**YOU DON’T KNOW BITCOIN**

**INTRODUCTION**

Bitcoin was the first application built on the blockchain technology and has revolutionalized with the introduction of a fully decentralized cryptocurrency called bitcoin (notice the small **b**).

NB: Bitcoin with capital **B** is to refer to the Bitcoin blockchain or protocol while bitcoin with small **b** is to refer to the digital currency or cryptocurrency. This cryptocurrency is transacted on the Bitcoin blockchain from one node to another, is used to reward miners etc.

In 2008, Bitcoin was introduced in a paper called Bitcoin: A Peer-to-Peer Electronic Cash System. The true identity of the author is unknown.

Link to the Satoshi paper: <https://bitcoin.org/bitcoin.pdf>

Satoshi Nakamoto withdrew from the public in April 2011, leaving the responsibility of developing the code and network to a group of volunteers.

However, neither Satoshi Nakamoto nor anyone else exerts individual control over the Bitcoin system. It operates based on fully transparent mathematical principles, open source code, and consensus among participants.

There are no physical coins or even digital coins per se. The coins are implied in transactions that transfer value from sender to recipient.

**WHAT IS BITCOIN?**

Bitcoin is:

1. A decentralized peer-to-peer network (Bitcoin protocol)
2. A distributed ledger (Bitcoin blockchain)
3. A set of rules for independent transaction validation and currency issuance (Consensus mechanism)
4. A mechanism for reaching global decentralized consensus on the valid blockchain (Proof of Work)

The bitcoin protocol includes built-in algorithms that regulate the mining function across the network. The difficulty of the processing task that miners must perform is

adjusted dynamically so that, on average, someone succeeds every 10 minutes regardless of how many miners (and how much processing) are competing at any moment.

The protocol also halves the rate at which new bitcoin are created every 4 years, and limits the total number of bitcoin that will be created to a fixed total just below 21

million coins. The result is that the number of bitcoin in circulation closely follows an easily predictable curve that approaches 21 million by the year 2140. Due to bitcoin’s

diminishing rate of issuance, over the long term, the bitcoin currency is deflationary. Furthermore, bitcoin cannot be inflated by “printing” new money above and beyond

the expected issuance rates.

Bitcoin aims to solve problems that other digital currencies before it such b-money, hashcash, Bitcoin Gold etc. couldn’t overcome:

1. Byzantine General’s Problem: In a nutshell, the problem consists of trying to agree on a course of action or the state of a system by exchanging information over an unreliable and potentially compromised network. He solved this problem by the using the concept of Proof of Work to achieve consensus without a trusted central authority.
2. Double-spending problem: briefly, this is when a user sends coins to two different users at the same time and they are verified independently as a valid transaction. This problem is solved by using a distributed ledger (blockchain) where every transaction is recorded permanently and by implementing a transaction validation and confirmation mechanism.

**Main Components of a Bitcoin.**

1. Digital keys
2. Addresses
3. Transactions
4. Blockchain
5. Miners
6. The Bitcoin Network
7. Wallets

**HOW THE BITCOIN BLOCKCHAIN WORKS? – OVERVIEW**

1. The transaction starts with a sender signing the transaction with their private key
2. The transaction is serialized so that it can be transmitted over the network
3. The transaction is broadcast to the network
4. Miners listening for transaction pick up the transaction
5. The transaction is verified for its legitimacy by the miners
6. The transaction is added to the candidate block for mining
7. Once mined, the result is broadcast to all nodes on the Bitcoin network.
8. Usually, users wait for confirmations to be received before a transaction is considered final.

The smallest Bitcoin denomination is the Satoshi.

**KEYS AND ADDRESSES**

On the Bitcoin network, possession of Bitcoins and the transfer of value via transactions are reliant upon private keys, public keys and addresses.

Elliptic Curve Cryptography (ECC) is used to generate public and private key pairs in the Bitcoin network.

**PRIVATE KEYS**

Private keys are required to be kept safe and normally reside only on the owner’s side. They’re used to digitally sign the transactions, proving ownership of the bitcoins.

Private keys have their corresponding public keys. Public keys on their own are useless as well as private keys.

