PV204 Security technologies

Bitcoin II.

Petr Švenda ✉ ***svenda@fi.muni.cz*** ***@rngsec***
Centre for Research on Cryptography and Security, Masaryk University

Please provide any corrections and comments here (thank you!):
https://drive.google.com/file/d/15z8k8zltcBaxEcF18DGwdoTtUNQFd-9c/view?usp=sharing

CROCS
Centre for Research on
Cryptography and Security

Phase III

- Finalize the implementation
- Prepare and record a presentation of your project (10 minutes)
  - Project design
  - Implementation
  - Issues and solutions
  - Application demonstration
- Deadline: 16. 4. 2023
  - Submit presentation slides and the recording to IS
  - Submission from this phase will be made available to reviewing teams

Task: Questions to ask

- Write 2-3 questions you want to discuss about Bitcoin
- https://sli.do #pv204_2023
- We will cover it together towards second half of this lecture
  - (and possibly during seminar)

slido

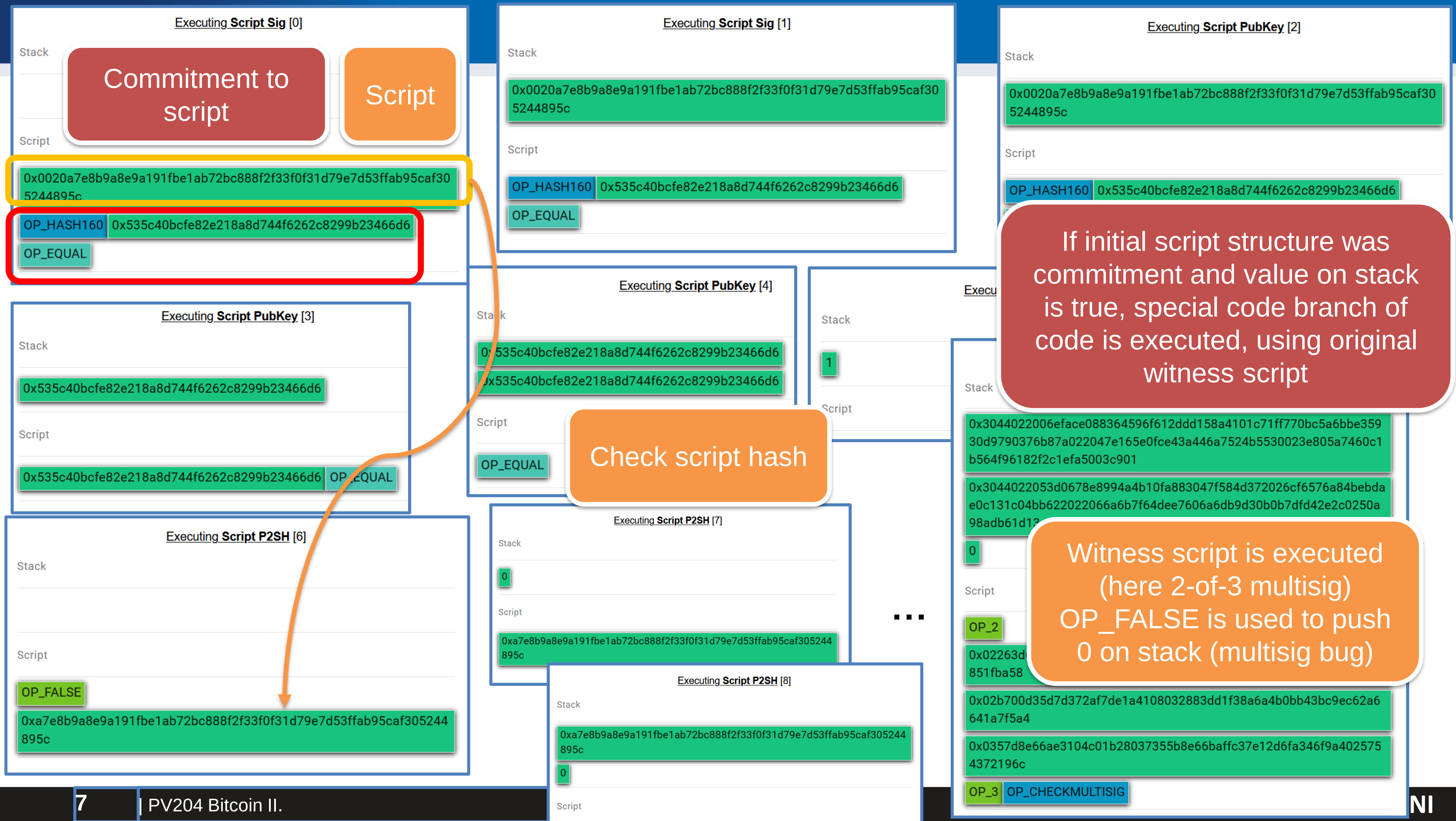
Audience Q&A Session

ⓘ Start presenting to display the audience questions on this slide.

LEFTOVER FROM PREVIOUS LECTURE ☺

Pay to script hash (P2SH), BIP16, starts with '3'

- Lock script separated into two parts
  - 1) commitment to the script (hash value, checked later)
  - 2) actual lock script (hash value must match the commitment)
- Sending tx sets output's ScriptPub to the commitment
  - Shorter as only hash is posted, not whole lock script
  - Lock script is provided only later when spending (privacy, fee to be paid)
  - Lock script can have multiple spending paths (Merkle tree) and only the one used is posted (better for privacy)
- Redeeming tx provides actual lock script + unlock script



Executing Script Sig [0]

Stack

Script

0x0020a7e8b9a8e9a191fbe1ab72bc888f2f33f0f31d79e7d53ffab95caf305244895c

OP_HASH160 0x535c40bcfe82e218a8d744f6262c8299b23466d6

OP_EQUAL

Commitment to script

Script

Executing Script Sig [1]

Stack

0x0020a7e8b9a8e9a191fbe1ab72bc888f2f33f0f31d79e7d53ffab95caf305244895c

Script

OP_HASH160 0x535c40bcfe82e218a8d744f6262c8299b23466d6

OP_EQUAL

Executing Script PubKey [2]

Stack

0x0020a7e8b9a8e9a191fbe1ab72bc888f2f33f0f31d79e7d53ffab95caf305244895c

Script

OP_HASH160 0x535c40bcfe82e218a8d744f6262c8299b23466d6

If initial script structure was commitment and value on stack is true, special code branch of code is executed, using original witness script

Executing Script PubKey [3]

Stack

0x535c40bcfe82e218a8d744f6262c8299b23466d6

Script

0x535c40bcfe82e218a8d744f6262c8299b23466d6 OP_EQUAL

Executing Script PubKey [4]

Stack

0x535c40bcfe82e218a8d744f6262c8299b23466d6

0x535c40bcfe82e218a8d744f6262c8299b23466d6

Script

OP_EQUAL

Check script hash

Execu

Stack

1

Script

Executing Script P2SH [6]

Stack

Script

OP_FALSE

0xa7e8b9a8e9a191fbe1ab72bc888f2f33f0f31d79e7d53ffab95caf305244895c

Executing Script P2SH [7]

Stack

0

Script

0xa7e8b9a8e9a191fbe1ab72bc888f2f33f0f31d79e7d53ffab95caf305244895c

Executing Script P2SH [8]

Stack

0xa7e8b9a8e9a191fbe1ab72bc888f2f33f0f31d79e7d53ffab95caf305244895c

0

Script

. . .

Stack

0x3044022006eface088364596f612ddd158a4101c71ff770bc5a6bbe35930d9790376b87a022047e165e0fce43a446a7524b5530023e805a7460c1b564f96182f2c1efa5003c901

0x3044022053d0678e8994a4b10fa883047f584d372026cf6576a84bebdae0c131c04bb622022066a6b7f64dee7606a6db9d30b0b7dfd42e2c0250a98adb61d13

0

Script

OP_2

0x02263d
851fba58

0x02b700d35d7d372af7de1a4108032883dd1f38a6a4b0bb43bc9ec62a6641a7f5a4

0x0357d8e66ae3104c01b28037355b8e66baffc37e12d6fa346f9a4025754372196c

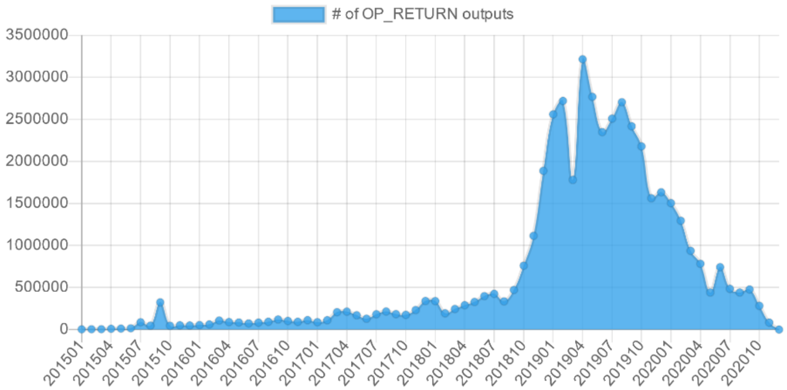
OP_3 OP_CHECKMULTISIG

Witness script is executed (here 2-of-3 multisig) OP_FALSE is used to push 0 on stack (multisig bug)

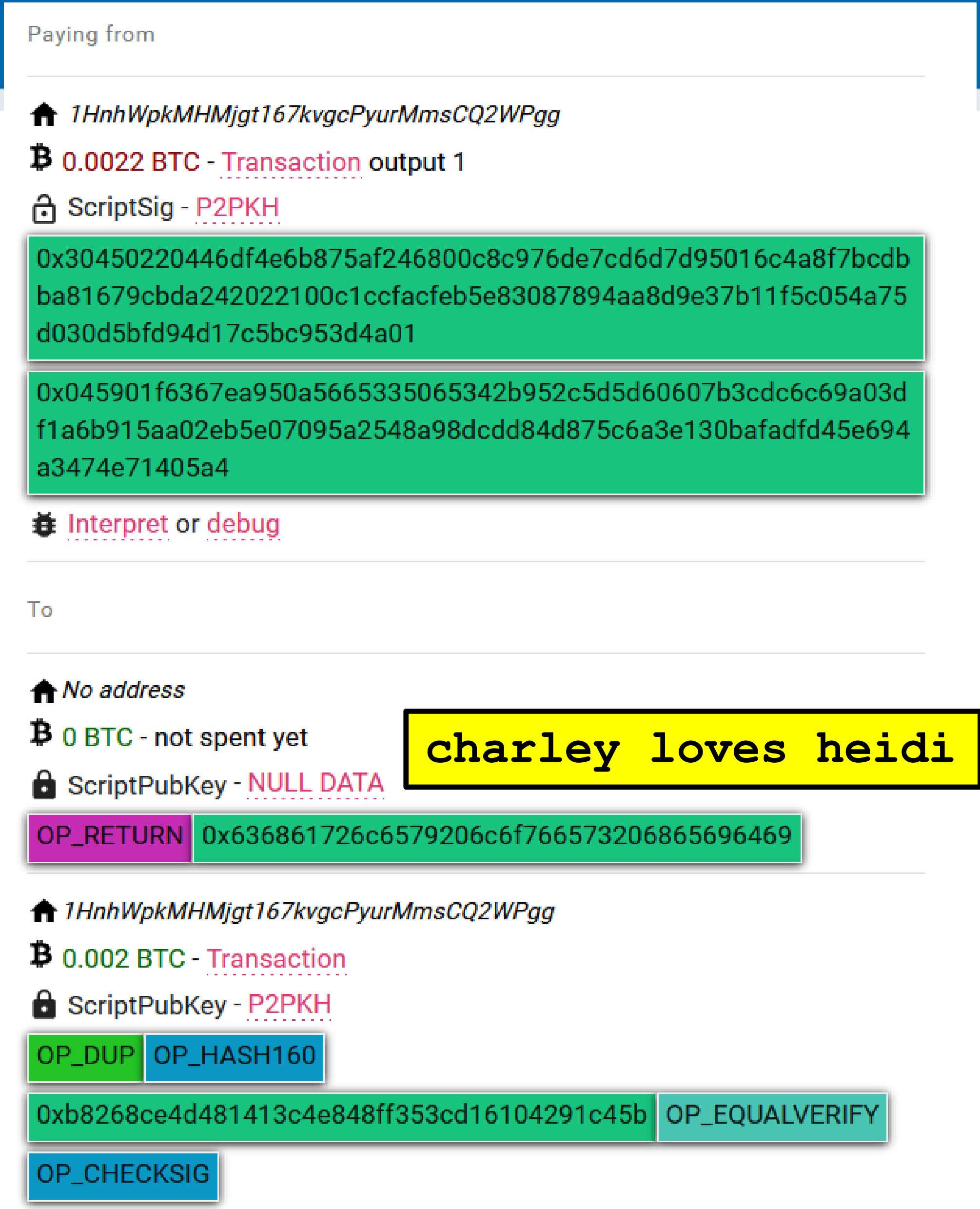
Interesting, non-standard scripts

- SHA1 collision bounty
  - Bitcoins locked to script requiring two different inputs hashed to same SHA1 hash
  - Redeemed shortly after Google published SHA1 collision blocks
    - https://blockstream.info/tx/8d31992805518fd62daa3bdd2a5c4fd2cd3054c9b3dca1d78055e9528cff6adc
    - https://nioctib.tech/#/transaction/8d31992805518fd62daa3bdd2a5c4fd2cd3054c9b3dca1d78055e9528cff6adc
    - More details: https://bitcoinjs-guide.bitcoin-studio.com/bitcoinjs-guide/v5/part-three-pay-to-script-hash/puzzles/computational_puzzle_sha1_collision_p2sh.html
- Similar bounties for other hash collisions

OP_RETURN



- If OP_RETURN is encountered during execution of unlock+lock script, it is FALSE
  - Such output is provably unspendable => not in UTXO set
- Somewhat controversial instruction
  - Some feels, that blockchain shall not be used for non-financial data (USDT was initially on Bitcoin via OP_RETURN)
  - But there were already ways how to store arbitrary data into blockchain anyway (e.g., bytes of value, invalid address)
- Analysis of OP_RETURN data
  - https://www.blockchainresearchlab.org/2020/03/13/how-do-op-return-transactions-impact-bitcoin/
  - https://opreturn.org/
- Relevant recent discussion with Inscriptions (later)



Paying from

1HnhWpkMHMjgt167kvgcPyurMmsCQ2WPgg

0.0022 BTC - Transaction output 1

ScriptSig - P2PKH

0x30450220446df4e6b875af246800c8c976de7cd6d7d95016c4a8f7bcdbba81679cbda242022100c1ccfacfeb5e83087894aa8d9e37b11f5c054a75d030d5bfd94d17c5bc953d4a01

0x045901f6367ea950a5665335065342b952c5d5d60607b3cdc6c69a03df1a6b915aa02eb5e07095a2548a98dcdd84d875c6a3e130bafadfd45e694a3474e71405a4

Interpret or debug

To

No address

0 BTC - not spent yet

ScriptPubKey - NULL DATA

OP_RETURN 0x636861726c6579206c6f766573206865696469

charley loves heidi

1HnhWpkMHMjgt167kvgcPyurMmsCQ2WPgg

0.002 BTC - Transaction

ScriptPubKey - P2PKH

OP_DUP OP_HASH160

0xb8268ce4d481413c4e848ff353cd16104291c45b OP_EQUALVERIFY

OP_CHECKSIG

https://nioctib.tech/#/transaction/f2f398dace996dab12e0cfb02fb0b59de0ef0398be393d90ebc8ab397550370b

BLOCKS AND MINING

Problem: Who will include next block in blockchain?

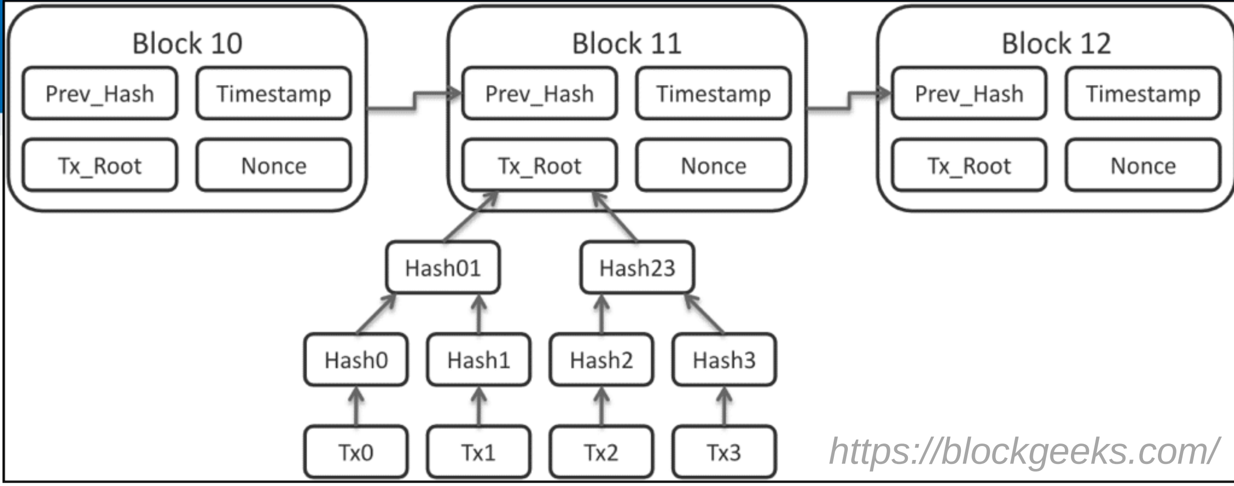
- Transactions (state updates) has to be included somehow into block to be "permanently" valid
- Entity including new block has special position and power
  - Can decide which transactions (state updates) will be included
    - May lead to censorship of certain transactions
    - May lead to transactions reordering impacting the financial value (e.g. MEV)
  - Can decide where new block is appended
    - Shall be last previous block, but can cause malicious forks abandoning part of previously extended blockchain (e.g., 51% attack to rewrite history)
  - Typically receive some reward (motivation for participation)
    - May cause long-term centralized accumulation of underlying token

Who can include next block to blockchain?

