Transactions: A transfer of value between Bitcoin wallets, which get recorded in the blockchain.
Miners: Participants in the network who validate transactions, add them to the blockchain, and receive newly minted bitcoins as a reward.
Bitcoin wallets: Software or hardware that store the private keys needed to access and transact with bitcoins.

Module -2
Introducing Bitcoin

Bitcoin is the first application of blockchain technology.

-name of the Bitcoin inventor Satoshi Nakamoto is believed to be a pseudonym, as the true identity of Bitcoin inventor is unknown
-Bitcoin is built on decades of cryptographic research such as the research in Merkle trees, hash functions, public key cryptography, and digital signatures.

Bitcoin definition

It is a combination of peer-to-peer network, protocols, software that facilitates the creation and usage of the digital currency named bitcoin. Nodes in this peer-to-peer network talk to each other using the Bitcoin protocol.

Decentralization of currency is made possible for the first time with the invention of bitcoin

Double spending problem is resolved in Bitcoin by using a distributed ledger (blockchain) where every transaction is recorded permanently and by implementing transaction validation and confirmation mechanism.

Bitcoin is composed of the elements listed below:

- Digital keys
- Addresses
- Transactions
- Blockchain
- Miners
- The Bitcoin network
- Wallets (client software)

Bitcoin network- from an end user's perspective

Sending a payment to someone
An Example that demonstrates how money can be sent using Bitcoin network from one user to another

Steps

1. Payment is requested from a user by sending his Bitcoin address to the sender via email or SMS, chat applications.
-sender initiates a transfer to send money to another user
-the address of beneficiary is required.
-from the Blockchain wallet a payment request is created

2. Sender enters the receiver's address or scans the QR code that has the Bitcoin address, amount and optional description encoded in it.
- wallet application recognizes this QR code and decodes it into
 Please send <Amount> BTC to the Bitcoin address <receiver's Bitcoin address>.
 Please send 0.00033324 BTC to the Bitcoin address 1JzouJCVhMQBnTcd8K4YSBP36gEFNn1ZJ3.

3. In the wallet application of the sender, this transaction is constructed and broadcasted to the Bitcoin network.
- This transaction is digitally signed using the private key of the sender before broadcasting it.
 -a number of fields such as From, To, BTC, and Fee are shown in Blockchain wallet.
Fee is calculated based on the size of the transaction and a fee rate is a value that depends on the volume of the transaction in the network, is represented in Satoshis/byte.

Get the Receiver's Bitcoin Address
Initiate the Payment
Broadcast the Transaction
Confirm the Transaction
Payment Complete:

4. The transaction will be picked up by miners to be verified and included in the block.
- Transaction confirmations will start to appear as soon as the transaction is verified,included in the block, and mined.
- appropriate fee will be deducted from the original value to be transferred and will be paid to the miner who has included it in the block for mining.

Segregated Witness (SegWit) is an important upgrade to the Bitcoin blockchain protocol that was implemented in 2017. SegWit transactions are an important aspect of this upgrade because they provide several key benefits over traditional Bitcoin transactions.

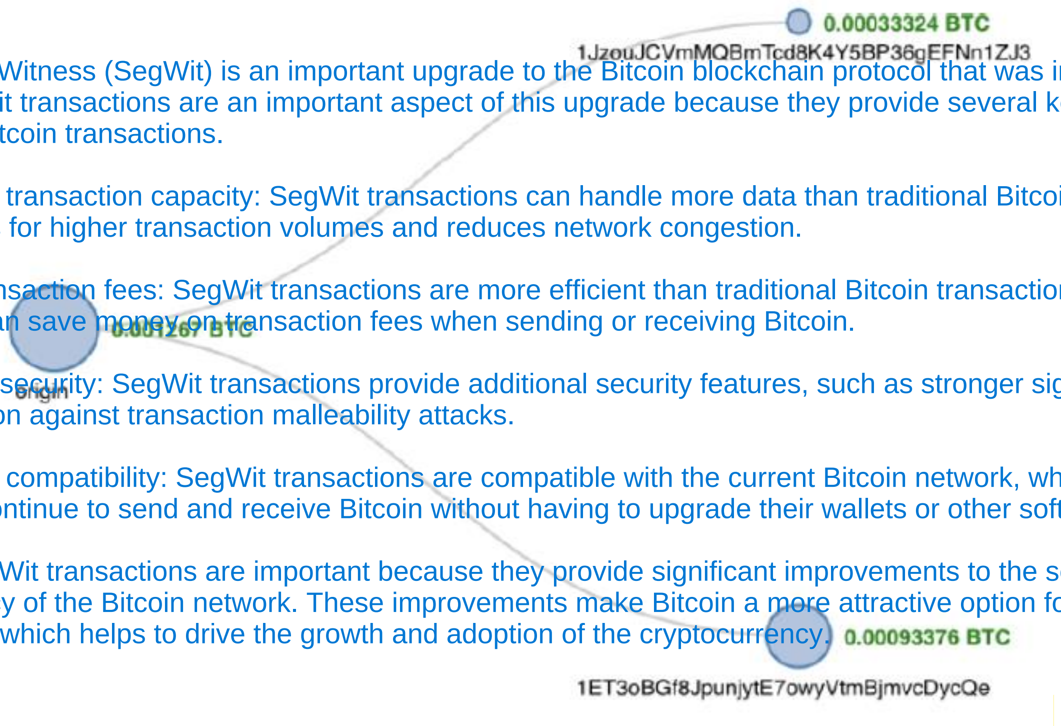
1. Increased transaction capacity: SegWit transactions can handle more data than traditional Bitcoin transactions, which allows for higher transaction volumes and reduces network congestion.

2. Lower transaction fees: SegWit transactions are more efficient than traditional Bitcoin transactions, which means that users can save money on transaction fees when sending or receiving Bitcoin.

3. Improved security: SegWit transactions provide additional security features, such as stronger signature validation and protection against transaction malleability attacks.

