Перевод цифровых ценностей

Мешков Дмитрий Александрович
Осень 2017

Упрощенная транзакция

- У каждого участника сети есть своя пара ключей (PK, SK)
- Отпрвка денег кому-то – это отправка на его публичный ключ
- Транзакция подписана ключем отправителя
- Элис опрпавляет Amount монет Бобу:

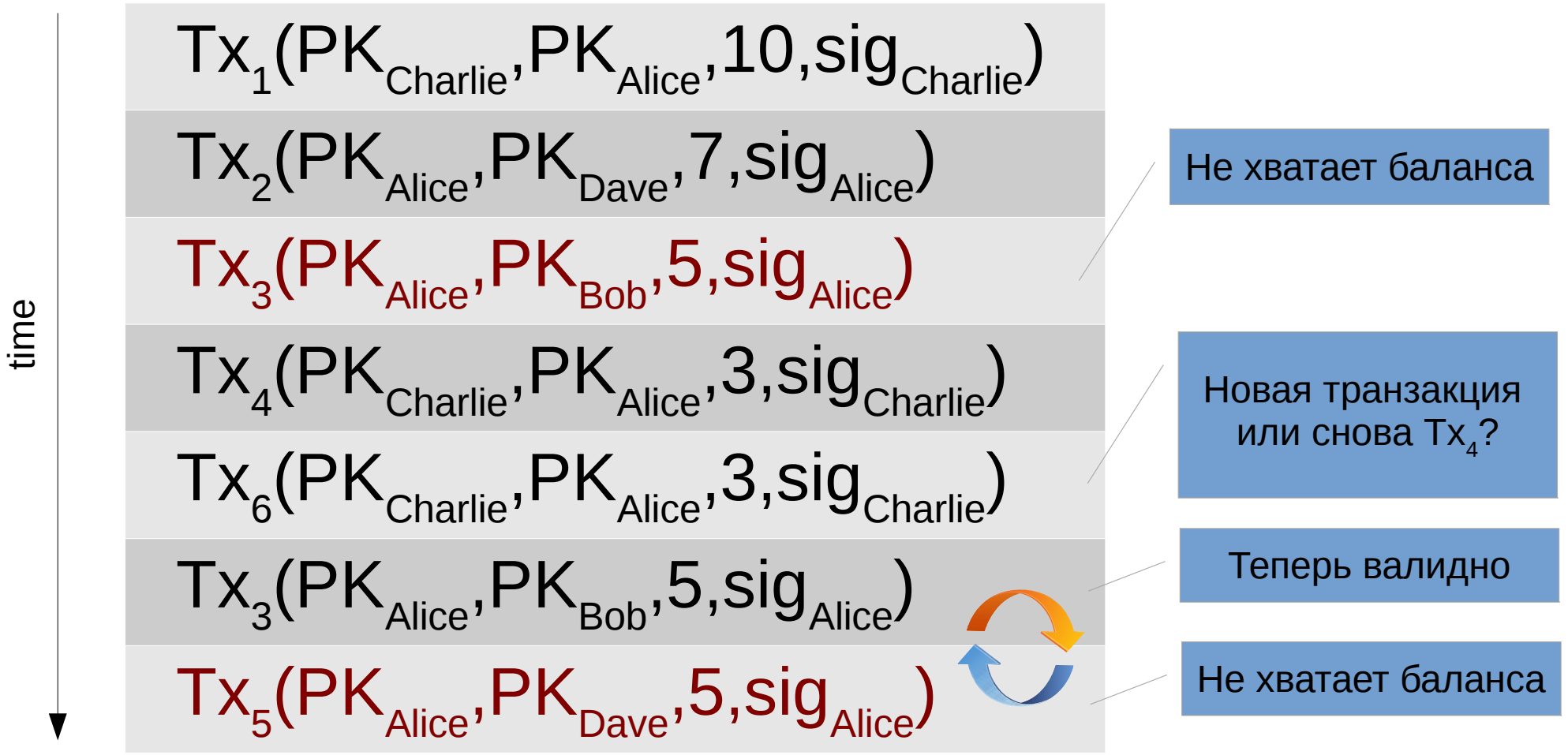
$Tx(sender = PK_{Alice},$
$recipient = PK_{Bob},$
$amount = Amount,$
$signature = sig_{Alice})$

Валидация транзакции

Проверки:

- Amount > 0
- PK_{Bob} – валидный публичный ключ
- Sig_{Alice} – валидная подпись данных (Amount,PK_{Bob}) относительно публичного ключа PK_{Alice}
- **Баланс (сумма приходов – сумма расходов) PK_{Alice} больше Amount**
- **Транзакция замечена впервые**

История транзакций



- **Tx_3 стала валидной спустя какое-то время**
- **Нужно уметь различать Tx_4 и Tx_6**
- Валидна только одна из двух транзакций Tx_3 и Tx_5

Упрощенная транзакция

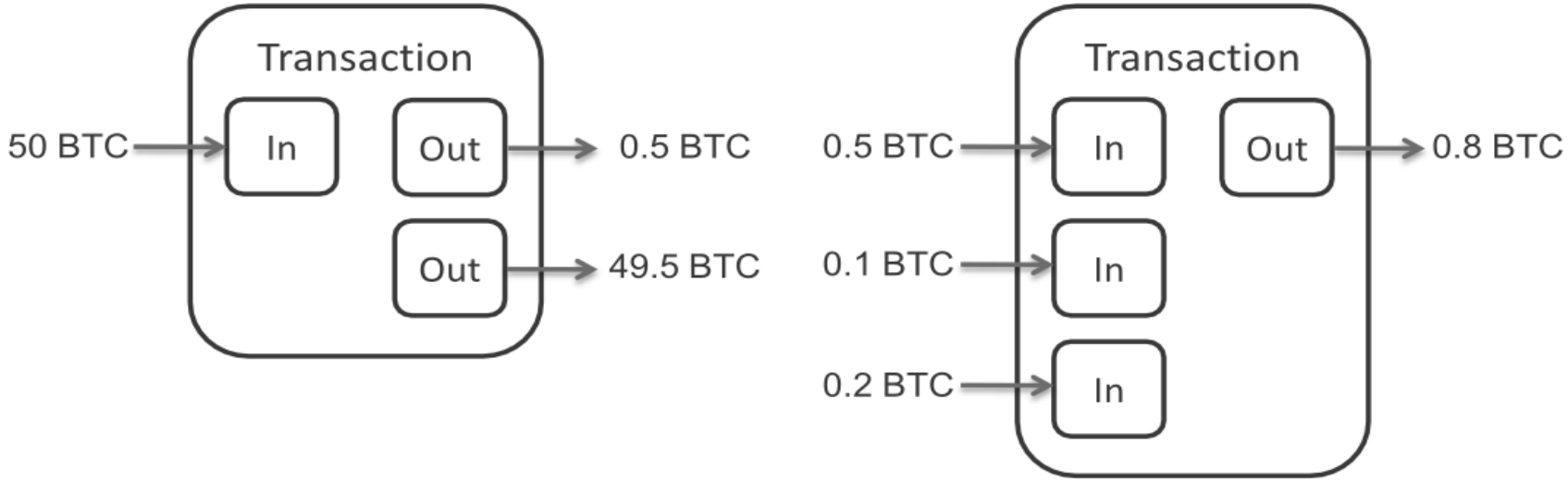
Tx_3 стала валидной спустя какое-то время