We will focus mainly on Proof Of Work used in Bitcoin

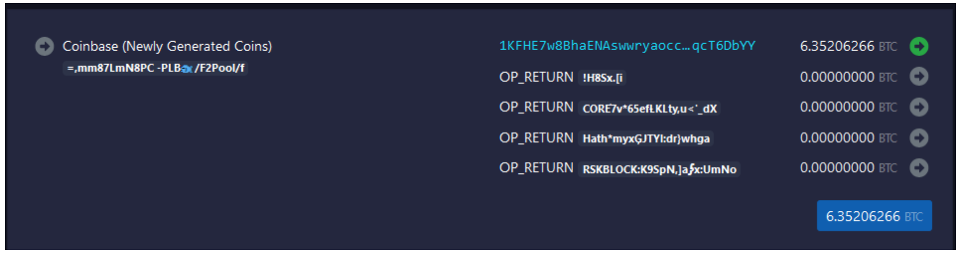
- Proof of Work (PoW, Bitcoin, Ethereum 1.0, Zcash…)
  - Solver of computationally hard puzzle can include new block
- Proof of Stake (PoS, Zcoin, Cardano, BNB, Ethereum 2.0…)
  - More coins you own, higher the probability you will be selected to include next block
  - Various variants, Stake pools…
- Merged Mining (Namecoin…)
  - Hash of block from the chain is included in coinbase tx of other chain (typically Bitcoin)
  - The chain is not performing own mining, Bitcoin miners are getting reward for inclusion of other chains
- Proof of Proof (PoP)
  - Hash of block from other chain is included in Bitcoin transaction (typically OP_RETURN)
  - Security of other chain is improved by security of Bitcoin blockchain
- Proof of Authority (PoA)
  - Small number of trusted actors create new blocks

Bitcoin block



https://blockgeeks.com/

- Header (80 B) + data (up to ~4MB)
  - Version
  - Previous block hash (linking to past blockchain)
  - Merkle root of all included transactions (Coinbase tx + others)
  - Timestamp (unix time)
  - Bits (specification of required mining difficulty)
  - Nonce (variable part for mining, now insufficient)



```
Coinbase (Newly Generated Coins)        1KFHE7w8BhaENAswwryaocc…qcT6DbYY       6.35206266 BTC
=,mm87LmN8PC -PLB /F2Pool/f             OP_RETURN  !H8Sx.[i                     0.00000000 BTC
                                        OP_RETURN  CORE7v*65efŁKLty,u<'_dX      0.00000000 BTC
                                        OP_RETURN  Hath*myxGJTYI:dr}whga        0.00000000 BTC
                                        OP_RETURN  RSKBLOCK:K9SpN,]aƒx:UmNo     0.00000000 BTC
                                                                                6.35206266 BTC
```

- Coinbase transaction (reward for miners, emission of new bitcoins)
  - First transaction in every block (only one)
  - Only one input, previous TX ID = 0x0000..00, prev. TX index = 0xffffffff
  - (Typically) equal to block reward + all fees from included transactions

Bitcoin’s Proof of Work (SHA256 function)

- Crucial for security of blockchain (no rewrite of history)
- Initially on CPU (Satoshi: “Everyone can participate 1 CPU 1 vote”`)
- CPU→GPU →FPGA →ASIC
- Initially solo mining, later collaborative mining (too little chance alone)
- First mining pool: SlushPool in Prague (now Braiins Pool)
  - Miners join their hashrate, fraction of reward based on number of partial solutions
- Cambridge university centre for alternative finance (CBECI)
  - Where are the miners? https://cbeci.org/mining_map/
  - More mining details: https://cbeci.org/cbeci/methodology

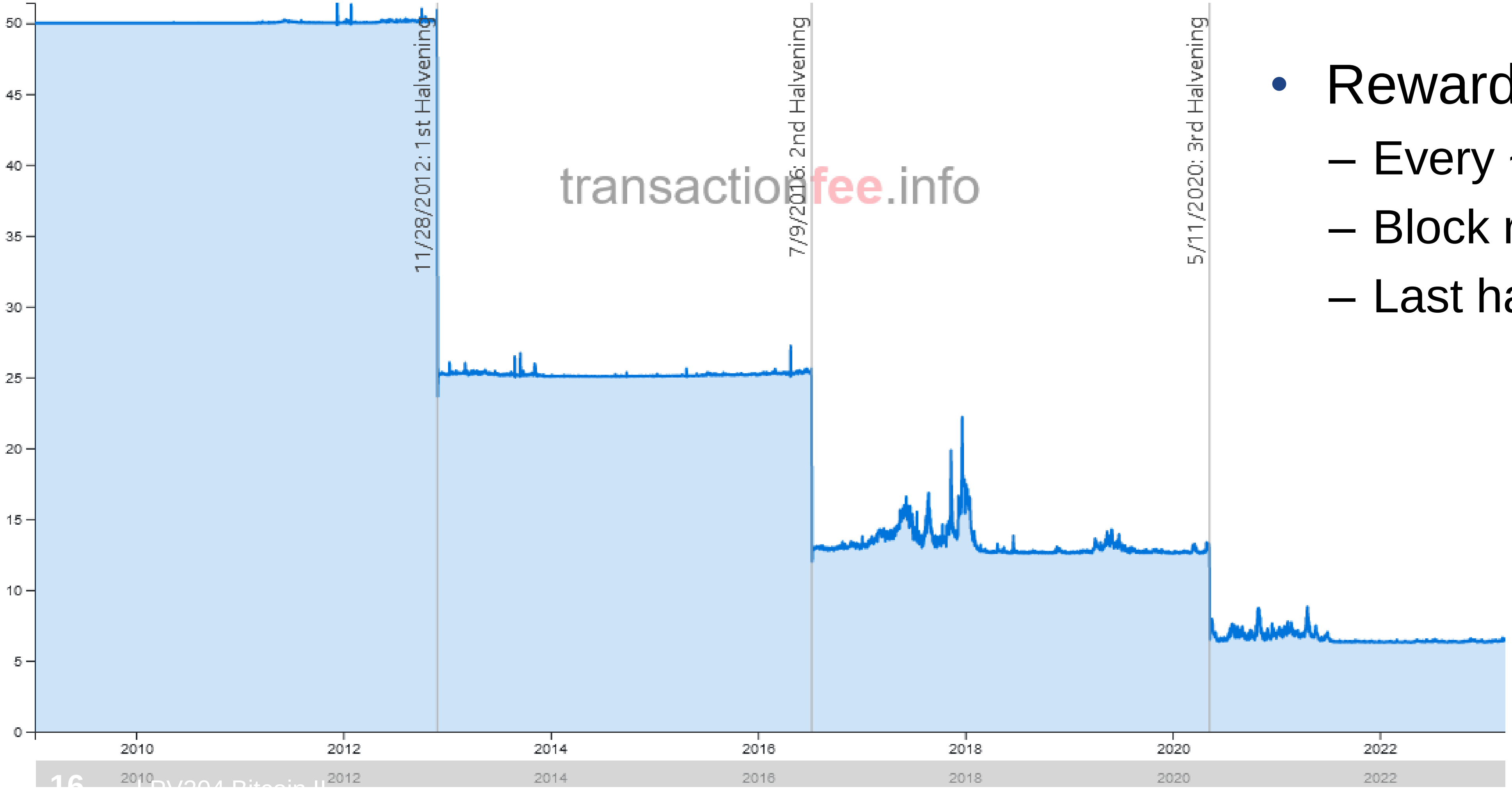
https://transactionfee.info/charts/block-coinbase-amount/?start=2009-01-09

DEMO: SHOW EVOLUTION OF REWARDS

https://transactionfee.info/charts/block-coinbase-amount/?start=2009-01-09

Miner reward – coinbase output: block + fees

Shows the average coinbase transaction output amount.



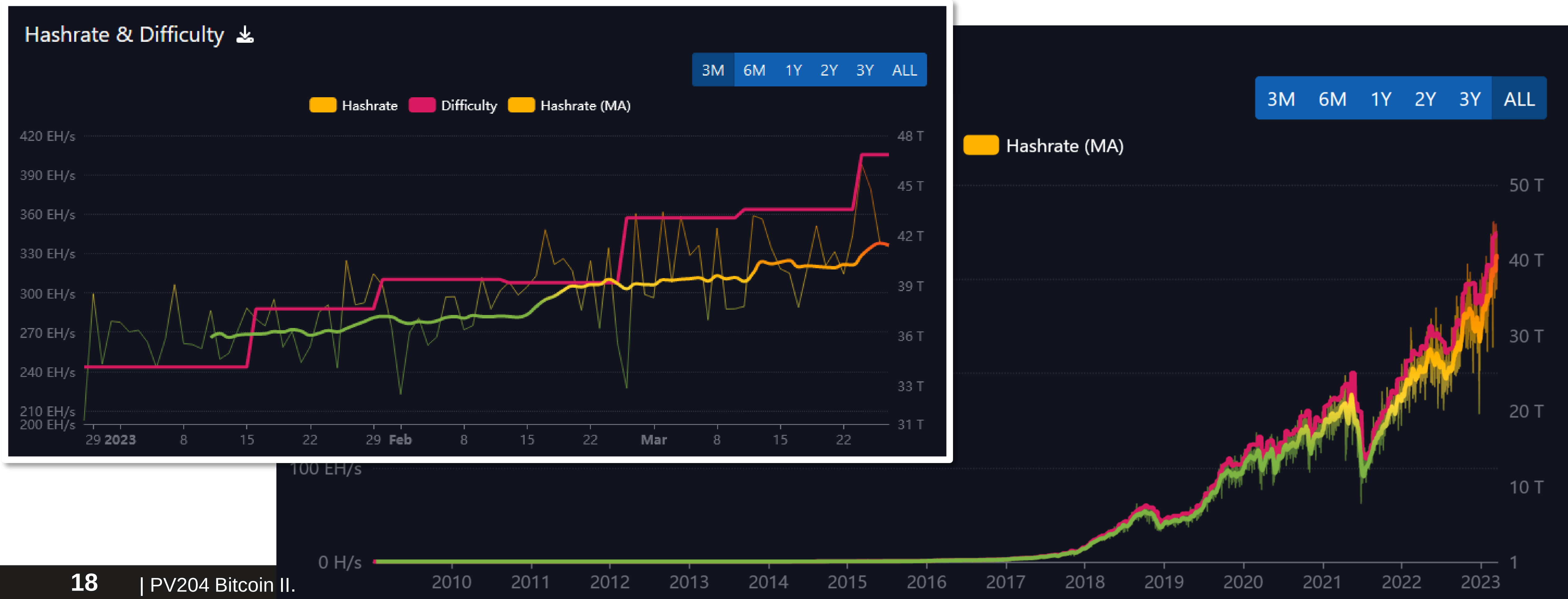
- Reward halving
  - Every ~4 years
  - Block reward drops to ½
  - Last halving in year 2140

Difficulty adjustment

- Bitcoin shall have one block every ten minutes (on average)
- Block must have overall hash with specific number of leading zeroes (March 2023 ~75 binary 0)
  - Miners change part of block header to try different hashes until required found
- How to specify the number of leading zeroes for decades in future?
  - Speed of new blocks found depends on the overall speed of hashing
  - Overall speed of hashing depends on technology advancements (single chip) and number of chips deployed
  - Impossible to predict technology and interest into distant future
  - If # zeroes is too low => blocks are found too fast (and vice versa)
- Idea of difficulty adjustment (part of consensus protocol), https://en.bitcoin.it/wiki/Difficulty
  - Check number of actually mined blocks every 2016 blocks (shall be ~14 days)
    - Increase/decrease difficulty for next period based on actual number of mined blocks
  - Every full node can deterministically compute expected difficulty (lower # zeroes rejected)
- Block hash must be below the “Target” number (computed to avg keep 1 block / ~10 min)
  - “Target” is transformed to “Bits” (condensed 4 bytes number – coefficient (3B) + exponent (1B))
  - Current difficulty is relative number of current Target with respect to Target of Genesis block

https://mempool.space/graphs/mining/hashrate-difficulty#all

Hashrate in time (>320EH/s = $3.2*10^{20}$ hash/sec = 2^{66} /sec) Peak 26.3.2023: 400,000,000,000,000,000,000x SHA256 computations per second



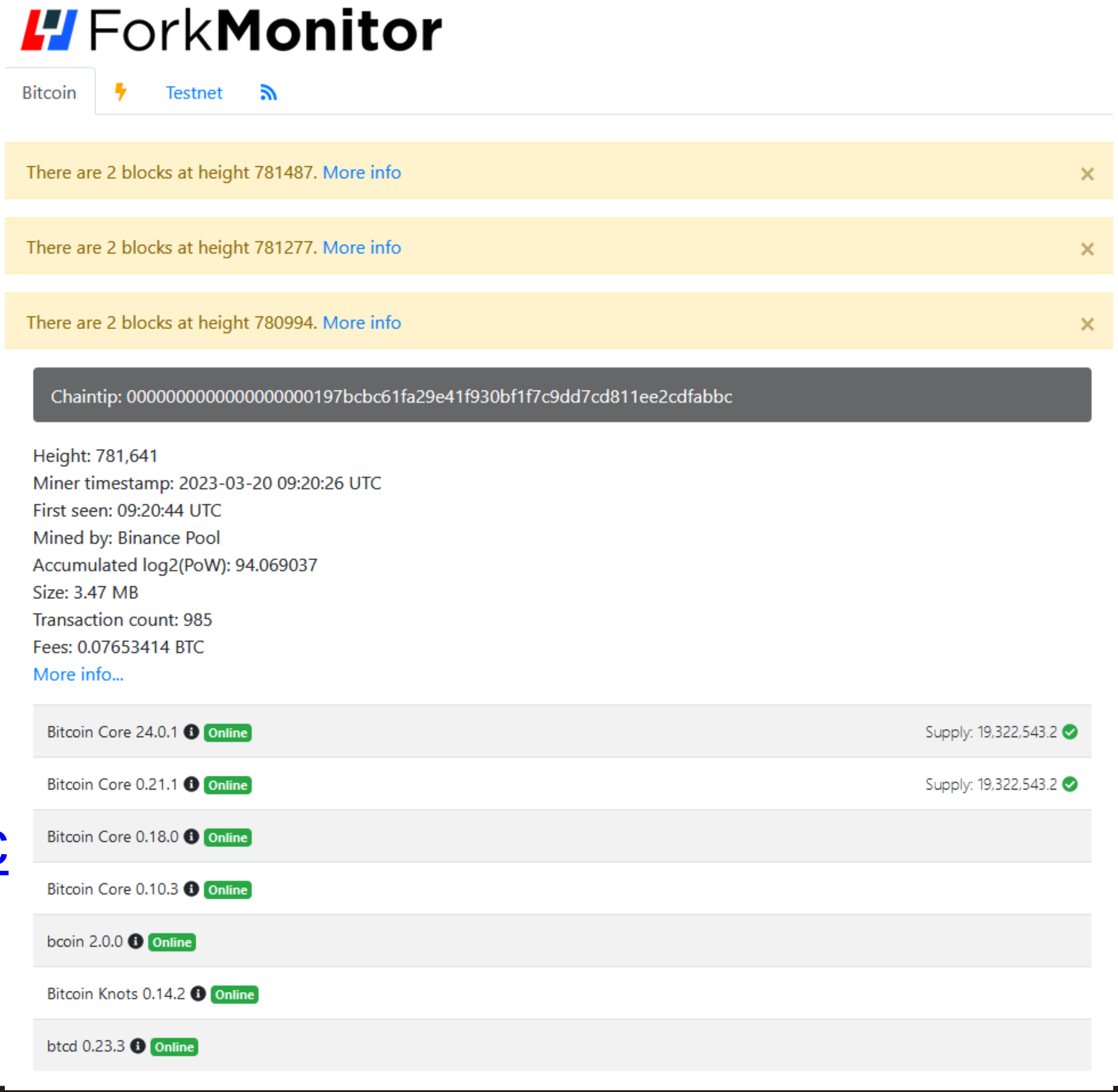
https://mempool.space/

https://mempool.space/graphs/mining/hashrate-difficulty#all

DEMO: SHOW DIFFICULTY ADJUSTMENT, HASHRATE

Blockchain forks

- Occasional natural forks happen
  - (not to be confused with softforks)
- Quickly resolved
  - usually, next block
- Sometimes temporary double-spent can occur
  - Same input used in different txs
- https://forkmonitor.info/nodes/btc



ForkMonitor

There are 2 blocks at height 781487. More info ×

There are 2 blocks at height 781277. More info ×

There are 2 blocks at height 780994. More info ×

Chaintip: 0000000000000000000197bcbc61fa29e41f930bf1f7c9dd7cd811ee2cdfabbc

Height: 781,641
Miner timestamp: 2023-03-20 09:20:26 UTC
First seen: 09:20:44 UTC
Mined by: Binance Pool
Accumulated log2(PoW): 94.069037
Size: 3.47 MB
Transaction count: 985
Fees: 0.07653414 BTC
More info...