4. Increased compatibility: SegWit transactions are compatible with the current Bitcoin network, which means that users can continue to send and receive Bitcoin without having to upgrade their wallets or other software.

Overall, SegWit transactions are important because they provide significant improvements to the scalability, security, and efficiency of the Bitcoin network. These improvements make Bitcoin a more attractive option for both users and businesses, which helps to drive the growth and adoption of the cryptocurrency.


0.00033324 BTC
1JzouJCVmMQBmTcd8K4Y5BP36gEFNn1ZJ3
0.001267 BTC
Origin
0.00093376 BTC
1ET3oBGf8JpunjytE7owyVtmBjmvcDycQe

Transaction flow visualization (Blockchain.info)

Flow is shown in the diagram, where a payment of 0.001267 BTC (approximately 11 USD is originated from the sender's address and been paid to receiver's address (starting with 1Jz). Fee of 0.00010622 (approximately 95 cents) is also deducted from the transaction as mining fee.
Important fields are listed here with their purpose and explanation:
Size: This is the size of the transaction in bytes.
Weight: new metric given for block and transaction size since the introduction of Segregated Witness (SegWit) version of Bitcoin.
Received Time: time when the transaction is received.
Included In Blocks: block number on the blockchain in which the transaction is included.
Confirmations:This is the number of confirmations by miners for this transaction.
Total Input: number of total inputs in the transaction.
Total Output: number of total outputs in the transaction.
Fees: total fees charged.
Fee per byte: This field represents the total fee divided by the number of bytes in a transaction. example 10 Satoshis per byte.
Fee per weight unit: For legacy transaction it is calculated using total number of bytes * 4.
For SegWit transactions it is calculated by combining SegWit marker, flag, and witness field as one weight unit and each byte of other fields as four weight units.

In summary, the payment transaction in the Bitcoin network can be divided into the following steps:
1. Transaction starts with a sender signing the transaction with their private

key
2. Transaction is serialized so that it can be transmitted over the network
3. Transaction is broadcasted to the network
4. Miners listening for the transactions picks up the transaction
5. Transaction are verified for their validity by the miners
6. Transaction are added to the candidate/proposed block for mining
7. Once mined, the result is broadcasted to all nodes on the Bitcoin network

Private Key: A private key is a secret code that allows you to access and control your Bitcoin holdings. It is a 256-bit long number that is generated randomly and kept secret. The private key is used to create a digital signature that is required to authorize a transaction on the Bitcoin network.

Public Key: A public key is a code that is generated from a private key and is used to receive Bitcoins. The public key can be shared publicly, and it acts as an address to which others can send Bitcoins. The public key is used in combination with the private key to perform a secure transaction.

2.1 Digital keys and addresses

On the Bitcoin network, possession of bitcoins and transfer of value via transactions is 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.**

2.1.1 Private keys in Bitcoin

- are to be kept safe and normally resides only on the owner's side.

-are used to digitally sign the transactions proving the ownership of the bitcoins.
- are fundamentally 256-bit numbers randomly chosen in the range specified by the secp256k1 ECDSA curve recommendation
- are usually encoded using Wallet Import Format (WIF) in order to make them easier to copy and use.
-WIF can be converted into a private key and vice versa
- mini private key format/ minikey is sometimes used to create the private key with a maximum of up to 30 characters in order to allow storage where physical space is limited, for example, etching on physical coins or encoding in damage-resistant QR codes.
- first character of mini private key is always uppercase letter S.

2.1.2 Public keys in Bitcoin

-Public keys exist on the blockchain and all network participants can see it.
- are derived from private keys due to their special mathematical relationship with the private keys.
-Once a transaction signed with the private key is broadcasted on the Bitcoin network, public keys are used by the nodes to verify that the transaction has indeed been signed with the corresponding private key. This process of verification proves the ownership of the bitcoin.
- Bitcoin uses DCC based on the secp256k1 standard.
A public key is 256-bits in length.
-can be represented in an uncompressed or compressed format.
- are fundamentally x and y coordinates on an elliptic curve.
- Keys are identified by various prefixes,:
-Incompressed public keys use 0x04 as the prefix
-Compressed public key starts with 0x03 if the y 32-bit part of the public key is odd
-Compressed public key starts with 0x02 if the y 32-bit part of the public key is even

2.1.3 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 SHA-256 algorithm and then withRIPEMD-160. The resultant 160-bit hash is then prefixed with a version number and finally encoded with a Base58Check encoding scheme. The bitcoin addresses are 26-35 characters long and begin with digit 1 or 3.
There are two types of addresses, the commonly used P2PKH and another P2SH type, starting with number 1 and 3, respectively.
Addresses should not be used more than once; otherwise, privacy and security issues can arise. Avoiding address reuse circumvents anonymity issues to an extent,

2**.1.3.1 Base58Check encoding** : Bitcoin addresses are encoded using the Base58Check encoding. The encoding basically takes the binary byte arrays and converts them into human-readable strings. This string is composed by utilizing a set of 58 alphanumeric symbols.

2.1.3.2 Vanity addresses: is a type of cryptocurrency address that containes a personalized human readable message. Vanity addresses are generated using a purely brute-force method. The paper wallets can be stored physically as an alternative to electronic storage of private keys.

2.1.3.3 Multisignature addresses(M-of-N MultiSig): These addresses require multiple private keys. Here M represents threshold or the minimum number of signatures required from N number of keys to release the bitcoins. Sending bitcoins from this address requires signature from atleast M keys.

2.2 Transactions

-are at the core of the bitcoin ecosystem

-sending some bitcoins to a bitcoin address, or 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.

-Coins are unspent transaction outputs represented in Satoshis.

-Transactions are not encrypted and are publicly visible in the blockchain.

-Blocks are made up of transactions and these can be viewed using any online blockchain explorer

2.2.1 The transaction life cycle

The following steps describe the transaction life cycle:

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 transaction are placed in the block they are placed in a special memory buffer called **transaction pool**.

5. Mining starts, which is a process by which the blockchain is secured and new coins are generated as a reward for the miners who spend appropriate computational resources.

