50.020 Security
Lecture 19 - Introduction to Blockchain

This lecture

This lecture, two applications related to blockchain technology:

- Bitcoin
  - Design Challenge: Digital Currency
  - Bitcoin
- Ethereum and Smart Contract

Bitcoin

Design challenge: currency

- Let's design a digital currency
  - Using content we learned about
  - Ideally, decentralized without server
- Every student here starts with 100 C$
- How can some of this money be transferred to others?
- How can a third party observe a payment?
- How can you prevent double-spending of money?
- How can new money be created?

Simple example (Traditional Currency)

- Preshared symmetric keys and a server
- To send money to Bob, Alice sends message to Server
  - E("Hi, I'm Alice, send C$5 to Bob", k_{AS})
- Server will update database with new account values
- Server will send message to Bob:
  - E("Hi, I'm Server, you received C$5 from Alice", k_{BS})
- Nonces or time-stamps should be added to prevent replay attacks
- This is more or less how online banking works
- Everyone trusts Server, Server can create new money

Why Bitcoin?

Problems with Traditional Currency and Payment

- Inefficient
- Expensive
- Vulnerable

Bitcoin:

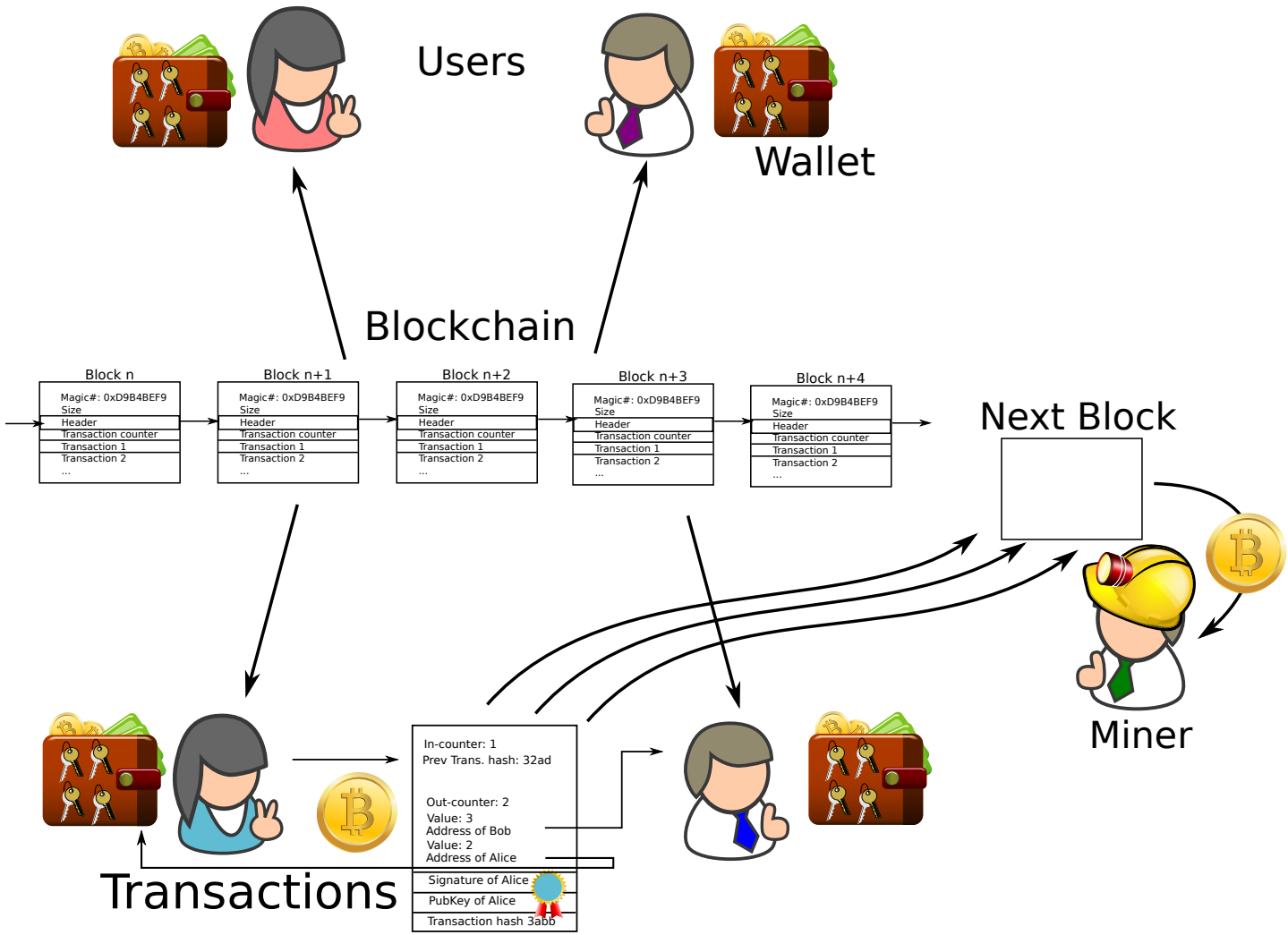
- Saving our time
- Removing/reducing costs (charged by third-party servers)
- Reducing risk/Increasing trust