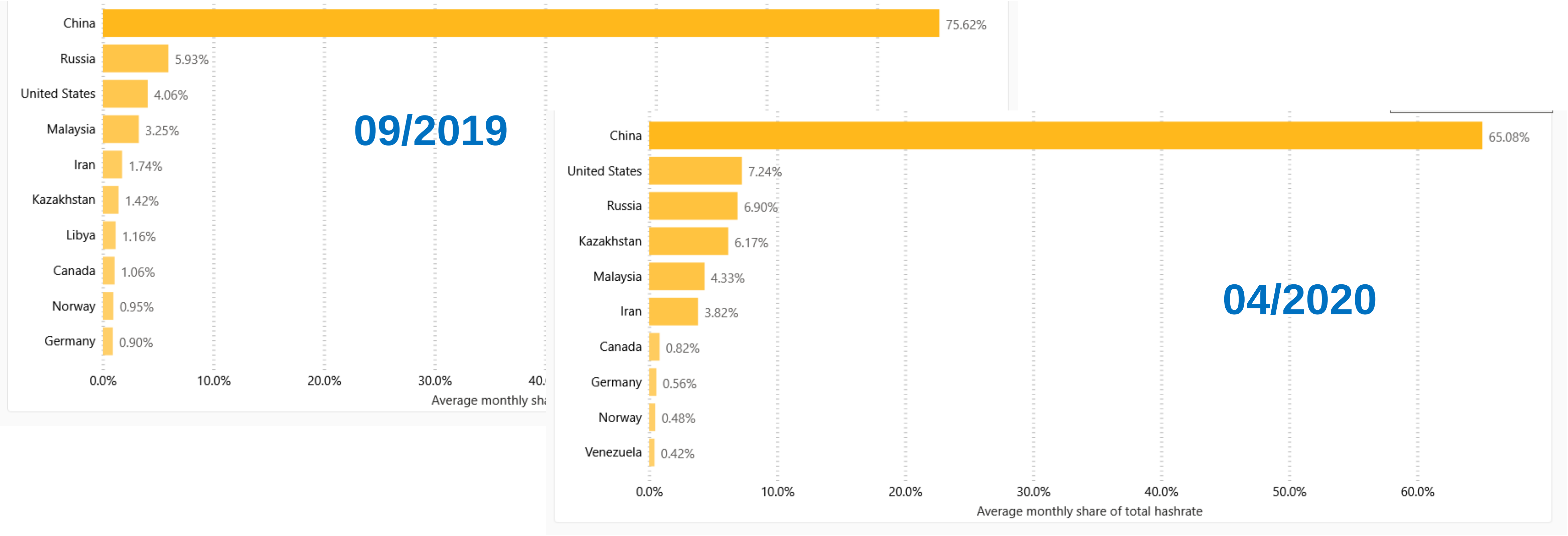
Bitcoin Core 24.0.1 Online	Supply: 19,322,543.2 ✓
Bitcoin Core 0.21.1 Online	Supply: 19,322,543.2 ✓
Bitcoin Core 0.18.0 Online	
Bitcoin Core 0.10.3 Online	
bcoin 2.0.0 Online	
Bitcoin Knots 0.14.2 Online	
btcd 0.23.3 Online	

https://forkmonitor.info/nodes/btc

Double spent tx https://forkmonitor.info/stale/btc/782129

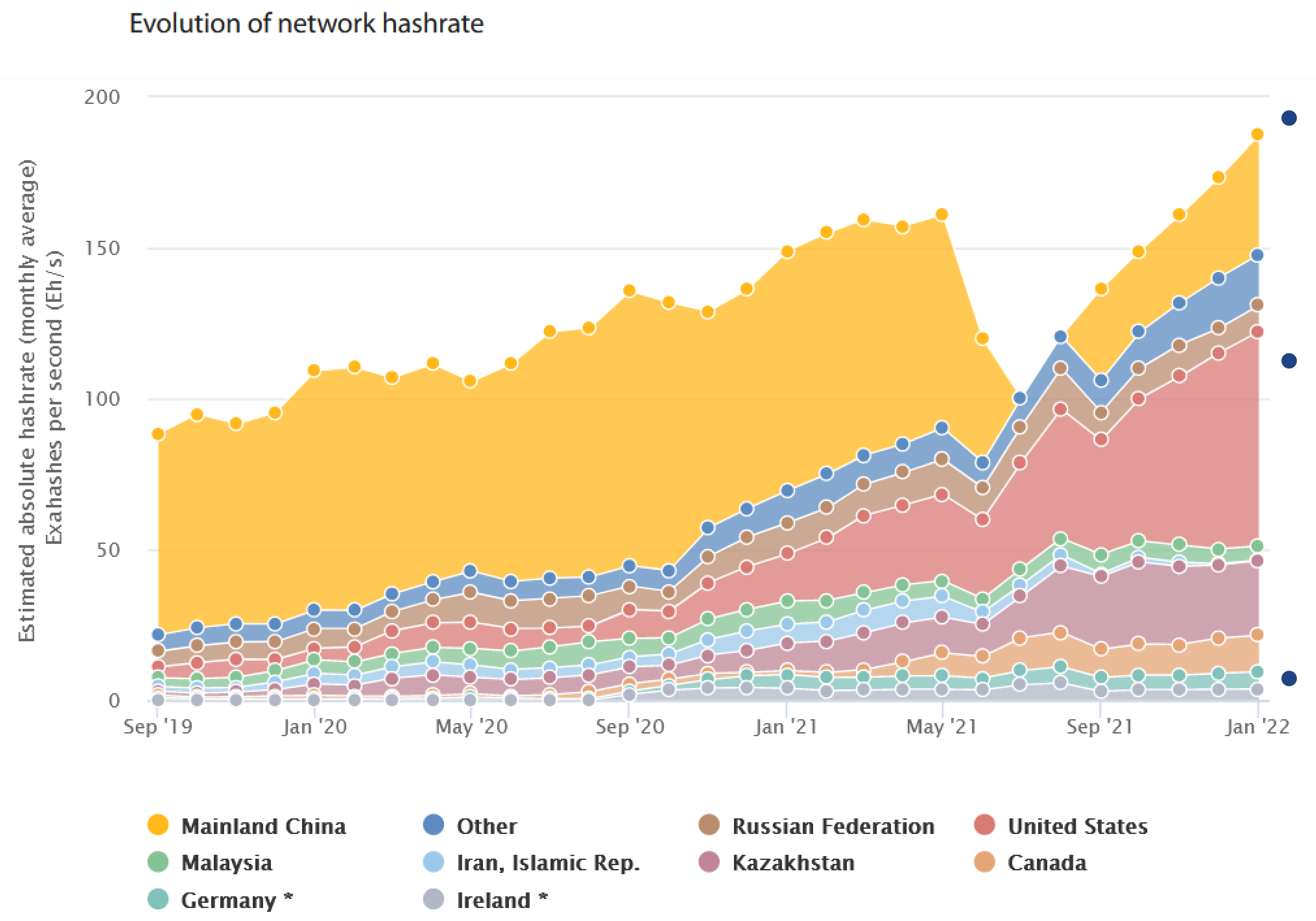
DEMO: SHOW NATURAL FORKS

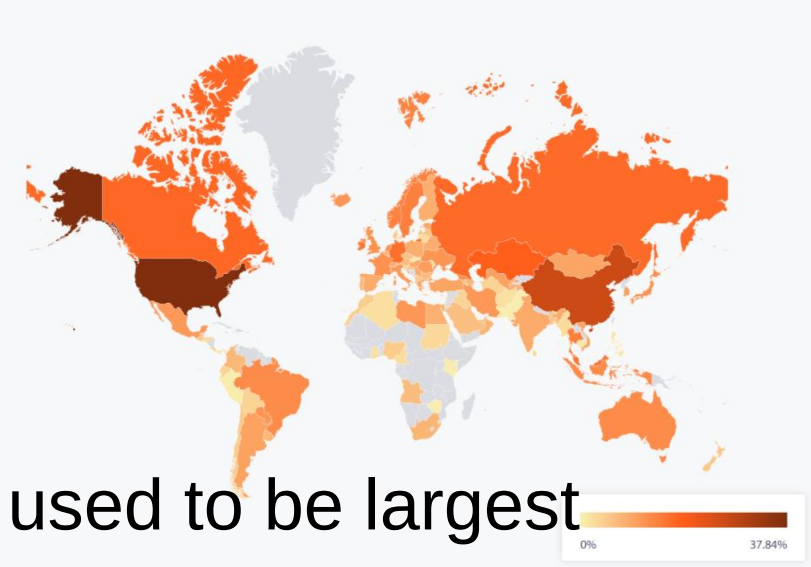
China mining dominance (09/2019→04/2020: 75.6%->65%)



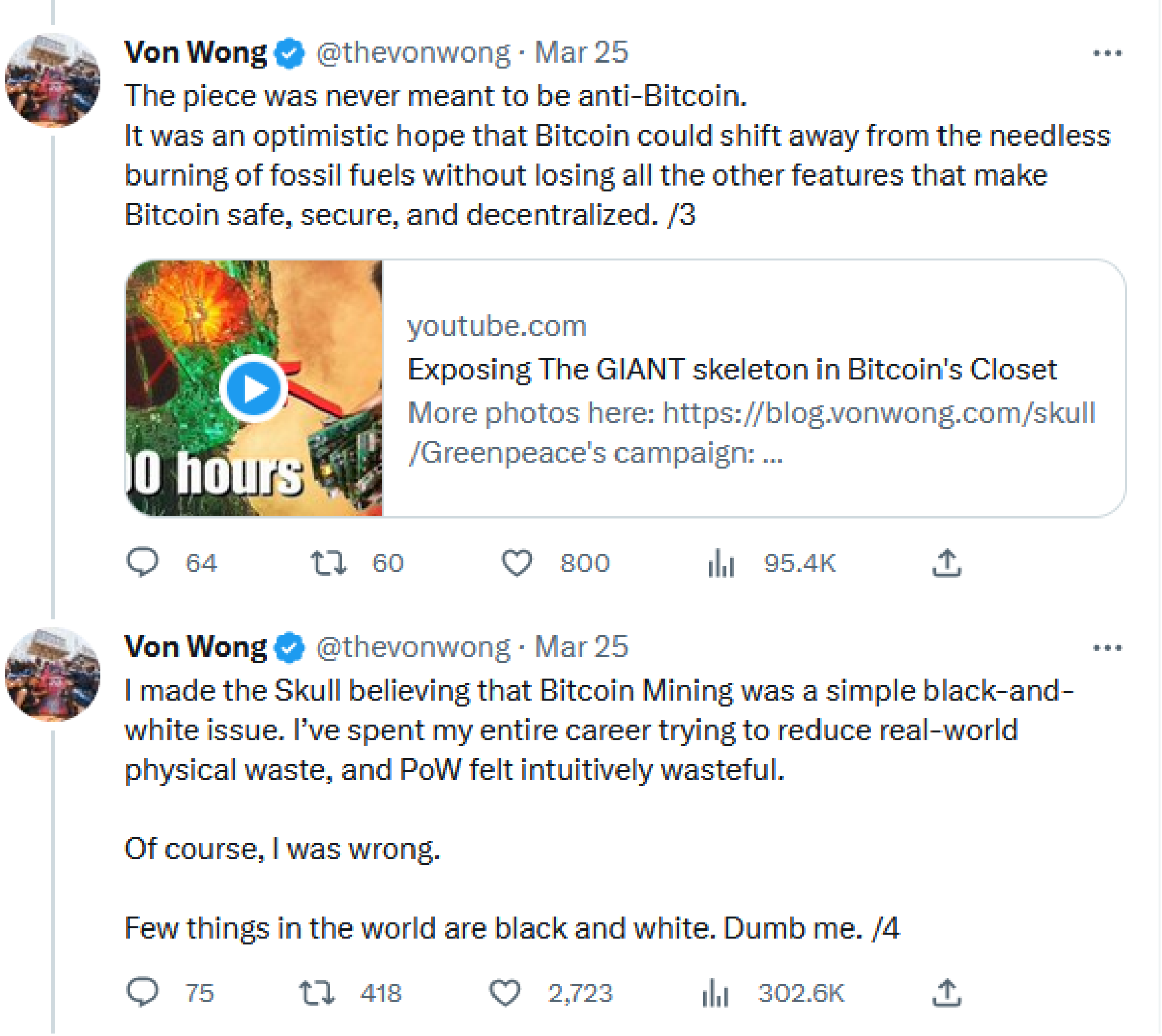
https://cbeci.org/mining_map/

Bitcoin mining map (January 2022)





- China used to be largest
  - – >80% (till 2018, slow decrease)
  - – Mining ASICS made in China
- China evicted “all” miners in May 2021
  - – Officially 0% (unofficially still active)
  - – Now coming back 21.11%
- Resulted in strong increase in:
  - – US 37.84%, Kazakhstan 13.22%
  - – Canada 6.48%, other 9% …



Von Wong @thevonwong · Mar 25

The piece was never meant to be anti-Bitcoin.
It was an optimistic hope that Bitcoin could shift away from the needless burning of fossil fuels without losing all the other features that make Bitcoin safe, secure, and decentralized. /3

youtube.com
Exposing The GIANT skeleton in Bitcoin's Closet
More photos here: https://blog.vonwong.com/skull /Greenpeace's campaign: ...

64 60 800 95.4K

Von Wong @thevonwong · Mar 25

I made the Skull believing that Bitcoin Mining was a simple black-and-white issue. I've spent my entire career trying to reduce real-world physical waste, and PoW felt intuitively wasteful.

Of course, I was wrong.

Few things in the world are black and white. Dumb me. /4

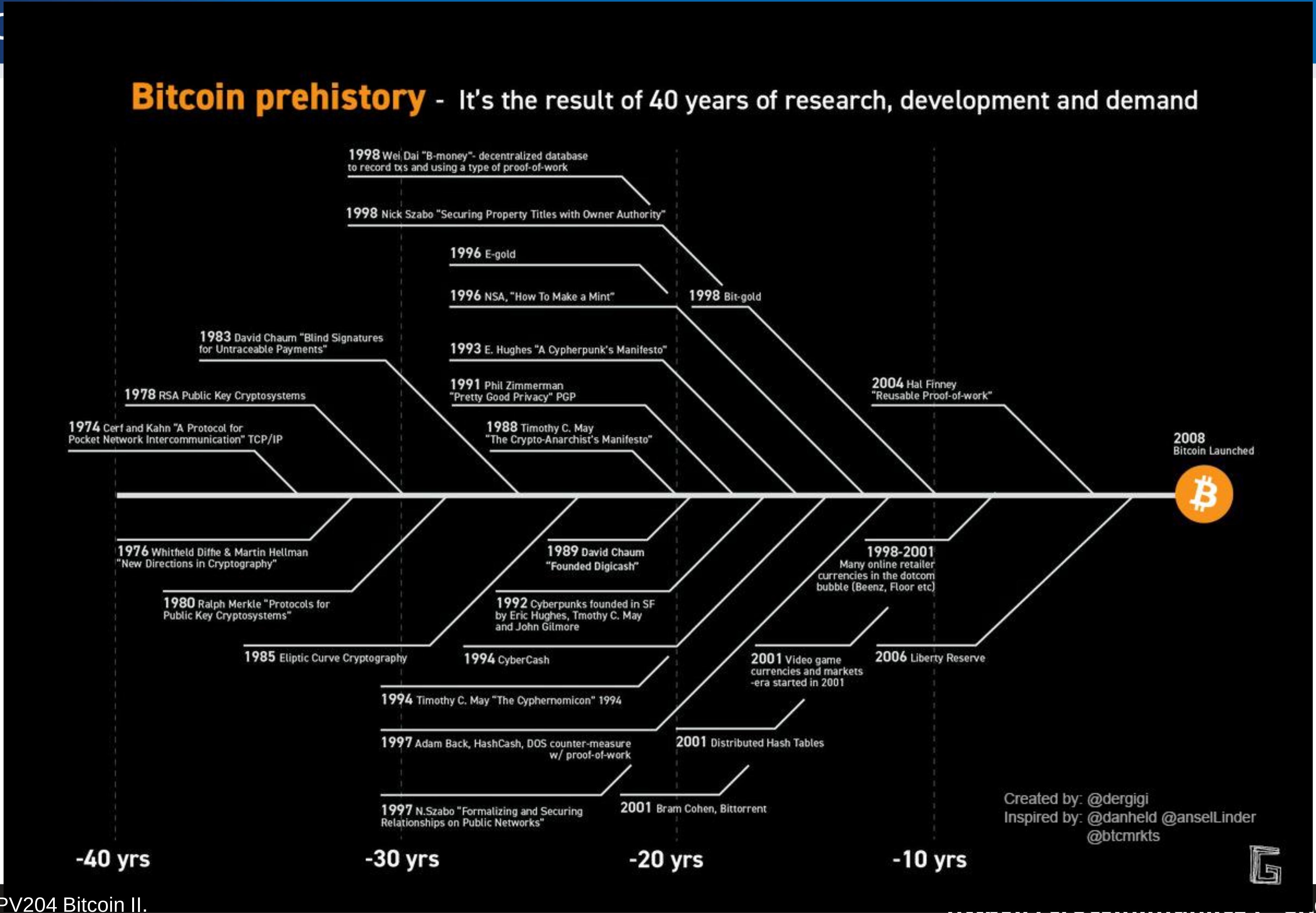
75 418 2,723 302.6K

https://twitter.com/thevonwong/status/1639690663846375424

Is Bitcoin mining wasteful?

