the online tool, you can simulate a blockchain with 5 blocks. Figure 6 shows you a blockchain in which block# 5 has some data in the **Data** text box:

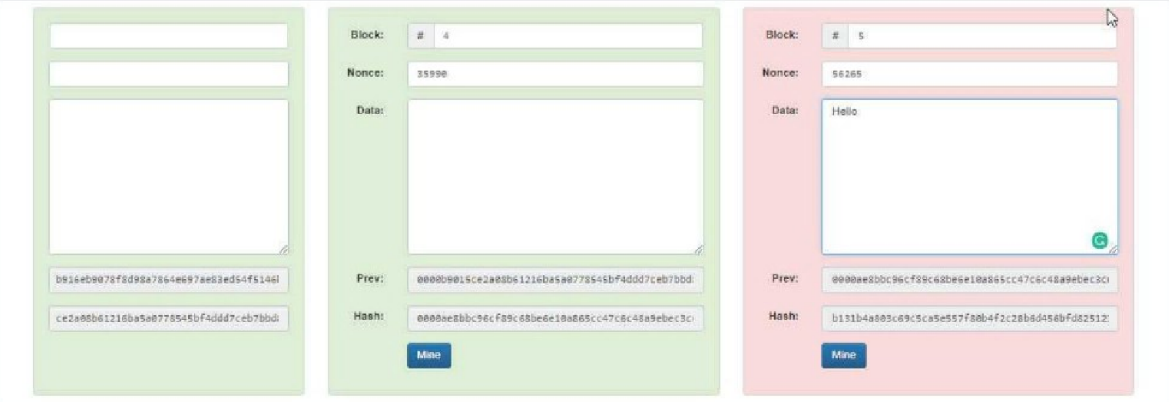

Figure 6: **A Block**

Now, as soon as you click the **Mine** button, the hash value of the block will be changed.

Suppose, you make some changes in block# 3. It will make blocks# 4 and 5 invalid, as shown in Figure 7:

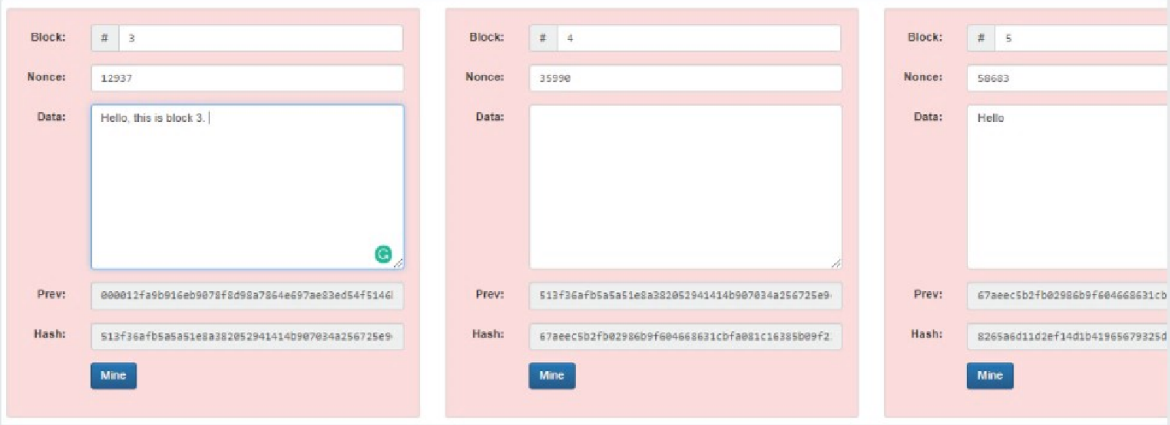

Figure 7: **Invalid Blockchain**

Now, you are required to mine block# 3, 4 and 5 blocks. It means that if any block in the past is modified, then it will make all the succeeding blocks invalid.

If you want to check if the previous blocks or any block is remined, then it can be confirmed by the blockchains of other nodes in the blockchain network. Suppose there are some changes in the blockchain of Peer A and the blockchain is remined to make it valid. If you check the blockchain of Peer B on the same network, then you will find it is different.