- Ограниченное время жизни тразакции
- Многие не решают этой проблемы

Нужно уметь различать Tx_4 и Tx_6

- Данные в транзакции должны быть уникальными
- Дополнительное поле в транзакции (nonce, timestamp, …)

Транзакции сети Биткоин



- Есть только входы и выходы
- Входы – это непотраченные выходы из предыдущих транзакций
- Выходы тратятся только полностью
- ∑входов≥∑выходов (кроме coinbase транзакции)
- К выходу привязывается скрипт его траты
- Чтобы потратить выход, нужно подать на вход скрипта такие данные, чтобы получить true

Транзакции сети Биткоин

metadata

input(s)

output(s)


```
{
    "hash":"5a42590fbe0a90ee8e8747244d6c84f0db1a3a24e8f1b95b10c9e050990b8b6b",
    "ver":1,
    "vin_sz":2,
    "vout_sz":1,
    "lock_time":0,
    "size":404,
    "in":[
     {
      "prev_out":{
       "hash":"3be4ac9728a0823cf5e2deb2e86fc0bd2aa503a91d307b42ba76117d79280260",
       "n":0
      },
        "scriptSig":"30440..."
     },
     {
      "prev_out":{
       "hash":"7508e6ab259b4df0fd5147bab0c949d81473db4518f81afc5c3f52f91ff6b34e",
       "n":0
      },
      "scriptSig":"3f3a4ce81...."
     }
    ],
    "out":[
     {
      "value":"10.12287097",
      "scriptPubKey":"OP_DUP OP_HASH160 69e02e18b5705a05dd6b28ed517716c894b3d42e OP_EQUALVERIFY OP_CHECKSIG"
     }
    ]
}
```

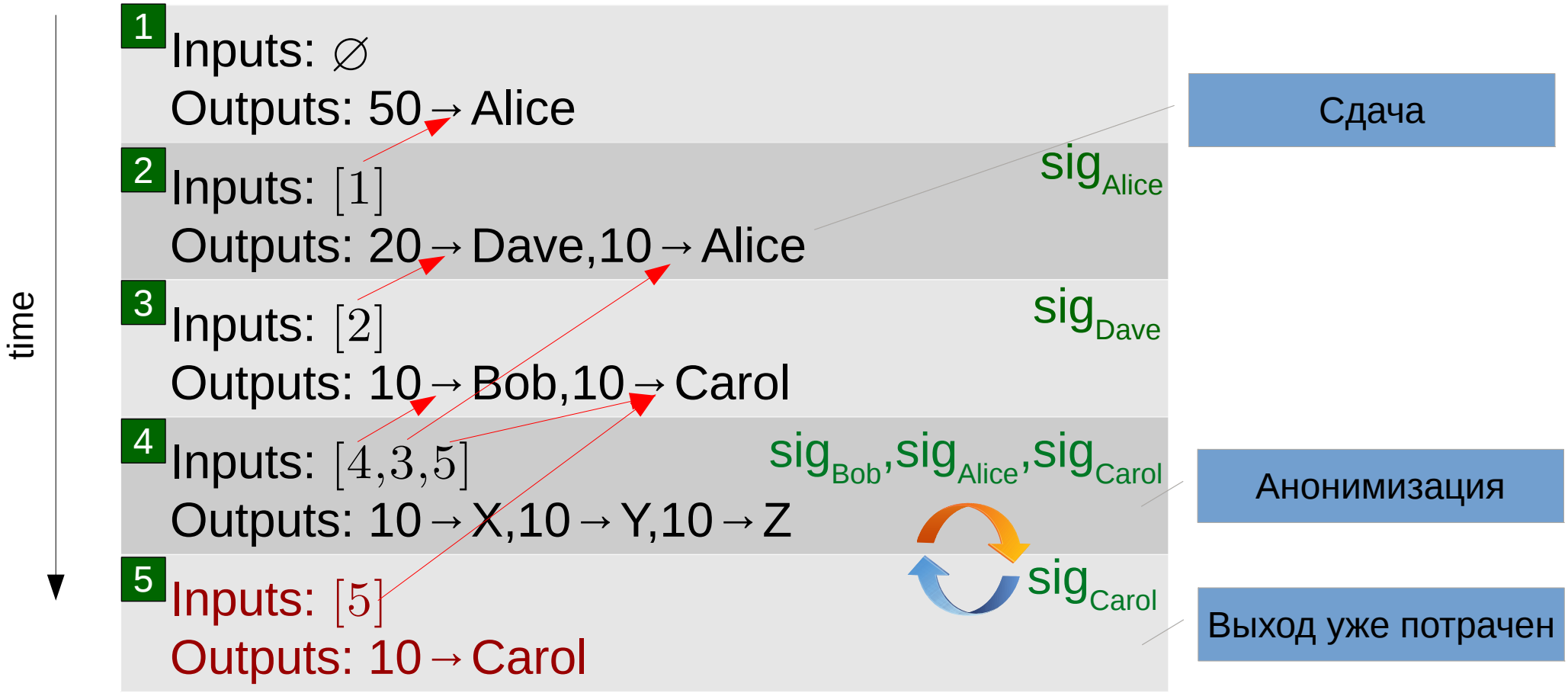
Транзакции сети Эфириум

- Изменяемый баланс
- Аутентификация – только подпись secp256k1
- Дополнительная защита от replay атак – поле nonce
- Nonce увеличивается строго на 1