6. Once a miner solves the PoW problem it broadcasts the newly mined lock to the network

7. The nodes verify the block and propagate the block further, and confirmations start to generate.

8. Finally, the confirmations start to appear in the receiver's wallet and after approximately three confirmations, the transaction is considered finalized and confirmed. 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 three confirmations.

Transaction fee

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 and the sum of the outputs.

fee = sum(inputs) - sum(outputs)

fees are used as an incentive for miners to encourage them to include a user transaction in the block the miners are creating

The time for transaction confirmation usually ranges from 10 minutes to over 12 hours in some cases.

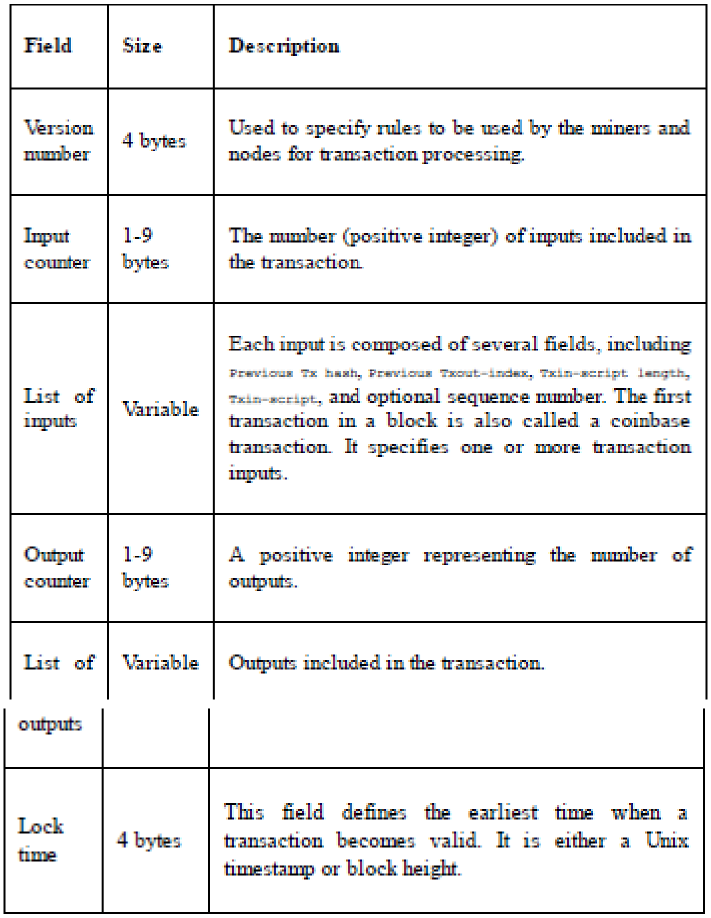
Transaction pools

Also known as memory pools, these are basically created in local memory (computer RAM) by nodes in order to maintain a temporary list of transactions that are not yet confirmed in a block. Transactions are included in a block after passing verification and based on their priority.

2.2.2 The transaction data structure

A transaction at a high level contains metadata, inputs, and outputs. Transactions are combined to create a block.

The transaction data structure is shown in the following table:



Field	Size	Description
Version number	4 bytes	Used to specify rules to be used by the miners and nodes for transaction processing.
Input counter	1-9 bytes	The number (positive integer) of inputs included in the transaction.
List of inputs	Variable	Each input is composed of several fields, including Previous Tx hash, Previous Txout-index, Txin-script length, Txin-script, and optional sequence number. The first transaction in a block is also called a coinbase transaction. It specifies one or more transaction inputs.
Output counter	1-9 bytes	A positive integer representing the number of outputs.
List of outputs	Variable	Outputs included in the transaction.
Lock time	4 bytes	This field defines the earliest time when a transaction becomes valid. It is either a Unix timestamp or block height.

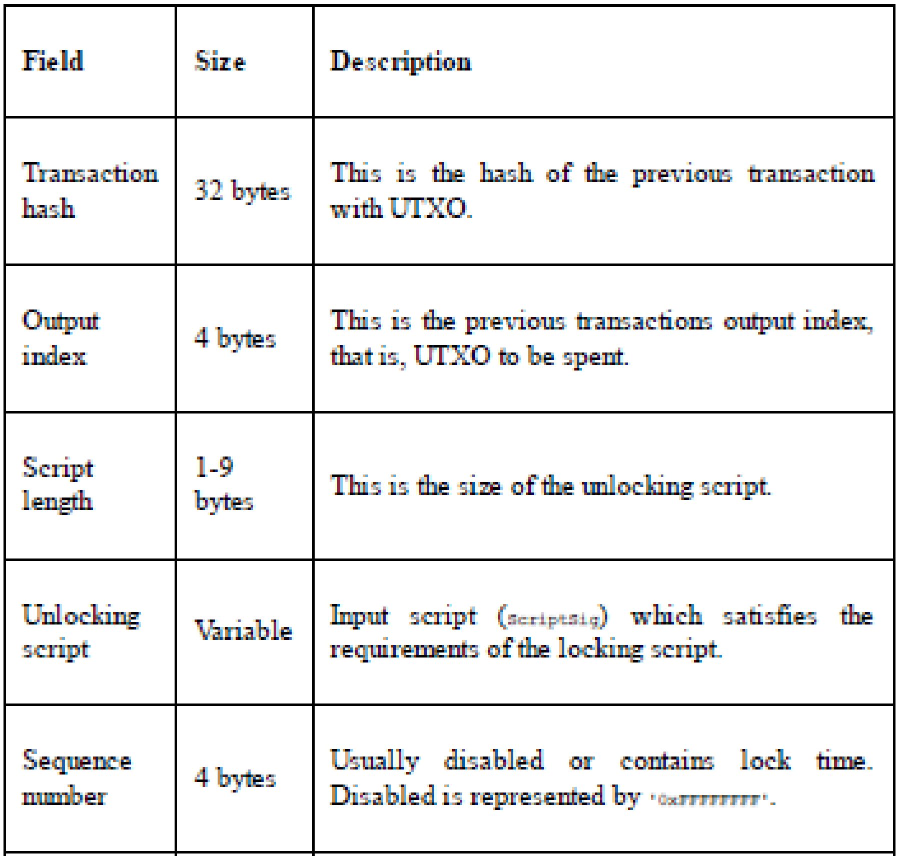
composed of several field of previous input.
the first tx in the block is called coinbase tx.

