Introduction to Blockchain: Bitcoin and Ethereum

Alex Miao

Bitcoin

Digital currency with a fixed supply (21 million by year 2140)

Whitepaper released October 31, 2008 pseudonymously under the name Satoshi Nakamoto

Network went live January 3rd, 2009, with the genesis block of 50 BTC being mined

Bitcoin

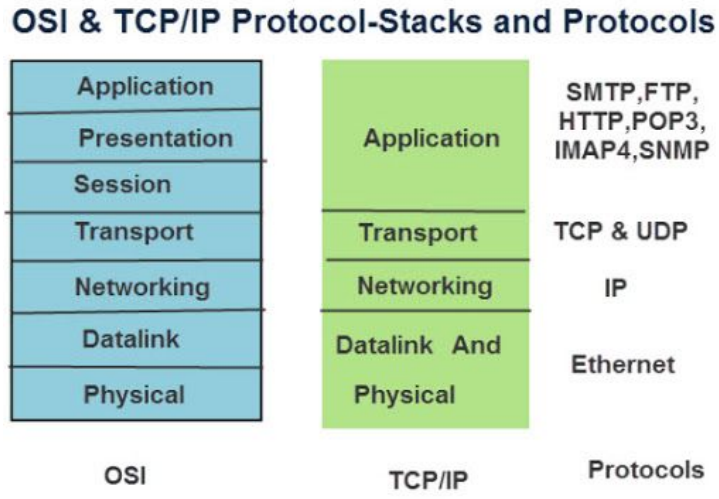
Bitcoin is a protocol over TCP

Individuals download software (a client) that implements this protocol (or parts of it)

The Internet and Protocols

A protocol is a set of rules that people follow

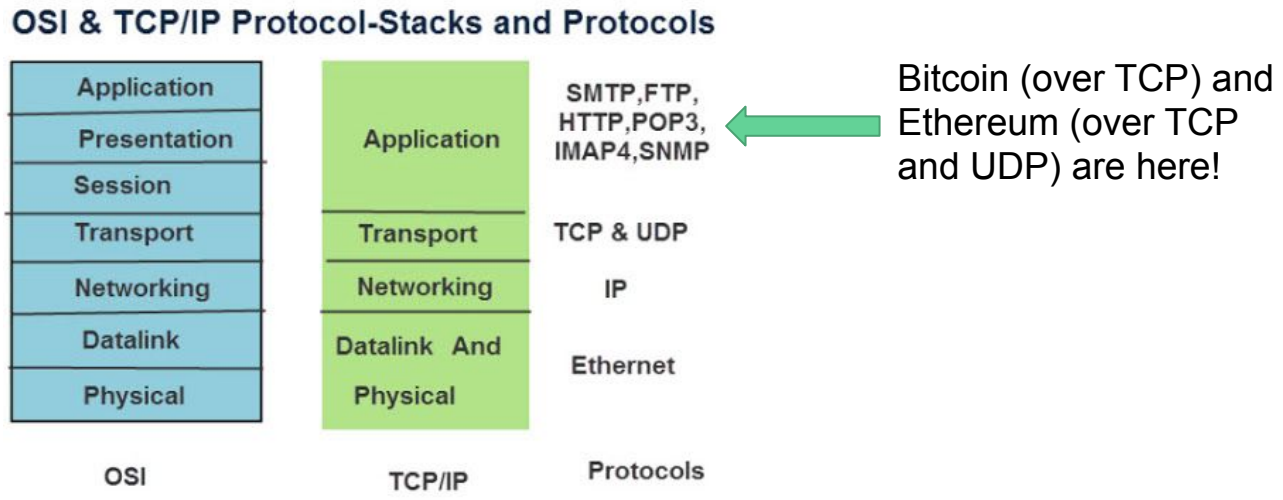
Computer networking uses layers of protocols to provide useful abstraction



The Internet and Protocols

A protocol is a set of rules that people follow

Computer networking uses layers of protocols to provide useful abstraction



Bitcoin Cryptography

Before 1976, cryptography was about securing communication channels

Two breakthrough papers in cryptography:

Diffie-Hellman Key Exchange (1976) - exchange of cryptographic keys over public channel