,

Introduction:

- Bitcoin is a distributed currency/payment system
- Bitcoin was proposed by an anonymous person (Satoshi Nakamoto)
- Bitcoin relies on public-key cryptography and crypto-hashes
- Money is transferred between *wallets*, with public/private key(s)
- Transactions are
  - signed using spending wallet's private key
  - addressed to the receiver's public key (anonymous?)
  - Broadcasted to the bitcoin network
- Global consensus on all valid transactions is achieved through *blockchain*

Bitcoin overview



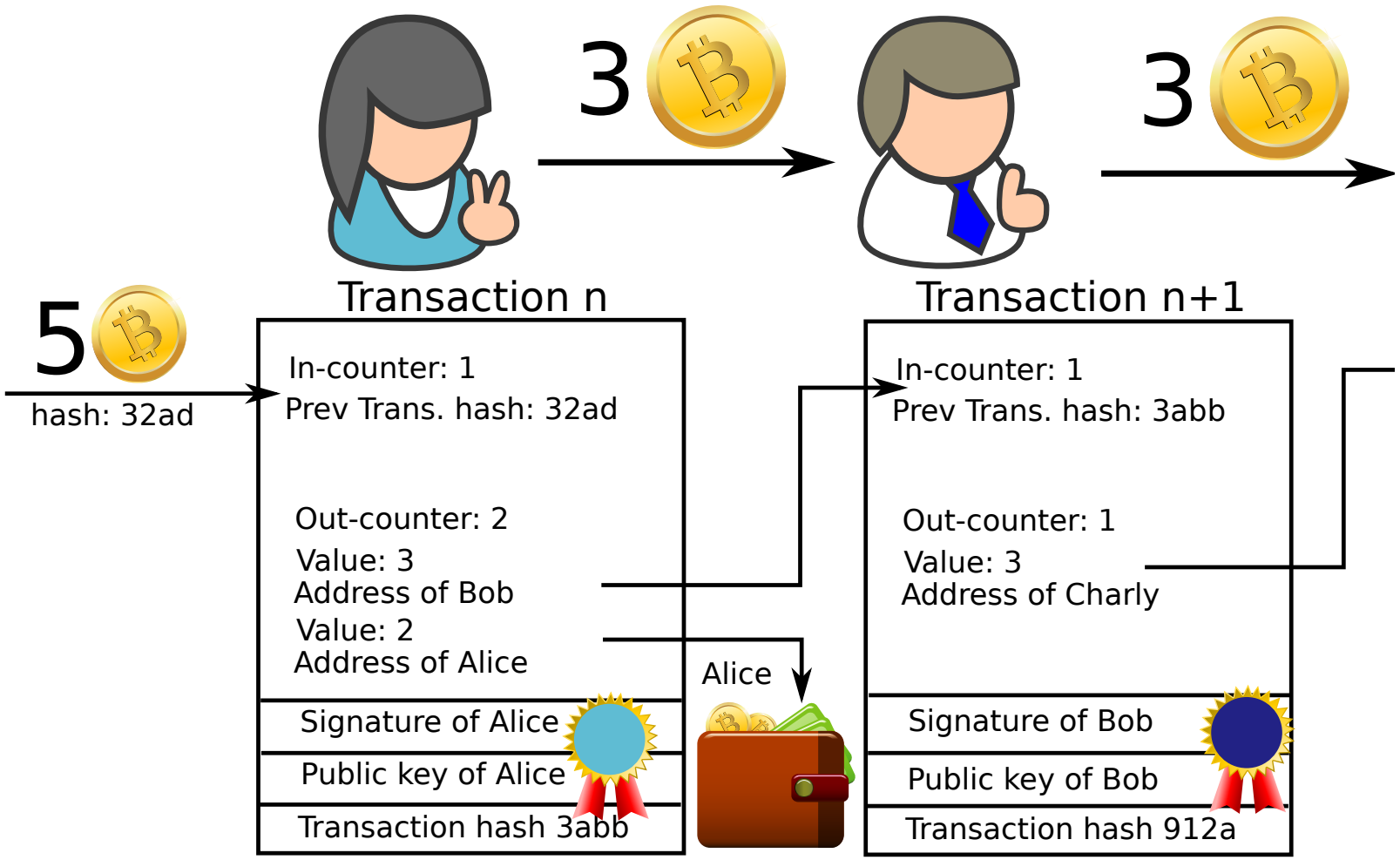
Wallets

- Wallets have 1+ addresses (each 26-35 alphanumeric characters)
- Each Address is actually a 160-bit hash of the public key (in Base58 representation), plus checksum and version number
  - Address=version||base58(RIPEMD-160(SHA256(publicKey)))||checksum