Pairs of public and private keys are required for the normal functioning of any public key cryptography based systems such as Bitcoin blockchain.

**PUBLIC KEYS IN BITCOIN**

Public keys exist on the blockchain and all nodes can see them. They’re derived from private keys due to their special mathematical relationship with those private keys. Once a transaction signed with the private key is broadcast on the network, public keys are used by nodes to verify that the transaction has indeed been signed with the corresponding private key.

**ADDRESSES IN BITCOIN**

A Bitcoin address is created by taking the corresponding public key of a private key and hashing it twice, first with the SHA256 algorithm and then with RIPEMD160.

Bitcoin addresses are 26-35 characters long and begin with digits 1 or 3. Give example:

**HOW BITCOIN ADDRESS IS GENERATED?**

1. In the first step, we have a randomly generated ECDSA private key.

2. The public key is derived from the ECDSA private key.

3. The public key is hashed using the SHA-256 cryptographic hash function.

4. The hash generated in *step 3* is hashed using the RIPEMD-160 hash function.

5. The version number is prefixed to the RIPEMD-160 hash generated in *step 4*.

6. The result produced in *step 5* is hashed using the SHA-256 cryptographic hash function.

7. SHA-256 is applied again.

8.The first 4 bytes of the result produced from *step 7* is the address checksum.

9. This checksum is appended to the RIPEMD-160 hash generated in *step 4*.

10. The resultant byte string is encoded into a Base58-encoded string by applying the Base58 encoding function.

11. Finally, the result is a typical Bitcoin address.

**TRANSACTIONS**

Transactions are at the core of the Bitcoin ecosystem. Transactions can be as simple as just sending some bitcoins to a Bitcoin address, or can be quite complex, depending on the requirements. Each transaction is composed of at least one input and output. Inputs can be thought of as coins being spent that have been created in a previous transaction, and outputs as coins being created. If a transaction is minting new coins, then there is no input, and therefore no signature is needed. If a transaction should send coins to some other user (a Bitcoin address), then it needs to be signed by the sender with their private key. In this case, a reference is also required to the previous transaction to show the origin of the coins. Coins are unspent transaction outputs represented in Satoshis.

Transactions are not encrypted and are publicly visible on the blockchain. Blocks are made up of transactions, and these can be viewed using any online blockchain explorer.

**THE TRANSACTION LIFECYCLE**

Lifecycle of a Bitcoin transaction in the following steps:

1. A user/sender sends a transaction using wallet software or some other interface.

2. The wallet software signs the transaction using the sender's private key.

3. The transaction is broadcasted to the Bitcoin network using a flooding algorithm.

4. Mining nodes (miners) who are listening for the transactions verify and include this transaction in the next block to be mined. Just before the transactions are placed in the block, they are placed in a special memory buffer called the transaction pool. The purpose of the transaction pool is explained in the next section.

5. Next, the mining starts, which is the process through which the blockchain is secured and new coins are generated as a reward for the miners who spend appropriate computational resources. Once a miner solves the PoW problem, it broadcasts the newly mined block to the network. PoW is explained in detail in the *Mining* section. The nodes verify the block and propagate the block further, and confirmations start to generate.

6. Finally, the confirmations start to appear in the receiver's wallet and after approximately three confirmations, the transaction is considered finalized and confirmed. However, three to six is just a recommended number; the transaction can be considered final even after the first confirmation. The key idea behind waiting for six confirmations is that the probability of double spending is virtually eliminated after six confirmations.

When a transaction is created by a user and sent to the network, it ends up in a special area on each Bitcoin software client. This special area is called the transaction pool or memory pool.

**TRANSACTION POOL**

Also known as memory pools, these pools are basically created in local memory (computer RAM) by nodes (Bitcoin clients) in order to maintain a temporary list of transactions that have not yet been added to a block. Miners pick up transactions from these memory pools to create candidate blocks. Miners select transactions from the pool after they pass the verification and validity checks. The selection of which transactions to choose is based on the fee and their place in the order of transactions in the pool. Miners prefer to pick up transactions with higher fees.

To send transactions on the Bitcoin network, the sender needs to pay a fee to the miners. This fee is an incentive mechanism for the miners.