Rivest, Shamir, Adleman (1977) - public/private keys

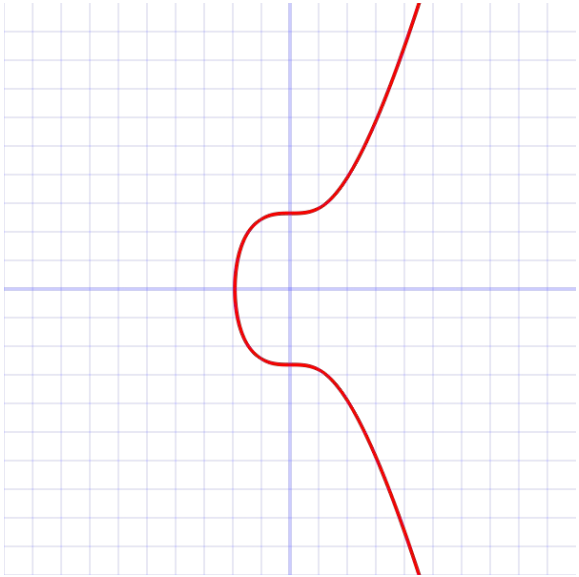
Bitcoin Cryptography

Elliptic Curve Digital Signature Algorithm (ECDSA)

Similar to RSA - creates a public/private key pair, but uses a different underlying algebraic structure (elliptic curves over finite fields rather than cyclic groups of prime order)

Use private key to sign transactions to verify sender identity

Secp256k1 elliptic curve over Real values



Bitcoin Cryptography

Hashing - “one-way” deterministic function that is easy to compute, but hard to find inverse.

Bitcoin mostly uses SHA-256, a specific hash.

> Converts input of arbitrary length into a 256 bit value.

https://xorbin.com/tools/sha256-hash-calculator

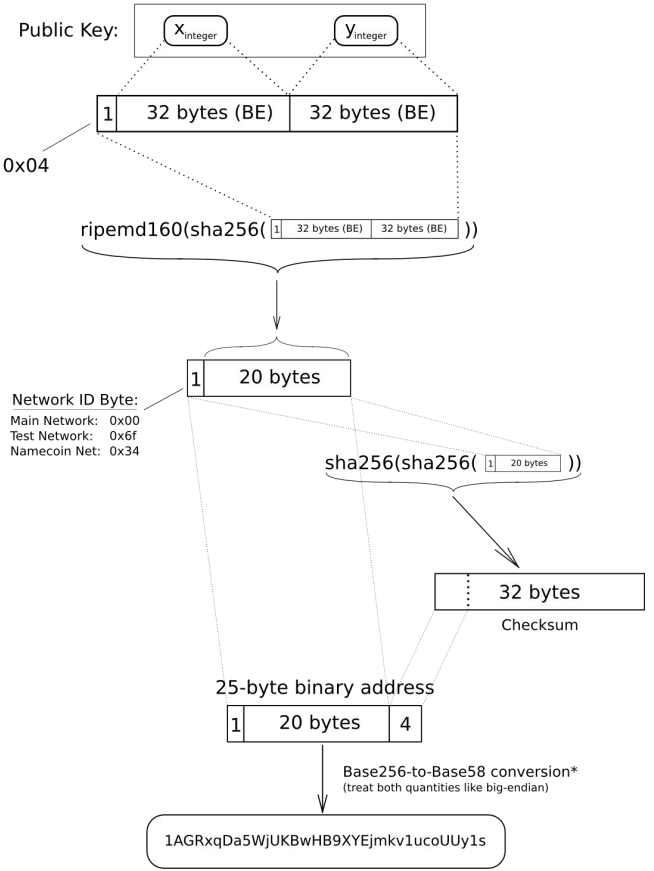
Bitcoin Keys and Addresses

Private Key - randomly generated 256 bit value. Used to sign transactions.

