Proof of Stake 1

Murat Osmanoglu

Designing a Security Protocol

- Objective
- Resources
- Threat Model
- Algorithm
- Assumption

Designing a Security Protocol

- define an Objective that you would like to achieve

Designing a Security Protocol

- define an Objective that you would like to achieve
  - designing a ledger that blockchain protocol is used to construct
  - designing an unforgeable digital signature scheme

Designing a Security Protocol

- define an Objective that you would like to achieve
  - designing a ledger that blockchain protocol is used to construct
  - designing an unforgeable digital signature scheme
- analyze the Resources that are available to the parties which are using the algorithm to meet the objective

Designing a Security Protocol

- define an Objective that you would like to achieve
  - designing a ledger that blockchain protocol is used to construct
  - designing an unforgeable digital signature scheme
- analyze the Resources that are available to the parties which are using the algorithm to meet the objective
- design the Threat Model to describe what the adversary is allowed to do and what it is not allowed to do

Designing a Security Protocol

- define an Objective that you would like to achieve
  - designing a ledger that blockchain protocol is used to construct
  - designing an unforgeable digital signature scheme
- analyze the Resources that are available to the parties which are using the algorithm to meet the objective
- design the Threat Model to describe what the adversary is allowed to do and what it is not allowed to do
  - to have a good threat model, think exactly what will happen when the algorithm are being executed in the real world (it should reflect the real-time scenario)

Designing a Security Protocol

- design the Algorithm (or Protocol) that uses the available resources, and achieves the objective given the threat model

Designing a Security Protocol

- design the Algorithm (or Protocol) that uses the available resources, and achieves the objective given the threat model
  - formally prove that this is true!

Designing a Security Protocol

- design the Algorithm (or Protocol) that uses the available resources, and achieves the objective given the threat model
  - formally prove that this is true!
- find a proper Assumption, difficult to solve, that is used when we argue that the algorithm achieves the objective

Designing a Security Protocol

- design the Algorithm (or Protocol) that uses the available resources, and achieves the objective given the threat model
  - formally prove that this is true!
- find a proper Assumption, difficult to solve, that is used when we argue that the algorithm achieves the objective

establish the proof : the algorithm meets the objective given the resources in the threat model we have specified, under the assumption we have described

Designing a Security Protocol

- design the Algorithm (or Protocol) that uses the available resources, and achieves the objective given the threat model
  - formally prove that this is true!
- find a proper Assumption, difficult to solve, that is used when we argue that the algorithm achieves the objective

establish the proof : the algorithm meets the objective given the resources in the threat model we have specified, under the assumption we have described

implementation

Designing a Security Protocol

- Objective – Distributed Ledger with two properties that refer to a log of transactions

Designing a Security Protocol

- Objective – Distributed Ledger with two properties that refer to a log of transactions
  - Persistence(safety) :
  - Liveness :

Designing a Security Protocol

- Objective – Distributed Ledger with two properties that refer to a log of transactions
  - Persistence(safety) : view of the log of the transactions is stable. If you ask any node the log of the transactions, it will give you the same version of the log.
  - Liveness :

Designing a Security Protocol

- Objective – Distributed Ledger with two properties that refer to a log of transactions
  - Persistence(safety) : view of the log of the transactions is stable. If you ask any node the log of the transactions, it will give you the same version of the log.
  - Liveness : new transactions are added to the ledger at regular frequency. There is an upper bound at which the transactions send to the nodes will be included in the log.

Designing a Security Protocol

- Resources – the parties have a private memory
  - the ability to sample random strings
  - access to a network that is relatively synchronous

Designing a Security Protocol

- Resources – the parties have a private memory
  - the ability to sample random strings
  - access to a network that is relatively synchronous

  - the private memory can only be accessed by the associated party

Designing a Security Protocol

- Resources – the parties have a private memory
  - the ability to sample random strings
  - access to a network that is relatively synchronous

  - the private memory can only be accessed by the associated party

  - sampling randomness should also be private in the sense that an attacker from outside cannot subvert the randomness

Designing a Security Protocol

- Resources – the parties have a private memory
  - the ability to sample random strings
  - access to a network that is relatively synchronous

  - the private memory can only be accessed by the associated party
  - sampling randomness should also be private in the sense that an attacker from outside cannot subvert the randomness
  - when a party sends a message, it will be going to reach to other parties

Designing a Security Protocol

- Threat Model – the adversary controls a number of parties running the protocol

Designing a Security Protocol

- Threat Model – the adversary controls a number of parties running the protocol
  - there should be a bound on how many parties the adversary can control.
  - depending on the setting the bound is adjusted to meet the objectives

Designing a Security Protocol

- Threat Model – the adversary controls a number of parties running the protocol

  - there should be a bound on how many parties the adversary can control.