Transactions

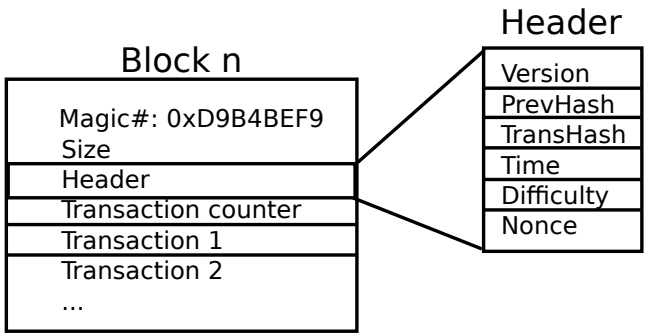
- Transactions are using digital signatures
  - Implemented with ECDSA
- A transaction consists of:
  - Input transaction(s) hash(es)
  - One (or more) target addresses, and respective amounts
  - Signature of spending wallet
- Transactions might have a small fee (more on that later)

Transaction example



Blocks

- A block is a record of some or all of the most recent Bitcoin transactions
  - That have not yet been recorded in any prior blocks
- Tied to the previous block by including the previous block's hash
- A valid block contains a header with the following (some omitted):
  - prevHash: The hash value of previous block
  - transHash: The hash value of all transactions in block (root of Merkle tree)
  - nonce: An incrementing nonce