Public Key - 256 bit value that can be derived from private key, but not the other way around

Address - A hash of the public key with a checksum

Elliptic-Curve Public Key to BTC Address conversion



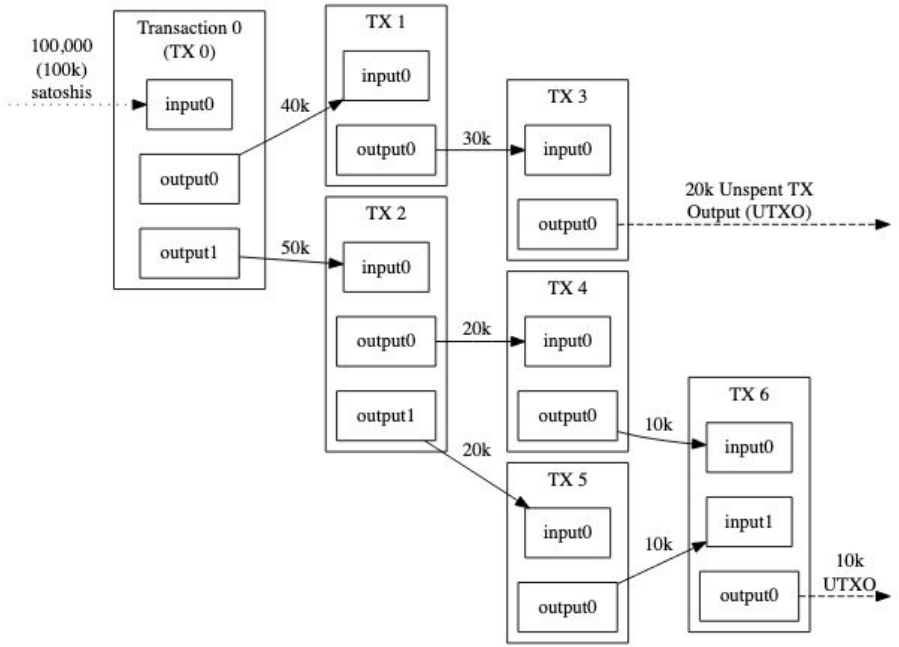
Bitcoin Transactions

Bitcoin is based on unspent transaction outputs (UTXOs)

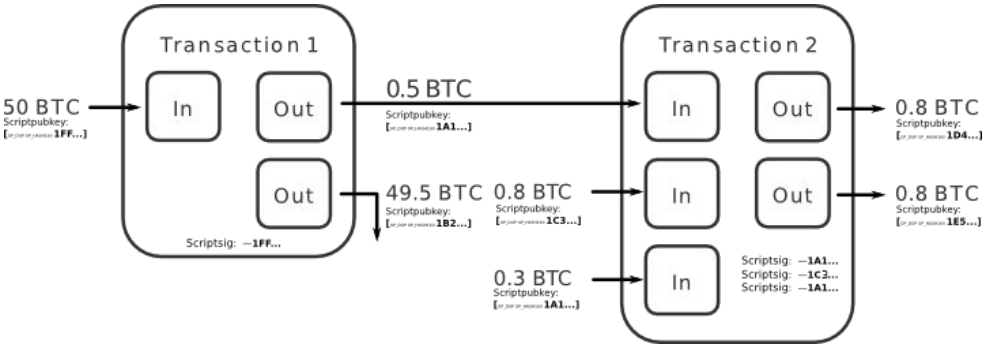
Specify old outputs you own as inputs, and declare new outputs

Broadcast transaction to network

Transactions waiting to be confirmed make up the mempool



Triple-Entry Bookkeeping (Transaction-To-Transaction Payments) As Used By Bitcoin



Anatomy of a Bitcoin Block

Miners take transactions from mempool and put it in a block

Add a coinbase transaction that pays the miner (currently 12.5 BTC)

Block header:

- Previous Block Header Hash
- Merkle Root Hash - A hash representing all transactions in the block
- Nonce - 32 bits used for mining
- Also timestamp, version, and bits (mining difficulty)

https://www.blockchain.com/explorer

Bitcoin Mining

Miners try hashing the block header by changing the nonce until they get a result that has a certain number of zeros

