Distributed Trust and Blockchains	Date:	*11th Sept '18*
Instructor: *Sujit Prakash Gujar*	Scribes:	Abhisagar Khatri, Sai Snehith, Vikas Kamineni

Lecture 10: Bitcoin Blockchain Structure and User Correlation

1 Recap

In the last lecutre, the following topics were covered -

1.1 Bitcoin scripts execution

- Transaction can be redeemed by the execution of Bitcoin scripts.
- The script is concatenation of scriptPubKey and scriptSig where as scriptPubKey specifies public key and scriptSig signature with that public key.

1.2 Bitcoin payment methods

- Pay-to-Public-Key-Hash (P2PKH)
  - It is common form of transaction in bitcoin network.
  - Transactions that pay to a Bitcoin address contain P2PKH scripts which are resolved by sending the public key and a digital signature created by the corresponding private key.
  - Then the Validation of the script is done by following Procedure.Only two outputs are possible for validation i.e. True or False.

    $< Signature >< PublicKey >$ DUP HASH160 $< PublicKeyHash >$ EQUALVERIFY CHECKSIG
- Pay-to-Public-Key
  - The public key is presented as elliptic curve point instead of hash.So hash and dup operators are not required.

    Example of a transaction:

    $$ScriptPubKey =< PublicKey > \text{ OP\_CHECKSIG}$$
    $$ScripSig =< Signature >$$
  - Here the OP_CHECKSIG validates the signature and unlocks the transaction.
- Pay-to-ScriptHash
  - Pay-to-Script-hash(P2SH) is an extension of the multisignature idea which reduces the storage required by storing the hash of the details of transactions.
  - In P2SH, the input is paid to the hash of the script and the transaction can be redeemed by the person having the script which matches the same hash.

1.3 Applications of Bitcoin scripts

- Escrow transactions
  - A trusted intermediate party(Escrow) is present during a transaction between two parties, who verifies the whole transaction and then confirms it.
  - Implemented using MULTISIG. Suppose Alice wants to purchase some goods from Bob. Instead of paying directly to Bob, Alice creates a MULTISIG transaction which can be redeemed when two or more signs it. When Alice recieves the goods, both sign the transaction and money will be sent to the Bob redeeming funds from Escrow.
  - Judy come into play in case theres any dispute.
- Green addresses
  - Green address is trusted by both merchant and customer. Customer uses green addresses to pay to the merchant.
  - It is guaranteed that green address will not double spend this money.
  - Resolves problems related to the need of wait for confirmations and problem of checking whether the transaction is added in the blockchain or not.
- Efficient Micro-payments
  - Micro-payments helps to reduce the huge transaction fees even for sending micro payments.
  - Suppose a customer wants to pay small amounts very frequently to some merchant's service, the customer starts a MULTISIG transaction which can be redeemed only when both the customer and the merchant signs. Based on the amount of service used till now, customer keeps on signing the transactions with the corresponding amount.
  - Customer uses the input script in all these transactions which points to same hash. It might appear as double spending transactions but the transactions which the merchant signs first gets redeemed and published into the blockchain and other transactions becomes invalid transactions. As the merchant wants the entire money for the service used, he signs the last transaction sent by the customer and redeems it.
- LockTime
  - Locktime is the time at which a particular transaction can be added to the block chain.
  - This is the earliest time that miners can include the transaction in their hashing of the Merkle root to attach it in the latest block to the block chain.

2 Overview

The following topics were covered in the lecture. We describe these in detail in the following sections.

1. Merkle Trees
2. Bitcoin Block chain
   - Bitcoin block structure(high-level and low-level)
3. Coinbase Transactions
4. Bitcoin Network
5. Thin or Simple Payment Verification(SPV) Clients
6. Gossip Protocol
7. Role of Bitcoin Node
8. Unspent Transaction Output

3 Merkle Trees

A Merkle tree is a tree which takes large amounts of data and makes it more manageable to process. In Bitcoin technology, merkle trees are used to organize transaction in such a way that it minimizes the resources utilized.