  - depending on the setting the bound is adjusted to meet the objectives

    In PoW setting, the adversary is allowed to control the minority of the total hash power which is available to network

    In PoS setting, the adversary is allowed to control the minority of the total stake that is recorded in the ledger

Bitcoin

Bitcoin

- a public ledger, maintained by the nodes in peer-to-peer network, which records all the transactions

Bitcoin

- a public ledger, maintained by the nodes in peer-to-peer network, which records all the transactions
- transactions are chosen from the 'mempool', and added to the ledger by Proof of Work at certain frequency

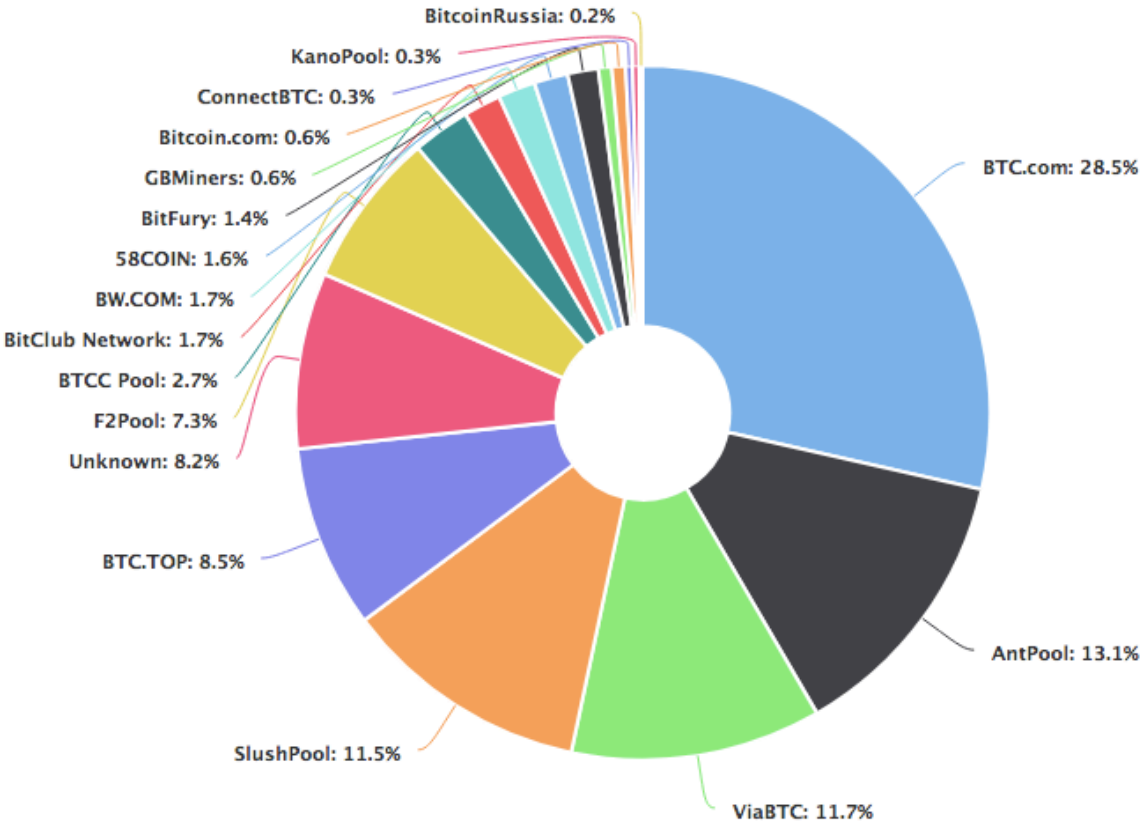
Bitcoin

- a public ledger, maintained by the nodes in peer-to-peer network, which records all the transactions
- transactions are chosen from the 'mempool', and added to the ledger by Proof of Work at certain frequency
- Proof of Work can be considered as solving some difficult problem so that nodes cannot add new blocks arbitrarily

Bitcoin

- a public ledger, maintained by the nodes in peer-to-peer network, which records all the transactions
- transactions are chosen from the 'mempool', and added to the ledger by Proof of Work at certain frequency
- Proof of Work can be considered as solving some difficult problem so that nodes cannot add new blocks arbitrarily
- the difficulty level of the puzzle is adjusted continually according to the current computation power of the network