Figures 8 and 9 show the blockchain on Peer A and Peer B:

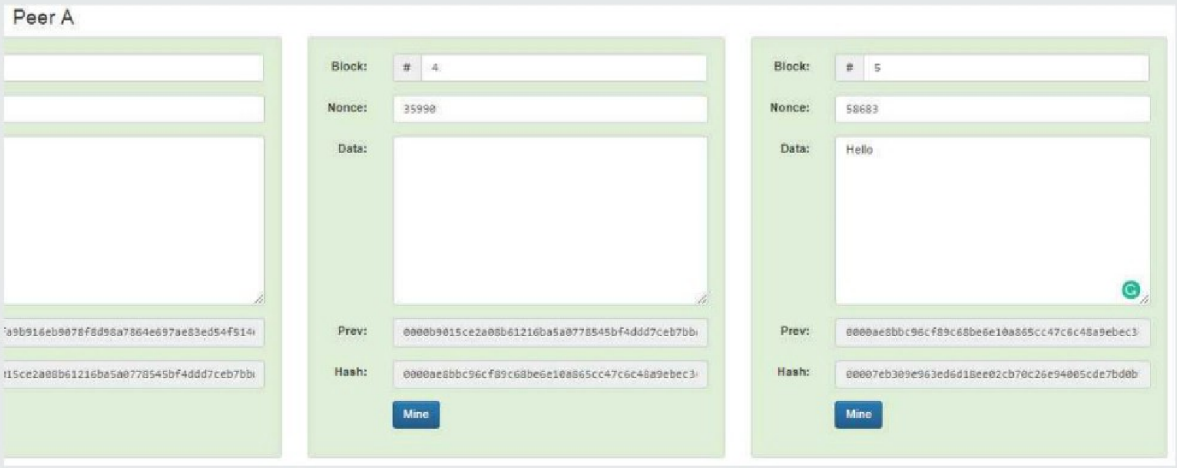

Figure 8: **Peer A Blockchain**

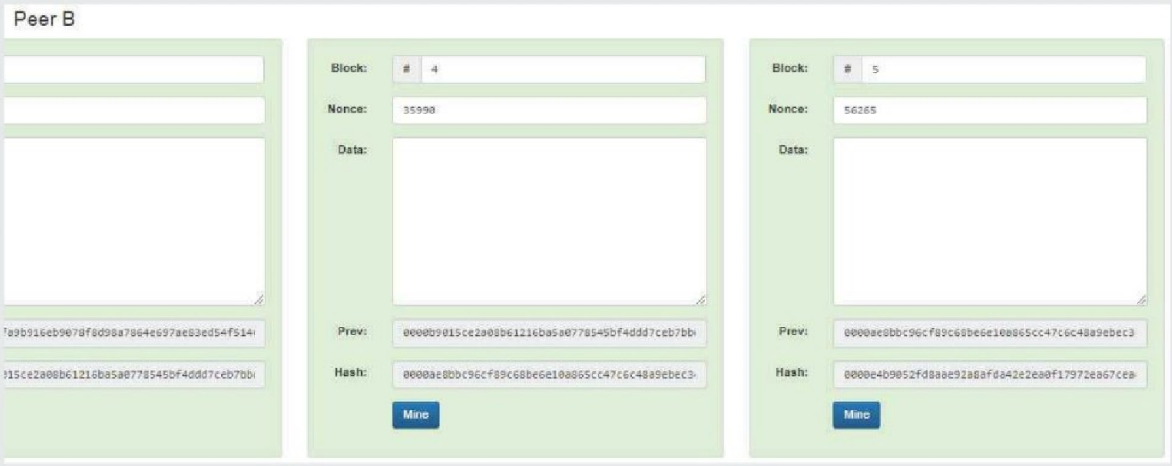

Figure 9: **Peer B Blockchain**

As shown in Figure 8 and 9, the hashes of block# 5 in the blockchains of Peer A and Peer B are different. If you check against the hashes of block# 5 in the blockchain of Peer C, you will see the hash value matches with hash value of Peer B. It means that the blockchain of Peer B is correct. This is how a distributed copy of blockchain works.