Definition 1. *Merkle tree is a tree in which every leaf node is hash of the data block and every non-leaf node is the hash of its child nodes [1].*

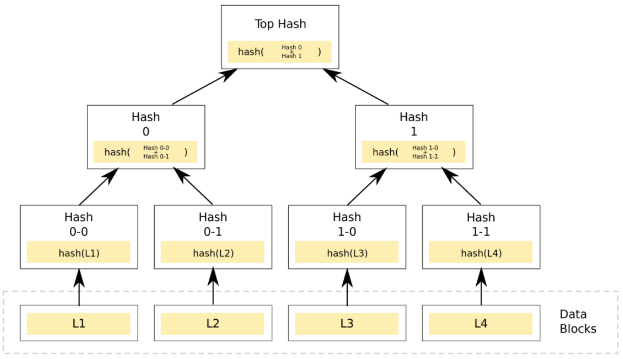


Figure 1: Merkle tree [1]

3.1 Properties of Merkle trees

- In a Binary Merkle tree, if there are 2^k data blocks, then the depth of the Merkle tree will be k.
- File modification or data tampering:
  - Suppose you want to check if a file or data (say L2 data block) is modified or not. We can verify this efficiently using Merkle trees. In Figure 1, if $L2$ data block is modified, nodes all the way till the root($Hash0-1$, $Hash0$ and $TopHash$) are modified. To check if some data is modified or not, we can just check the hash against the root node of the Merkle tree instead of checking the hash of whole data block.
  - In case of data tampering, suppose if an adversary tries to tamper with some data block, then he will have to change the hash pointers all the way upto the root of the tree where he cannot change the hash pointer of the root of the tree which we have stored. So, any attempt to tamper with the data block in a Merkle tree can be detected by just storing the hash pointer of the root of the Merkle tree.
- To prove that a data block is present in the Merkle tree having n data blocks(leaf nodes), we would just need $log(n)$ hash pointers rather than whole Merkle tree. In Figure 1, suppose we need to prove that $L2$ data block is in the Merkle tree. $Hash(L2)$ is stored in $Hash0-1$ node. We would need $Hash0-0$ and concatenate it with $Hash0-1$ to get $Hash0$. Then we would need $Hash1$ and concatenate it with $Hash0$ to get the $TopHash$. As we can see, there are 4 data blocks in total, so in order to prove that a data block is present in a Merkle tree, we need two ($log(4)$) hash pointers ($Hash0-0$ and $Hash1$).

[1]Image-credits: `https://en.wikipedia.org/wiki/Merkle_tree`

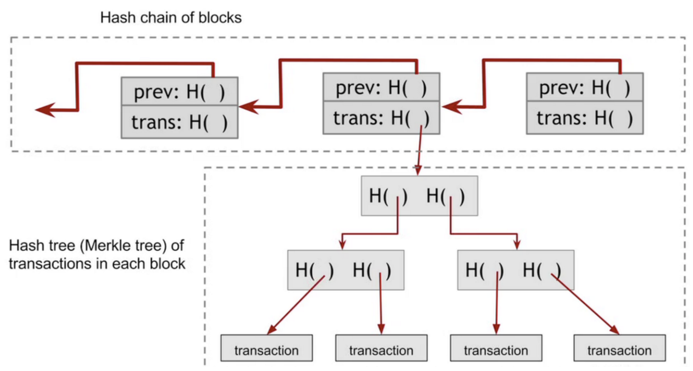


Figure 2: All transactions in a block are stored as leaf nodes in a Merkle tree.[2]

4 Bitcoin Blockchain

In this section, we discuss the need to group multiple transactions into a block.

1. A block creates a single unit of work for miners which is bigger than a single transaction. It is very expensive to hash and store metadata for each transaction.

2. Limit length of hash-chain of blocks. This helps in efficient verification of block chain data structure.

The transactions are stored in a block, which are the leaf nodes of a Merkle tree.

4.1 Bitcoin Block Structure

Bitcoin block structure is a combination of two different data structures, described as follows,

1. Hash chain of blocks in which each block contains a block header, a hash pointer to the transaction data and hash pointer to the previous block in the sequence.

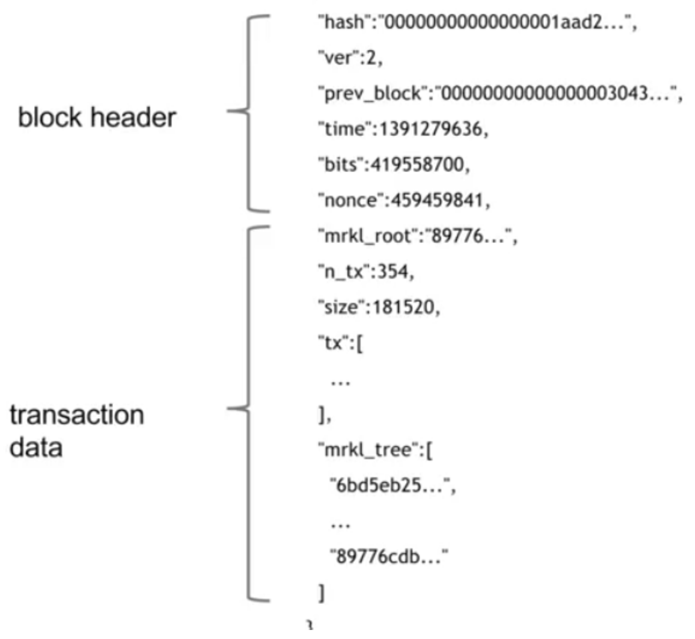
2. Merkle tree of transactions in each block.