(Previous Tx hash, Previous Tx out Index, Txin-script length,Txin-script)

There are a number of structures that make up the transaction. All these elements are described in the following subsections.

Metadata:This part of the transaction contains some 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.

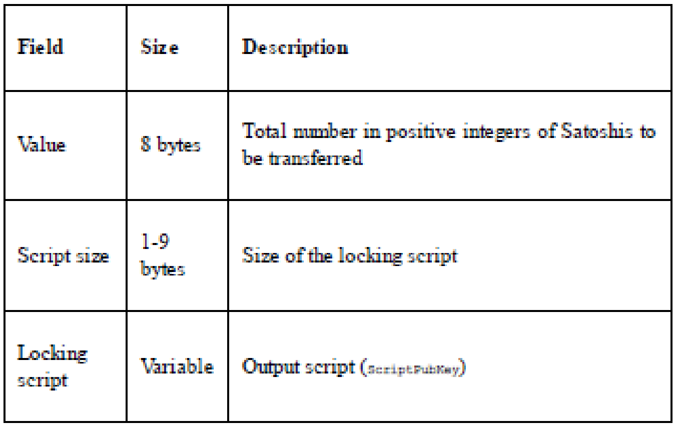
Inputs :Each input spends a previous output. Each output is considered as **Unspent Transaction Output (UTXO)** until an input consumes it. UTXO is an unspent transaction output that can be spent as an input to a new transaction.



Field	Size	Description
Transaction hash	32 bytes	This is the hash of the previous transaction with UTXO.
Output index	4 bytes	This is the previous transactions output index, that is, UTXO to be spent.
Script length	1-9 bytes	This is the size of the unlocking script.
Unlocking script	Variable	Input script (scriptSig) which satisfies the requirements of the locking script.
Sequence number	4 bytes	Usually disabled or contains lock time. Disabled is represented by '0xFFFFFFFF'.

Outputs: Outputs have three fields.

First field contains the amount of Satoshis whereas the second field contains the size of the locking script. The third field contains a locking script that holds the conditions that need to be met in order for the output to be spent.



Field	Size	Description
Value	8 bytes	Total number in positive integers of Satoshis to be transferred
Script size	1-9 bytes	Size of the locking script
Locking script	Variable	Output script (ScriptPubKey)

Verification:Verification is performed using Bitcoin's scripting language.

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.

It is evaluated from the left to the right using a Last In, First Out (LIFO) stack.

2.2.3Types of Transactions

Standard transactions are evaluated using **IsStandard()** and **IsStandardTx()** tests and only standard transactions that pass the test are generally allowed to be mined or broadcasted on the Bitcoin network.

The following are the standard transaction types:

- **Pay to Pubkey**: This script is a very simple script that is commonly used in coinbase transactions. It is now obsolete and was used in an old version of bitcoin. **The P2PK locking script expects the unlocking script to push a signature to the stack. If the signature is valid for the specified public key in the locking script, the output is allowed to be spent.**
- **Pay to Public Key Hash (P2PKH)**: P2PKH is the most commonly used transaction type and is used to send transactions to the bitcoin addresses. **This works similarly to P2PK but instead of pushing the public key, it pushes a hash of the public key, commonly referred to as an address**.
- **Pay to Script Hash (P2SH):** P2SH is used in order to send transactions to a script hash (that is, the addresses starting with 3). **Pay to Script Hash is used to require the spender of an output to include a specific set of operations in their unlocking script.**
- **MultiSig (Pay to MultiSig)**: M-of-N MultiSig transaction script is a complex type of script where **it is possible to construct a script that required multiple signatures to be valid in order to redeem a transaction.**
- **Null data/OP_RETURN**: This script is used to store arbitrary data on the blockchain for a fee. The limit of the message is 40 bytes. The output of this script is unredeemable because OP_RETURN will fail the validation in any case. **Data Output scripts are used to create outputs that are not spendable but instead are used purely to add data to a transaction.**

Coin base transactions

-A coinbase transaction or generation transaction is always created by a miner
- is the first transaction in a block
- is used to create new coins
- includes a special field called coinbase, which acts as an input to the coinbase transaction.
- allows up to 100 bytes of arbitrary data that can be used to store arbitrary data.
- A coinbase transaction input has the same number of fields as usual transaction input, but the structure contains coinbase data size and coinbase data
-it does not have a reference pointer to the previous transaction. This structure is shown in the following table:

Field	**Size**	**Description**
Transaction hash	32 bytes	Set to all zeroes as no hash reference is used
Output index	4 bytes	Set to 0xFFFFFFFF
Coinbase data length	1-9 bytes	2 bytes-100 bytes
Data	Variable	Any data
Sequence number	4 bytes	Set to 0xFFFFFFFF

Contracts

Contracts are transactions that use the Bitcoin system to enforce a financial agreement.
-it allows users to design complex contracts that can be used in many real-world scenarios.
-allow the development of a completely decentralized, independent, and reduced risk platform.
-Various contracts, such as escrow, arbitration, and micropayment channels, can be built using the Bitcoin scripting language.

-For example, the release of funds only if multiple parties sign the transaction or perhaps the release of funds only after a certain time has elapsed. Both of these scenarios can be realized using multisig and transaction lock time options.

2.2.4 Transaction Verification

Verification process is performed by Bitcoin nodes.