As you have seen, there is no information in the Data section. Therefore, this section is not useful. Now, let's move on to the token section. As shown in Figure 10, now there are some transactions instead of the Data section:

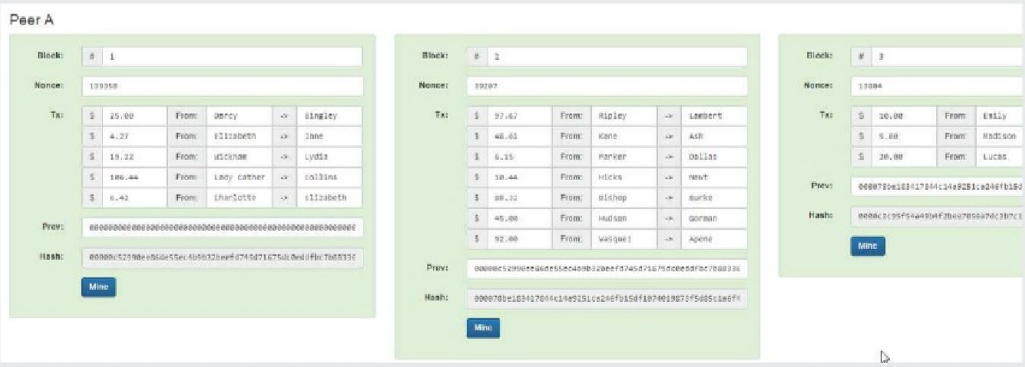

Figure 10: **Peer A Blockchain with Some Transactions**

Every block has different number of transactions in the following form:

$ 25.00 From: Darcy -> Bingley

It means that 25 dollars are sent from Darcy to Bingley.

Now, if you check the blockchain of Peer B, you will find that it also has the same transactions, as shown in Figure 11:

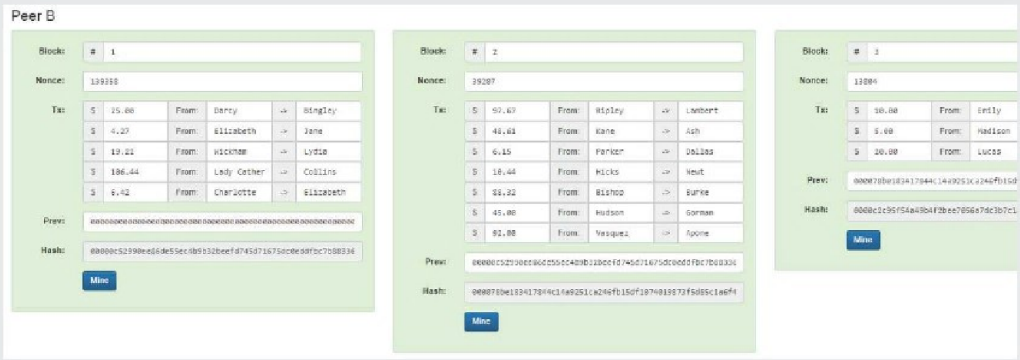

Figure 11: **Blockchain on Peer B**

If you change some value in block# 4 of Peer A, you will find that the hashes on blockchain of Peer B will be different than that of Peer A. Figure 12 shows the change in block# 4 of Peer A:

Figure 12: **Change in Block# 4 of Peer A**

Figure 13 shows the blockchain of Peer B:

Figure 13: **Blockchain of Peer B**

If you compare the hashes of block# 4 in Peer A and Peer B, you will find them different. This helps in preventing any loss of track and resisting any modifications which could have been done in the past. This is one of the reasons why tokens are used in blockchains.

As you have seen, the transactions only list that 25 dollars are sent from Darcy to Bingley. However, it does not state that Darcy has 25 dollars. This is the problem with this version of the blockchain. It does not remember the account balance. It only records the money transactions. Now, let's move on to the Coinbase transactions.