To send transactions on the Bitcoin network, the sender needs to pay a fee to the miners. This fee is an incentive mechanism for the miners.

**TRANSACTION FEES**

Transaction fees are charged by the miners. The fee charged is dependent upon the size and weight of the transaction. Transaction fees are calculated by subtracting the sum of the inputs from the sum of the outputs.

The fees are used as an incentive for miners to encourage them to include users' transactions in the block the miners are creating. All transactions end up in the memory pool, from where miners pick up transactions based on their priority to include them in the proposed block. The calculation of priority is introduced later in this chapter; however, from a transaction fee point of view, a transaction with a higher fee will be picked up sooner by the miners. There are different rules based on which fee is calculated for various types of actions, such as sending transactions, inclusion in blocks, and relaying by nodes.

Fees are not fixed by the Bitcoin protocol and are not mandatory; even a transaction with no fee will be processed in due course, but may take a very long time. This is, however, no longer practical due to the high volume of transactions and competing investors on the Bitcoin network, therefore it is advisable to always provide a fee. The time for transaction confirmation usually ranges from 10 minutes to over 12 hours in some cases. Transaction time is dependent on transaction fees and network activity. If the network is very busy, then naturally, transactions will take longer to process, and if you pay a higher fee then your transaction is more likely to be picked by miners first due to the additional incentive of the higher fee.

A transaction on the Bitcoin network is represented by a data structure that consists of several fields. We will now introduce the transaction data structure.

**TRANSACTION DATA STRUCTURE**

1. Metadata: This part of the transaction contains values such as the size of the transaction, the number of inputs and outputs, the hash of the transaction, and a lock\_time field. Every transaction has a prefix specifying the version number
2. Inputs: Generally, each input spends a previous output. Each output is considered an **Unspent Transaction Output** (**UTXO**) until an input consumes it. A UTXO is an unspent transaction output that can be spent as an input to a new transaction.
3. Outputs: Outputs have three fields, and they contain instructions for sending bitcoins. The first field contains the amount of Satoshis, whereas the second field contains the size of the locking script. Finally, the third field contains a locking script that holds the conditions that need to be met in order for the output to be spent. More information on transaction spending using locking and unlocking scripts and producing outputs is discussed later in this section.

**VERIFICATION**

Verification is performed using Bitcoin's scripting language, which we will now describe in the next section in detail.

**The Script language**

Bitcoin uses a simple stack-based language called **Script** to describe how bitcoins can be spent and transferred. It is not Turing complete and has no loops to avoid any undesirable effects of long-running/hung scripts on the Bitcoin network. This scripting language is based on a Forth programming language-like syntax and uses a reverse polish notation in which every operand is followed by its operators. It is evaluated from left to right using a **Last in, First Out** (**LIFO**) stack.

Scripts are composed of two components, namely elements and operations. Scripts use various operations (**opcodes**) or instructions to define their operations. Elements simply represent data such as digital signatures. Opcodes are also known as words, commands, or functions

Scripting is quite limited and can only be used to program one thing—the transfer of bitcoins from one address to other addresses. However, there is some flexibility possible when creating these scripts, which allows for certain conditions to be put on the spending of the bitcoins. This set of conditions can be considered a basic form of a financial contract. In other words, we can say that there is a basic support for building contracts in Bitcoin. However, it is not same as **smart contracts**, which allow the writing of arbitrary programs on the blockchain.

**CONTRACTS**

Contracts are Bitcoin scripts that use the Bitcoin blockchain to enforce a financial agreement. This is a simple definition but has far-reaching consequences as it allows users to programmatically create complex contracts that can be used in many real-world scenarios. Contracts allow the development of completely decentralized, independent, and reduced-risk platforms by programmatically enforcing different conditions for unlocking the money (bitcoins). With the security guarantees provided by the Bitcoin blockchain, it is almost impossible to circumvent these conditions. Various contracts, such as escrow, arbitration, and micropayment channels, can be built using the Bitcoin scripting language.

**COINBASE TRANSACTION**

A coinbase transaction or generation transaction is always created by a miner and is the first transaction in a block. It is used to create new coins. It includes a special field, also called the coinbase, which acts as an input to the coinbase transaction.