- Heavily discussed topic (“Bitcoin boils the oceans by 2020”)
- Some questions to ask (**Do your own research!**)
  - What value you are getting for the energy expended? (neutral decentralized monetary system)
  - Miners want the cheapest energy available to maximize profits => energy nobody wants => waste energy
  - What is the source of the energy used? (btc mining ~60% “green” energy due to its low cost)
  - Can mining help to stabilize electrical grid with intermittent (solar, wind) sources? (instant turn on/off of mining ASICs, consumption only when cheap (= not demanded) energy)
  - How long is mining hardware profitable before dismantling? (depends on energy price, 5+ years)
  - Can miners finance construction of energy sources (hydro…) at places otherwise not viable financially (stranded energy)?
  - Can miners incentivize higher portion of intermittent (solar, wind) sources? (bigger source even when low sun/wind?)

UPGRADING BITCOIN FUNCTIONALITY



Bitcoin prehistory - It's the result of 40 years of research, development and demand

1998 Wei Dai "B-money"- decentralized database to record txs and using a type of proof-of-work

1998 Nick Szabo "Securing Property Titles with Owner Authority"

1996 E-gold

1996 NSA, "How To Make a Mint"

1998 Bit-gold

1983 David Chaum "Blind Signatures for Untraceable Payments"

1993 E. Hughes "A Cypherpunk's Manifesto"

1978 RSA Public Key Cryptosystems

1991 Phil Zimmerman "Pretty Good Privacy" PGP

2004 Hal Finney "Reusable Proof-of-work"

1974 Cerf and Kahn "A Protocol for Pocket Network Intercommunication" TCP/IP

1988 Timothy C. May "The Crypto-Anarchist's Manifesto"

2008 Bitcoin Launched

1976 Whitfield Diffie & Martin Hellman "New Directions in Cryptography"

1989 David Chaum "Founded Digicash"

1998-2001 Many online retailer currencies in the dotcom bubble (Beenz, Floor etc)

1980 Ralph Merkle "Protocols for Public Key Cryptosystems"

1992 Cyberpunks founded in SF by Eric Hughes, Tmothy C. May and John Gilmore

1985 Eliptic Curve Cryptography

1994 CyberCash

2001 Video game currencies and markets -era started in 2001

2006 Liberty Reserve

1994 Timothy C. May "The Cyphernomicon" 1994

1997 Adam Back, HashCash, DOS counter-measure w/ proof-of-work

2001 Distributed Hash Tables

1997 N.Szabo "Formalizing and Securing Relationships on Public Networks"

2001 Bram Cohen, Bittorrent

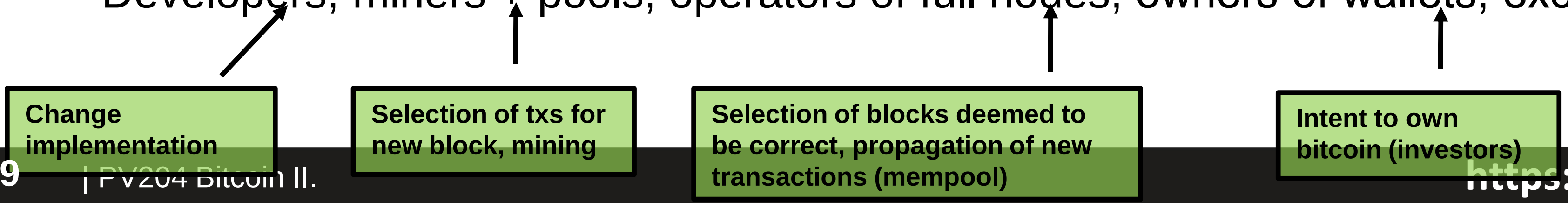
Created by: @dergigi
Inspired by: @danheld @anselLinder @btcmrkts

Options for upgrades

- Upgrading software in centralized environment is relatively easy
- Who sets rules for upgrades of Facebook WhatsApp?
  - Decision by FB management, implementation by FB developers
  - State regulation (e.g., customer protection laws, GDPR…)
  - Pressure of users (e.g., postponed date for EULA acceptation from February to May 2021)
- Who can influence technical consensus rules of Bitcoin?
  - Bitcoin Core full node is open-source (everybody can modify)
  - Softfork – backward compatible (**non-**actualized nodes will accept)
    - Old nodes only cannot use new functionality
  - Hardfork – backward incompatible (**non-**actualized nodes will reject)
    - E.g., change of block reward, block size, mining difficulty (=> average time per block), PoW vs. PoS…
- Bitcoin performs only softforks (to keep valid rules from time when you acquired your "bitcoins")

How to agree on code changes in Bitcoin Core?

- Changes NOT influencing consensus rules
  - E.g., code optimizations, changes in GUI/CLI…
  - „Common“ development process, pull requests + discussion + thorough review
- Changes influencing consensus rules
  - Discussion of the proposed change idea + example implementation
  - After some time, an attempt to include change into main branch repository
- How to decide if change shall be accepted or rejected?
  - Initially direct changes by Satoshi Nakamoto
  - Later small group of Bitcoin Core developers
  - Later development of various methods for signalization of readiness for acceptance or rejection
- Basic economic actors of Bitcoin ecosystem
  - Developers, miners + pools, operators of full nodes, owners of wallets, exchanges…



Change implementation

Selection of txs for new block, mining

Selection of blocks deemed to be correct, propagation of new transactions (mempool)

Intent to own bitcoin (investors)

Segregated witness (Segwit), softfork, August 2017

- Backward-compatible upgrade (soft fork) activated in August 2017
- Introduced the following changes:
  - Block size increase (up to ~4MB, witness data bytes discounted by 1/4)
  - Fixes transaction malleability (signature excluded from transaction ID computation)
  - Support for clear future versioning (special code rule for `OP_0 <20-byte hash>`)
- Additional witness data send only to node which requests it
  - Backward compatible, older nodes will not ask
  - Segwit transaction looks to them as “anyone-can-spend” script
- Significant controversy called “Blocksize wars” (several hardforks)
  - “Big blockers” wanted larger blocks or dynamic blocks to keep transaction fee low “forever”
    - But larger blocks increase blockchain size => less people able to run fullnode => centralization
  - New York Agreement, User Activate Soft Fork (UASF)
  - Demonstrates problems of decentralized social consensus (Schelling point)

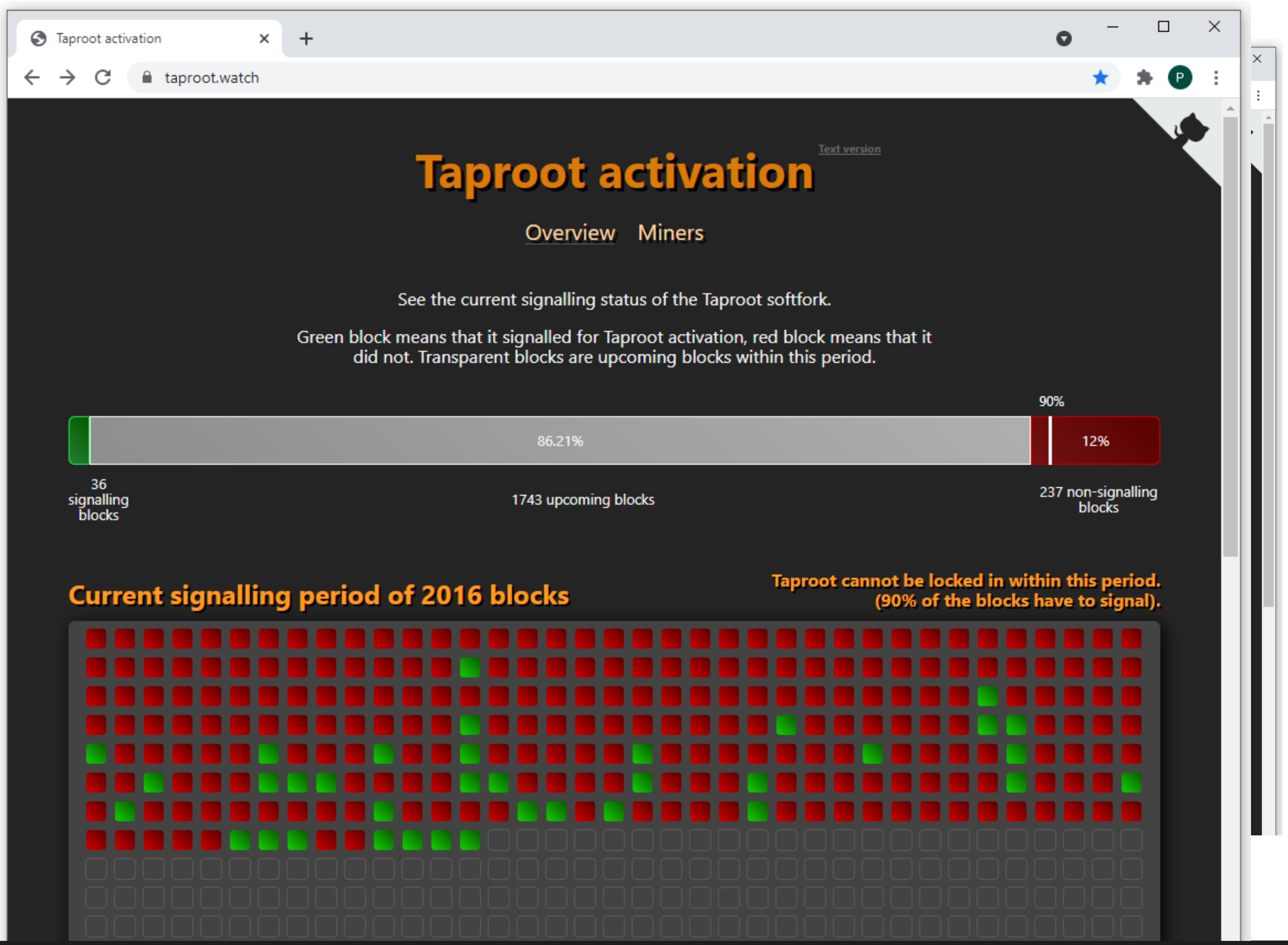
Taproot, softfork, October 2021

- Backward-compatible upgrade (soft fork) activated in October 2021
- Introduced the following changes:
  - Added support for Schnorr signatures (more compact, easier MPC…)
  - Increased privacy (Schnorr-based multiparty signature, Musig2, FROST…)
  - More powerful “Tapscript” added
- Not controversial, generally believed to be wanted improvement
  - “Speedy trial” used

Path to Taproot softfork

- 1990 publication of Schnorr signatures, 2008 expiration of patent
- Discussion for use in Bitcoin for a very long time
- 2018 formal Schnorr Bitcoin Improvement Proposals (BIPs 340, 341, 342)
- 2019 discussion of parameters, update of BIPs
- 2020 example implementation, available for testing on signet
- 2020/2021 discussion how to activate: signalization by miners vs. full nodes (UASF)
- Speedy trial – signalization by miners (according to BIP9)
  - Approximately 3 months to achieve >90% blocks in last 2016 blocks
- Possible option: User Activated Soft Fork (UASF), BIP8
  - If not activated by miners, some full nodes will start to reject mined blocks without Taproot signalization (possible fork of blockchain, mined block will not receive reward if abandoned => economic pressure on miners to activate)
- 12.6.2021 Taproot locked-in
  - Actualized implementations started to enforce Taproot rules from block 709632
- 14.11.2021 6:15UTC Taproot “activated” (first block mined)

https://taproot.watch/



Taproot activation

Text version

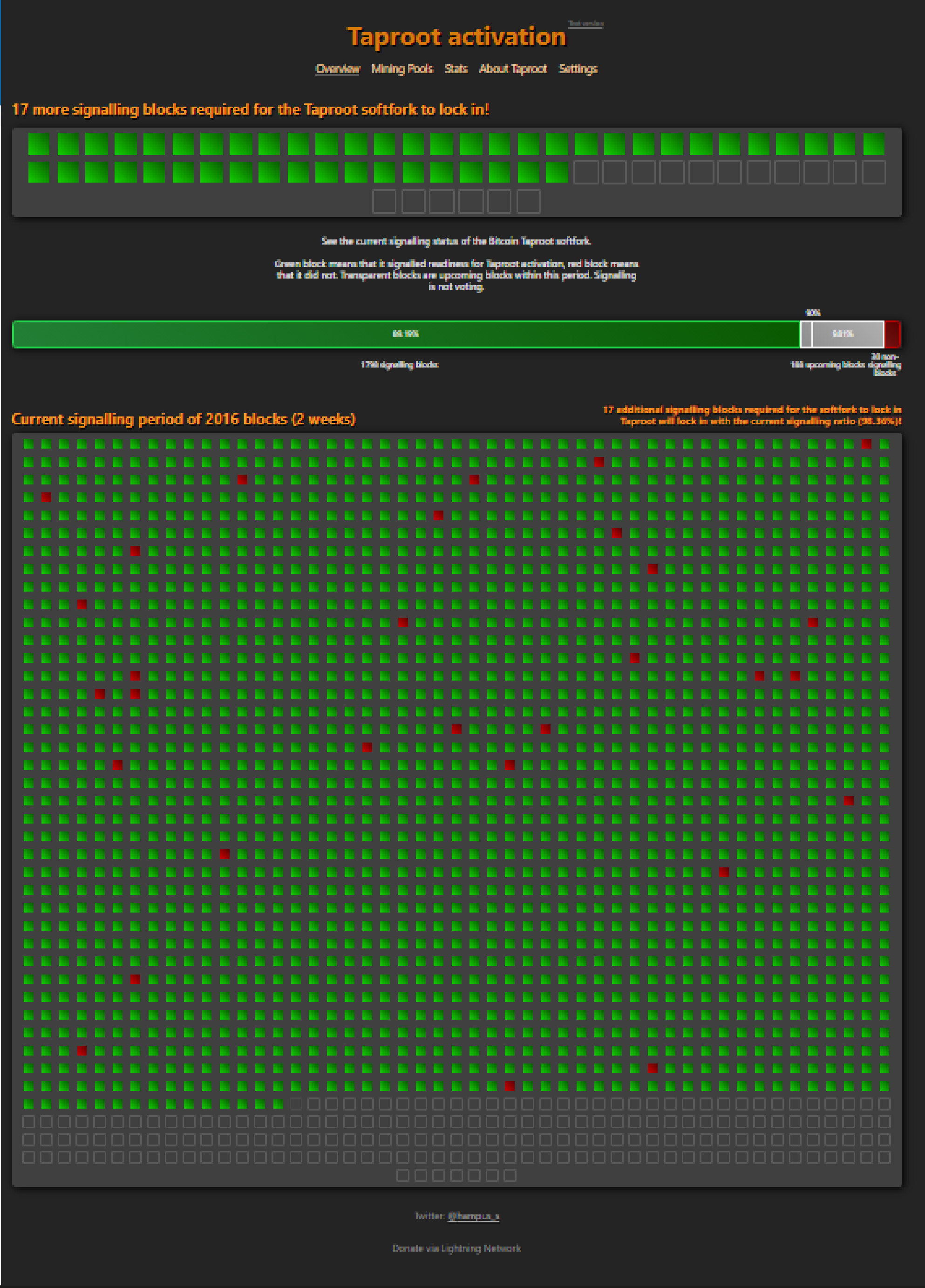
Overview Miners

See the current signalling status of the Taproot softfork.

Green block means that it signalled for Taproot activation, red block means that it did not. Transparent blocks are upcoming blocks within this period.

Current signalling period of 2016 blocks

Taproot cannot be locked in within this period. (90% of the blocks have to signal).



Taproot activation

Text version

Overview Mining Pools Stats About Taproot Settings

17 more signalling blocks required for the Taproot softfork to lock in!

See the current signalling status of the Bitcoin Taproot softfork.

Green block means that it signalled readiness for Taproot activation, red block means that it did not. Transparent blocks are upcoming blocks within this period. Signalling is not voting.

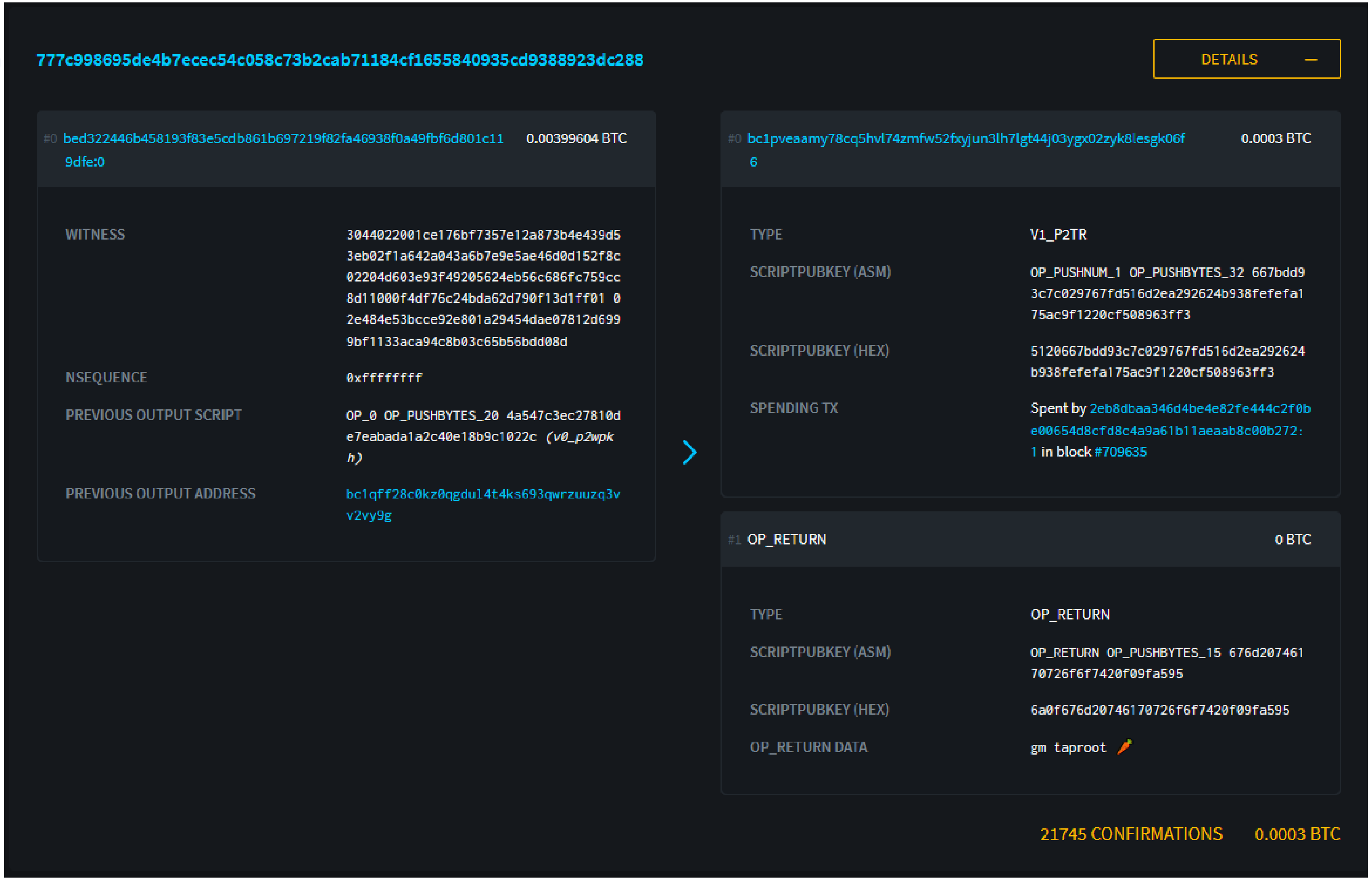
Current signalling period of 2016 blocks (2 weeks)

17 additional signalling blocks required for the softfork to lock in
Taproot will lock in with the current signalling ratio (98.38%)!