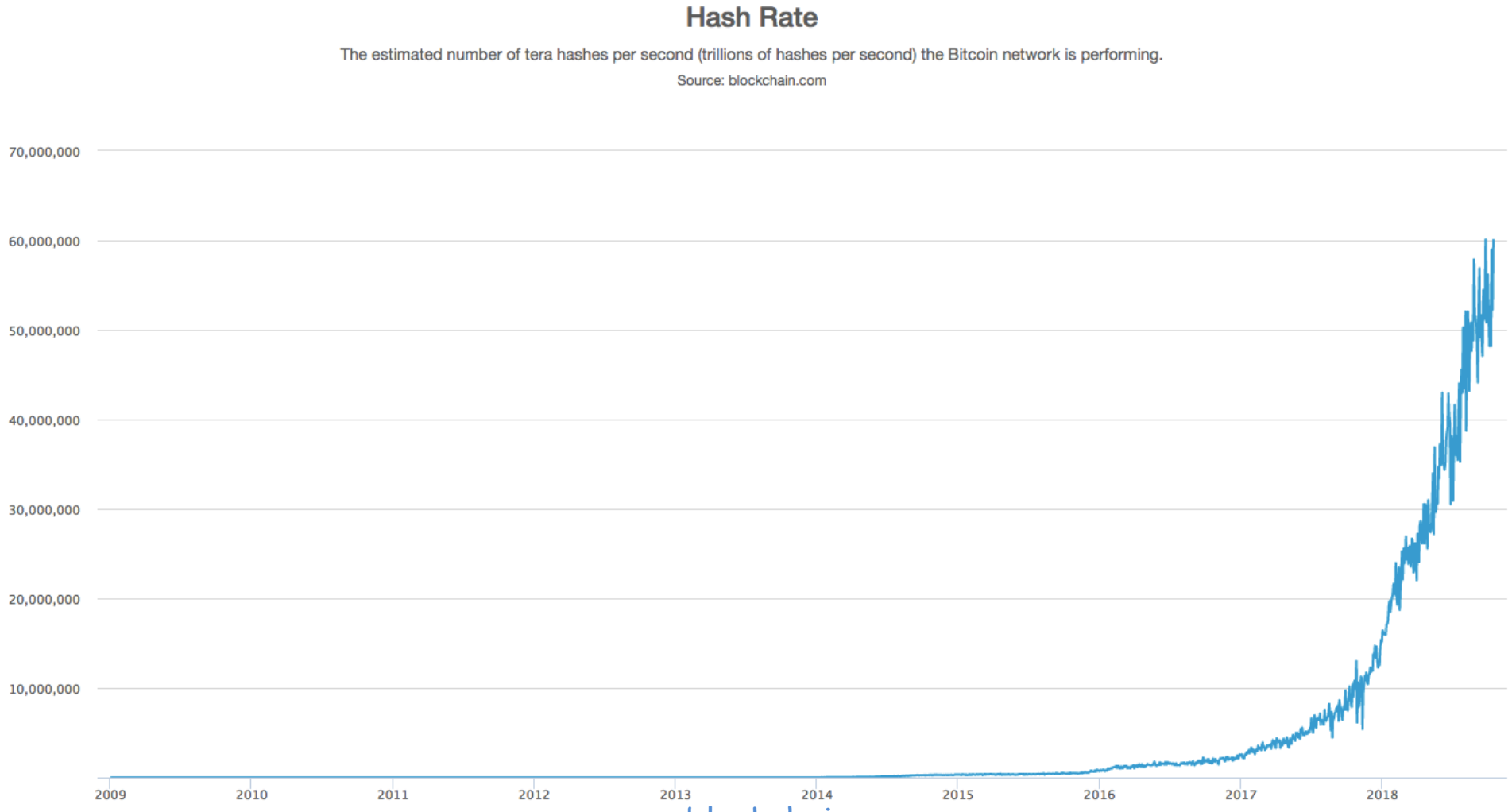
Disadvantages of Bitcoin



www.blockchain.com

Hashrate distribution
among the largest pools

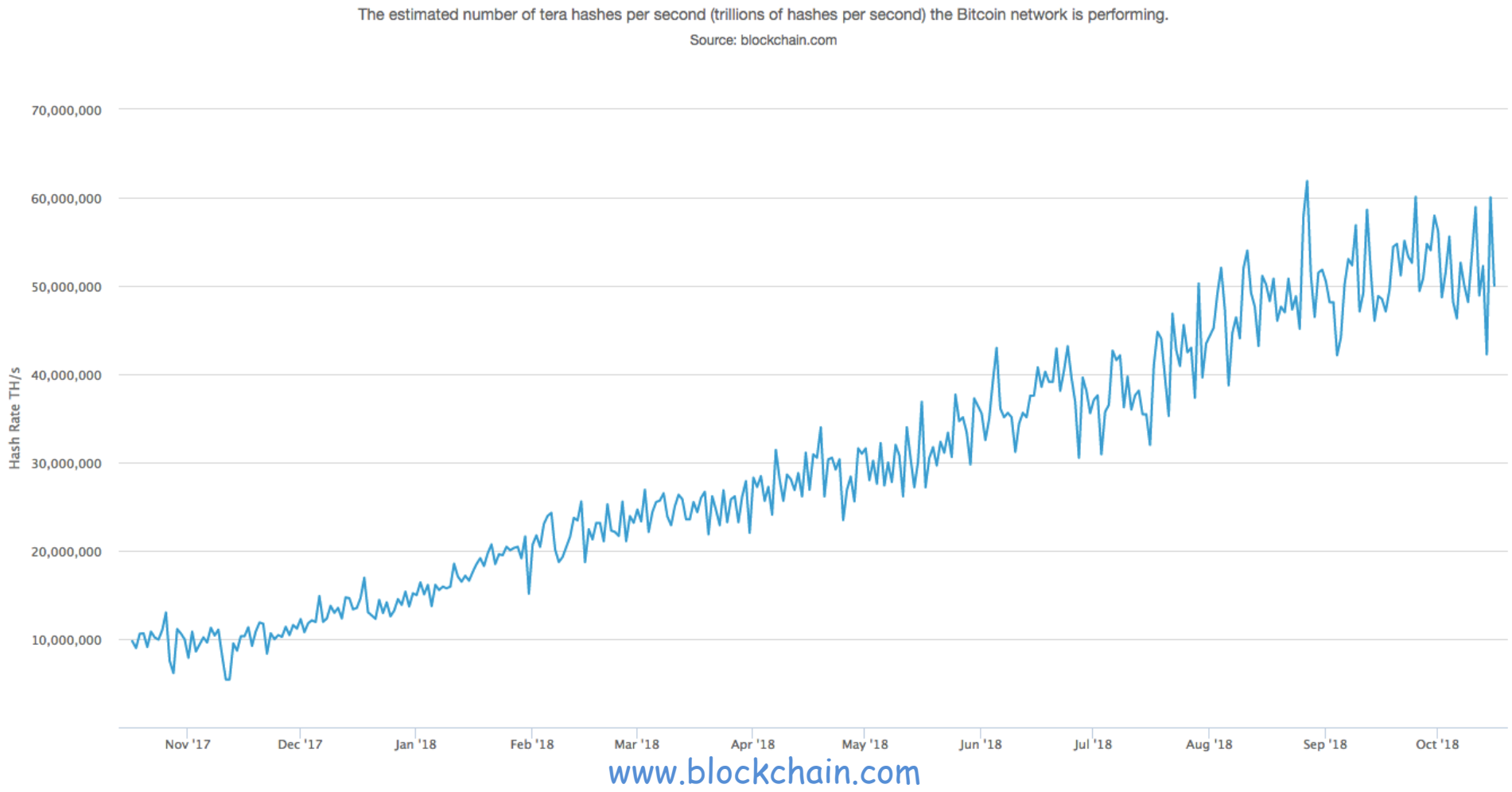
Disadvantages of Bitcoin



www.blockchain.com

The estimated number of tera hashes per second
(trillions of hashes per second) the Bitcoin network is performing

Disadvantages of Bitcoin



Last recorded one (daily) : 54,105,921,231 GH/s

Disadvantages of Bitcoin

- last recorded one(daily) : 54.105.921.231 GH/s

Disadvantages of Bitcoin

- last recorded one(daily) : 54.105.921.231 GH/s
- estimated the electricity consumption : a rate of 650 watts per GH/s

Disadvantages of Bitcoin

- last recorded one(daily) : 54.105.921.231 GH/s
- estimated the electricity consumption : a rate of 650 watts per GH/s
- ≈ 35,1 gigawatts ➔ ≈ 12.811 gigawatt-hours per year