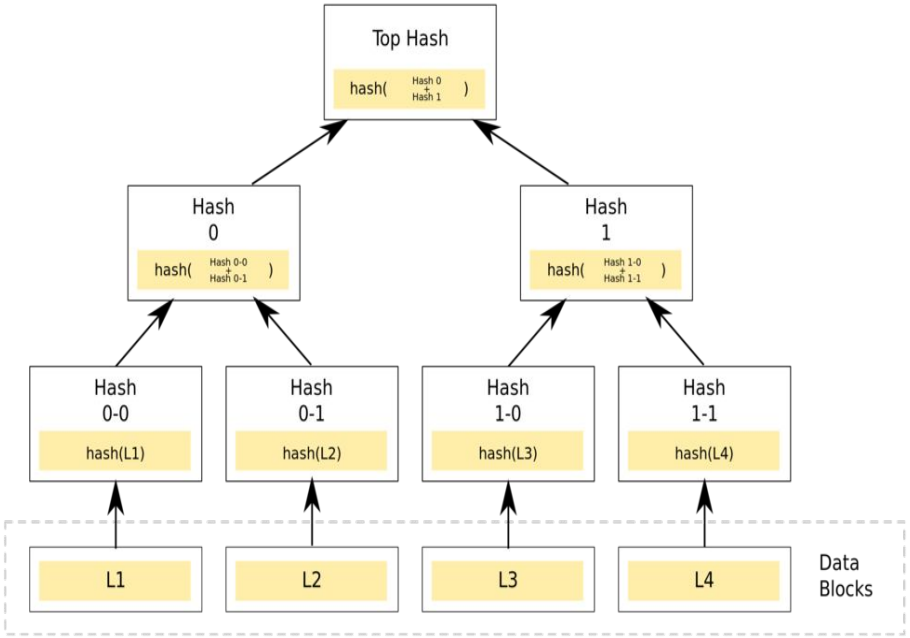
If they get it an answer, they can broadcast the block to the network. Rest of network checks their work, and if it’s ok, everyone moves to next block.

The protocol updates difficulty every 2016 blocks so that blocks average 10 minutes

Merkle Trees

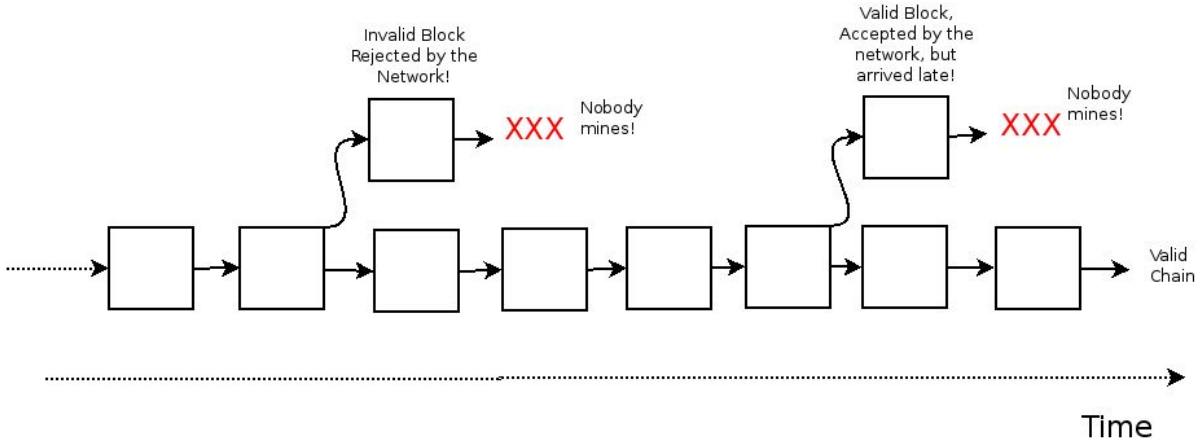
Merkle Trees - Data structure to “summarize” transactions in a block

Transactions are hashed together repeatedly to obtain a single Merkle root used in the block header



Bitcoin Consensus

Everyone in the network agrees that the longest chain is the true ledger



Protocol Forking

Thousands to millions of people independently download different Bitcoin clients, what happens if the system wants to make an update?

Soft fork - backwards compatible upgrade

Hard fork - non-backwards compatible upgrade

Bitcoin Mining in Practice

Application specific integrated circuit (ASIC)

Emergence of large mining pools to decrease variance

Uses lots of electricity (0.28% of worldwide energy consumption)

Bitcoin Scripts

Bitcoin has a simple scripting language that is not Turing-Complete (cannot do loops)

UTXOs can be owned by script rather than individual private key

- Useful in implementing multisig transactions
- Colored coins

Ethereum

Whitepaper released 2013 by Vitalik Buterin

Response to Bitcoin scripts':

- Lack of Turing-completeness
- Value-blindness
- Lack of state
- Blockchain-blindness

Ethereum

Bitcoin - state transition system where state is a set of transactions

Ethereum - state transition system where state tracks the execution of programs.