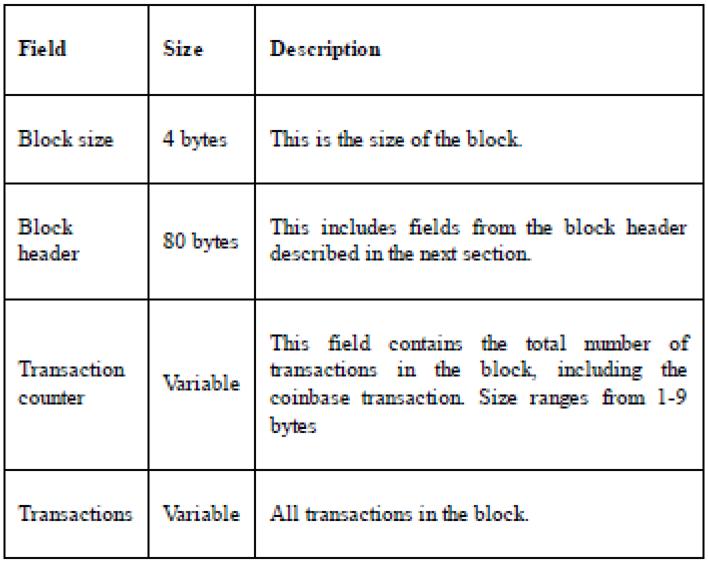
1. Check the syntax and ensure that the syntax and data structure of the transaction conforms to the rules provided by the protocol.
2. Verify that no transaction inputs and outputs are empty.
3. Check whether the size in bytes is less than the maximum block size.
4. The output value must be in the allowed money range (0 to 21 million BTC).
5. All inputs must have a specified previous output, except for coinbase transactions, which should not be relayed.
6. Verify that nLockTime must not exceed 31-bits. (nLockTime specifies the time before which transaction will not be included in the block.)
7. For a transaction to be valid, it should not be less than 100 bytes.
8. The number of signature operations in a standard transaction should be less than or not more than two.
9. Reject nonstandard transactions; for example, ScriptSig is allowed to only push numbers on the stack.. isStandard() checks specify that only standard transactions are allowed.
10. A transaction is rejected if there is already a matching transaction in the pool or in a block in the main branch.
11. The transaction will be rejected if the referenced output for each input exists in any other transaction in the pool.
12. For each input, there must exist a referenced output unspent transaction.
13. For each input, if the referenced output transaction is the coinbase, it must have at least 100 confirmations; otherwise, the transaction will be rejected.
14. For each input, if the referenced output does not exist or has been spent already, the transaction will be rejected.
15. Using the referenced output transactions to get input values, verify that each input value, as well as the sum, is in the allowed range of 0-21 million BTC. Reject the transaction if the sum of input values is less than the sum of output values.
16. Reject the transaction if the transaction fee would be too low to get into an empty block.
17. Each input unlocking script must have corresponding valid output scripts.

2.2.5 Transaction malleability

Transaction malleability in Bitcoin was introduced due to a bug in the bitcoin implementation. Due to this bug, it became possible for an adversary to change the transaction ID of a transaction, thus resulting in a scenario where it would appear that a certain transaction has not been executed. This can allow scenarios where double deposits or withdrawals can occur.