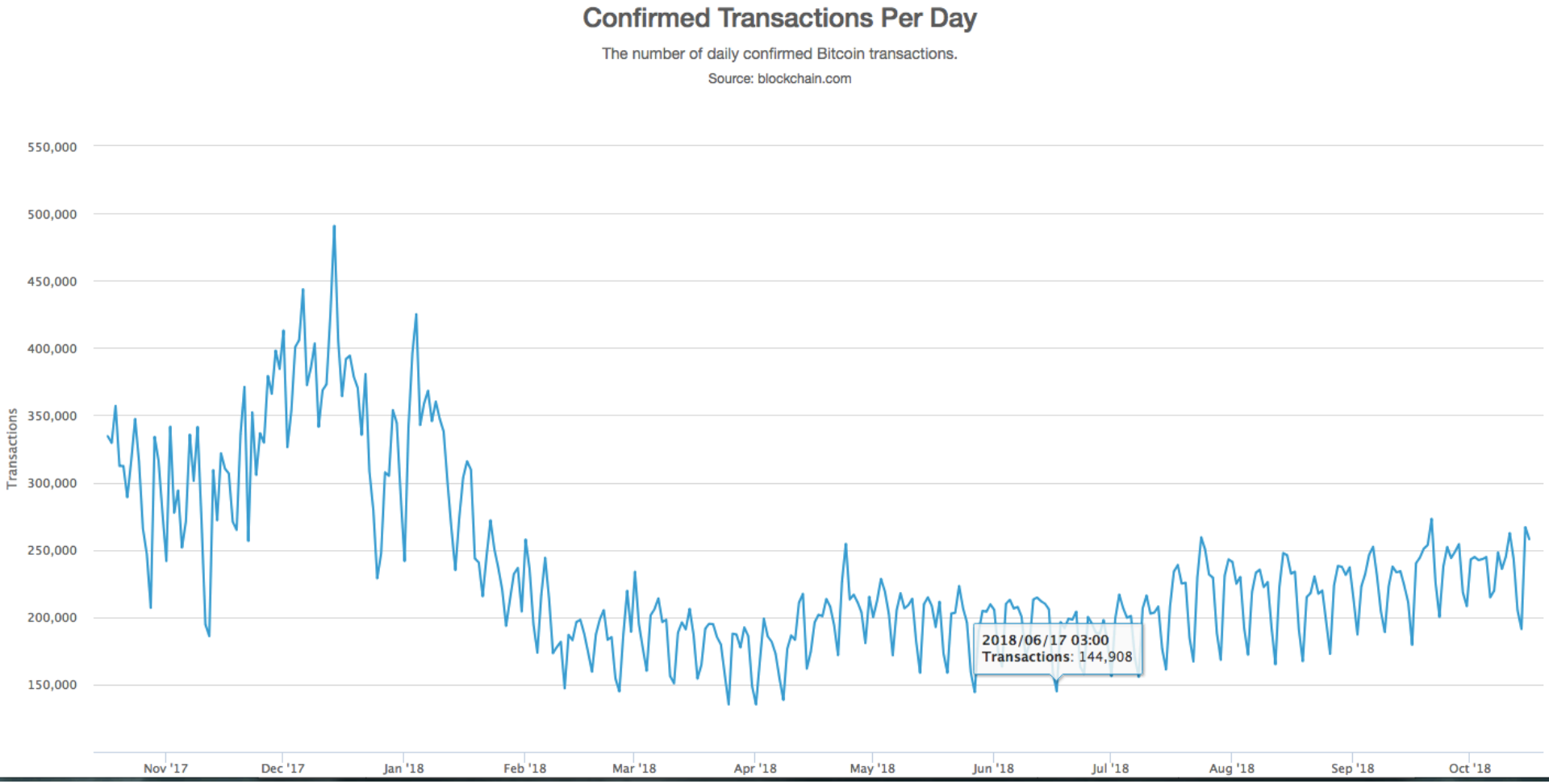
Disadvantages of Bitcoin

- last recorded one(daily) : 54.105.921.231 GH/s
- estimated the electricity consumption : a rate of 650 watts per GH/s
- ≈ 35,1 gigawatts ➔ ≈ 12.811 gigawatt-hours per year
- total consumption of Bosnia ≈ 11.400 gWh

Disadvantages of Bitcoin

- last recorded one(daily) : 54.105.921.231 GH/s
- estimated the electricity consumption : a rate of 650 watts per GH/s
- ≈ 35,1 gigawatts ➔ ≈ 12.811 gigawatt-hours per year
- total consumption of Bosnia ≈ 11.400 gWh