Транзакции сети Эфириум

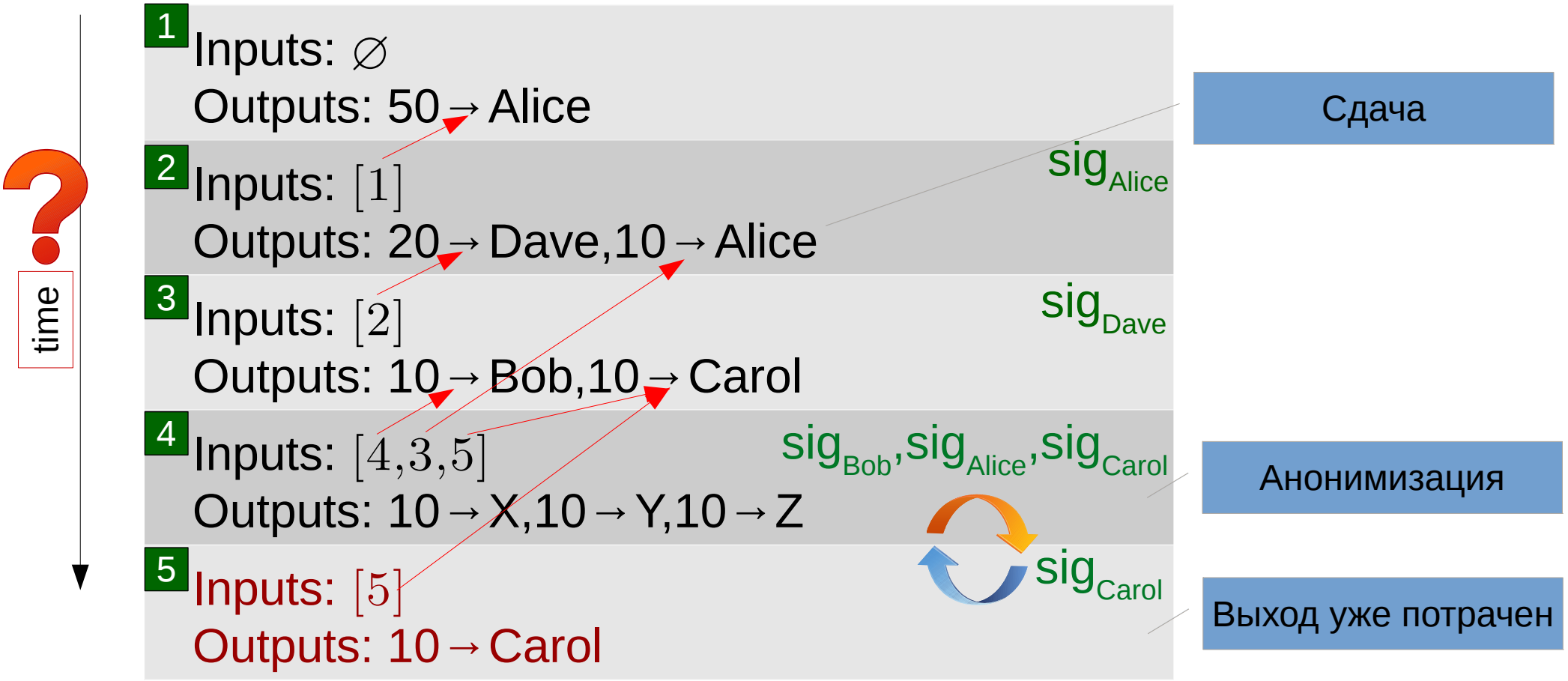
```
{
    "hash":"0xc6ef2f...",
    "nonce":"0x",
    "from":"0x407d73...",
    "to":"0x85h43d...",
    "value":"0x7f110" // 520464
    "gas": "0x7f110" // 520464
    "gasPrice":"0x09184e72a000",
    "input":"0x603880...",
}
```

История транзакций Биткоин



- Не потратить еще не полученных монет
- Транзакции всегда различимы
- **Валидна только одна из транзакций Tx_4 и Tx_5**