**TRANSACTION VALIDATION**

Steps performed by Bitcoin nodes to verify transactions.

1. That transaction inputs are previously unspent. This validation step prevents double spending by verifying that the transaction inputs have not already been spent by someone else.

2. That the sum of the transaction outputs is not more than the total sum of the transaction inputs. However, both input and output sums can be the same, or the sum of the input (total value) could be more than the total value of the outputs. This check ensures that no new bitcoins are created out of thin air.

3. That the digital signatures are valid, which ensures that the script is valid.

**BITCOIN BLOCKCHAIN**

As we discussed in the first module: You Don’t Know Blockchain: Introduction, a blockchain is a distributed ledger of transactions. Specifically, from Bitcoin's perspective, the blockchain can be defined as a public, distributed ledger holding a timestamped, ordered, and immutable record of all transactions on the Bitcoin network. Transactions are picked up by miners and bundled into blocks for mining. Each block is identified by a hash and is linked to its previous block by referencing the previous block's hash in its header.

**DATA STRUCTURE OF A BITCOIN BLOCK**

1. Block size: The size of the block
2. Block header: Includes fields from the block header
3. Transaction counter: Contains the total number of transactions in the block.
4. Transactions: All transactions in the block
5. Version: Block version number that dictates the validation rules to follow
6. Previous Block’s header hash: This is a double SHA256 hash of the previous block’s header
7. Merkle root hash
8. Timestamp
9. Difficulty target: current difficulty target of the network/block.
10. Nonce: The number miners change repeatedly to produce a hash that is lower than the difficulty target

**THE GENESIS BLOCK**

This is the first block in the Bitcoin blockchain. The genesis block was hardcoded in the Bitcoin core software.

Bitcoin provides protection against double-spending by enforcing strict rules on transaction verification and via mining. Transactions and blocks are added to the blockchain only after the strict rule-checking explained in the Transaction validation section on successful PoW solutions.

**STALE AND ORPHAN BLOCKS**

Stale blocks are old blocks that have already been mined. Miners who keep working on these blocks due to a fork, where the longest chain (main chain) has already progressed beyond those blocks, are said to be working on a stale block. In other words, these blocks exist on a shorter chain, and will not provide any reward to their miners.

Orphan blocks are a slightly different concept. Their parent blocks are unknown. As their parents are unknown, they cannot be validated. This problem occurs when two or more miners discover a block at almost the same time. These are valid blocks and were correctly discovered at some point in the past but now they are no longer part of the main chain. The reason why this occurs is that if there are two blocks discovered at almost the same time, the one with a larger amount of PoW will be accepted and the one with the lower amount of work will be rejected. Similar to stale blocks, they do not provide any reward to their miners.

**FORK**

A fork is a condition that occurs when two different versions of the blockchain exist.

**TYPES OF FORK**

1. Temporary forks

2. Soft forks

3. Hard forks

Forks in a blockchain can also occur with the introduction of changes to the Bitcoin protocol. In the case of a **soft fork**, a client that chooses not to upgrade to the latest version supporting the updated protocol will still be able to work and operate normally. In this case, new and previous blocks are both acceptable, thus making a soft fork backward compatible. Miners are only required to upgrade to the new soft fork client software in order to make use of the new protocol rules. Planned upgrades do not necessarily create forks because all users should have updated already.

A **hard fork**, on the other hand, invalidates previously valid blocks and requires all users to upgrade. New transaction types are sometimes added as a soft fork, and any changes such as block structure changes or major protocol changes result in a hard fork.

**SIZE OF THE BITCOIN BLOCKCHAIN**

Bitcoin is an ever-growing chain of blocks and is increasing in size.

Research the current bitcoin blockchain size.

**NETWORK DIFFCULTY**

Network difficulty refers to a measure of how difficult it is to find a new block, or in other words, how difficult it is to find a hash below the given target.

New blocks are added to the blockchain approximately every 10 minutes, and the network difficulty is adjusted dynamically every 2,016 blocks in order to maintain a steady addition of new blocks to the network.