- total consumption of Turkey ≈ 213.200 gWh

Disadvantages of Bitcoin



www.blockchain.com

Last recorded one (daily) : 266.308

≈ 1850 per block

Disadvantages of Bitcoin

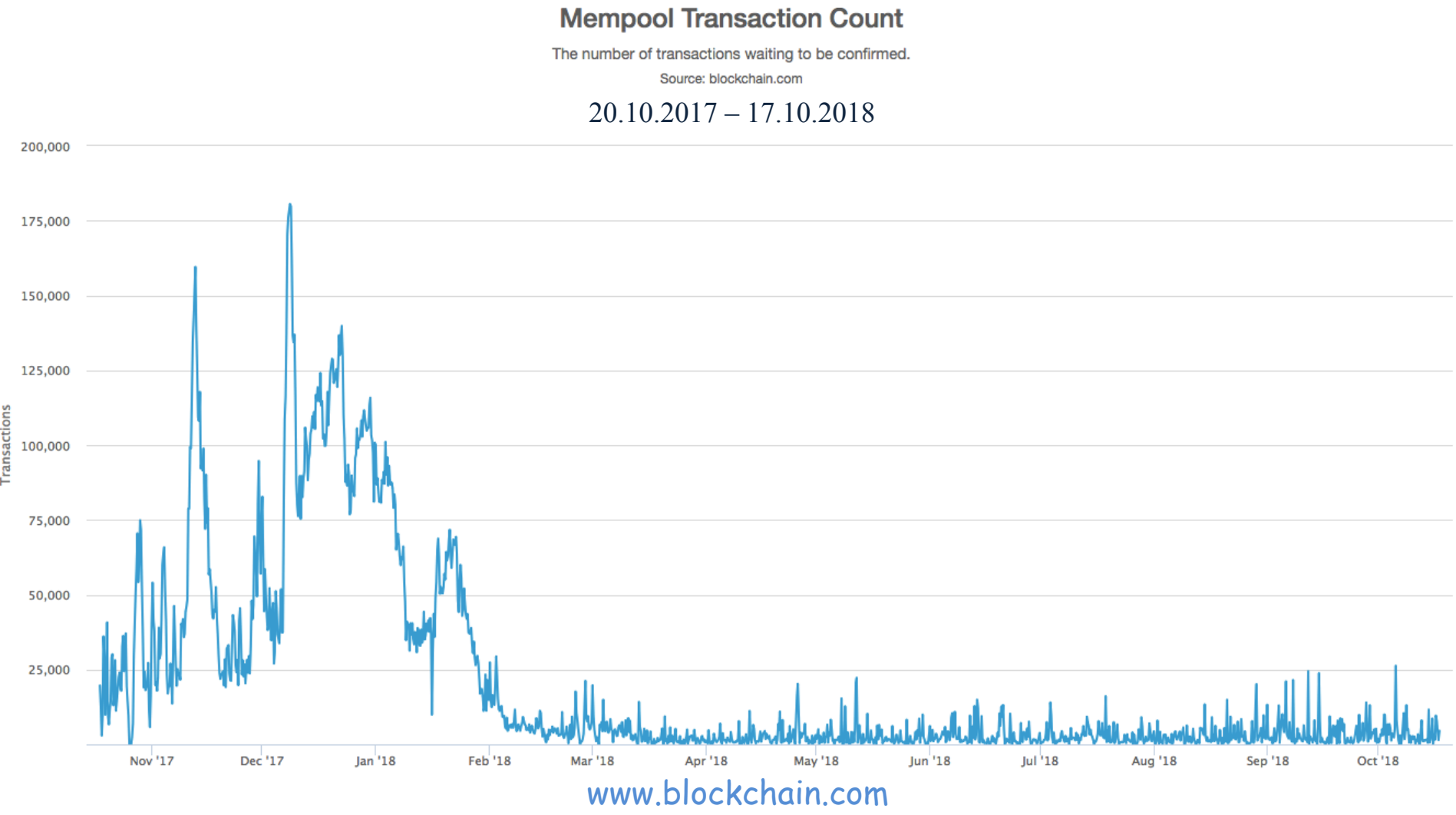
- ≈ 1850 transactions per block
  ≈ 3 transactions per second
- 25 transactions per second for Ethereum
- 61 transactions per second for BitcoinCash
- 193 transactions per second for PayPal
- 1500 transactions per second for Ripple
- 1667 transactions per second for Visa