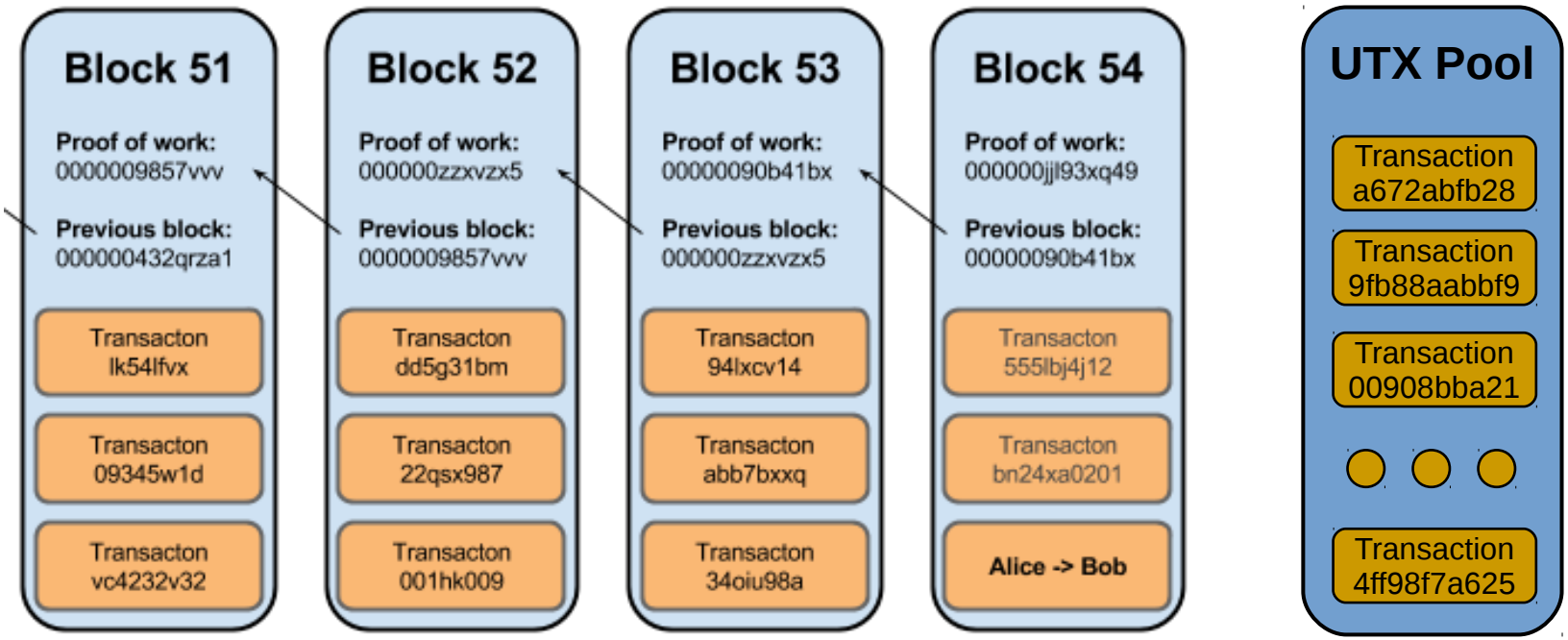
История транзакций Биткоин



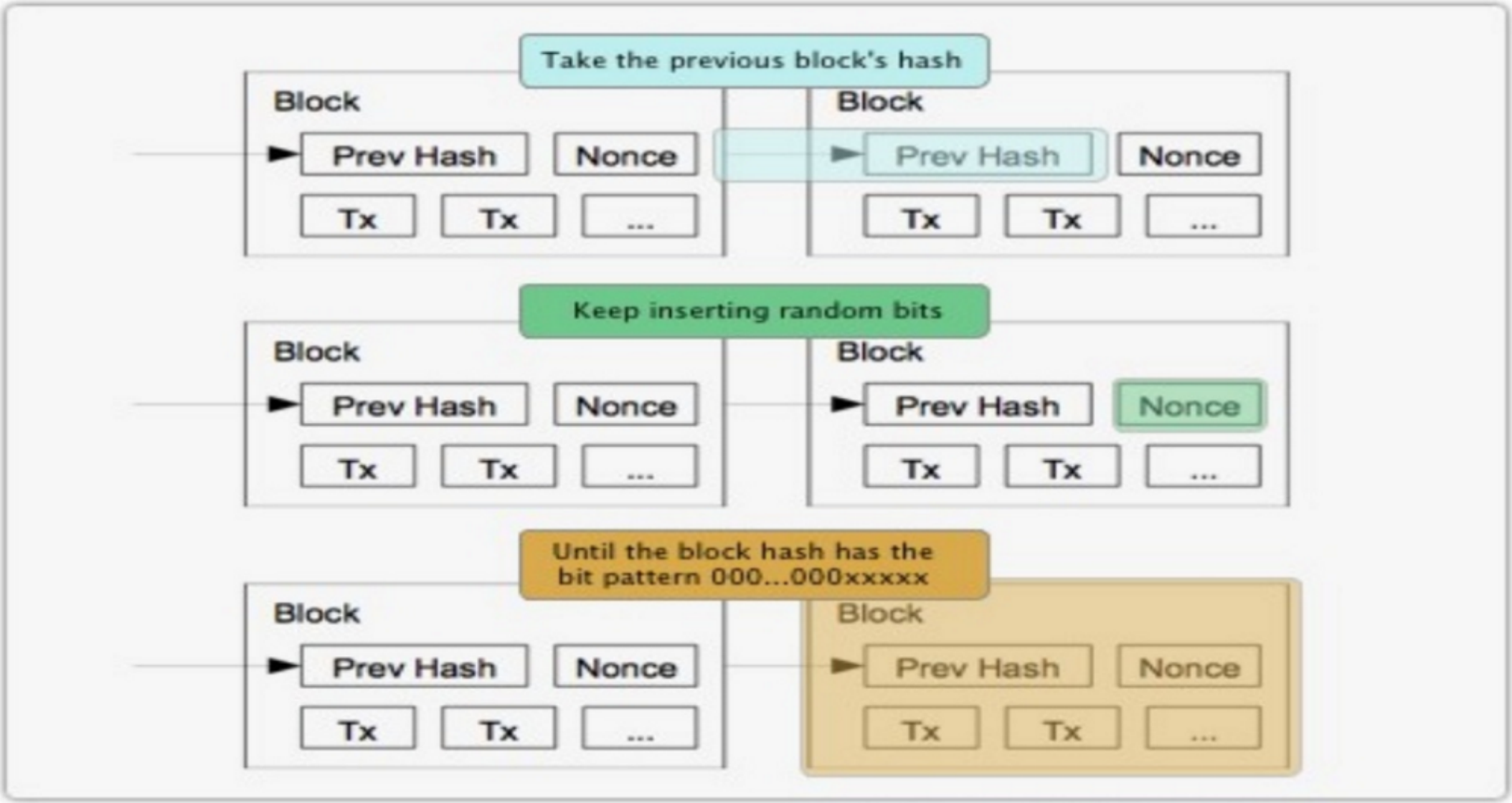
- Не потратить еще не полученных монет
- Транзакции всегда различимы
- **Валидна только одна из транзакций Tx_4 и Tx_5**

Блокчейн

- Проблема упорядочения транзакций по времени решается путем их добавления в связанные в цепочку блоки