Using Merkle trees, it is easy to provide path to the transaction in the tree, which is logarithmic in size in order prove that the transaction is included in a specific block (as described in Section 3.1). Figure 2 is the high level view of the Bitcoin block structure. Now, let us discuss about a Bitcoin block at the low-level.

At low-level, a block consists of the following,

- Block header
  - Header contains all the metadata for the block. It contains the hash of the block and hash of the previous block. It contains a nonce field that miners can change. It also contains time field and bits field.
  - Block header mostly has the information related to mining puzzles. Only headers are hashed during mining the block. So, we only need to look at headers to verify the chain of blocks.
  - The only transaction data which is stored in the block header is the root of the Merkle tree.
- Transaction data

[2]Image-credits: `https://medium.com/@skj48817/merkle-trees-introduction-to-blockchain-c80c0247046`
[3]Image-credits: `http://learningspot.altervista.org/bitcoin-blocks/block_structure/`



```
"hash":"00000000000000001aad2...",
"ver":2,
"prev_block":"00000000000000003043...",
"time":1391279636,
"bits":419558700,
"nonce":459459841,
"mrkl_root":"89776...",
"n_tx":354,
"size":181520,
"tx":[
  ...
],
"mrkl_tree":[
  "6bd5eb25...",
  ...
  "89776cdb..."
]
}
```

Figure 3: Low-level Bitcoin block structure[3]

- It contains a long list of transactions, number of transactions, size of the block.
- It contains all the hashes in the `mrkl_tree` field which are arranged in tree structure. This helps in efficiently proving which transactions are in the block.
- There is one special transaction in Merkle tree, which is unlike all other transactions called Coinbase Transaction. We will discuss it in next section.

5 Coinbase Transactions

There are two aspects of mining where you get money, the block reward and the transaction fee. The block reward is what we call *Coinbase.*

Definition 2 (Coinbase Transaction)**.** *The* coinbase transaction *is the transaction inside a block that pays the miner his block reward. Inside the coinbase transaction is a field that is called the* coinbase *[4].*

5.1 Properties of Coinbase Transaction

1. This is analogous to CreateCoins in ScroogeCoin. So this is where the creation of new coins in bitcoins happen.

2. This seems like a normal transaction but there are several differences between them:

   (a) It always contains a single input and a single output.
   (b) The input doesn't redeem a previous output and thus contains a null hash pointer, since it is minting a new bitcoins and not spending existing ones.
   (c) The output value in the miner's block consists of two components: a flat mining reward, which we just talked over along with all the transaction fees received from all the transactions included in the block.
   (d) There is a special *coinbase* parameter, it provides up to 100 bytes of arbitrary data.

3. The value of the output currently is 12.5 Bitcoins. This value started from 50 Bitcoins in 2009, it halves every 210,000 blocks or 4 years(as bitcoin's block time is 10 minutes per block). The maximum number of halving allowed is 64 times so when the block reward has halved 64 times the value becomes 0.

4. The *Genesis Block* marks the start of Bitcoin blockchain. The coinbase field in it was an article title: "*The Times 03/Jan/2009 Chancellor on brink of second bailout for banks*".

6 Bitcoin Network

6.1 Introduction

- The bitcoin network is a peer-to-peer payment network that operates on a cryptographic protocol. Users send and receive bitcoins, the units of currency, by broadcasting digitally signed messages to the network using bitcoin cryptocurrency wallet software.
- The network changes over time and is dynamic due to nodes entering and leaving at any time. There isnt an explicit way to leave the network. Instead, if a node hasn't been heard from three hours,other nodes start to forget it. In this way, the network gracefully handles nodes going offline.
- It runs over TCP and has a random topology, where each node peers with other random nodes.

6.2 How to join a Bitcoin Network?

∗ To join the network, you need to know is how to contact your seed node i.e the one node thats already on the network and there are a few different ways you can look up lists of seed nodes. You send a special message, saying, Tell me the addresses of all the other nodes in the network that you know about.