Disadvantages of Bitcoin

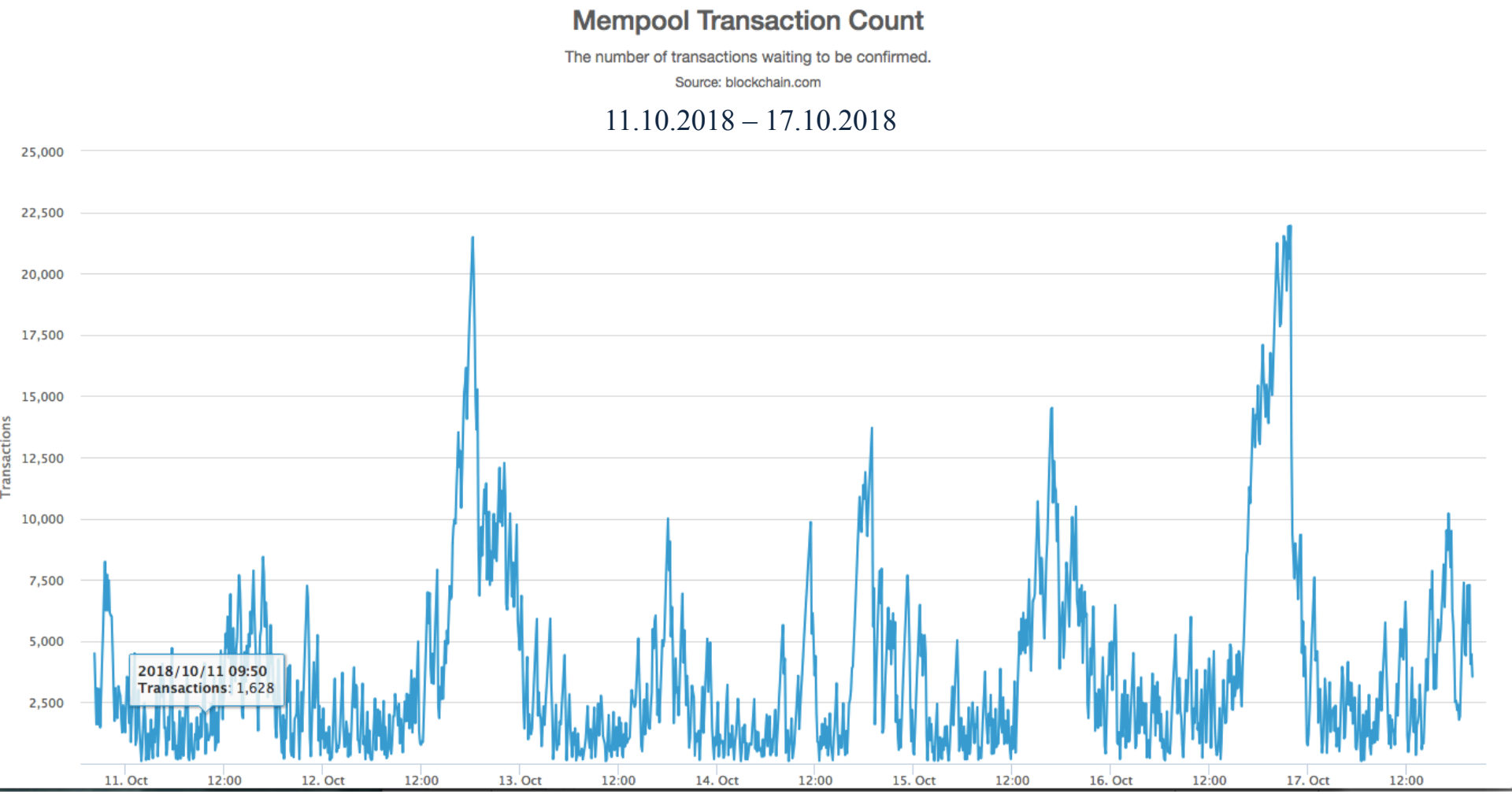
24.10.2017 – 16.10.2018

www.blockchain.com

Disadvantages of Bitcoin



Disadvantages of Bitcoin



www.blockchain.com

Bitcoin

- there is a genesis block which is extended by parties as they find Proof of Work