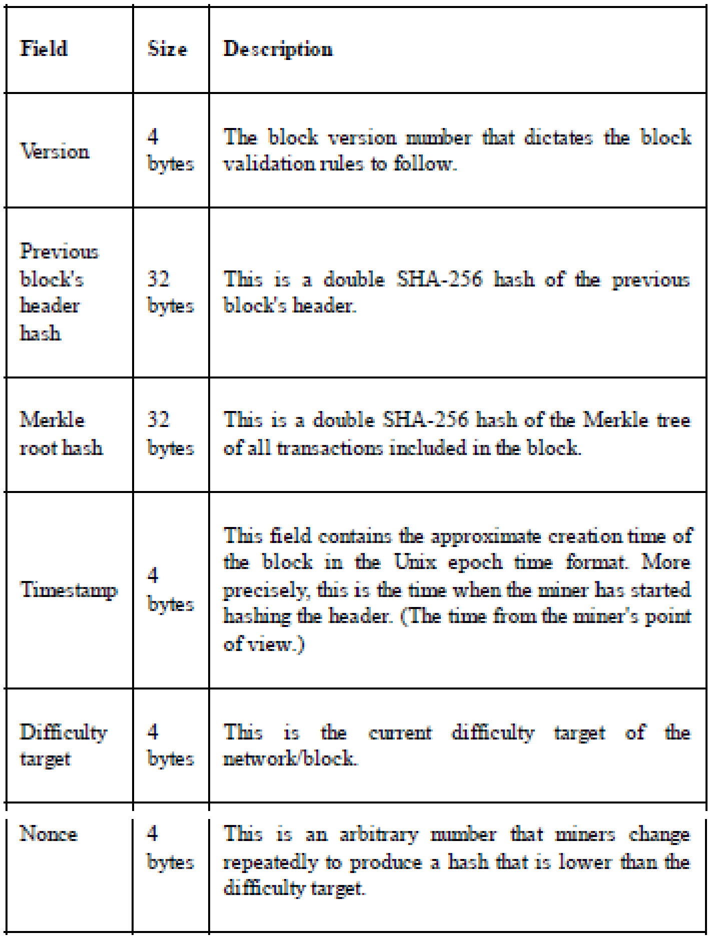
2.3 Blockchain

The structure of a block



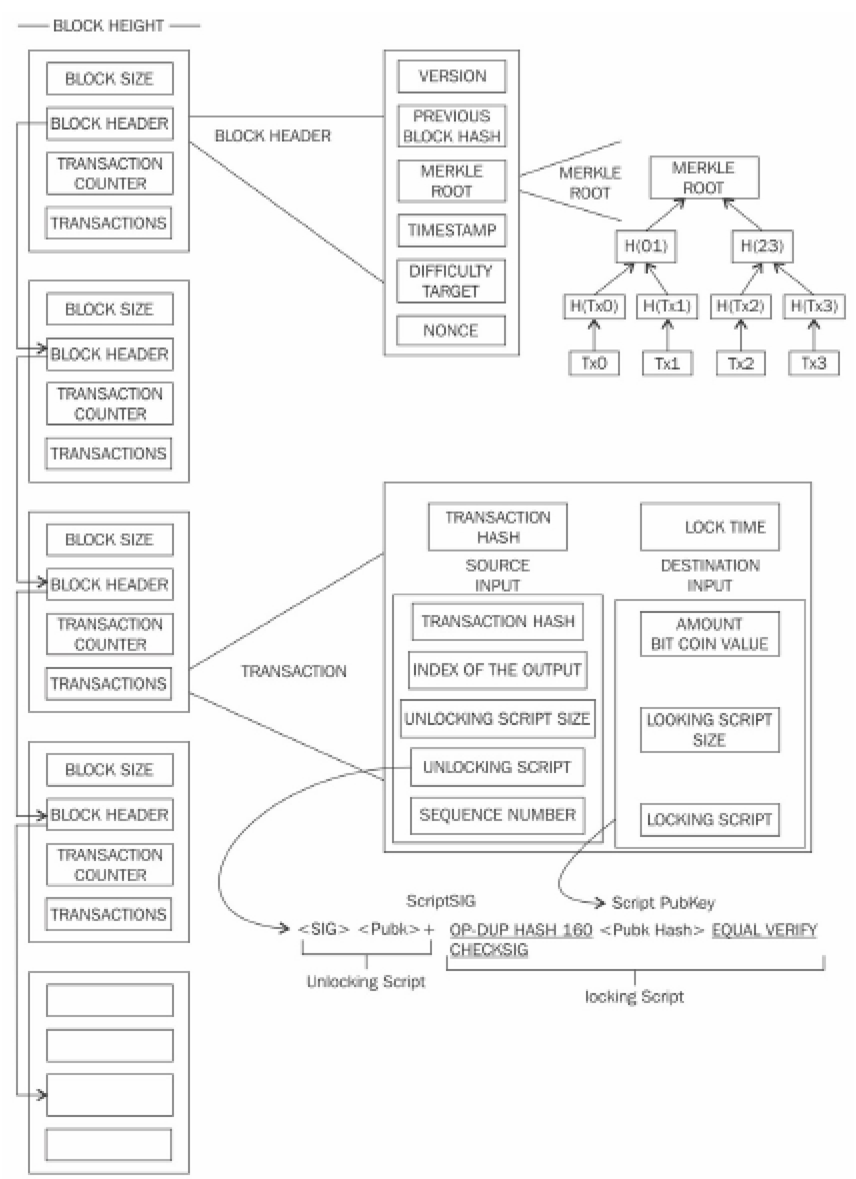
Field	Size	Description
Block size	4 bytes	This is the size of the block.
Block header	80 bytes	This includes fields from the block header described in the next section.
Transaction counter	Variable	This field contains the total number of transactions in the block, including the coinbase transaction. Size ranges from 1-9 bytes
Transactions	Variable	All transactions in the block.

The structure of a block header



Field	Size	Description
Version	4 bytes	The block version number that dictates the block validation rules to follow.
Previous block's header hash	32 bytes	This is a double SHA-256 hash of the previous block's header.
Merkle root hash	32 bytes	This is a double SHA-256 hash of the Merkle tree of all transactions included in the block.
Timestamp	4 bytes	This field contains the approximate creation time of the block in the Unix epoch time format. More precisely, this is the time when the miner has started hashing the header. (The time from the miner's point of view.)
Difficulty target	4 bytes	This is the current difficulty target of the network/block.
Nonce	4 bytes	This is an arbitrary number that miners change repeatedly to produce a hash that is lower than the difficulty target.

High-level overview of the Bitcoin blockchain



On the left-hand side blocks are shown starting from top to bottom. Each block contains transactions and block headers which are further magnified on the right-hand side. On the top, first, block header is expanded to show various elements within the block header. Then on the right-hand side the Merkle root element of the block header is shown in magnified view which shows that how Merkle root is calculated.

Genesis block:This is the first block in the Bitcoin blockchain
Block height is the number of blocks before a particular block in the blockchain.

PoW is used to secure the blockchain. Each block contains one or more transactions, out of which the first transaction is a coinbase transaction. There is a special condition for coinbase transactions that prevent them from being spent until at least 100 blocks in order to avoid a situation where the block may be declared stale later on

Stale blocks are created when a block is solved and every other miner who is still working to find a solution to the hash puzzle is working on that block.

Orphan blocks are also called detached blocks and were accepted at one point in time by the network as valid blocks but were rejected when a proven longer chain was created that did not include this initially accepted block.

Forks in blockchain can also occur with the introduction of changes in the Bitcoin protocol. In case of a soft fork, a client which chooses not to upgrade to the latest version supporting the updated protocol will still be able to work and operate normally

Network difficulty basically means how hard it is for miners to find a new block, that is, how difficult the hashing puzzle is.
Network difficulty is calculated using the following equation:
Target = Previous target * Time/2016 * 10 minutes

Previous target represents the old target value, and time is the time spent to generate previous 2016 blocks

2.4 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.
Roughly one new block is created (mined) every 10 minutes to control the frequency of generation of bitcoins. Miners are rewarded with new coins if and when they discover new blocks by solving PoW. Miners are paid transaction fees in return for including transactions in their proposed blocks. New blocks are created at an approximate fixed rate of every 10 minutes. Approximately 144 blocks, that is, 1,728 bitcoins are generated per day. The number of actual coins can vary per day.

Tasks of the miners
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.
2. Transaction validation: Transactions broadcasted 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 broadcasted on the network after validating them.
5. Perform Proof of Work: 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.