As shown in Figure 14, there is a Coinbase transaction in block# 1 of Peer A:

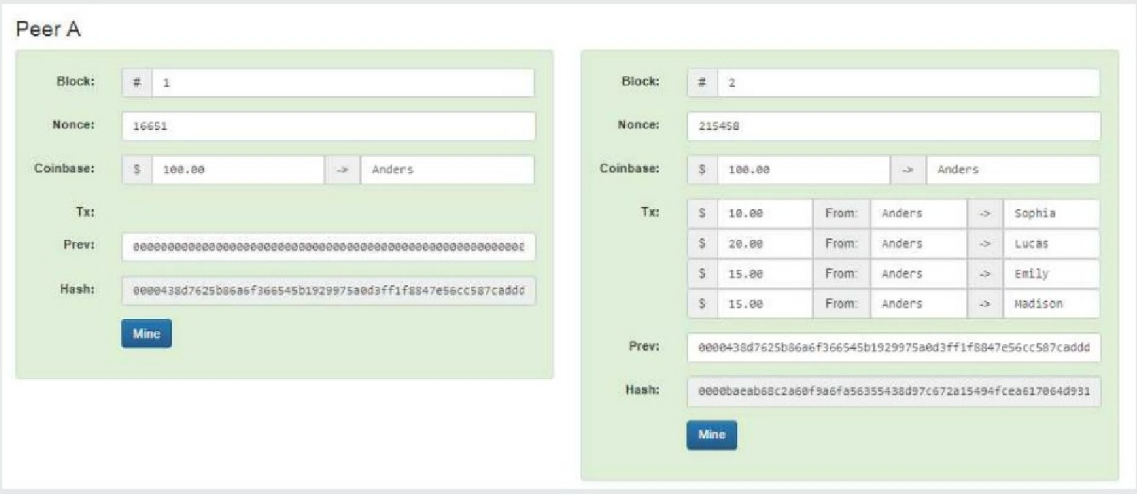

Figure 14: **Coinbase Transaction in Block# 1 of Peer A**

The Coinbase transaction states that 100 dollars are assigned to Anders. As shown in Figure 14, there are no other transactions in the block because no one had any money before this transaction.

As shown in Figure 14, block# 2 also has a Coinbase transaction which states that 100 dollars are assigned to Anders and it also has some other transactions where Anders is transferring money to Sophia, Lucas, Emily and Madison because only Anders has money. Now, while transferring 10 dollars to Sophia, you can check that Anders has 10 dollars because block# 1 has a Coinbase transaction.

Now, as shown in Figure 15, block# 5 has a transaction where Jackson has sent 2 dollars to Alexander:

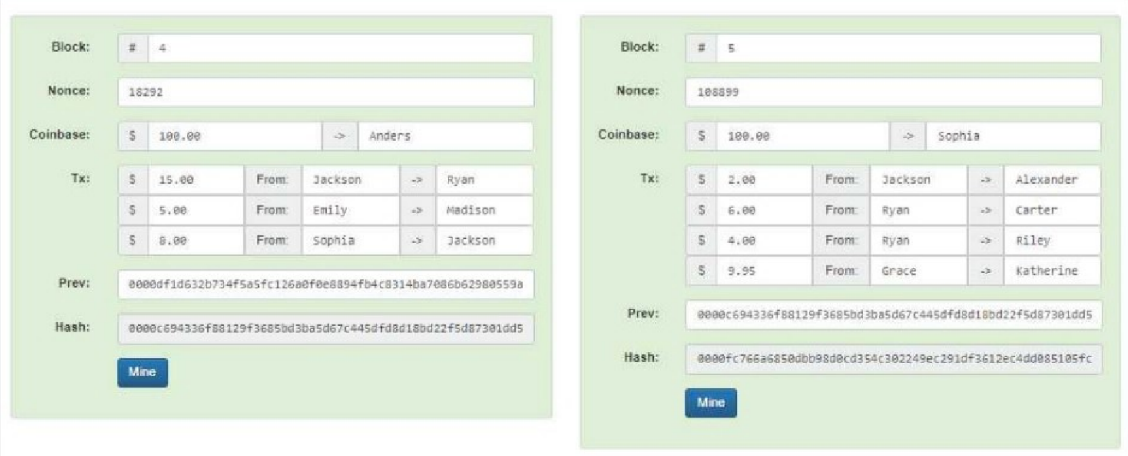

Figure 15: **Blockchain on Peer A**

If you want to check if Jackson has 2 dollars, then you can move backwards using the hash of the previous block and see that Sophia has transferred 8 dollars to Jackson. This allows you to trace the provenance of any coin you want.