Bitcoin

- there is a genesis block which is extended by parties as they find Proof of Work
- each block that is appended to the chain, contains transactions

Bitcoin

- there is a genesis block which is extended by parties as they find Proof of Work
- each block that is appended to the chain, contains transactions
- consider Proof of Work as an election

Bitcoin

- there is a genesis block which is extended by parties as they find Proof of Work
- each block that is appended to the chain, contains transactions
- consider Proof of Work as an election
  - at any given moment, one of the parties is elected to create the next block based on the hash power the parties have

Bitcoin

- there is a genesis block which is extended by parties as they find Proof of Work
- each block that is appended to the chain, contains transactions
- consider Proof of Work as an election
  - at any given moment, one of the parties is elected to create the next block based on the hash power the parties have
  - random sampling by those trying to create the block

Proof of Stake

Proof of Stake

- first introduced by user 'QuantumMechanic' at bitcointalk.org in 2011

Proof of Stake

- first introduced by user 'QuantumMechanic' at bitcointalk.org in 2011

- the parties who hold the stake in the system are well-suited to maintain the ledger since their stake will diminish in value when the security of the system collapses

Proof of Stake

- first introduced by user 'QuantumMechanic' at bitcointalk.org in 2011
- the parties who hold the stake in the system are well-suited to maintain the ledger since their stake will diminish in value when the security of the system collapses
- adding the concept 'stake' on the consensus mechanism

Proof of Stake

- first introduced by user 'QuantumMechanic' at bitcointalk.org in 2011
- the parties who hold the stake in the system are well-suited to maintain the ledger since their stake will diminish in value when the security of the system collapses
- adding the concept 'stake' on the consensus mechanism

Tendermint

peercoin

CARDANO

NXT

NeuCoin

Peercoin

PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake

Sunny King, Scott Nadal
(sunnyking9999@gmail.com, scott.nadal@gmail.com)

August 19th, 2012

Abstract

A peer-to-peer crypto-currency design derived from Satoshi Nakamoto's Bitcoin. Proof-of-stake replaces proof-of-work to provide most of the network security. Under this hybrid design proof-of-work mainly provides initial minting and is largely non-essential in the long run. Security level of the network is not dependent on energy consumption in the long term thus providing an energy-efficient and more cost-competitive peer-to-peer crypto-currency. Proof-of-stake is based on coin age and generated by each node via a hashing scheme bearing similarity to Bitcoin's but over limited search space. Block chain history and transaction settlement are further protected by a centrally broadcasted checkpoint mechanism.

Introduction

Since the creation of Bitcoin (Nakamoto 2008), proof-of-work has been the predominant design of peer-to-peer crypto currency. The concept of proof-of-work has been the backbone of minting and security model of Nakamoto's design.

Peercoin

" . . . the crypto-currency is dependent on energy consumption, thus introducing significant cost overhead in the operation of such networks, which is borne by users wia a combination of inflation and transaction fees. As the mint slows in Bitcoin network, eventually it could put pressure on raising transaction fees to sustain a preferred level of security."

Peercoin

- Coin Age : currency amount X holding period

Peercoin

- Coin Age : currency amount X holding period
  - if Bob received 10 coins from Alice and kept it for 10 days, then Bob has accumulated 1000 coin days of coin age
  - when Bob spent these coins, the coin age had been consumed
  - coin age can be considered as a form of PoS

Peercoin

- Coin Age : currency amount X holding period
  - if Bob received 10 coins from Alice and kept it for 10 days, then Bob has accumulated 1000 coin days of coin age
  - when Bob spent these coins, the coin age had been consumed
  - coin age can be considered as a form of PoS
- It's a hybrid design such that both PoS and PoW are used in the consensus

Peercoin

- Coin Age : currency amount X holding period
  - if Bob received 10 coins from Alice and kept it for 10 days, then Bob has accumulated 1000 coin days of coin age
  - when Bob spent these coins, the coin age had been consumed
  - coin age can be considered as a form of PoS
- It's a hybrid design such that both PoS and PoW are used in the consensus
  - In PoW, there is fixed target value applying to everynode
  - In PoS, target value changes for each node. Coin Age determines the target value of the node

Peercoin

- Coin Age : currency amount X holding period
  - if Bob received 10 coins from Alice and kept it for 10 days, then Bob has accumulated 1000 coin days of coin age
  - when Bob spent these coins, the coin age had been consumed
  - coin age can be considered as a form of PoS
- It's a hybrid design such that both PoS and PoW are used in the consensus
  - In PoW, there is fixed target value applying to everynode
  - In PoS, target value changes for each node. Coin Age determines the target value of the node
- In bitcoin, mint-rate is determined by block height. However, in peercoin, it is determined by difficulty.

Peercoin

- Coin Age : currency amount X holding period
  - if Bob received 10 coins from Alice and kept it for 10 days, then Bob has accumulated 1000 coin days of coin age
  - when Bob spent these coins, the coin age had been consumed
  - coin age can be considered as a form of PoS
- It's a hybrid design such that both PoS and PoW are used in the consensus
  - In PoW, there is fixed target value applying to everynode
  - In PoS, target value changes for each node. Coin Age determines the target value of the node
- In bitcoin, mint-rate is determined by block height. However, in peercoin, it is determined by difficulty. Each 16x raise of mining difficulty halves the block mint amount.