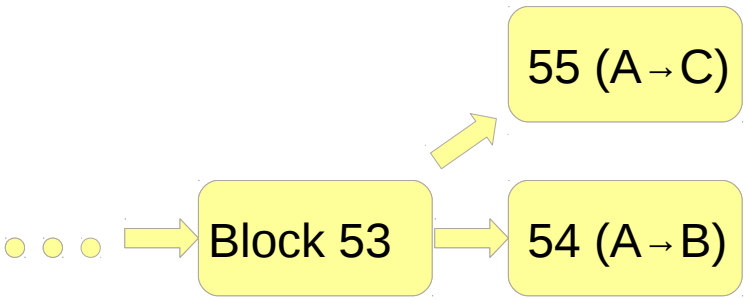


Proof of Work



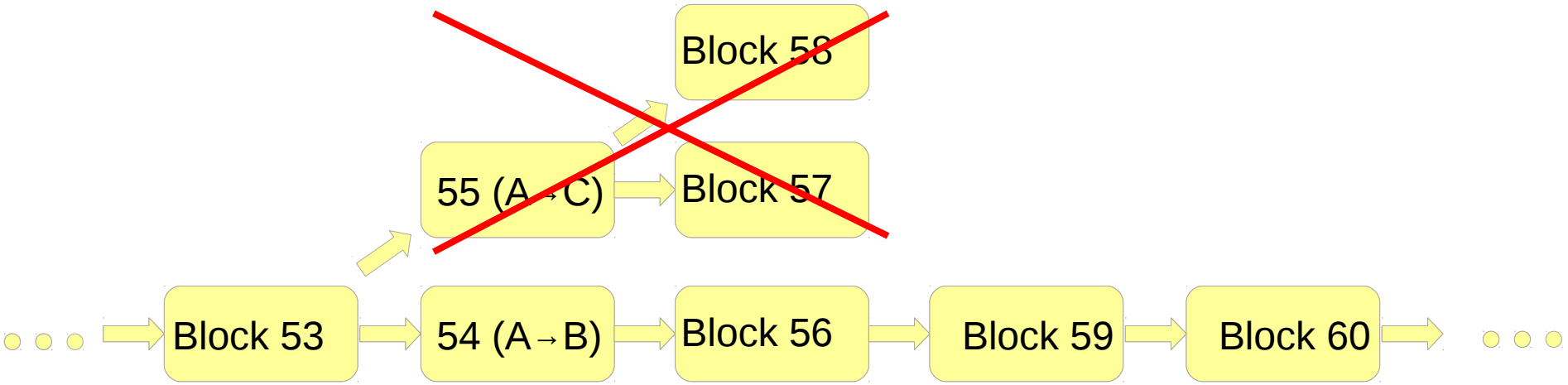
Форки

- Создани блока – случайный процесс
- Могут возникать форки



Форки

- Правильная цепочка та, в которую вложено больше всего работы
- При постоянной сложности это самая длинная цепочка
- Вероятность форков экспоненциально убывает с длиной

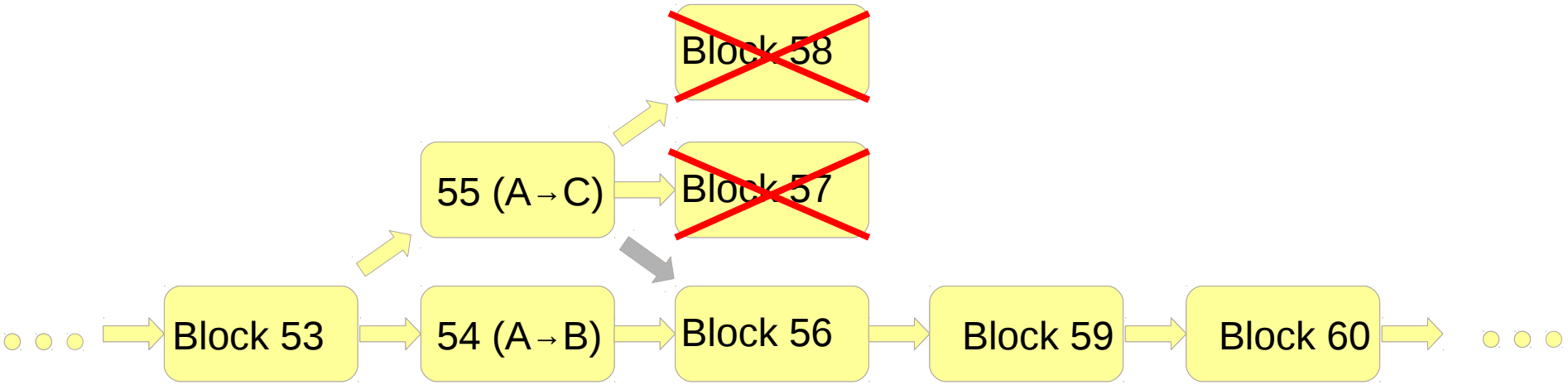


Форки в сети Биткоин

- В биткоине ~ 2% блоков в форках
- Известны случаи форков в 6 и 3 блоков из-за ошибок в настройке софта
- Реально 1, очень редко 2 блока
- Нужно ждать подтверждение транзакций! Чем больше сумма – тем дольше
- За блок майнеры получают награду ~ $35000