∗ You can repeat the process with the new nodes you learn about as many times as you want.

∗ Then you can choose which ones to peer with, and youll be a fully functioning member of the Bitcoin network.

∗ After becoming the member of the Network, you would be required to answer such queries yourselves.

∗ Also if the network finds you inactive for more than 3 hours, it will forget you.

The various options to join the Bitcoin Network are:

(a) Can join just as *thin client* to receive payments in bitcoin.

(b) Or as a miner or a full node

(c) About 90% of nodes run *Core Bitcoin* (C++) (serves as a bitcoin node and provides a wallet to verify bitcoin payments).

(d) Other implementations running successfully are:

    i. BitcoinJ (Java)
    ii. Libbitcoin (C++)
    iii. btcd (Go)

(e) **Original Satoshi Client**: Bitcoin Core is free open-source software that serves as a bitcoin node (the set of which form the bitcoin network) and provides a bitcoin wallet which fully verifies payments

7 Thin or Simple Payment Verification Clients

Lightweight Bitcoin clients are gaining increasing adoption among Bitcoin users, owing to their reduced resource and bandwidth consumption.
The SPV client submits a bloom filter (a filter to request only matching transactions and merkle blocks) to a full node providing SPV services.

Definition 3 (Lightweight node)**.** *A lightweight node, also called thin or SPV client does not download the whole blockchain, instead it downloads the block headers only to validate the authenticity of the transactions that it care about.[2]*

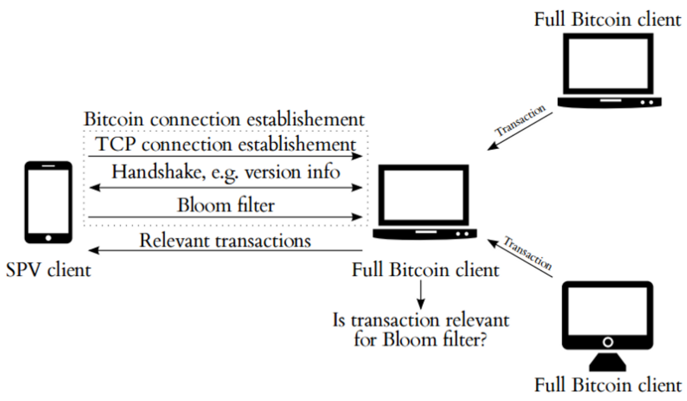

Full Bitcoin client
Transaction
Bitcoin connection establishement
TCP connection establishement
Handshake, e.g. version info
Bloom filter
Relevant transactions
SPV client
Full Bitcoin client
Transaction
Is transaction relevant for Bloom filter?
Full Bitcoin client

Figure 4: Sketch of the operation undergone by an SPV client. SPV clients connect to a full Bitcoin node, which only forwards to the SPV clients the transactions relevant to their Bloom filters [7].

(a) They use the method of Simplified Payment Verification to verify transactions.

(b) They are served by full nodes to connect to bitcoin network thus being effectively dependant on them to function.

(c) An SPV client constructs a Bloom filter by embedding all the Bitcoin addresses which appear in its wallets. Upon connection to a full Bitcoin node, the constructed Bloom filter is outsourced to the full node following an initial handshake protocol (Figure 4)

(d) The cost savings of being an SPV node are huge. The block headers are only about 1/1,000 the size of the block chain. So instead of storing a few tens of gigabytes, its only a few tens of megabytes. Even a smartphone can easily act as an SPV node in the Bitcoin network.

(e) Lightweight wallets do not validate the rules of bitcoin. If somebody pays a lightweight wallet user with fake or invalid bitcoins, then wallet will happily accept them and the user will be left out of pocket.

(f) Lightweight wallets typically send addresses to a trusted third party and receive wallet balance and history. This allows the trusted third party to spy on all the users past and future transactions. Full node wallets avoid this serious privacy leak by downloading the entire blockchain and scanning it locally.

(g) Lightweight node wallets skip several security steps which can leave the user vulnerable.

(h) These downsides can all be avoided by configuring a lightweight node wallet to connect only to your own full node.