Twitter: @hampus_s

Donate via Lightning Network

První ,



777c998695de4b7ecec54c058c73b2cab71184cf1655840935cd9388923dc288

#0 bed322446b458193f83e5cdb861b697219f82fa46938f0a49fbf6d801c11 9dfe:0 0.00399604 BTC

WITNESS	3044022001ce176bf7357e12a873b4e439d53eb02f1a642a043a6b7e9e5ae46d0d152f8c02204d603e93f49205624eb56c686fc759cc8d11000f4df76c24bda62d790f13d1ff01 02e484e53bcce92e801a29454dae07812d6999bf1133aca94c8b03c65b56bdd08d
NSEQUENCE	0xffffffff
PREVIOUS OUTPUT SCRIPT	OP_0 OP_PUSHBYTES_20 4a547c3ec27810de7eabada1a2c40e18b9c1022c *(v0_p2wpkh)*
PREVIOUS OUTPUT ADDRESS	bc1qff28c0kz0qgdul4t4ks693qwrzuuzq3vv2vy9g

#0 bc1pveaamy78cq5hvl74zmfw52fxyjun3lh7lgt44j03ygx02zyk8lesgk06f6 0.0003 BTC

TYPE	V1_P2TR
SCRIPTPUBKEY (ASM)	OP_PUSHNUM_1 OP_PUSHBYTES_32 667bdd93c7c029767fd516d2ea292624b938fefefa175ac9f1220cf508963ff3
SCRIPTPUBKEY (HEX)	5120667bdd93c7c029767fd516d2ea292624b938fefefa175ac9f1220cf508963ff3
SPENDING TX	Spent by 2eb8dbaa346d4be4e82fe444c2f0be00654d8cfd8c4a9a61b11aeaab8c00b272:1 in block #709635

#1 OP_RETURN 0 BTC

TYPE	OP_RETURN
SCRIPTPUBKEY (ASM)	OP_RETURN OP_PUSHBYTES_15 676d20746170726f6f7420f09fa595
SCRIPTPUBKEY (HEX)	6a0f676d20746170726f6f7420f09fa595
OP_RETURN DATA	gm taproot 🥕

21745 CONFIRMATIONS 0.0003 BTC

https://blockstream.info/tx/777c998695de4b7ecec54c058c73b2cab71184cf1655840935cd9388923dc288

Future Bitcoin upgrades

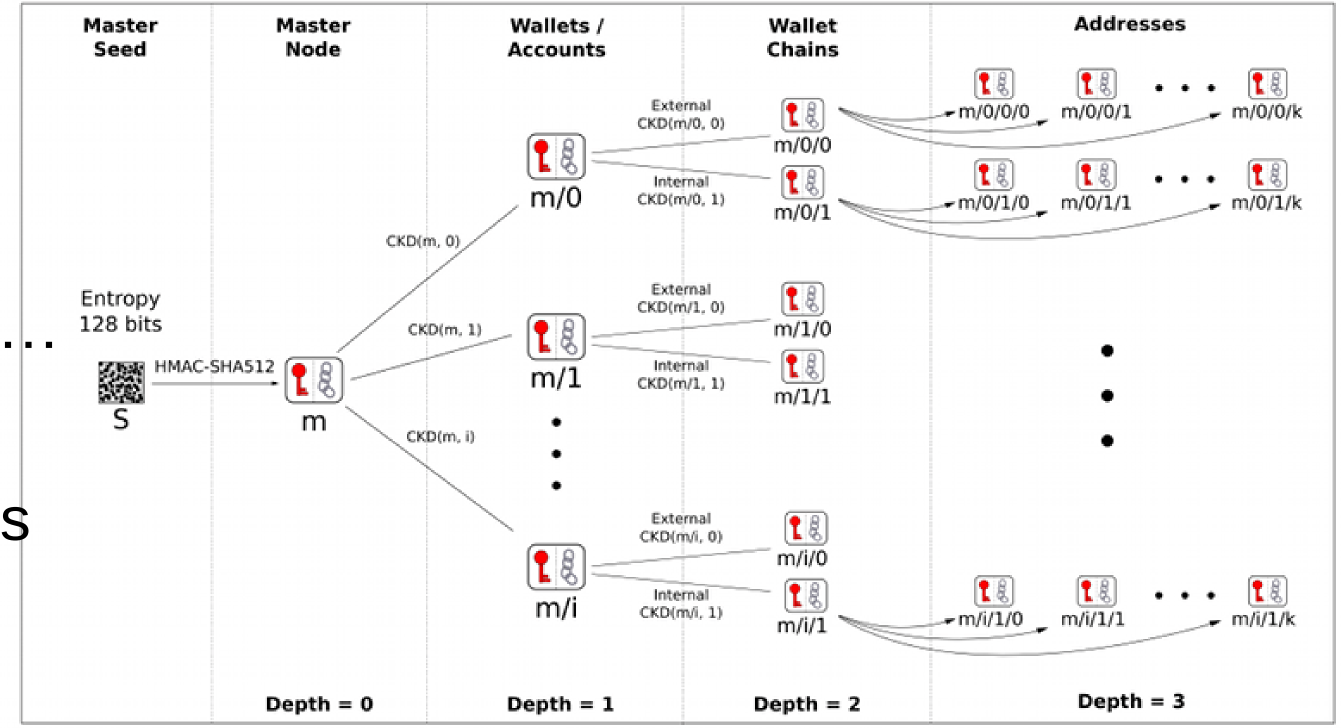
- Protocols tend to ossify with adoption (e.g., TCP protocol)
  - Difficult to update software at once, increased probability of problems after change
- Many discussed future changes (some already tested on Signet)
  - OP_VAULT (convenance)
  - SIGHASH_ANYPREVOUT (Eltoo, channel factories)
  - Cross-input signature aggregation (one signature for multiple inputs)
  - Drivechains, spacechains…
  - (Time representation in block will overflow in 2106)
- Potential hardforks?
  - (Quantum computer breaking ECDSA)

THRESHOLD SIGNATURES VS. MULTISIG VS. MULTI-PARTY COMPUTATION

Making fresh private keys (with backup) BIP32, BIP44…

- Deterministic derivation from:
  - – master seed (key)
  - – derivation path (data)
    - m/purpose/coin/account/receive…
- Single master seed allows:
  - – Generate many distinct private keys
  - – Sharing sub-tree value allows:
    - Generate keys in sub-trees
    - Cannot generate keys from other trees
- Deterministic generation, Master Seed enough to recover whole tree

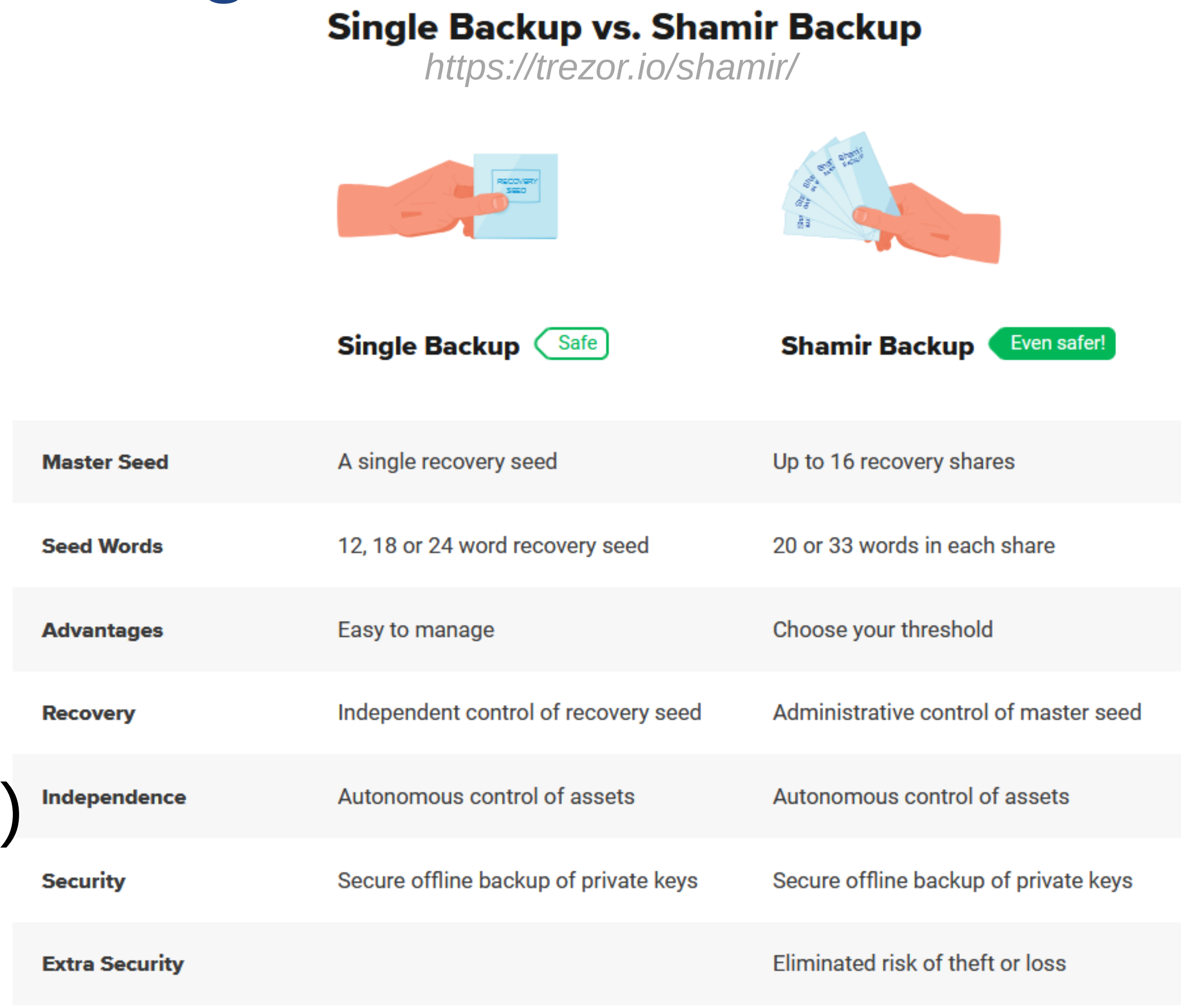
BIP 32 - Hierarchical Deterministic Wallets



Child Key Derivation Function ~ CKD(x,n) = HMAC-SHA512(x_{Chain}, x_{PubKey} || n)

1. Shamir’s threshold secret sharing scheme

- Private key is recovered from multiple shares
  - Then used at single place
  - An attacker can compromise private key after its recovery from shares
- Network is unaware of key split, single public key used in lock script
- Can be used to backup wallet seed (e.g., Trezor wallet https://trezor.io/shamir/)
  - n-out-of-n or k-out-of-n



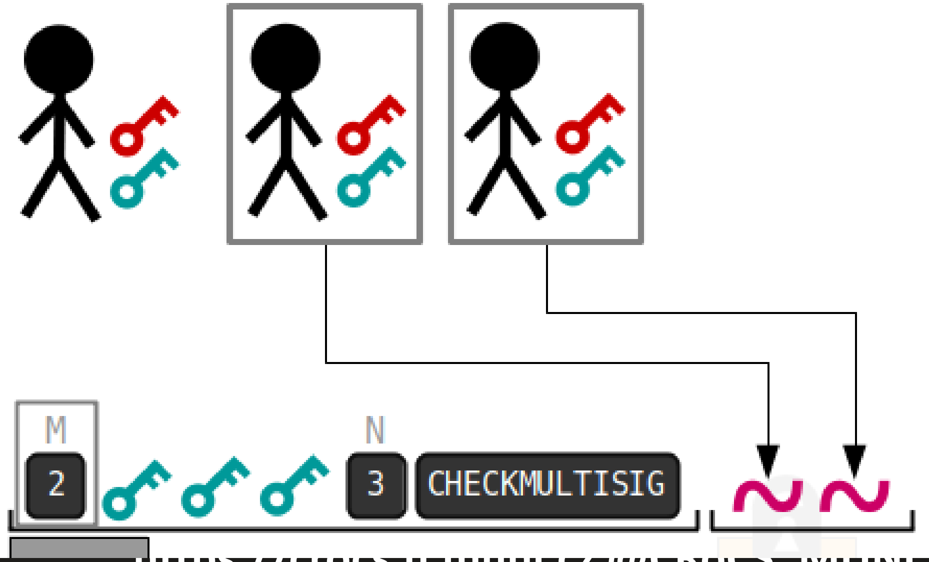
Single Backup vs. Shamir Backup

https://trezor.io/shamir/

	Single Backup (Safe)	Shamir Backup (Even safer!)
Master Seed	A single recovery seed	Up to 16 recovery shares
Seed Words	12, 18 or 24 word recovery seed	20 or 33 words in each share
Advantages	Easy to manage	Choose your threshold
Recovery	Independent control of recovery seed	Administrative control of master seed
Independence	Autonomous control of assets	Autonomous control of assets
Security	Secure offline backup of private keys	Secure offline backup of private keys
Extra Security		Eliminated risk of theft or loss

Multisignatures

- Lock script constructed to require multiple signatures (OP_CHECKMULTISIG)
  - – transaction valid only if multiple signers provide signatures for unlock script
- n-out-of-n or k-out-of-n, https://en.bitcoin.it/wiki/Multisignature
- P2MS, P2MS wrapped in P2SH
  - – https://learnmeabitcoin.com/technical/p2ms

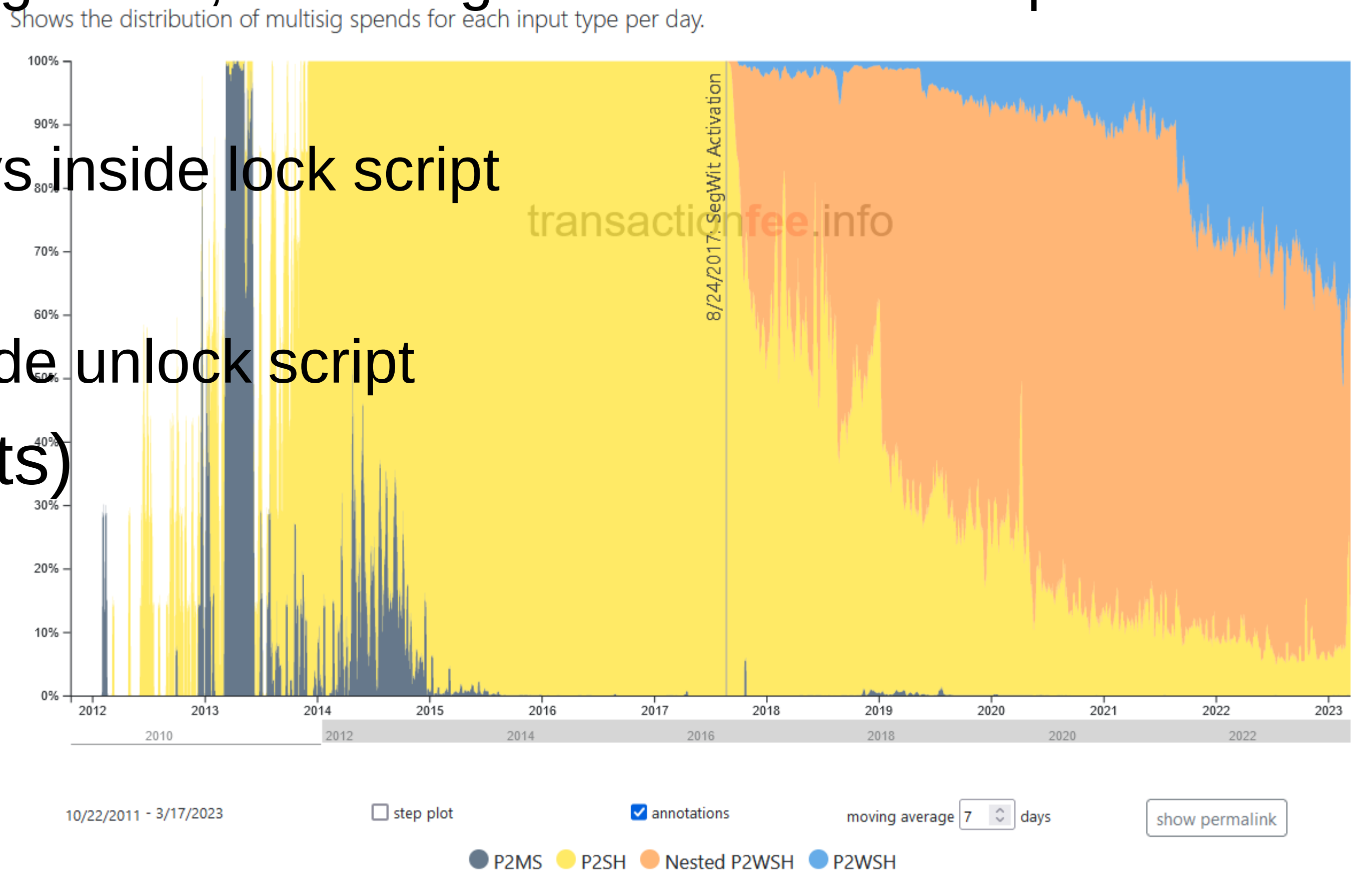


Secure multi-party computation (MPC)

- Single signature computed using multiple separated signers
  - Each signer has own private key
  - An attacker must comprise more than one entity
- Communication between signers
  - During initial key generation
  - Optionally during signing
- Legacy compatible schemes (produces valid ECDSA signature)
  - 2-party ECDSA, n-out-of-n or k-out-of-n ECDSA (only since 2016)
- Taproot-compatible schemes (activated since Nov 2021)
  - Schorr signatures, MuSig2, FROST…
- https://academy.binance.com/en/articles/threshold-signatures-explained

Frequency of different multisignature scripts

- Cannot tell for Shamir, MPC ECDSA and Schnorr (e.g., MuSig)!
  - Resulting signature is standard signature, no change to lock/unlock scripts
- Can tell for P2MS
  - Threshold and allowed public keys inside lock script
- Can tell for P2SH (if spent)
  - Multisig script and used keys inside unlock script
- (analogically for Segwit variants)



Shows the distribution of multisig spends for each input type per day.