Ethereum Accounts

Two types of accounts:

- External accounts - Owned by someone who owns the private key
- Contract accounts - No private key, has code that dictates behavior

Ether balance

Contract code

Storage - data for contracts

Account nonce - number of transactions that have been sent by an account

Ethereum Transactions

An Ethereum transaction consists of:

Transaction recipient

Sender signature

Amount of ether to transfer

Data field (for contract code to access)

StartGas - Maximum number of computational steps the program invoked can run

GasPrice - Price per computational step

Ethereum Mining

Total transaction fee is calculated as StartGas * GasPrice

This amount is subtracted from sender account, and increment account nonce

Let Gas = StartGas. Run the contract code, subtracting the appropriate amount from Gas for each line of code executed

If Gas goes to 0, revert transaction, but miner takes the spent Gas

If code finishes with enough Gas, return unused Ether

Smart Contracts

Smart contracts on the Ethereum blockchain

They usually written in Solidity and then compiled down into EVM opcodes

```
function createReward(uint256 _fracNum, uint256 _fracDenom) external onlyBondingManager whenSystemNotPaused returns (uint256) {
    // Compute and mint fraction of mintable tokens to include in reward
    uint256 mintAmount = MathUtils.percOf(currentMintableTokens, _fracNum, _fracDenom);
    // Update amount of minted tokens for round
    currentMintedTokens = currentMintedTokens.add(mintAmount);
    // Minted tokens must not exceed mintable tokens
    require(currentMintedTokens <= currentMintableTokens);
    // Mint new tokens
    livepeerToken().mint(this, mintAmount);

    // Reward = minted tokens
    return mintAmount;
}
```

Sample Solidity function from Livepeer Bonding Manager

Basic Smart Contract Example

```
pragma solidity ^0.4.0;

contract SimpleStorage {
    uint storedData;

    function set(uint x) public {
        storedData = x;
    }

    function get() public view returns (uint) {
        return storedData;
    }
}
```

Simple Storage contract from Solidity documentation

Ethereum Virtual Machine (EVM)

There are many miners all running different computers

The Ethereum Virtual Machine is a low level interface that specifies how programs should be executed

Consists of opcodes

```
PUSH1 0x60 PUSH1 0x40 MSTORE PUSH1 0x18 PUSH1 0x0 SSTORE CALLVALUE
ISZERO PUSH1 0x13 JUMPI PUSH1 0x0 DUP1 REVERT JUMPDEST JUMPDEST
PUSH1 0x36 DUP1 PUSH1 0x21 PUSH1 0x0 CODECOPY PUSH1 0x0 RETURN STOP
PUSH1 0x60 PUSH1 0x40 MSTORE JUMPDEST PUSH1 0x0 DUP1 REVERT STOP
LOG1 PUSH6 0x627A7A723058 KECCAK256 SLT 0xc9 0xbd STOP ISZERO 0x2f
LOG1 0xc4 DUP1 0xf6 DUP3 PUSH32
0x81515BB19C3E63BF7ED9FFBB5FDA0265983AC79800290000000000000000000
```

EVM opcodes

ERC-20 Tokens

An interface for Ethereum Smart Contracts implementing tokens

Shared interface allows for compatibility with programs

Example: 0x Token (ZRX)

6 functions:
totalSupply(): Total supply of Token.

balanceOf(address _owner): The balance in the _owner address.

Transfer(address _to, uint256 _value): Sends a token of _value to address_to, triggering the Transfer event.

transferFrom(address _from, address _to, uint256 _value): Sends a pass from the address_from _value to address_to, triggering the Transfer event.

Approve (address _spender, uint256 _value): Approve _spender to extract a certain amount of money.

Allowance(address _owner, address _spender): Returns the amount that _spender extracted from _owner.

Greedy Heaviest Observed Subtree

Bitcoin block times average 10 minutes

- Takes about 12 seconds to have block propagate to over half the network

But what happens if we have really fast block times?

Greedy Heaviest Observed Subtree (GHOST)

- Count uncle blocks

Use the most “difficult” chain