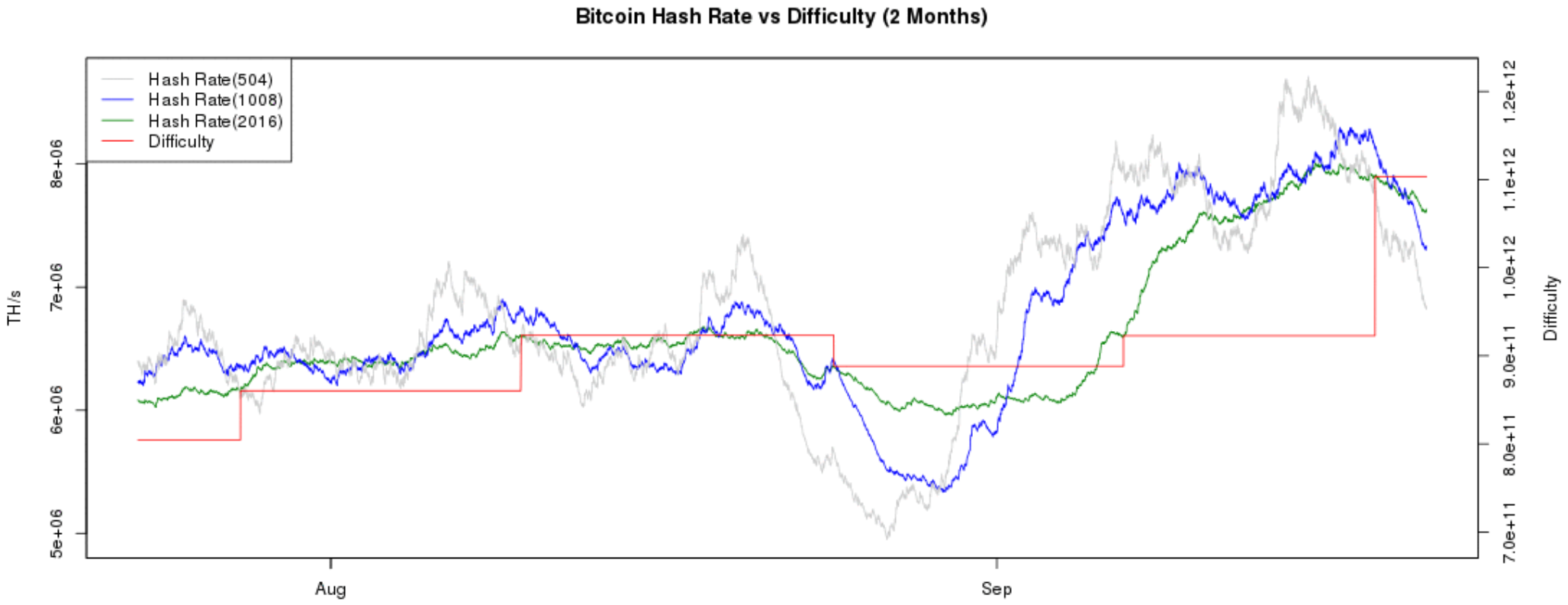
Ghost

- Возможно добавить ссылки не только на родителей



Bitcoin difficulty recalculation

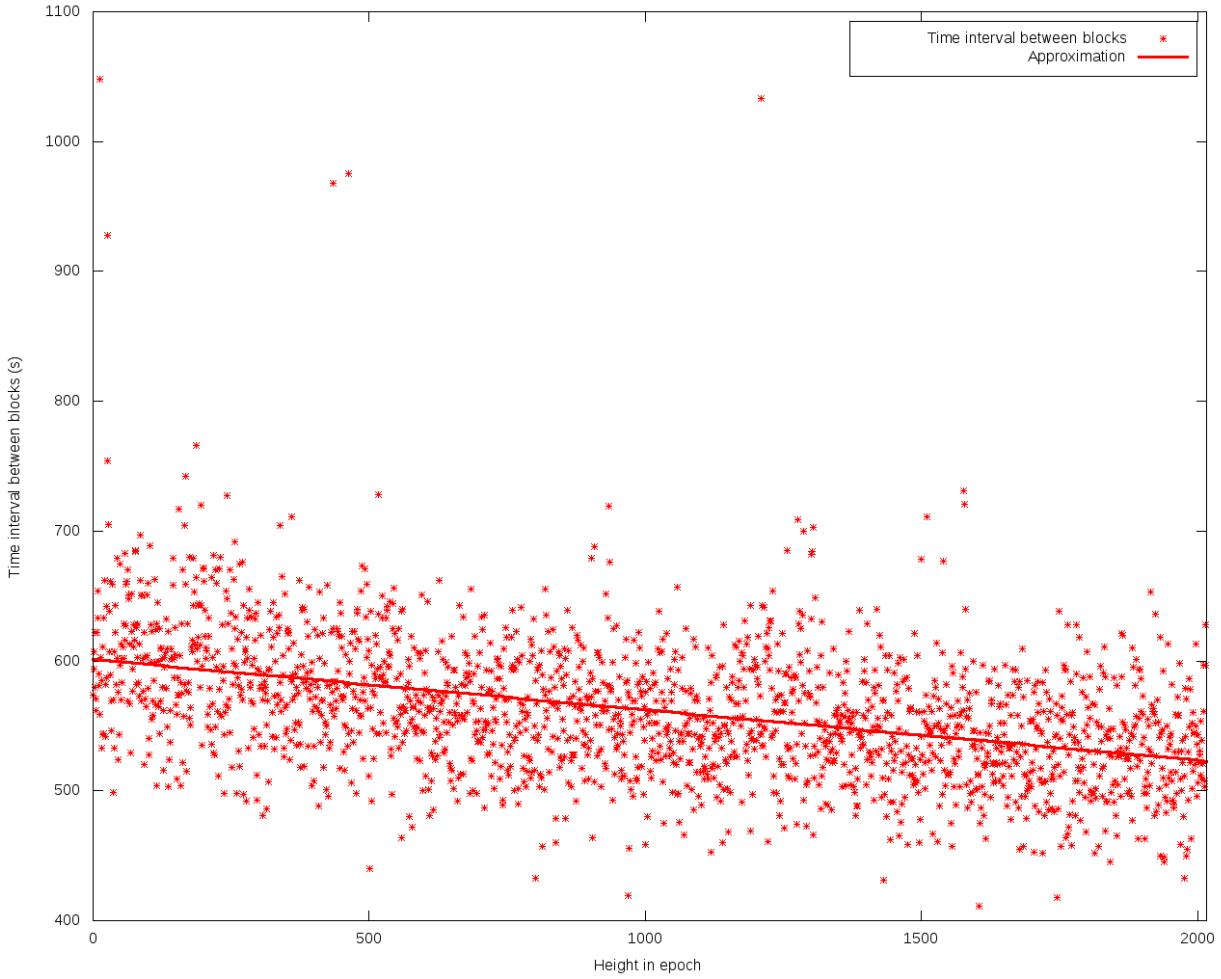
- Пересчитывается каждые 2016 блоков
- Предполагает статический хэшрейт



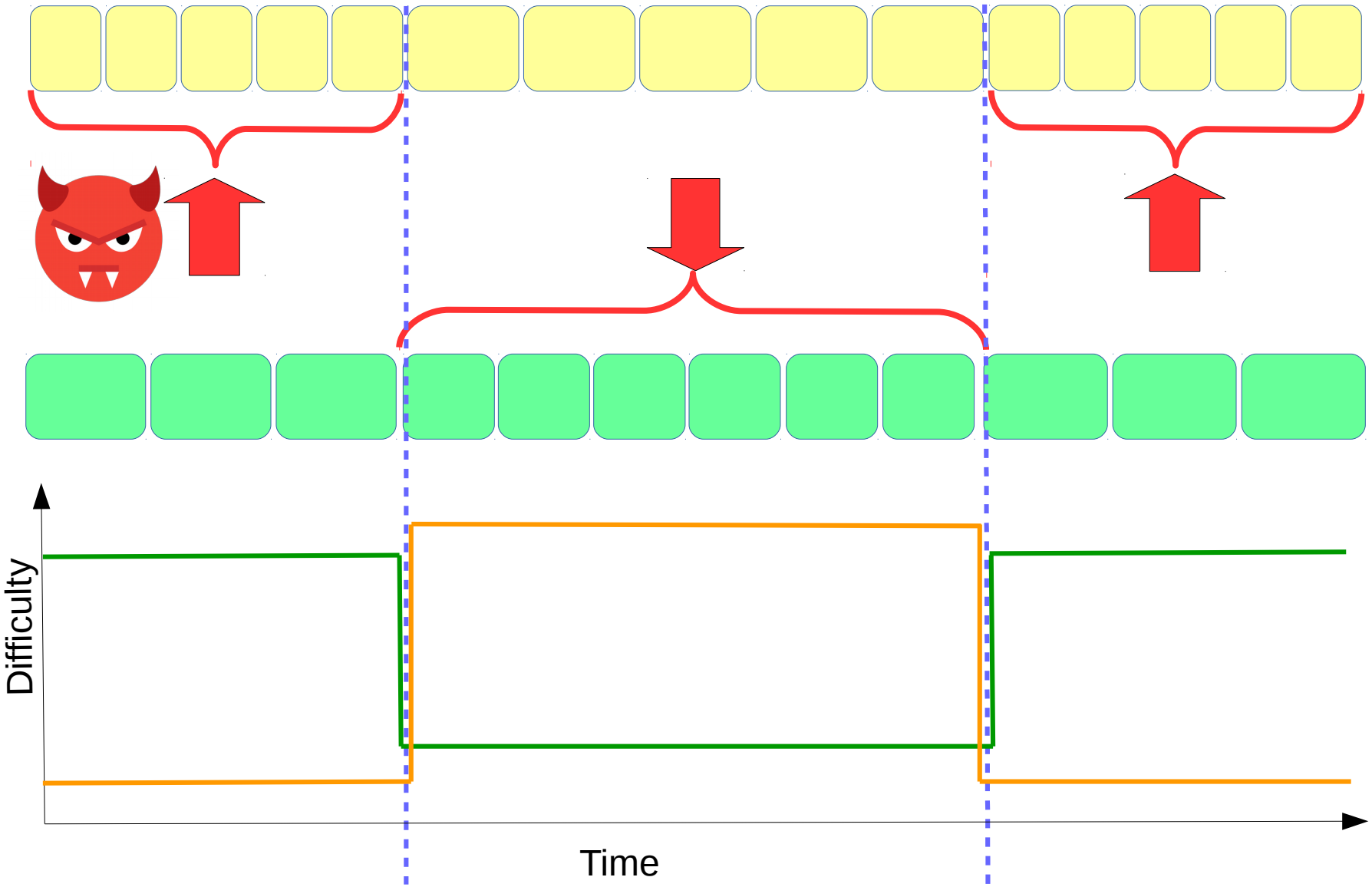
Данные с https://bitcoinwisdom.com/bitcoin/difficulty