8 Gossip Protocol

Definition 4 (Gossip Protocol)**.** *A gossip protocol is a procedure or process of computer-computer communication that is based on the way social networks disseminate information or how epidemics spread [5].*

- In order to publish a transaction,we want to get the entire network to hear about it.It can be done through a simple flooding algorithm also called as Gossip Protocol.
- Suppose if Alice wants to pay Bob, her client creates and her node sends this transaction to all the nodes its peered with.
- Each of those nodes executes a series of checks to determine whether or not to accept and relay the transaction. If the checks pass, the node in turn sends it to all of its peer nodes.
- Nodes that hear about a transaction put it in a pool of transactions which theyve heard about but that arent on the block chain yet.
- If a node hears about a transaction thats already in its pool, it doesnt further broadcast it. This ensures that the flooding protocol terminates and transactions dont loop around the network forever.
- Every transaction is identified uniquely by its hash, so its easy to look up a transaction in the pool.

9 Role of Bitcoin Node

Definition 5 (Bitcoin Node)**.** *A Full node or Bitcoin node is a program that fully validates transactions and blocks. Almost all full nodes also help the network by accepting transactions and blocks from other full nodes, validating those transactions and blocks, and then relaying them to further full nodes [6].*

- Most full nodes also serve lightweight clients by allowing them to transmit their transactions to the network and by notifying them when a transaction affects their wallet.
- If not enough nodes perform this function, clients wont be able to connect through the peer-to-peer networktheyll have to use centralized services instead.

How do nodes decide whether or not to propagate a new transaction?
There following 4 checks are performed:

1. The nodes need to check if the transaction inputs are in UTXO (Unspent Transactions Output).
2. The nodes need to ensure that it is not a double spending transaction.
3. The nodes should execute the output script of input transactions along with ScriptSig.The transaction is added to the transaction pool only if it is true.
4. The nodes must broadcast the transaction only if it is true in the previous transaction.Else if the transaction is repeat then do not broadcast the transaction to avoid flooding.

10 Unspent Transaction Output

- When anyone creates a new transaction, they are both destroying old bills of various denominations (called the inputs) and creating new bills of other denominations (called the outputs).
- UTXOs are created when we have more outputs than inputs.

Definition 6 (UTXO)**.** *In cryptocurrencies, an unspent transaction output is an output of a blockchain transaction that has not been spent, that is used as an input in a new transaction.[3]*

- Miners should keep them in fast access memory, while other parts of the blockchain ledger can be stored in external drives.
- We can reduce them if we have more inputs than outputs.
- But since in the Bitcoin's history the transactions of second type are more than the transactions of first type thus the number of UTXOs has been growing as we can see here(in Figure 5).

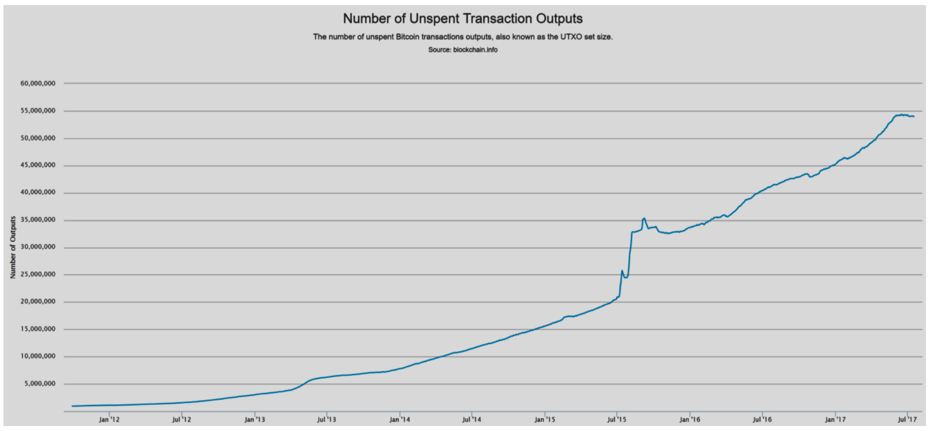


Figure 5: The total number of UTXOs on the system[4]

References

[1] https://en.wikipedia.org/wiki/Merkle_tree

[2] *Bitcoin and Cryptocurrency Technologies*, Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder

[3] https://blog.blockonomics.co/the-economics-of-transaction-fees-b6a7a0365753

[4] https://www.cryptocompare.com/coins/guides/what-is-a-coinbase-or-generation-transac

[5] https://www.quora.com/What-is-a-Gossip-protocol

[6] https://bitcoin.org/en/full-node

[7] Gervais, Arthur, et al. "On the privacy provisions of bloom filters in lightweight bitcoin clients." Proceedings of the 30th Annual Computer Security Applications Conference. ACM, 2014.

[4]Image-credits: https://www.blockchain.com/charts/utxo-count?timespan=all