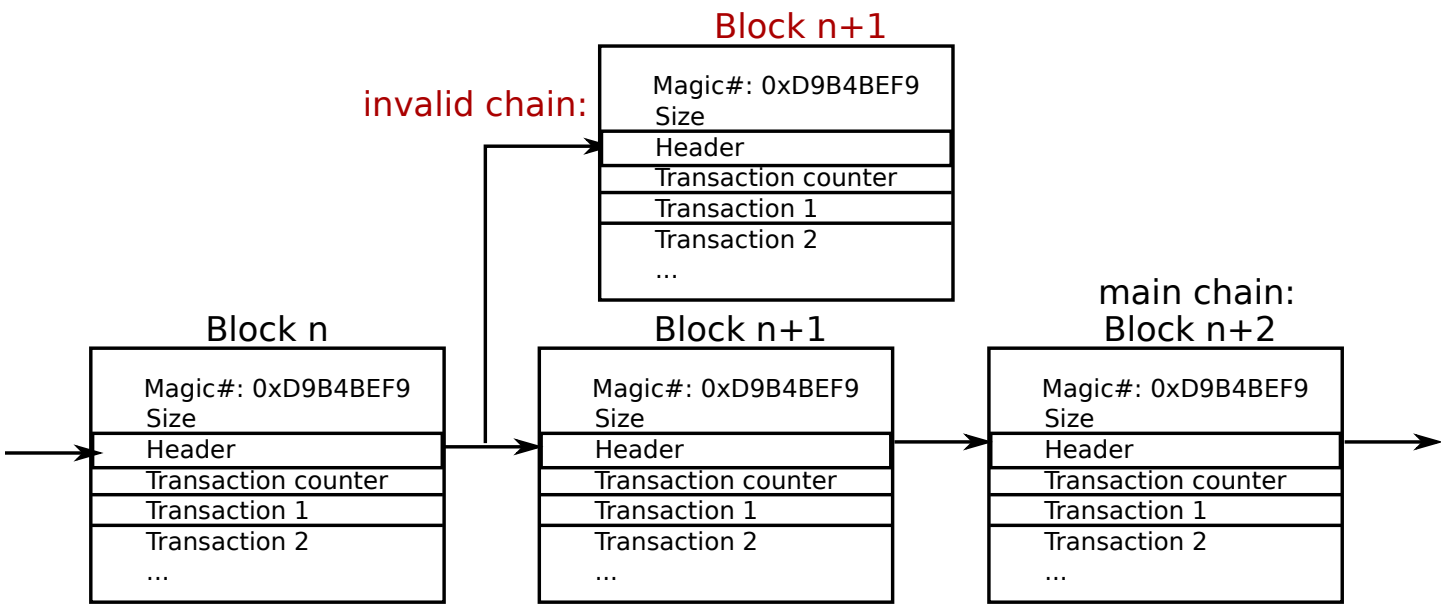


Why is block generation difficult

- Most of the content of new block is fixed
- Goal is to find a nonce such that:
  - The hash value of the block header is smaller than difficulty value
  - Current difficulty: 6,393,023,717,202.
  - Next difficulty estimate: 6,401,686,379,664 (Increased by 0.14%) `https://data.bitcoinity.org/bitcoin/difficulty/30d?t=l`
- This is probabilistic, so only trying all possible nonces will work
- The process of trying different solutions is called mining
- Once a working nonce is found, the finder broadcasts the block to the whole network
- There are two incentives for mining: new bitcoins, and transaction fees

Blockchain

- The blockchain is the *global state* of the bitcoin network
- It consists of valid blocks, chained together by their hashes
- When valid block is found, everyone starts mining for the next
- *The longest valid chain is the accepted one*

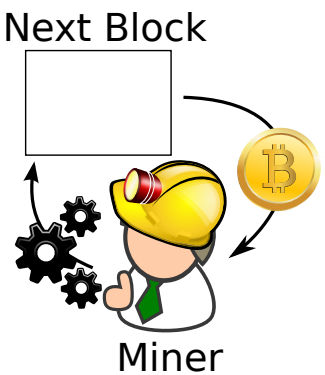


Difficulty

- Difficulty determines how small the hash needs to be
- Goal: block to be generated every 10 minutes on average
- Estimated network strength (Nov 2014): 0.288 PHash/s (PHash/s=10^12 Hash/s)
- Estimated network strength (April 2017): 4000 PHash/s
- Current estimated network strength (April 2019): 46080 PHash/s (28,4000 PHash/s in April 2018)
- Current block#: 571821 (330624 in Nov 14, 462239 in Apr 2017, 517463 in April 2018)
- Mining has become professional:
  - Mining pools
  - ASIC(Application-Specific Integrated Circuits)s and FPGA(Field-Programmable Gate Array)s for mining
- But due to increasing difficulty, hardware quickly becomes obsolete

Rewards for mining

- Every new block contains a genesis transaction that creates new bitcoins
- The miner will put his address as destination
- If the block gets accepted, the miner can spend those coins
- Current bitcoin price: 5,044 USD
- In addition, transaction fees are supposed to incentivise mining
  - Transactions with higher "bonus" fee get included faster
  - Currently, only about 1/10 BTC per block is awarded this way