Now, as you know, there are many copies of this blockchain on the network. If you change this blockchain, then it will become invalid as it will not agree with the blockchains on other peers of the network.

Introducing Blockchain Explorer

Discuss blockchain explorer]

A blockchain explorer is a tool or a website which is used to browse blocks of a blockchain. It also provides information about network wallet addresses, network hash rate, transaction data and other information about blockchain.

A blockchain explorer is just like a web browser. A web browser is used to browse the Internet, similarly, a blockchain explorer is used to browse the blocks of a blockchain. Bitcoin and Altcoin users depend on blockchain explorers to track their transactions. However, these explorers provide more information than just tracking transactions.

Every cryptocurrency has its own blockchain explorer. For example, you cannot track Bitcoin through Ethereum blockchain explorer. Figure 16 shows Ethereum blockchain explorer:

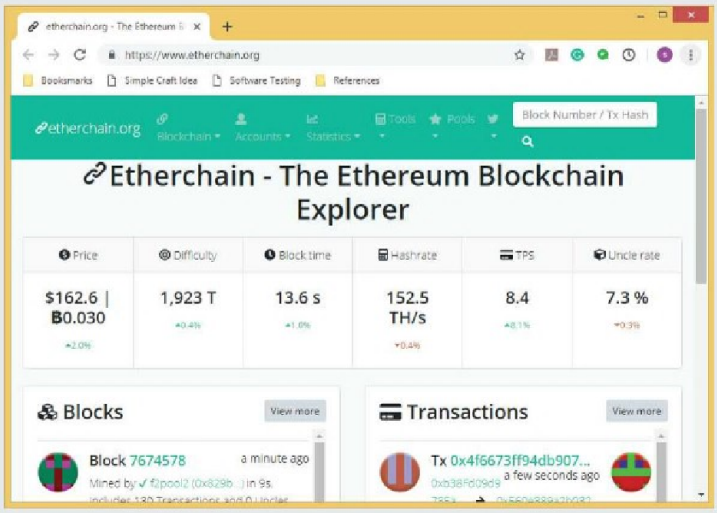

Figure 16: **Ethereum Blockchain Explorer**

A blockchain explorer can be used for the following:

- **Blocks feed**: It allows the user to explore recently mined blocks on a blockchain.
- **Transaction feed:** It also allows the user to view any transaction which has been already mined. Figure 17 shows you the blocks feed and transaction feed in Ethereum blockchain explorer:

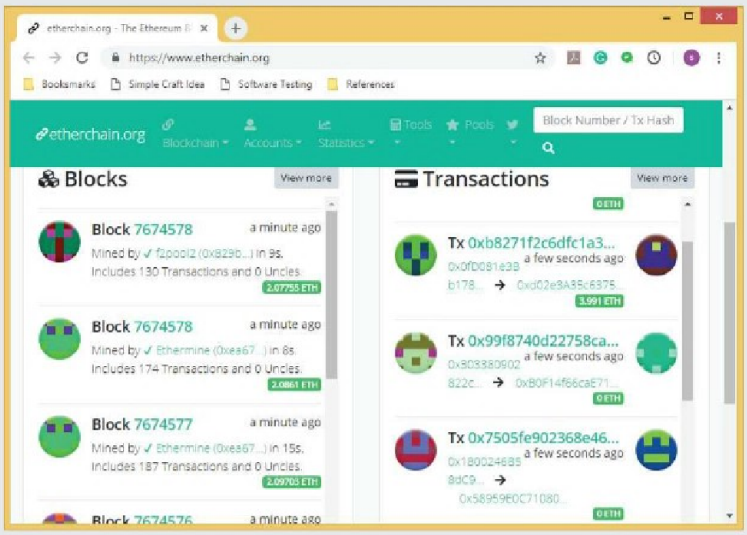

Figure 17: **Blocks Feed and Transaction Feed**

- **Transaction history of an address**: It allows the user to view the history of any public Bitcoin address.
- **Largest transaction**: Some blockchain explorers also allow the user to view the largest transactions performed in a day.