Bitcoin difficulty recalculation

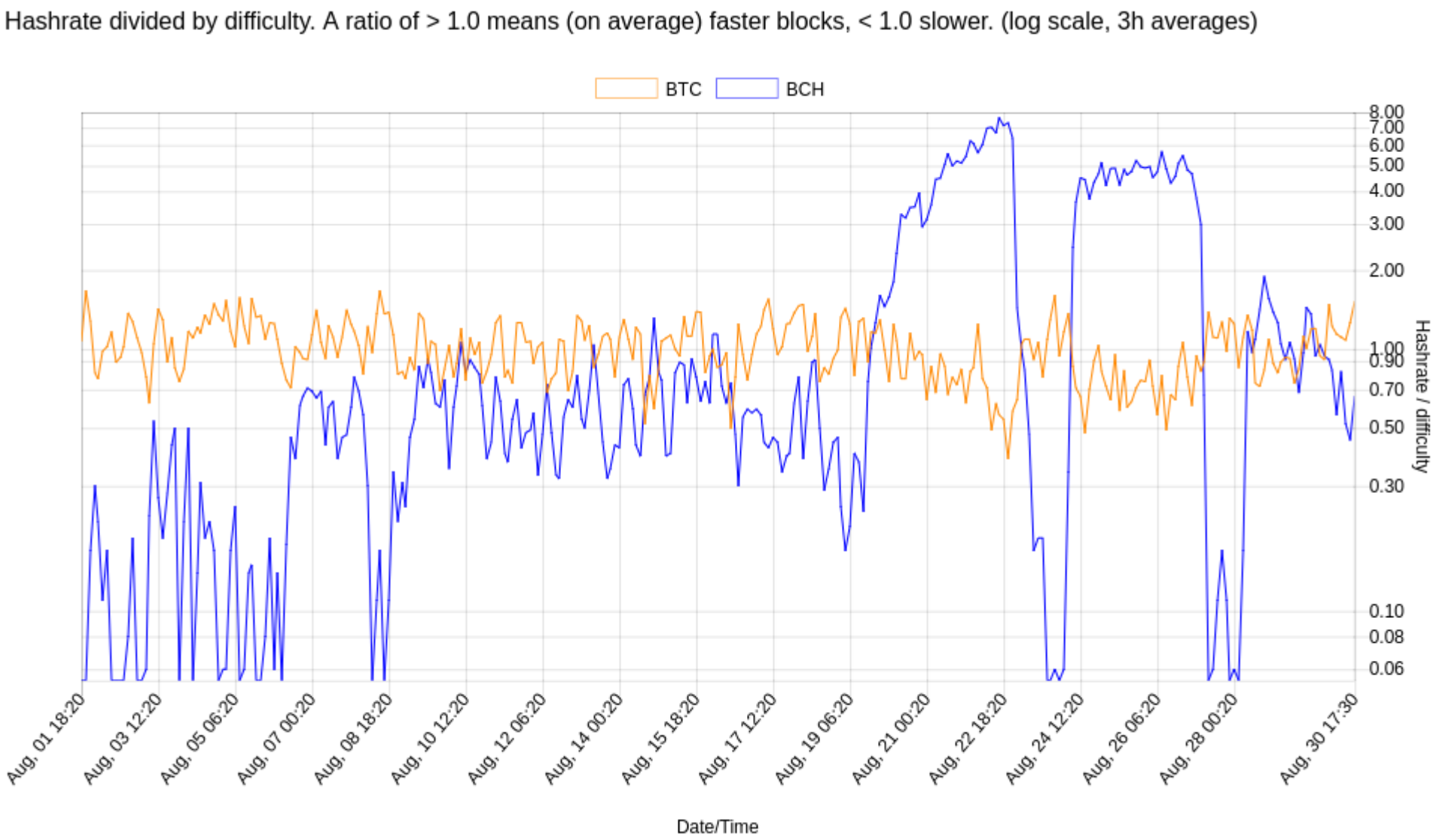
- Среднее время между блоками 9м 20с<10м
- Время в конце эпохи меньше 9 минут