Network difficulty is calculated using the following equation:

Target = Previous target \* Time/2016 \* 10 minutes

Difficulty and target are interchangeable and represent the same thing. The Previous target represents the old target value, and Time is the time spent to generate the previous 2,016 blocks. Network difficulty essentially means how hard it is for miners to find a new block; that is, how difficult the hashing puzzle is now.

**MINING**

Mining is a process by which new blocks are added to the blockchain. Blocks contain transactions that are validated via the mining process by mining nodes on the Bitcoin network. Blocks, once mined and verified, are added to the blockchain, which keeps the blockchain growing. This process is resource-intensive due to the requirements of PoW, where miners compete to find a number less than the difficulty target of the network. This difficulty in finding the correct value (also called sometimes the **mathematical puzzle**) is there to ensure that miners have spent the required resources before a new proposed block can be accepted. The miners mint new coins by solving the PoW problem, also known as the partial hash inversion problem. This process consumes a high amount of resources, including computing power and electricity. This process also secures the system against fraud and double-spending attacks while adding more virtual currency to the Bitcoin ecosystem.

Roughly one new block is created (mined) every 10 minutes to control the frequency of generation of bitcoins. This frequency needs to be maintained by the Bitcoin network. It is encoded in the Bitcoin Core client to control the "money supply."

Approximately 144 blocks, that is, 1,728 bitcoins, are generated per day. The number of actual coins can vary per day; however, the number of blocks remains at an average of 144 per day. Bitcoin supply is also limited. In 2140, all 21 million bitcoins will be finally created, and no new bitcoins can be created after that. Bitcoin miners, however, will still be able to profit from the ecosystem by charging transaction fees.

**WHAT MINERS DO**

1.**Synching up with the network**: Once a new node joins the Bitcoin network, it downloads the blockchain by requesting historical blocks from other nodes. This is mentioned here in the context of the Bitcoin miner; however, this not necessarily a task that only concerns miners.

2. **Transaction validation**: Transactions broadcast on the network are validated by full nodes by verifying and validating signatures and outputs.

3. **Block validation**: Miners and full nodes can start validating blocks received by them by evaluating them against certain rules. This includes the verification of each transaction in the block along with verification of the nonce value.

4. **Create a new block**: Miners propose a new block by combining transactions broadcast on the network after validating them.

5. **Perform PoW**: This task is the core of the mining process and this is where miners find a valid block by solving a computational puzzle. The block header contains a 32-bit nonce field and miners are required to repeatedly vary the nonce until the resultant hash is less than a predetermined target.

6. **Fetch reward**: Once a node solves the hash puzzle (PoW), it immediately broadcasts the results, and other nodes verify it and accept the block. There is a slight chance that the newly minted block will not be accepted by other miners on the network due to a clash with another block found at roughly the same time, but once accepted, the miner is rewarded with 12.5 bitcoins and any associated transaction fees.

**MINING REWARDS**

Miners are rewarded with new coins if and when they discover new blocks by solving the PoW. Miners are paid transaction fees in return, for the transactions in their proposed blocks. New blocks are created at an approximate fixed rate of every 10 minutes. The rate of creation of new bitcoins decreases by 50% every 210,000 blocks, which is roughly every 4 years. When Bitcoin started in 2009, the mining reward used to be 50 bitcoins. After every 210,000 blocks, the block reward halves. In November 2012 it halved down to 25 bitcoins. Currently, since May 2020, it is 6.25 bitcoins per block. This mechanism is hardcoded in Bitcoin to regulate and control inflation and limit the supply of bitcoins.

In order for miners to earn the reward, they have to show that they have solved the computational puzzle. This is called the PoW.

**PROOF OF WORK**

This is a proof that enough computational resources have been spent in order to build a valid block. PoW is based on the idea that a random node is selected every time to create a new block. In this model, nodes compete with each other in proportion to their computing capacity, in order to be selected.

**THE MINING ALGORITHM OR HOW MINING IS DONE**

The mining algorithm consists of the following steps:

1. The previous block's header is retrieved from the Bitcoin network.

2. Assemble a set of transactions broadcast on the network into a block to be proposed.

3. Compute the double hash of the previous block's header, combined with a nonce and the newly proposed block, using the SHA256 algorithm.