10/22/2011 - 3/17/2023

□ step plot

☑ annotations

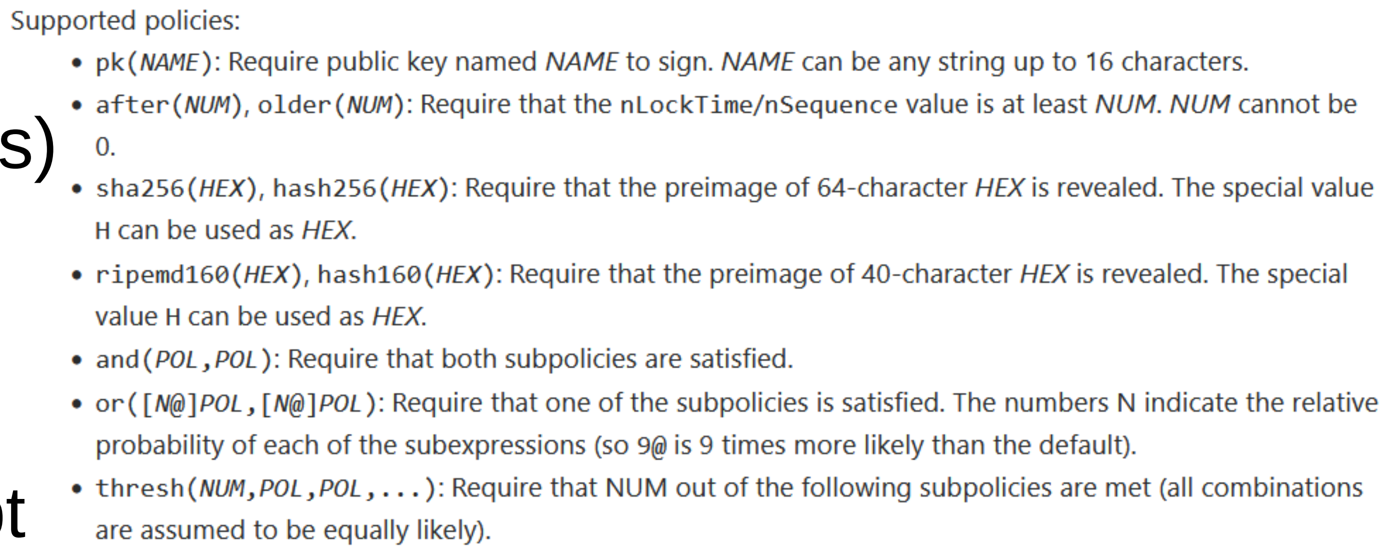
moving average 7 days

Partially Signed Bitcoin Transaction (PSBT), BIP-174

- Standardized way how to represent unsigned Bitcoin transaction
  - Transaction skeleton is created (inputs, outputs, amounts…)
  - But unlock script not completed (actual signature/signatures are missing)
- Suitable for sharing PSBT between multiple signers in secure way
  - Serialized transaction can be passed using USB, file, QRCode, NFC…
- When all signers sign, transaction can be broadcasted to p2p network
  - Unless fully signed, transaction is invalid and would be rejected
- Binary format (for compactness), but can be decoded to json
  - `bitcoin-cli decodepsbt "psbt_as_hex_format"`

Miniscript (A. Poelstra, P. Wuille, S. Kanjalkar, 2019)

- Language for easier and error-prone creation of Bitcoin scripts
  - Subset of Bitcoin script language
  - Human-readable, easy to express complex locking conditions
  - https://bitcoin.sipa.be/miniscript/
- Simple building blocks (policies)
  - Single-key, Multi-key,
  - Time-locks, Check-sequence,
  - Hash-lock…
- Compiler creates optimal script
  - And cost analysis

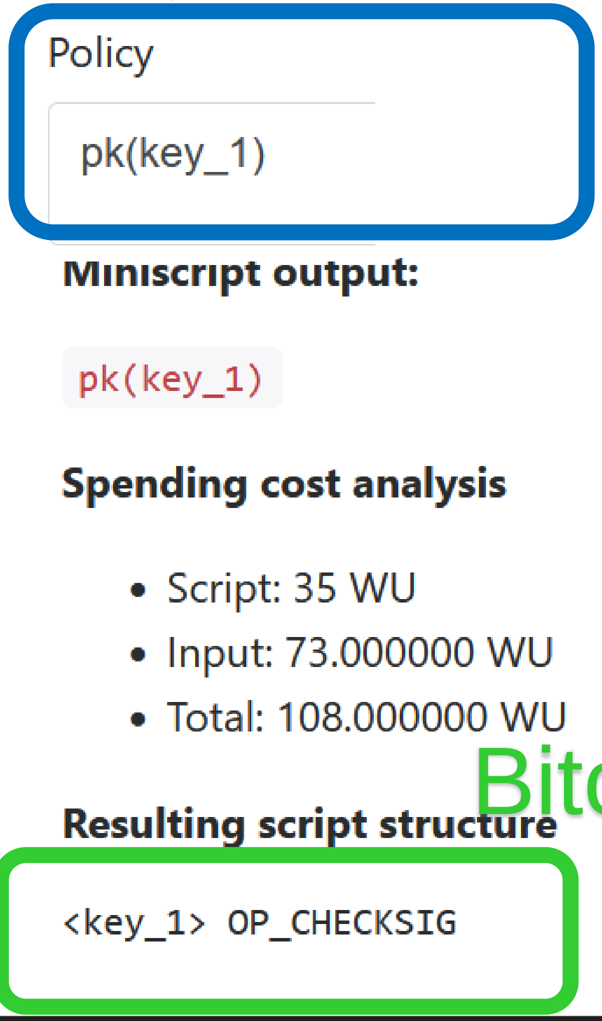


Supported policies:

- `pk(`*NAME*`)`: Require public key named *NAME* to sign. *NAME* can be any string up to 16 characters.
- `after(`*NUM*`)`, `older(`*NUM*`)`: Require that the `nLockTime`/`nSequence` value is at least *NUM*. *NUM* cannot be 0.
- `sha256(`*HEX*`)`, `hash256(`*HEX*`)`: Require that the preimage of 64-character *HEX* is revealed. The special value `H` can be used as *HEX*.
- `ripemd160(`*HEX*`)`, `hash160(`*HEX*`)`: Require that the preimage of 40-character *HEX* is revealed. The special value `H` can be used as *HEX*.
- `and(`*POL*`,`*POL*`)`: Require that both subpolicies are satisfied.
- `or([`*N*`@]`*POL*`,[`*N*`@]`*POL*`)`: Require that one of the subpolicies is satisfied. The numbers N indicate the relative probability of each of the subexpressions (so 9@ is 9 times more likely than the default).
- `thresh(`*NUM*`,`*POL*`,`*POL*`,...)`: Require that NUM out of the following subpolicies are met (all combinations are assumed to be equally likely).

Miniscript examples

A single key



Policy

pk(key_1)

Miniscript output:

```
pk(key_1)
```

Spending cost analysis

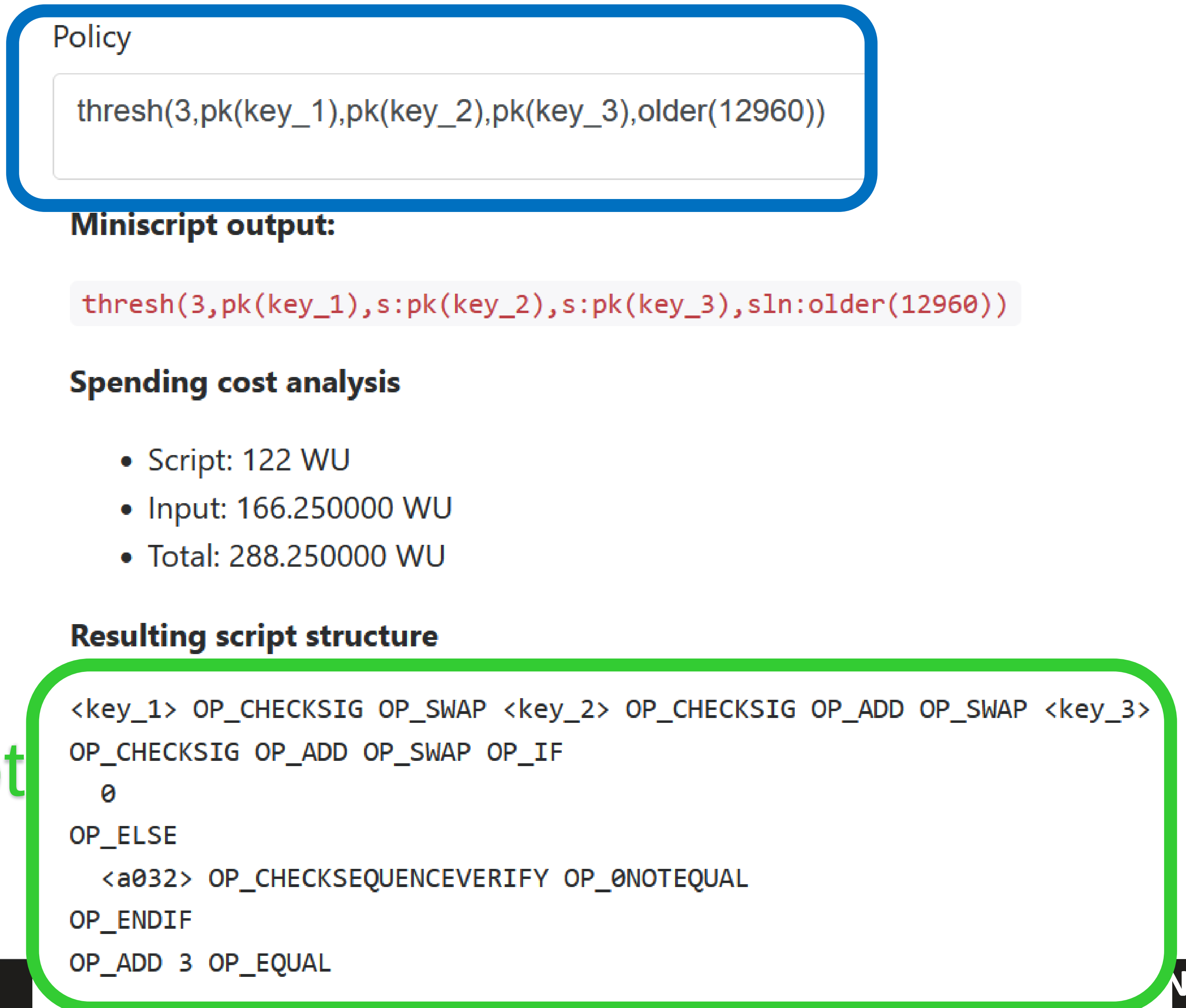
- Script: 35 WU
- Input: 73.000000 WU
- Total: 108.000000 WU

Resulting script structure

```
<key_1> OP_CHECKSIG
```

Bitcoin script

A 3-of-3 that turns into a 2-of-3 after 90 days



Policy

thresh(3,pk(key_1),pk(key_2),pk(key_3),older(12960))

Miniscript output:

```
thresh(3,pk(key_1),s:pk(key_2),s:pk(key_3),sln:older(12960))
```

Spending cost analysis

- Script: 122 WU
- Input: 166.250000 WU
- Total: 288.250000 WU

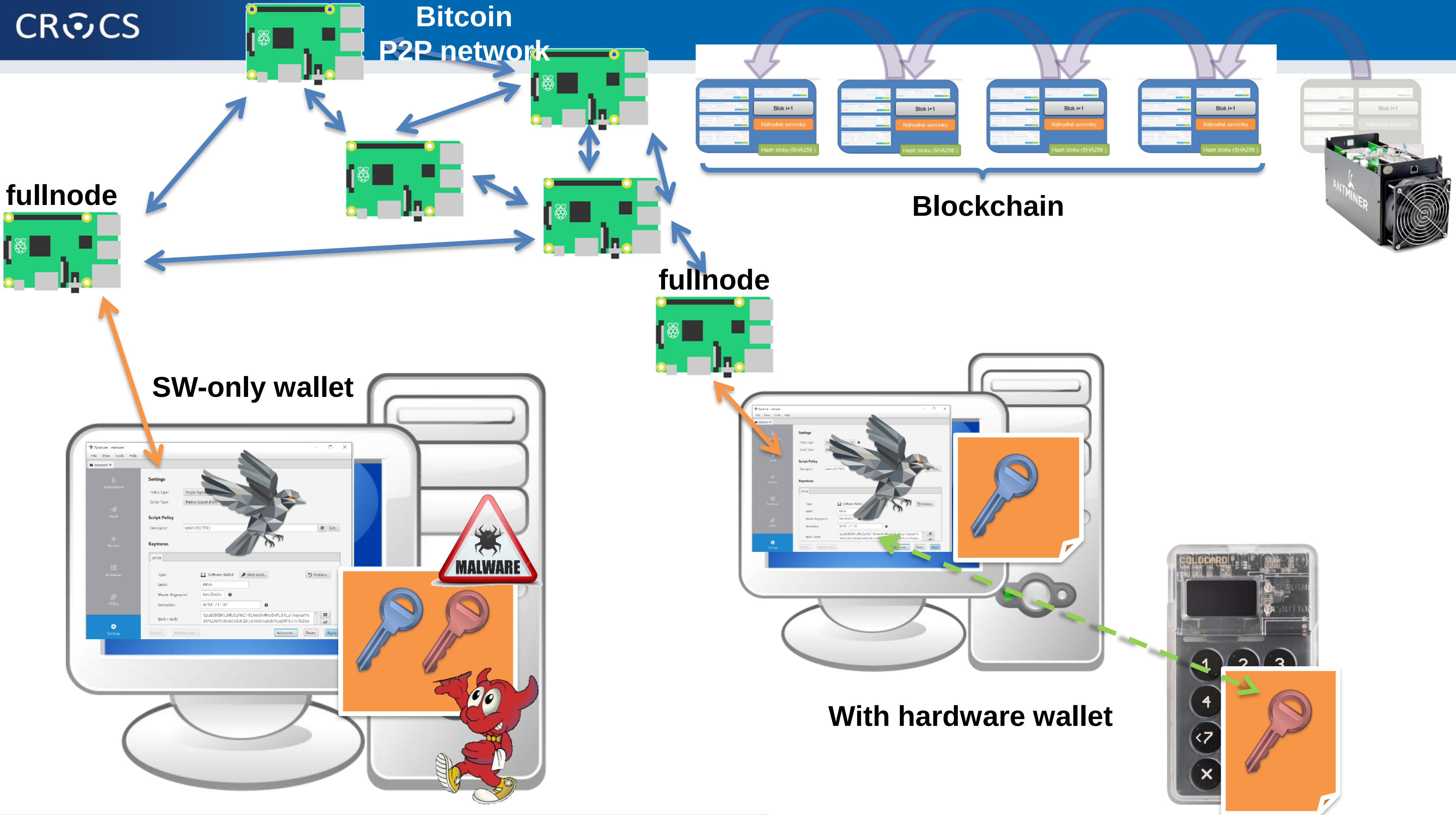
Resulting script structure

```
<key_1> OP_CHECKSIG OP_SWAP <key_2> OP_CHECKSIG OP_ADD OP_SWAP <key_3>
OP_CHECKSIG OP_ADD OP_SWAP OP_IF
  0
OP_ELSE
  <a032> OP_CHECKSEQUENCEVERIFY OP_0NOTEQUAL
OP_ENDIF
OP_ADD 3 OP_EQUAL
```

Inheritance problem – degradable multisig?