Coin-hopping attack



Example: BTC/BCH fork



Graph from http://fork.lol/pow/speed

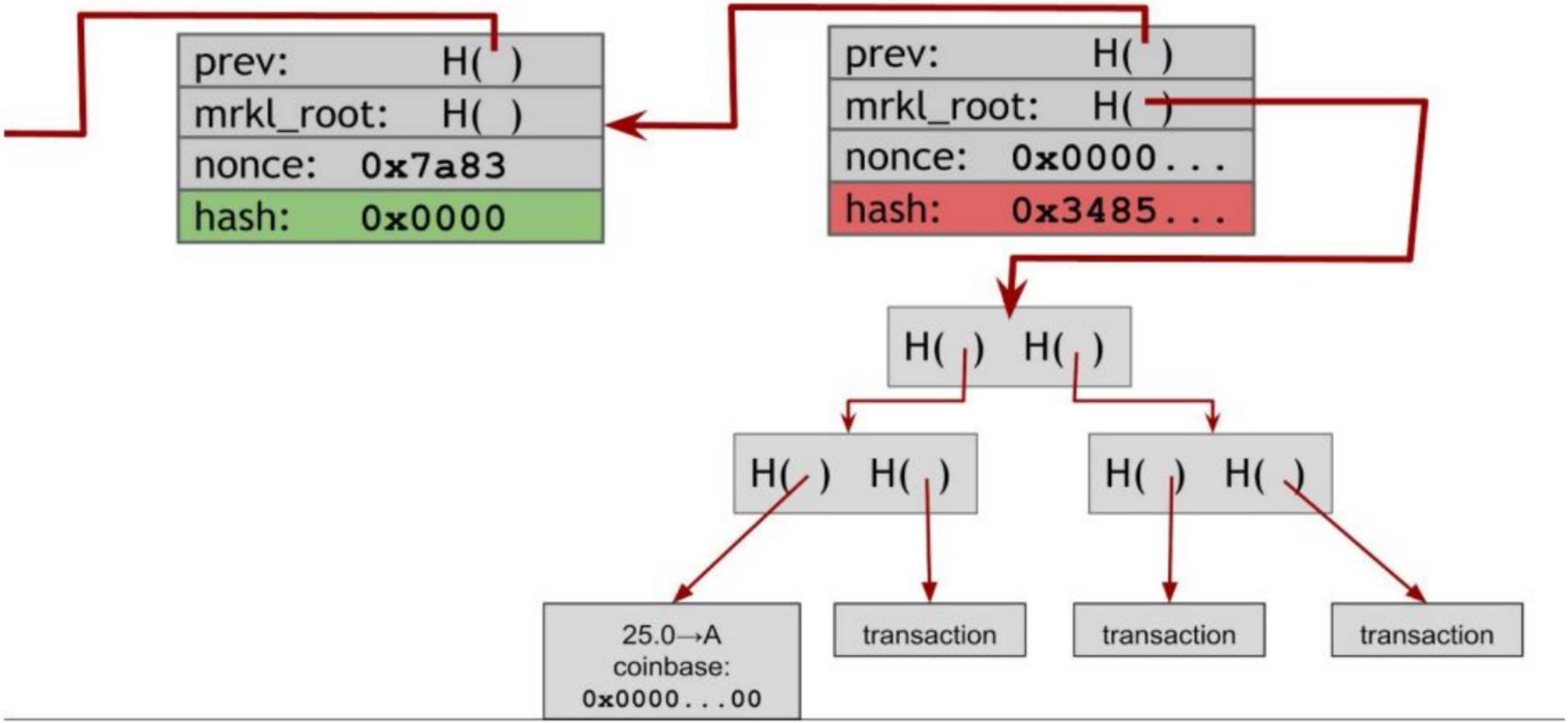
CAP теорема

В любой реализации распределённых вычислений возможно обеспечить не более двух из трёх следующих свойств:

- Согласованность данных (англ. consistency) — во всех вычислительных узлах в один момент времени данные не противоречат друг другу
- Доступность (англ. availability) — любой запрос к распределённой системе завершается корректным откликом, однако без гарантии, что ответы всех узлов системы совпадают
- Устойчивость к разделению (англ. partition tolerance) — расщепление распределённой системы на несколько изолированных секций не приводит к некорректности отклика от каждой из секций

- В блокчейне – консистентность в конечном счете

Структура блока



bitcoinwisdom.com

Структура блока

80 байт заголовка блока это

- 4 байта: версия
- 32 байта: хэш предыдушего блока
- 32 байта: рутхэш транзакций
- 4 байта: время
- 4 байта: сложность
- 4 байта: нонс

An example header in hex:

```
02000000 ............................ Block version: 2

b6ff0b1b1680a2862a30ca44d346d9e8
910d334beb48ca0c0000000000000000 ... Hash of previous block's header
9d10aa52ee949386ca9385695f04ede2
70dda20810decd12bc9b048aaab31471 ... Merkle root

24d95a54 ............................ Unix time: 1415239972
30c31b18 ............................ Target: 0x1bc330 * 256**(0x18-3)
fe9f0864 ............................ Nonce
```

Легкий клиент

- Качает только заголовки блоков
- При получении платежа получает транзакцию и доказательство ее пренадлежности к какому-то блоку

- Возможно даже качать лишь часть заголовков
- Возможно валидировать транзакции по стейту

Практика

Практика 1

Тестнет сети Биткон https://en.bitcoin.it/wiki/Testnet

- Установить кошелек https://copay.io/
- Создать кошелек для тестнета Биткоин
- Получить немного монет https://testnet.manu.backend.hamburg/faucet
- Сделать перевод соседу
- Найти транзакцию в эксплорере https://testnet.blockexplorer.com/
- Посмотреть транзакцию через API: https://api.blockcypher.com/v1/btc/test3/txs/<hash>2

Практика 2

- Создать кошелек с мультиподписью
- Отправить на него монет
- Отправить с него монет
- Посмотреть через эксплорер и API

Практика 2.5

Транзакция в тестнете вейвс

- Кошелек https://testnet.waveswallet.io/
- Фосет и эксплорер http://testnet.wavesexplorer.com/faucet
- API http://52.30.47.67:6869/transactions/info/<id>

Практика 3

- Подписать коммит на github
- https://github.com/blog/2144-gpg-signature-verification

Практика 4

- Создать подписанное и зашифрованное PGP письмо мне на почту dmitry.meshkov@iohk.io

- Create a PGP key associated with your email. Upload your PGP key to a PGP key server, like pgp.mit.edu. All email messages (internal and external) should be signed. All internal email messages should be signed and encrypted.
- Recommended PGP (Encryption and Signing) Tools
- OS X: https://ssd.eff.org/en/module/how-use-pgp-mac-os-x
- Linux: https://ssd.eff.org/en/module/how-use-pgp-linux
- Windows: https://ssd.eff.org/en/module/how-use-pgp-windows-pc
- iOS: https://itunes.apple.com/app/ipgmail/id430780873?mt=8
- Android: https://play.google.com/store/apps/details?id=org.sufficientlysecure.keychain
- Recommended Email Clients
- OS X: Mail (Pre-installed)
- Linux: Thunderbird (https://www.mozilla.org/en-US/thunderbird/)
- Windows: Thunderbird (https://www.mozilla.org/en-US/thunderbird/)
- iOS: Mail (Pre-installed)