Transaction acceptance

- Transactions become valid after
  - They are first listed in a valid block
  - That block has been followed by about 5 block in the blockchain
  - This will prevent potential double-spending with two parallel block chains
- Miners will check signatures in transactions before including
  - Invalid transaction lead to invalid blocks, miner loses effort
- De-facto, it is a majority vote on which current chain is the accepted one

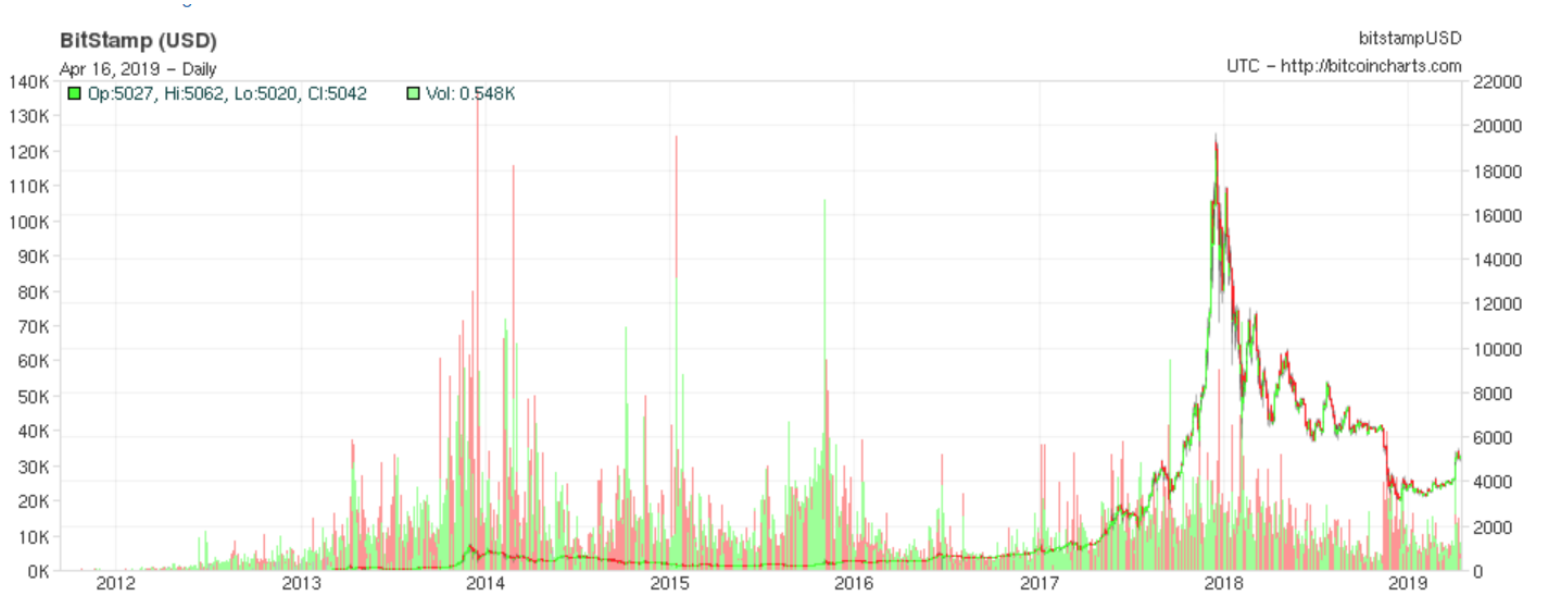
Losing wallets...

- You need your private key to spend any money
- If you lose your private key, your money is lost
- Internet is full of story of lost and found wallets
- Theoretically, no way to recover money or key
  - Unless you convince the global network to reject incoming transactions...