Peercoin

$$\text{hash}(\text{prev\_data}, \text{time}, \text{txout}_A) \leq d_0.\text{coins}(\text{txout}_A).\text{timeweight}(\text{txout}_A)$$

Peercoin

$$\text{hash}(\text{prev\_data}, \text{time}, \text{txout}_A) \leq d_0.\text{coins}(\text{txout}_A).\text{timeweight}(\text{txout}_A)$$

- prev_data : previous blocks data

Peercoin

$$\text{hash}(\text{prev\_data}, \text{time}, \text{txout}_A) \leq d_0.\text{coins}(\text{txout}_A).\text{timeweight}(\text{txout}_A)$$

- prev_data : previous blocks data
- time : time in seconds that restricts the hash attemps to 1 per second.

Peercoin

$$\text{hash}(\text{prev\_data}, \text{time}, \text{txout}_A) \leq d_0.\text{coins}(\text{txout}_A).\text{timeweight}(\text{txout}_A)$$

- prev_data : previous blocks data
- time : time in seconds that restricts the hash attemps to 1 per second.

  Note that nodes will regard a new block as invalid if the difference between its time and their local time is not within the bound

Peercoin

$$\text{hash}(\text{prev\_data}, \text{time}, \text{txout}_A) \leq d_0.\text{coins}(\text{txout}_A).\text{timeweight}(\text{txout}_A)$$

- prev_data : previous blocks data
- time : time in seconds that restricts the hash attemps to 1 per second.

  Note that nodes will regard a new block as invalid if the difference between its time and their local time is not within the bound
- d_0 : difficulty level that is readjusted according to a rule such that blocks should be created in 10 minutes on average.

Peercoin

$$\text{hash}(\text{prev\_data}, \text{time}, \text{txout}_A) \leq d_0.\text{coins}(\text{txout}_A).\text{timeweight}(\text{txout}_A)$$

- prev_data : previous blocks data
- time : time in seconds that restricts the hash attemps to 1 per second.

  Note that nodes will regard a new block as invalid if the difference between its time and their local time is not within the bound
- d_0 : difficulty level that is readjusted according to a rule such that blocks should be created in 10 minutes on average.
- $\text{coins}(\text{txout}_A)$: the amount of coins of some unspent transaction output txout_A

Peercoin

$$\text{hash}(\text{prev\_data}, \text{time}, \text{txout}_A) \leq d_0.\text{coins}(\text{txout}_A).\text{timeweight}(\text{txout}_A)$$

- prev_data : previous blocks data
- time : time in seconds that restricts the hash attemps to 1 per second.

  Note that nodes will regard a new block as invalid if the difference between its time and their local time is not within the bound
- d_0 : difficulty level that is readjusted according to a rule such that blocks should be created in 10 minutes on average.
- $\text{coins}(\text{txout}_A)$: the amount of coins of some unspent transaction output txout_A
- $\text{timeweight}(\text{txout}_A)$: it is proportional to the time elapsed since the transaction whose output is txout_A was included into a block

Peercoin

- Rational Forks : on every second Pr[some block is solved] ≈ 1/600, thus multiple blocks will be solved simultaneously every ≈ 4 days.

Peercoin

- Rational Forks : on every second Pr[some block is solved] ≈ 1/600, thus multiple blocks will be solved simultaneously every ≈ 4 days.

  Rational stakeholders can increase their expected reward by maintaining and trying to solve blocks on the multiple forked chains

Peercoin

- Rational Forks : on every second Pr[some block is solved] ≈ 1/600, thus multiple blocks will be solved simultaneously every ≈ 4 days.

  Rational stakeholders can increase their expected reward by maintaining and trying to solve blocks on the multiple forked chains

- Bribe Attacks : An attacker can publicly announce his intend to create a fork to reverse the last 6 blocks after the merchant sends the goods, and offer bribes to stakeholders in order to sign his blocks

Peercoin

- Rational Forks : on every second Pr[some block is solved] ≈ 1/600, thus multiple blocks will be solved simultaneously every ≈ 4 days.

  Rational stakeholders can increase their expected reward by maintaining and trying to solve blocks on the multiple forked chains

- Bribe Attacks : An attacker can publicly announce his intend to create a fork to reverse the last 6 blocks after the merchant sends the goods, and offer bribes to stakeholders in order to sign his blocks

- Timeweight : an attacker can boost his chances to generate consecutive blocks by waiting.

Peercoin

- Rational Forks : on every second Pr[some block is solved] ≈ 1/600, thus multiple blocks will be solved simultaneously every ≈ 4 days.

  Rational stakeholders can increase their expected reward by maintaining and trying to solve blocks on the multiple forked chains

- Bribe Attacks : An attacker can publicly announce his intend to create a fork to reverse the last 6 blocks after the merchant sends the goods, and offer bribes to stakeholders in order to sign his blocks

- Timeweight : an attacker can boost his chances to generate consecutive blocks by waiting.

  They changed the rule with *PPCoin v0.3* as timeweight stops growing after 90 days