- **Transaction pool status**: It allows the user to view the status of a transaction pool where the total number of unconfirmed transactions can be viewed.

Let's now explore a public bitcoin block on a blockchain explorer using the following steps.

1. Open a web browser and type the following URL in the address bar:

   https://www.blockchain.com/explorer

   Figure 18 shows the web page of the blockchain explorer:

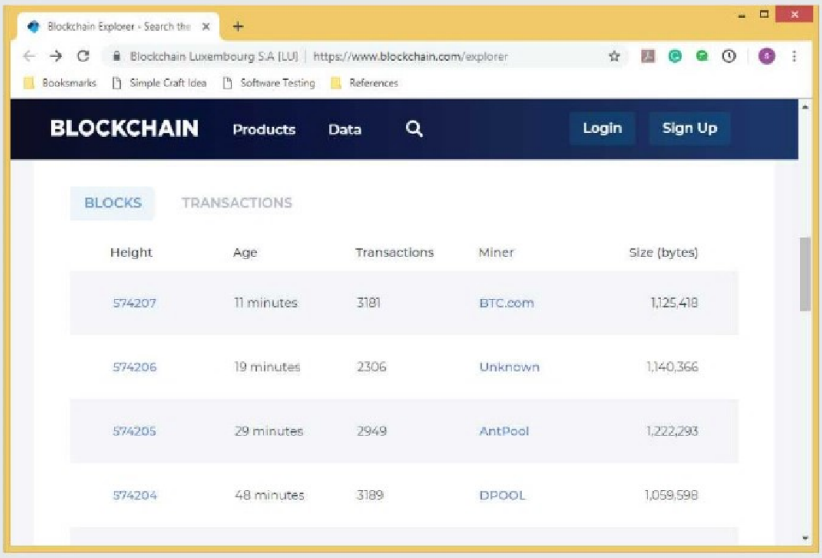

Figure 18: **Web Page of Blockchain Explorer**

2. Type height number as 574208 in the search box to search the block (Figure 19).
3. Click the BCH link for the block, as shown in Figure 19:

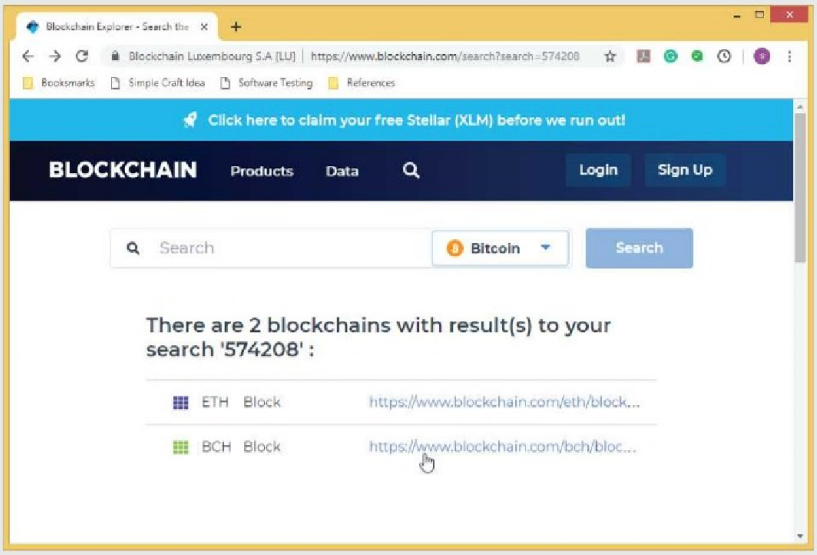

Figure 19: **Search Results**

Figures 20 (a) and 20 (b) show the details of the block:

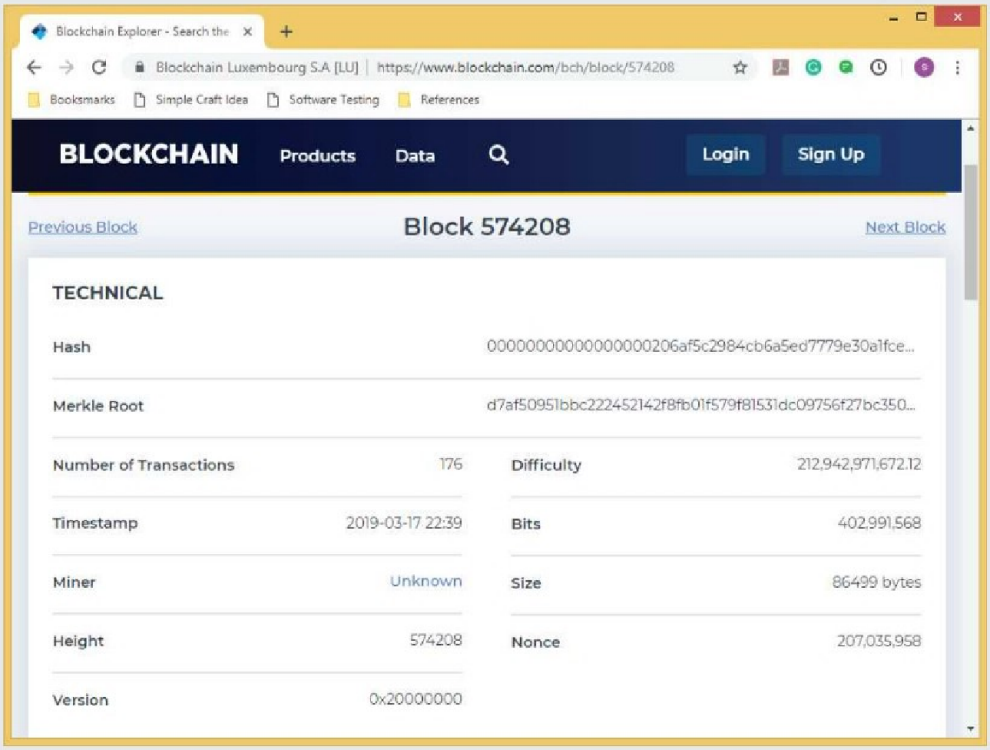

Figure 20: (a) **Details of Blockchain**

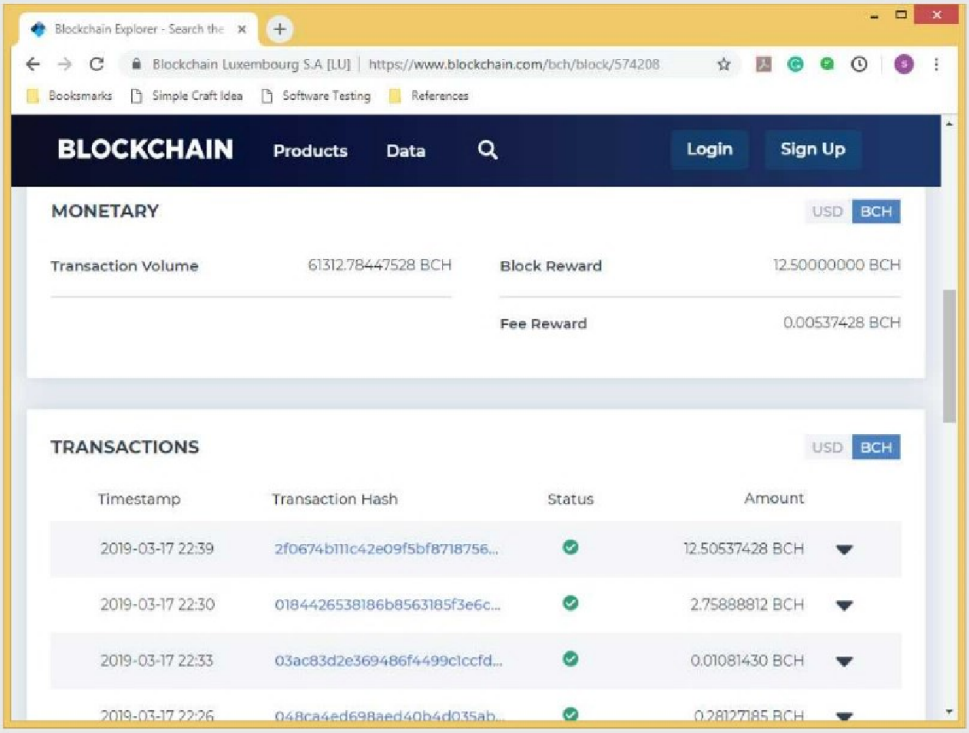

Figure 20: (b) **Details of Blockchain**

As you can see in Figures 20 (a) and 20 (b), you can view the number of transactions, estimated transaction volume, transaction fees, size, version, timestamp, miner, nonce, Merkle root, and the hash of the previous block.

4. Click any transaction to view the transaction details. Figure 21 shows the details of a transaction:

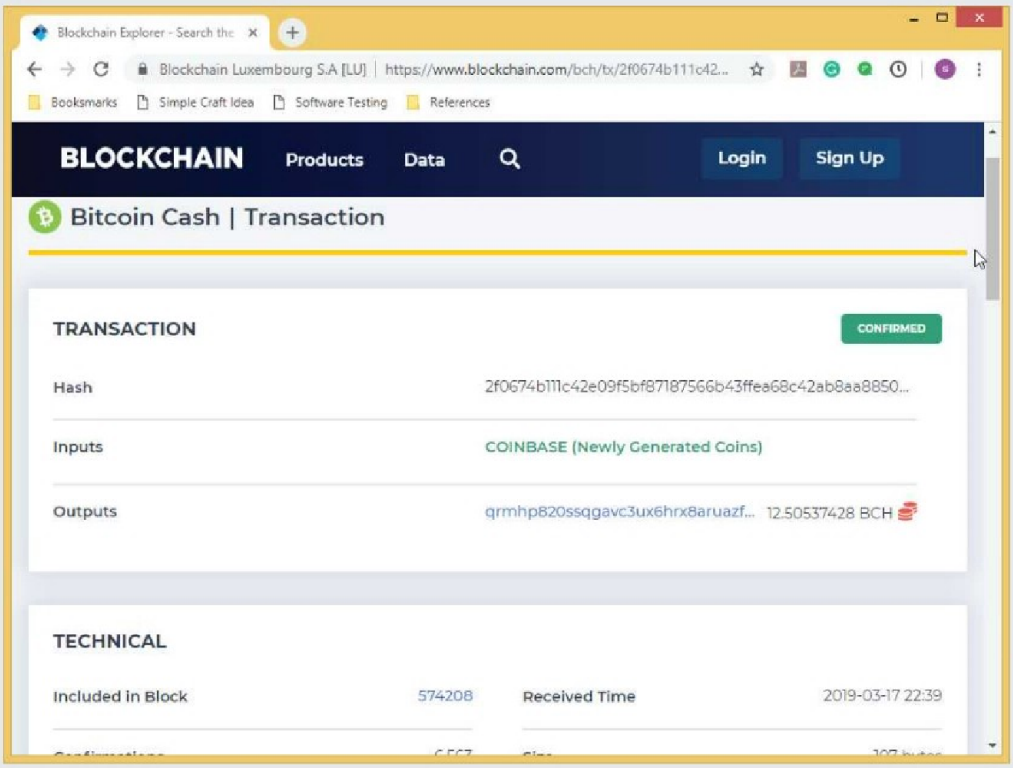

Figure 21: **Transaction Details**

SUMMARY

- A blockchain can be either stored in a flat file or in a database.
- Each block has the hash of the previous block that helps in linking all the blocks to the initial block known as the genesis block.
- A block is made up of a header and a body. The header contains the metadata which includes information about the data stored in the header.
- A Merkle tree is a binary hash tree that is used to summarize the transactions and to ensure the integrity of the transactions.
- A transaction pool is used to store unconfirmed transactions that are currently stuck in the Bitcoin network.
- A cryptocurrency is defined as a tradeable digital form of money which exists only online and is built on blockchain technology.
- A blockchain explorer is like a web browser which is used to browse the blocks of a blockchain.