- Copy of seeds – heirs can take your bitcoins
  - Not really solving the problem, just shifting it
- Notary – requires trust to people, can take your btc
- Timelock for usage of other keys
  - one key immediately, another key only after 1 year (timelock)
  - Liana wallet (example), https://github.com/wizardsardine/liana
    - https://wizardsardine.com/blog/liana-announcement/
  - Requires periodic resend of the money to prevent heirs access
  - Relative or absolute timelock use possible
- Degrading multisig
  - 4-of-4 immediately, 3-of-4 after 9 months, 2-of-6 after year and half

HARDWARE WALLETS



Bitcoin P2P network

fullnode

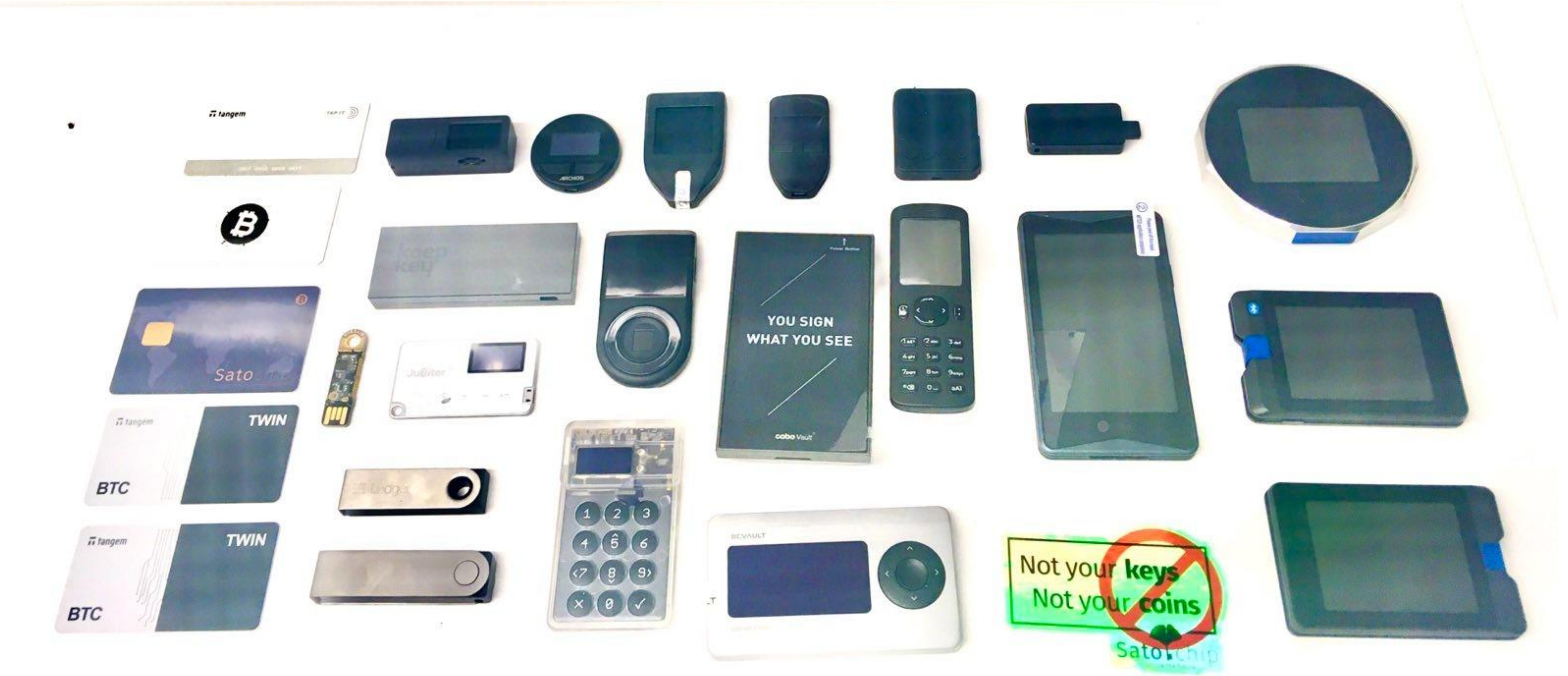
Blockchain

fullnode

SW-only wallet

With hardware wallet

Testbed with hardware wallets



Research done in CRoCS on hardware wallets

- Transparency of device build (HW/SW/TE)
- Multi-Party Computation (signatures, decryption) on limited hardware
- Analysis of cryptographic implementations (ECDSA, BIP32…)
- Entropy analysis of keys extracted from blockchain
- Usable security

BITCOIN PRIVACY

Risks

- Risk of lost coins
  - Lost wallet keys, forgotten access credentials
- Risk of stolen coins
  - Malware on computer (wallet keys), phishing/scam (recovery phrase)
  - Compromised trusted third party (exchange, web wallet…)
  - Random burglary (don’t know you have btc)
  - Targeted burglary (know you have btc), with(-out) you present
- Risk of traced coins
  - blockchain analysis, additional metadata correlation analysis (KYC/AML, scans, tx propagation, wallet peeling…)
  - Crooks, governments, wife…

Attacker models

- Blockchain-only analysis
- Malware, phishing
- Active network analysis, metadata
- Cryptographic analysis of used algorithms
- Side-channel analysis

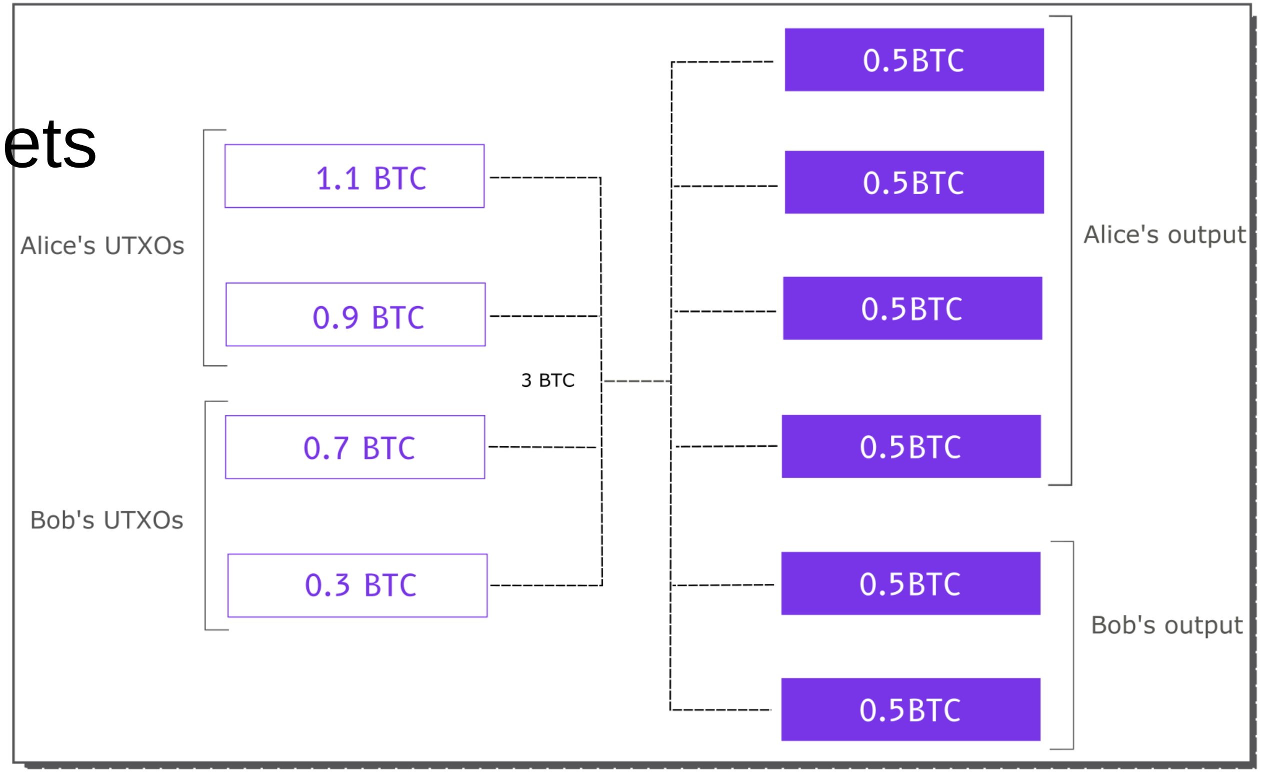
Improving privacy

- Hold your private keys (no custodial service like exchange…)
  - Cannot steal, cannot observe, cannot “vote” on your behalf
- Store private key in hardware wallet (Trezor, ColdCard, Ledger…)
  - Keys in “hot” software wallets are prone to malware attack
- Run own full node over Tor and connect your wallet to it
- Make on-chain analysis harder: https://en.bitcoin.it/wiki/Privacy
- Use manual coin selection, label coins by its origin (in your wallet only)
- Use CoinJoin, PayJoin (multiple users mix their inputs in single transaction)
- Have good opsec (no posting of own btc addresses, use Tor to broadcast tx, delink via CoinJoin after KYC…)

https://en.bitcoinwiki.org/wiki/CoinJoin
https://cryptotesters.com/blog/what-are-coinjoins-and-how-do-they-improve-bitcoin-privacy

CoinJoin

- Multiple users collaborates trustlessly in creating large transaction
- Outputs are all the same value => cannot be attributed to one of senders based on the value
- Supported by more advanced wallets
  – Wasabi wallet, Samurai wallet

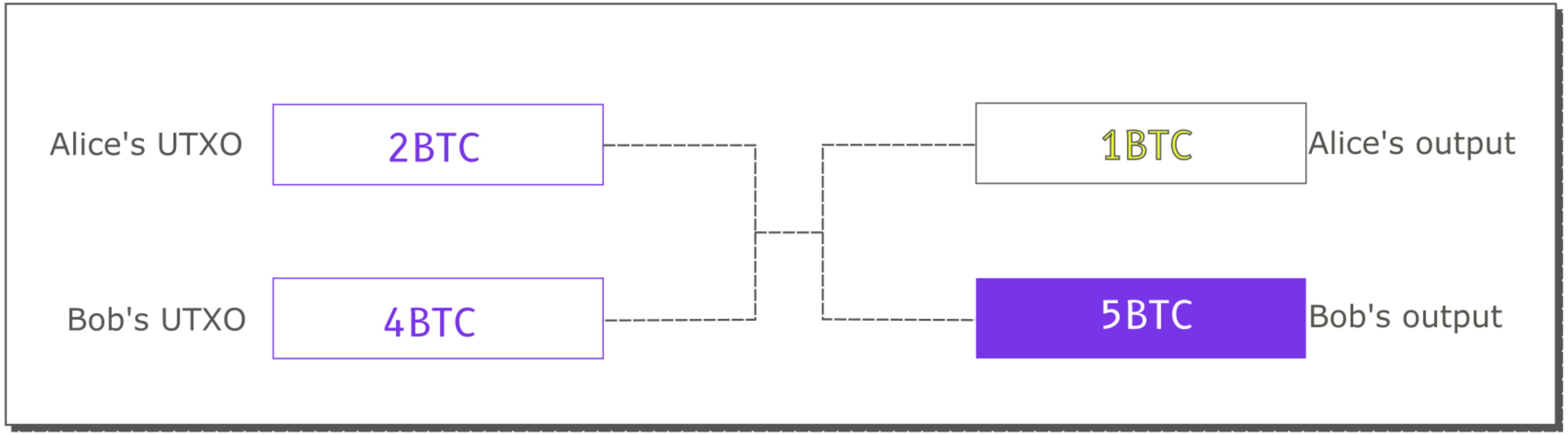


CoinJoin implementations

- Wasabi wallet https://github.com/zkSNACKs/WalletWasabi/
  - Centralized trustless coordinator, Tor, selected number of rounds executed within hours
    - https://docs.wasabiwallet.io/using-wasabi/CoinJoin.html
  - Wasabi 2.0 (beta) offer non-equal output coinjoin https://blog.wasabiwallet.io/privacy-guarantees-of-wasabi-wallet-2-0/
  - Anonymity set decrease over the time as people send their outputs to KYC exchanges
- Samourai Whirpool https://docs.samourai.io/en/whirlpool
  - CoinJoin with variable number of rounds, centralized trustless coordinator
  - CoinJoin runs until output is send away from Whirpool (days/months)
  - If not fullnode then xpub must be provided => privacy risk, decreased anonymity set
    - e.g., Samurai RoninDojo https://ronindojo.io/
  - Clients: Samourai wallet / Whirpool cli, SparrowWallet (using Samourai code)
- JoinMarket
  - No central coordinator, market Maker(s) run own fullnode and provide liquidity
  - Coinjoin transaction creation is coordinated by Taker who is paying also fee (on-chain and to the Maker)
  - JoininBox - JoinMarket cmdline-focused distribution https://github.com/openoms/joininbox

PayJoin

- PayJoin is special case of CoinJoin, but with less participants (typically only two: sender, receiver) and without equal UTXO sizes
- Faster than CoinJoin, done during a normal payment



- https://cryptotesters.com/blog/what-are-coinjoins-and-how-do-they-improve-bitcoin-privacy

ON-CHAIN BITCOIN ALTERNATIVES

Why search for other options (L2/sidechain/altcoins)?

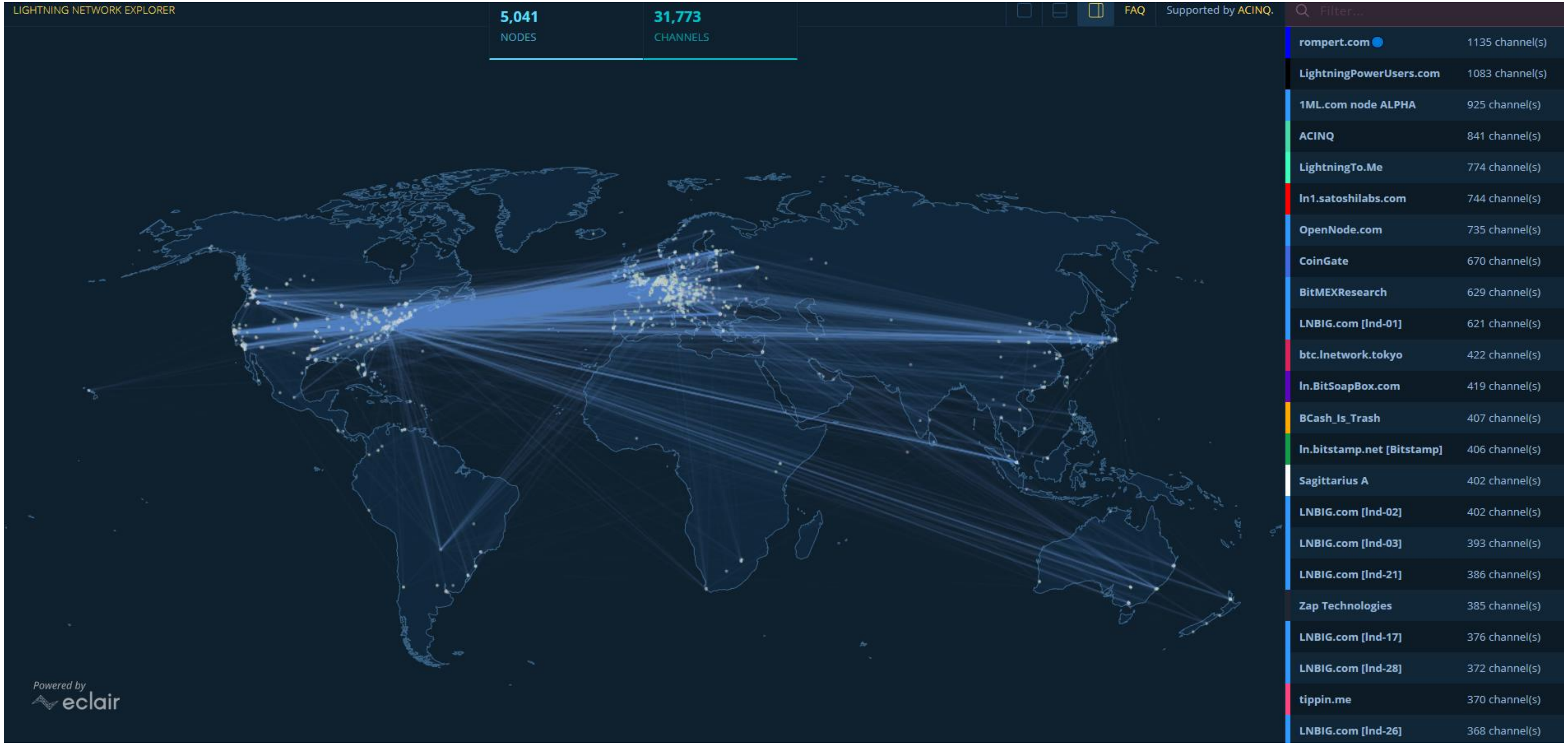
- Why something else than on-chain Bitcoin? List of typical reasons:

1. Cost of sending transaction
   - Peak was tens of dollars (for every transfer), variable (from 1sat/vB), but has to increase in future due to decreasing reward
2. Time to confirm transaction (+ limited block size)
   - 4 blocks inside chain commonly required, ~10 minutes per block => ~40 min
3. Traceability of transactions
   - Source, destination and amount is on public ledger
4. Limited scripting language (lock script)
   - For more complicated smart contracts
5. Mining requirements
   - Specialized mining equipment required (ASICs) => may cause centralization if not enough widespread
   - Proof of Work is energy intensive (what it means?)