4. Check if the resulting hash is lower than the current difficulty level (the target), then PoW is solved. As a result of successful PoW, the discovered block is broadcasted to the network and miners fetch the reward.

5. If the resultant hash is not less than the current difficulty level (target), then repeat the process after incrementing the nonce.

The reason why mining difficulty increases is because, in Bitcoin, the block generation time always has to be around 10 minutes. This means that if blocks are being mined too quickly because of faster hardware, the difficulty increases accordingly so that the block generation time can remain at roughly 10 minutes per block. This phenomenon is also true in reverse. If blocks take longer than 10 minutes on average to mine, then the difficulty is decreased. The difficulty is calculated every 2,016 blocks (around two weeks) and adjusted accordingly. If the previous set of 2,016 blocks were mined in a period of less than two weeks, then the difficulty increases. Similarly, if 2,016 blocks were found in more than two weeks (bearing in mind that if blocks are mined every 10 minutes, then 2,016 blocks take 2 weeks to be mined), then the difficulty is decreased.

The Bitcoin miners have to calculate hashes to solve the PoW algorithm. If the difficulty goes up then a higher hash rate is required to find the blocks. The difficulty increases accordingly if more hashing power is added due to more miners joining the network.

**THE HASH RATE**

The hash rate basically represents the rate of hash calculation per second. In other words, this is the speed at which miners in the Bitcoin network are calculating hashes to find a block.

In the early days of Bitcoin, it used to be quite small, as CPUs were used, which are relatively weak in mining terms. However, with dedicated mining pools and **Application Specific Integrated Circuits** (**ASICs**) now, this has gone up exponentially in the last few years. This has resulted in increased difficulty in the Bitcoin network.

**MINING SYSTEMS**

Over time, Bitcoin miners have used various methods to mine bitcoins. As the core principle behind mining is based on the double SHA256 algorithm, over time, experts have developed sophisticated systems to calculate the hash faster and faster. The following is a review of the different types of mining methods used in Bitcoin and how they evolved with time.

### CPU

CPU mining was the first type of mining available in the original Bitcoin client. Users could even use laptop or desktop computers to mine bitcoins. CPU mining is no longer profitable and now more advanced mining methods such as ASIC-based mining are used. CPU mining only lasted for around a year from the introduction of Bitcoin, and soon other methods were explored and tried by the miners.

### GPU

Due to the increased difficulty of the Bitcoin network and the general tendency of finding faster methods to mine, miners started to use the GPUs or graphics cards available in PCs to perform mining. GPUs support faster and parallelized calculations that are usually programmed using the OpenCL language.

This turned out to be a faster option as compared to CPUs. Users also used techniques such as overclocking to gain maximum benefit of the GPU power. Also, the possibility of using multiple graphics cards in parallel increased the popularity of graphics cards' usage for Bitcoin mining. GPU mining, however, has some limitations, such as overheating and the requirement for specialized motherboards and extra hardware to house multiple graphics cards. From another angle, graphics cards have become quite expensive due to increased demand and this has impacted gamers and graphics software users.

### FPGA

Even GPU mining did not last long, and soon miners found another way to perform mining using **Field Programmable Gate Arrays** (**FPGAs**). An FPGA is basically an integrated circuit that can be programmed to perform specific operations. FPGAs are usually programmed in **hardware description languages** (**HDLs**), such as Verilog and VHDL. Double SHA256 quickly became an attractive programming task for FPGA programmers and several open source projects were started too. FPGA offered much better performance compared to GPUs; however, issues such as accessibility, programming difficulty, and the requirement for specialized knowledge to program and configure FPGAs resulted in a short life for the FPGA era of Bitcoin mining.

Mining hardware such as the X6500 miner, Ztex, and Icarus were developed during the time when FPGA mining was profitable. Various FPGA manufacturers, such as Xilinx and Altera, produce FPGA hardware and development boards that can be used to program mining algorithms. The arrival of ASICs resulted in the quick phasing out of FPGA-based systems for mining. It should be noted that GPU mining is still profitable for some other cryptocurrencies to some extent, such as Zcoin (<https://zcoin.io/guide-on-how-to-mine-zcoin-xzc/>), but not Bitcoin, because the network difficulty of Bitcoin is so high that only ASICs (specialized hardware) running in parallel can produce reasonable profit.