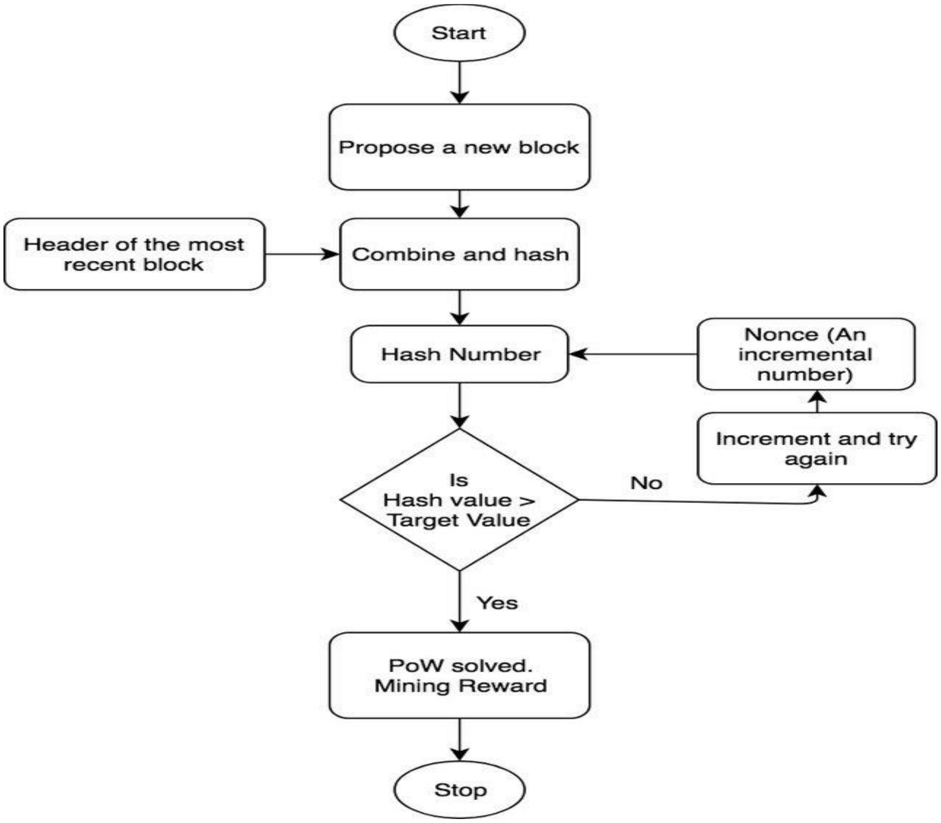
Proof of Work (PoW)
PoW is based on the idea that a random node is selected every time to create a new block. In this model, nodes compete with each other in order to be selected in proportion to their computing capacity.
The following equation sums up the PoW requirement in bitcoin:

$$H\,(N \,||\, P\_hash \,||\, Tx \,||\, Tx \,||\, .\,.\,.\; Tx) < Target$$

Where N is a nonce, P_hash is a hash of the previous block, Tx represents transactions in the block, and Target is the target network difficulty value. The only way to find this nonce is the brute force method. Once a certain pattern of a certain number of zeroes is met by a miner, the block is immediately broadcasted and accepted by other miners.

The mining algorithm

1. The previous block's header is retrieved from the bitcoin network.
2. Assemble a set of transactions broadcasted 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 SHA-256 algorithm.
4. Check if the resultant hash is lower than the current difficulty level (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.


Start
Propose a new block
Header of the most recent block
Combine and hash
Hash Number
Nonce (An incremental number)
Increment and try again
No
Is Hash value > Target Value
Yes
PoW solved. Mining Reward
Stop

Mining Process

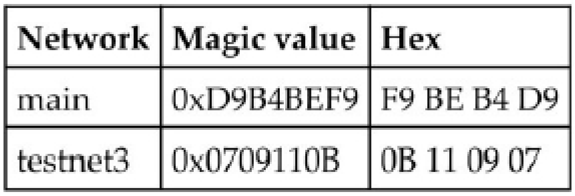
Bitcoin Network and Payments

2.5 The Bitcoin network

The Bitcoin network is a peer-to-peer network where nodes exchange transactions and blocks. There are different types of nodes on the network. There are two main types of nodes, **full nodes** and **SPV nodes**. **Full nodes** are implementations of Bitcoin core clients performing the wallet, miner, full blockchain storage, and network routing functions.

Simple Payment Verification (SPV) nodes or lightweight clients perform only wallet and network routing functionality. The latest version of Bitcoin protocol is 70015 and was introduced with Bitcoin core client 0.13.2.

The Bitcoin network is identified by its different magic values. A list is shown as follows:

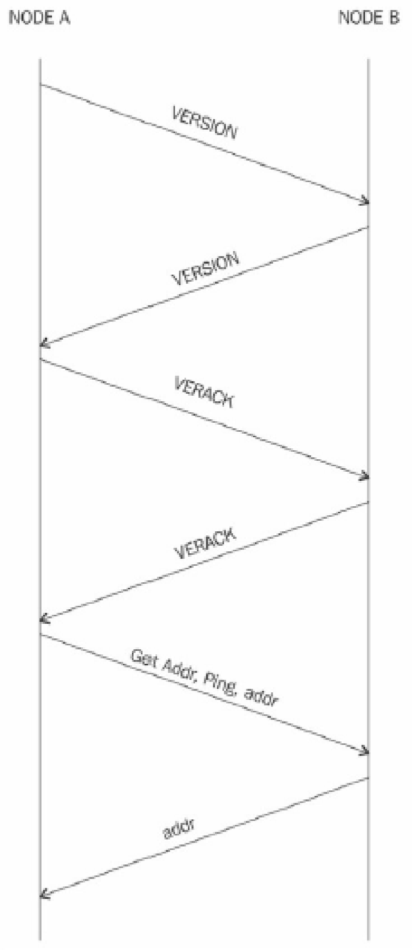


Network	Magic value	Hex
main	0xD9B4BEF9	F9 BE B4 D9
testnet3	0x0709110B	0B 11 09 07

Bitcoin network magic values

Magic values are used to indicate the message origin network.