Attacks on Bitcoin

- To fake transactions, attacker must
  - Either generate victim's signature
  - Or introduce manipulated blocks into blockchain
- Attacker can try to *double-spend*
  - Chain two transactions to the same source transaction
  - The second transaction will be rejected in block generation
  - In worst case, both victims could have different chains
    - This would lead to a split of the global state, be detected
- If attacker can somehow generate blocks faster than rest of network:
  - Attacker determines which transactions are accepted, and which are rejected!

Market development



- Bitcoin value *exploded* in 2013 and 2017
  - 2013: \$22, 2014: \$1000, 2017: \$1000, 2018: \$20,000
- Everyone is sad they didn't buy BTC pre 2013/2017...

Legal and economic aspects

- Bitcoins are not directly regulated
- No banks/governments are involved
- BTC could be used to do money laundry
- BTC has been used to buy/sell drugs, weapons, etc...
  - As wallets are anonymous
- But all transactions are public
- If wallet owner is somehow identified, all transactions can be reconstructed

What is Blockchain?

- Blockchain ($\neq$ bitcoin) is the underpinning technology that maintains the bitcoin transaction ledger.
- A blockchain is a continuously growing list of records, called blocks, which are linked and secured using cryptography
- A blockchain is an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way

Ethereum and Smart Contract

Why Ethereum and Smart Contract?

- Most public blockchains are cryptocurrencies
  - Can only transfer coins between users
- Smart contracts enable much more applications (asset issuance, crowdfunding, domain registration, title registration, gambling, prediction markets, internet of things, voting, etc.)

Ethereum is:

- a blockchain with expressive programming language Programming language makes it ideal for smart contracts
- the first blockchain-based smart contract platform

How Ethereum Works

Two types of account:

- Normal account like in Bitcoin
  - has balance and address
- Smart Contract account
  - like an object: containing (i) code, and (ii) private storage (key-value storage)
  - code can
    - Send ETH to other accounts. ETH is the cryptocurrency used in the Ethereum Platform
    - Read/write storage
    - Call (i.e., start execution in) other contracts

Ethereum Languages

Ethereum's smart contracts can be written in:

- Solidity (a language library with similarities to C and JavaScript),
- Serpent (similar to Python, but deprecated),
- LLL (a low-level Lisp-like language)
- Mutan (Go-based, but deprecated)
- Vyper (research-oriented language, under development)

Smart Contract Example (Solidity)


```
1   pragma solidity ^0.4.15;
2
3   contract Hello {
4
5     string public message;
6
7     function Hello(string initialMessage) public {
8       message = initialMessage;
9     }
10
11    function setMessage(string newMessage) public {
12      message = newMessage;
13    }
14  }
```

Transactions in Ethereum

- Normal transactions like Bitcoin transactions
  - Send tokens between accounts
- Transactions to contracts
  - like function calls to objects
  - specify which object you are talking to, which function, and what data (if possible)
- Transactions to create contracts

Transactions

- nonce (anti-replay-attack)
- to (destination address)
- value (amount of ETH to send)
- data (readable by contract code)
- gasprice (amount of ether per unit gas)
- startgas (maximum gas consumable)
- v, r, s (ECDSA signature values)

How to Create a Contract?

Submit a transaction to the blockchain with:

- nonce: previous nonce + 1
- to: empty
- value: value sent to the new contract
- data: contains the code of the contract
- gasprice (amount of ether per unit gas)
- startgas (maximum gas consumable)
- v, r, s (ECDSA signature values)

If transaction is successful, returns the address of the new contract

How to Interact With a Contract?

Submit a transaction to the blockchain with:

- nonce: previous nonce + 1
- to: contract address
- value: value sent to the new contract
- data: data supposed to be read by the contract
- gasprice (amount of ether per unit gas)
- startgas (maximum gas consumable)
- v, r, s (ECDSA signature values)

If transaction is successful, returns outputs from the contract (if applicable)

What is gas?

- Problem to solve: a malicious miner may DoS attack full nodes by including lots of computation in their transactions.
- Solution: Charge fee per computational step (i.e.,gas). Special gas fees for operations that take up storage

Conclusion

Two applications of blockchain technology:

- Bitcoin as a digital currency supporting distributed payement
- Etheruem and smart contracts for other blockchain-based applications