### ASICs

ASICs were designed to perform SHA-256 operations. These special chips were sold by various manufacturers and offered a very high hashing rate. This worked for some time, but due to the quickly increasing mining difficulty level, single-unit ASICs are no longer profitable.

Currently, mining is out of the reach of individuals due to the vast amounts of energy and money needed to be spent in order to build a profitable mining platform. Now, professional mining centers using thousands of ASIC units in parallel are offering mining contracts to users to perform mining on their behalf. There is no technical limitation—a single user can run thousands of ASICs in parallel—but it will require dedicated data centers and hardware, therefore the cost for a single individual can become prohibitive.

**MINING POOLS**

A mining pool forms when a group of miners work together to mine a block. The pool manager receives the coinbase transaction if the block is successfully mined, and is then responsible for distributing the reward to the group of miners who invested resources to mine the block. This is more profitable than solo mining, where only one sole miner is trying to solve the partial hash inversion function (hash puzzle) because, in mining pools, the reward is paid to each member of the pool regardless of whether they (or more specifically, their individual node) solved the puzzle or not.

**THE BITCOIN NETWORK AND PAYMENTS**

The Bitcoin network is a peer-to-peer (P2P) network where nodes perform transactions. The two main types of nodes on the network are full nodes and simple payment verification (SPV) nodes.

Full Nodes are the implementations of Bitcoin Core clients performing the wallet, miner, full blockchain storage and network routing functions.

SPV Nodes perform only wallet and network routing functionality.

**WALLETS**

The wallet software is used to generate and store cryptographic keys. It performs several functions such as sending and receiving Bitcoin, backing up keys and keeping track of the balance available. Wallet softwares provide some functions to the users to manage and carry out transactions on the Bitcoin network.

**TYPES OF WALLETS**

1. Non-deterministic Wallets: These wallets contain randomly generated private keys.
2. Deterministic Wallets: Here, keys are derived from a seed value via hash functions. This seed number is generated randomly and is commonly represented by human-readable **mnemonic code** words. These phrases can be used to recover all keys and make private key management comparatively easier.
3. Hierarchical Deterministic Wallets: These wallets store keys in a tree structure derived from a seed. The seed generates the parent key, which is used to generate child keys and subsequently grandchild keys.
4. Paper Wallets: This is a paper-based wallet with the required key material printed on it. It requires physical security to be stored.
5. Hardware Wallets: These are tamper-resistant devices used to store keys. Eg, Trezor.
6. Online Wallets: They’re stored entirely online and are provided as a service usually via the cloud. They provide a web interface to the users to manage their wallets and perform various functions such as making and receiving payments.
7. Mobile Wallets: They provide us with various methods to make payments, most especially, the ability to scan QR codes quickly and make payments.
8. Brain Wallets: The master private key can also be derived from the hashes of passwords that are memorized. The key idea is that this passphrase is used to derive the private key and if used in HD wallets, this can result in a full HD wallet that is derived from a single memorized password. This is known as a brain wallet. This method is prone to password guessing and brute-force attacks, but techniques such as key stretching can be used to slow down the progress made by the attacker.

**BITCOIN CLIENTS**

A Bitcoin client is a piece of software that is responsible for generating private/public key pairs and facilitates Bitcoin payments using the Bitcoin blockchain. In addition, a client can implement full synchronization function with a blockchain or choose to only implement basic wallet functionality or simple payment verification. A client can also provide other useful functions such as network monitoring, secure storage of keys, and user-friendly interfaces for interaction with the Bitcoin blockchain.

The Bitcoin core client can be installed from <https://bitcoin.org/en/download>. This is available for all platforms. Bitcoin clients has tools that enable you to run and manage the Bitcoin client and interact with the Bitcoin blockchain.

**TYPES OF BITCOIN CLIENTS AND TOOLS**

1. bitcoind: This is the core client software that runs as a service and it provides the JSON-RPC interface.
2. bitcoin-cli: This is the command-line feature-rich tool for interacting with the Bitcoin daemon; the daemon then interacts with the blockchain and performs various functions.
3. bitcoin-qt: This is the Bitcoin Core client GUI.