The network protocol sequence diagram shows communication between two Bitcoin nodes during initial connectivity. NODE A is shown on the left side and NODE B on the right. First, NODE A starts the connection by sending the version message which contains version number and current time to the remote peer NODE B. NODE B then responds with its own **version** message containing the version number and current time. NODE A and NODE B then exchange a **verack** message indicating that the connection has been successfully established. After the connection is successful the peers can exchange **getaddr** and **addr** messages to discover other peers on the network.



Visualization of node discovery protocol

2.6 Wallets

The wallet software is used to store private or public keys and Bitcoin address. It performs various functions, such as receiving and sending bitcoins. Software offers both functionalities: Bitcoin client and wallet. On the disk, the Bitcoin core client wallets are stored as the Berkeley CB file.
Private keys are generated by randomly choosing a 256-bit number by wallet software.
Private keys are used by wallets to sign the outgoing transactions. Wallets do not store any coins, and there is no concept of wallets storing balance or coins for a user.

Different types of wallets in Bitcoin

1. **Non-deterministic Wallets:** These wallets contain randomly generated private keys and are also called just a bunch of key wallets. The Bitcoin core client generates some keys when first started and generates keys as and when required.
2. **Deterministic Wallets:** In this type of wallet, keys are derived out of a seed value via hash functions. This seed number is generated randomly and is commonly represented by human-readable mnemonic coce words.
3. **Hierarchical Deterministic (HD) Wallets:** store keys in a tree structure derived from a seed. The seed generates the parent key (master key), which is used to generate child keys and grandchild keys. The hierarchy of private keys in an HC wallet is easily recoverable if the master private key is known. HC wallets are very easy to maintain and are highly portable. Example: Trezor , Jaxx , Electrum, Metamask

A brain wallet refers to a method of storing cryptocurrency keys in one's memory, rather than writing them down or saving them in a digital file.

4. **Brain Wallets**: The master private key can also be derived from the hash of passwords that are memorized. The key idea is that this passphrase is used to derive the private key and if used in HC wallets, this can result in a full HC wallet that is derived from a single memorized password. 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.
5. **Paper Wallets**: This is a paper-based wallet with the required key material printed on it.It requires physical security to be stored.
6. **Hardware Wallets:** use a tamper-resistant device to store keys. It can be custom-built or with the advent of **NFC-enabled phones**, this can also be a **Secure Element (SE)** in NFC phones. Trezor and Ledger wallets (various types) are the most commonly used Bitcoin hardware wallets.
7. **Online Wallets:** are 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. Example: GreenAddress
8. **Mobile Wallets:** are installed on mobile devices. Smartphone cameras are used to scan QR codes quickly and make payments. They are available for the Android platform and iOS, Example: Blockchain, breadwallet, Copay, and Jaxx.
9. **Hot wallets:** are online wallets through which cryptocurrencies can be transferred quickly. They are available online. Examples are Coinbase and Blockchain.info.
10. **Cold wallets:** are digital offline wallets where the transactions are signed offline and then disclosed online. They are not maintained in the cloud on the internet; they are maintained offline to have high security. Examples of cold wallets are Trezor and Ledger.

2.7 Bitcoin payments

Bitcoin is not recognized as a legal currency in many jurisdictions, but it is accepted as a payment method by many online merchants and ecommerce websites.

For example, in an online shop, Bitcoin merchant solutions can be used, whereas in traditional, physical shops, point of sale terminals and other specialized hardware can be used. Customers can simply scan the QR code with the seller's payment URI in it and pay using their mobile devices.

Uniform Resource Identifier (URI) Is basically a string that represents the transaction information. It is defined in BIP 21.

Various payment solutions, such **as XBTerminal** and 34 Bytes bitcoin **Point of Sale (POS)** terminal are available commercially.

These solutions work by following these steps:

1. The sales person enters the amount of money to be charged in Fiat currency, for example,US Dollars
2. Once the value is entered in the system the terminal prints a receipt with QR code on it and other relevant information such as amount
3. The customer can then scan this QR code using their mobile Bitcoin wallet to send the payment to the Bitcoin address of the seller embedded within the QR code
4. Once the payment is received on the designated Bitcoin address, a receipt is printed out as a physical evidence of sale

2.8 Innovation in Bitcoin

Bitcoin has undergone many changes and still evolving into a more and more robust and better system by addressing various weaknesses in the system. These improvement proposals are usually made in the form of BIPs or fundamentally new versions of Bitcoin protocols.

2.8.1 Bitcoin Improvement Proposals (BIPs) : These documents are used to propose or inform the Bitcoin community about the improvements suggested, the design issues, or information about some aspects of the bitcoin ecosystem. There are three types :

Standard BIP: Used to describe the major changes that have a major impact on the Bitcoin system, for example, block size changes, network protocol changes, or transaction verification changes.
Process BIP: deal with proposing a change in a process that is outside the core Bitcoin protocol. These are implemented only after a consensus among bitcoin users.
Informational BIP: These are usually used to just advise or record some information about the Bitcoin ecosystem, such as design issues.

2.8.2 Bitcoin Cash: Bitcoin Cash increases the block limit to 8 MB. This immediately increases the number of transactions that can be processed in one block to a much larger number as compared to 1 MB limit in original Bitcoin protocol. It uses PoW as consensus algorithm, and mining hardware is still ASIC based. The block interval is changed from 10 minutes to 10 seconds and up to 2 hours.

2.8.3 Bitcoin Gold: This has been implemented as a hard fork since block 49,1407 of the original Bitcoin blockchain. It resulted in a new blockchain, named Bitcoin Gold (BTG). BTG uses the Equihash algorithm as its mining algorithm instead of PoW.

2.8.4 Bitcoin Unlimited : In this proposal, the size of the block is increased but not set to a hard limit. Instead, miners come to a consensus on the block size cap over a period of time. Other concepts such as parallel validation and extreme thin blocks have also been proposed in Bitcoin Unlimited. Extreme thin blocks allow for a faster block propagation between Bitcoin nodes. Parallel validation allows nodes to validate more than one block along with new incoming transactions in parallel.