- …

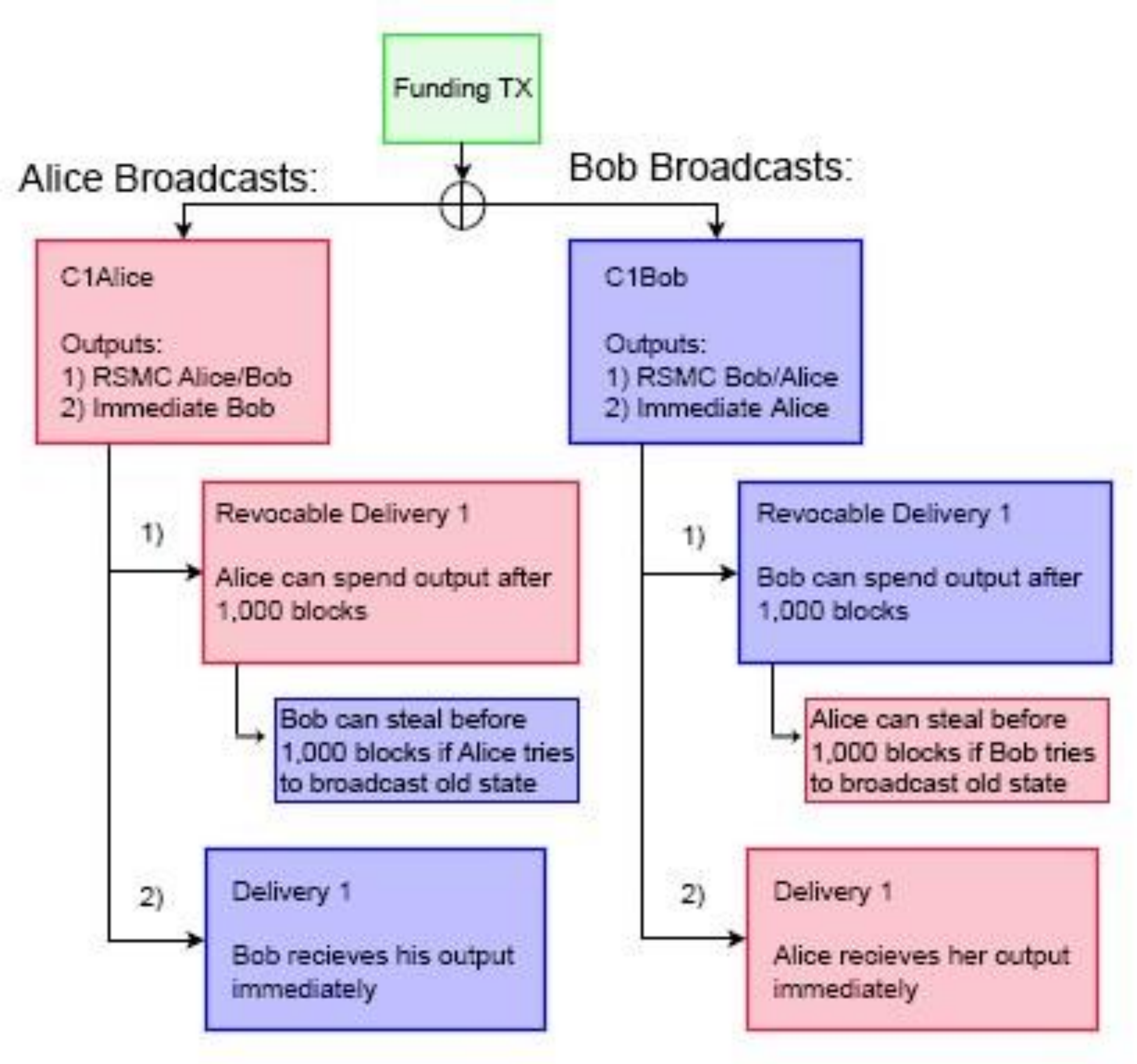
LIGHTING NETWORK

Lighting network https://explorer.acinq.co/



rompert.com ●	1135 channel(s)
LightningPowerUsers.com	1083 channel(s)
1ML.com node ALPHA	925 channel(s)
ACINQ	841 channel(s)
LightningTo.Me	774 channel(s)
ln1.satoshilabs.com	744 channel(s)
OpenNode.com	735 channel(s)
CoinGate	670 channel(s)
BitMEXResearch	629 channel(s)
LNBIG.com [lnd-01]	621 channel(s)
btc.lnetwork.tokyo	422 channel(s)
ln.BitSoapBox.com	419 channel(s)
BCash_Is_Trash	407 channel(s)
ln.bitstamp.net [Bitstamp]	406 channel(s)
Sagittarius A	402 channel(s)
LNBIG.com [lnd-02]	402 channel(s)
LNBIG.com [lnd-03]	393 channel(s)
LNBIG.com [lnd-21]	386 channel(s)
Zap Technologies	385 channel(s)
LNBIG.com [lnd-17]	376 channel(s)
LNBIG.com [lnd-28]	372 channel(s)
tippin.me	370 channel(s)
LNBIG.com [lnd-26]	368 channel(s)

Opening channel



Funding TX

Alice Broadcasts:

Bob Broadcasts:

C1Alice

Outputs:
1) RSMC Alice/Bob
2) Immediate Bob

1) Revocable Delivery 1

Alice can spend output after 1,000 blocks

Bob can steal before 1,000 blocks if Alice tries to broadcast old state

2) Delivery 1

Bob recieves his output immediately

C1Bob

Outputs:
1) RSMC Bob/Alice
2) Immediate Alice

1) Revocable Delivery 1

Bob can spend output after 1,000 blocks

Alice can steal before 1,000 blocks if Bob tries to broadcast old state

2) Delivery 1

Alice recieves her output immediately

Note: In future, P2TR will be used - opening and (collaborative) closing of channel looks same as ordinary payment

https://medium.com/@jkendzicky16/the-bitcoin-lightning-network-a-technical-primer-d8e073f2a82f

Some Lighting topics I.

- Custodial Lighting wallet (e.g., Wallet of Satoshi)
  - Service hold your private key, full trust in service
- Semi-custodial Lighting wallet (e.g., default BlueWallet, Zap…)
  - own key, but trust in 3rd party providing blockchain info
- Non-custodial (e.g., BlueWallet collected to own full node)
  - own key, blockchain info and monitoring by own full node
- Inbound, outbound capacity of channel between A and B
  - Initial value is given by initial on-chain 2-2 multisig transaction (x:0, x:y, 0:y)
  - Changes with every off-chain transaction executed (between A and B)

Some Lighting topics II.

- Sentinel service
  - trustless blockchain observer, broadcasts justice transaction in case of old state detected
  - No need for your full node to be always online
- Privacy considerations
  - Most of the transactions are NOT recorded on the blockchain
    - Good for speed as well as privacy
  - Doesn’t mean that payments are not traceable
    - Same as with internet connection => need to use Tor, ideally mixes…
    - Privacy of sender is significantly better than receiver
  - Taproot introduced ability to open channel indistinguishable from normal P2TR

Study materials

- Lighting network
  - ‘What is the Lightning Network?’ by 99Bitcoins (7 minutes)
    - https://www.youtube.com/watch?v=pBh4DcM-0pg
  - Watch ‘A Technical Introduction to The Lightning Network’ by A. Antonopoulos (43min)
    - https://www.youtube.com/watch?v=E1n3sKKPD_k

Lightning network – study more

- Description of Lighting Network basic principles
  - https://medium.com/@jkendzicky16/the-bitcoin-lightning-network-a-technical-primer-d8e073f2a82f
- Presentation by original Lighting creators
  - https://lightning.network/lightning-network.pdf
- List of Lighting nodes ready for channel opening
  - Bottom of the https://store.blockstream.com/

CASE STUDY: ORDINALS/INSCRIPTIONS

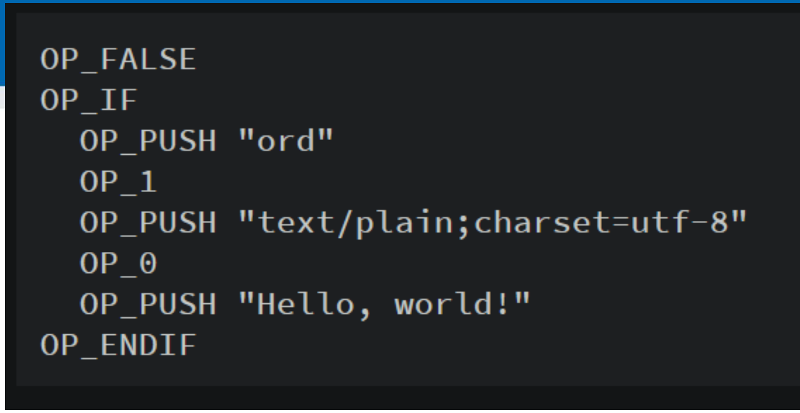
Case study: Ordinals/Inscription (NFT)

- Non-Fungible Token (NFT)
  - Unique digital asset, cannot be exchanged for other units of the same type
  - Dollars or satoshis are fungible (1$ = 1$), while NFT is non-fungible
  - Examples: jpegs, movie, music, numbered ticket, numbered equity…
  - Ownership can be transferred (methods depends on the underlying chain)
- Frequently, tied to blockchain like Bitcoin (Colored Coins) or Ethereum
  - Only URI and hash is stored in contract, actual picture/NFT stored elsewhere
  - Centralized DB (S3), Decentralized filesystem (e.g., IPFS)…
- Problem: What if storage place is erased?
  - Actual NFT is lost, only reference on-chain is kept

Case study: Ordinals/Inscriptions (NFT)

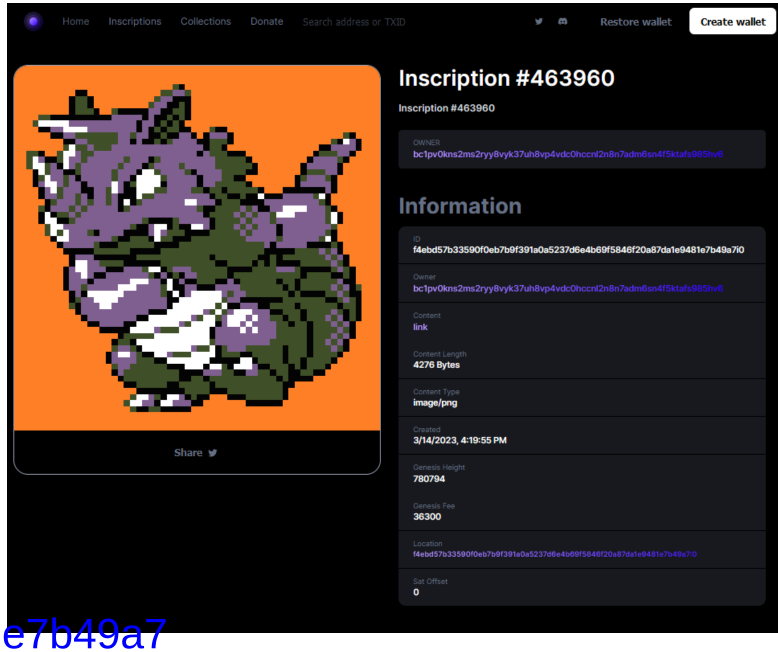
- NFT needs two principal components
  - Non-fungible (transferable) reference for NFT [Ordinals]
  - Storage place for actual NFT content (picture, movie, 3d model) [Inscriptions]
- Ordinals (https://docs.ordinals.com/overview.html)
  - Virtual unambiguous numbering scheme for every **`satoshi`** mined so far
    - xth **`satoshi`** mined, keeps its number
  - When UTXO is spent, all its satoshies (already numbered) are distributed on ordered basis (FIFO, first-satoshi-in-first-satoshi-out)
  - Important: no “serial number” is put on blockchain, everything is just virtual overlay
  - https://github.com/casey/ord/blob/master/bip.mediawiki

Case study: Ordinals/Inscriptions (NFT)



```
OP_FALSE
OP_IF
  OP_PUSH "ord"
  OP_1
  OP_PUSH "text/plain;charset=utf-8"
  OP_0
  OP_PUSH "Hello, world!"
OP_ENDIF
```

- Inscriptions (https://ordinals.com/inscriptions)
  - – Requires Taproot (P2TR address), first tx spends sats, second reveals script
  - – Embedding of data into witness script in non-spendable path (OP_FALSE)
  - – Inscription is on first sat of first output
  - – Ownership can be transferred to other person (ordinals)
- Transaction fee needs to be paid
  - – Data are in discounted Segwit bytes (¼ price)
  - – But inscriptions are typically significantly larger than tx
    - Ordinary tx is 100-200 sats, data 1024 sats / kB
  - – Significant number of transactions in Jan-March 2023



Inscription #463960

Inscription #463960

OWNER
bc1pv0kns2ms2ryy8vyk37uh8vp4vdc0hccnl2n8n7adm6sn4f5ktafs985hv6

Information

ID
f4ebd57b33590f0eb7b9f391a0a5237d6e4b69f5846f20a87da1e9481e7b49a7i0

Owner
bc1pv0kns2ms2ryy8vyk37uh8vp4vdc0hccnl2n8n7adm6sn4f5ktafs985hv6

Content
link

Content Length
4276 Bytes

Content Type
image/png

Created
3/14/2023, 4:19:55 PM

Genesis Height
780794

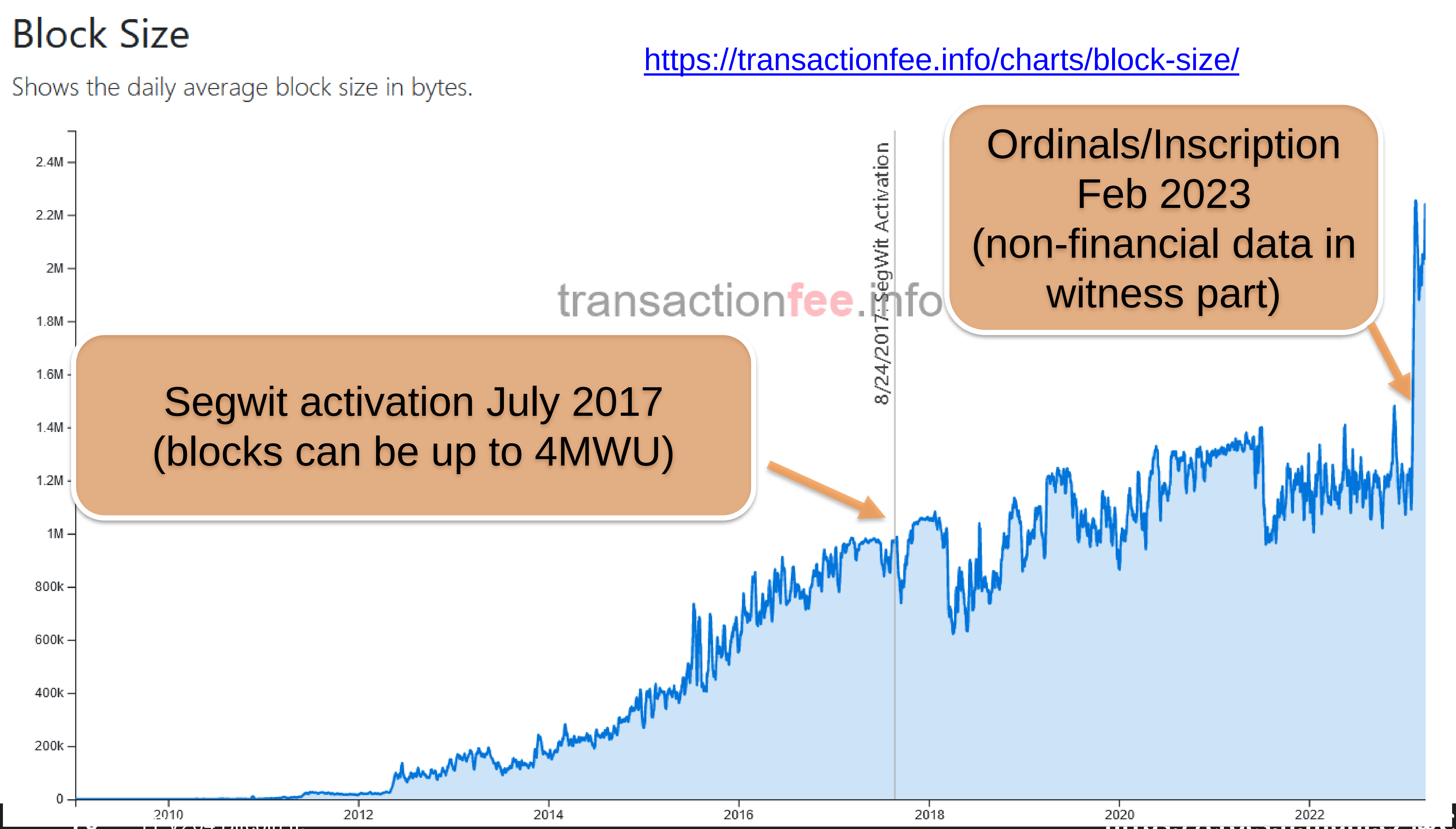
Genesis Fee
36300

Location
f4ebd57b33590f0eb7b9f391a0a5237d6e4b69f5846f20a87da1e9481e7b49a7:0

Sat Offset
0

https://mempool.space/tx/f4ebd57b33590f0eb7b9f391a0a5237d6e4b69f5846f20a87da1e9481e7b49a7

https://ordinals.com/inscription/f4ebd57b33590f0eb7b9f391a0a5237d6e4b69f5846f20a87da1e9481e7b49a7i0



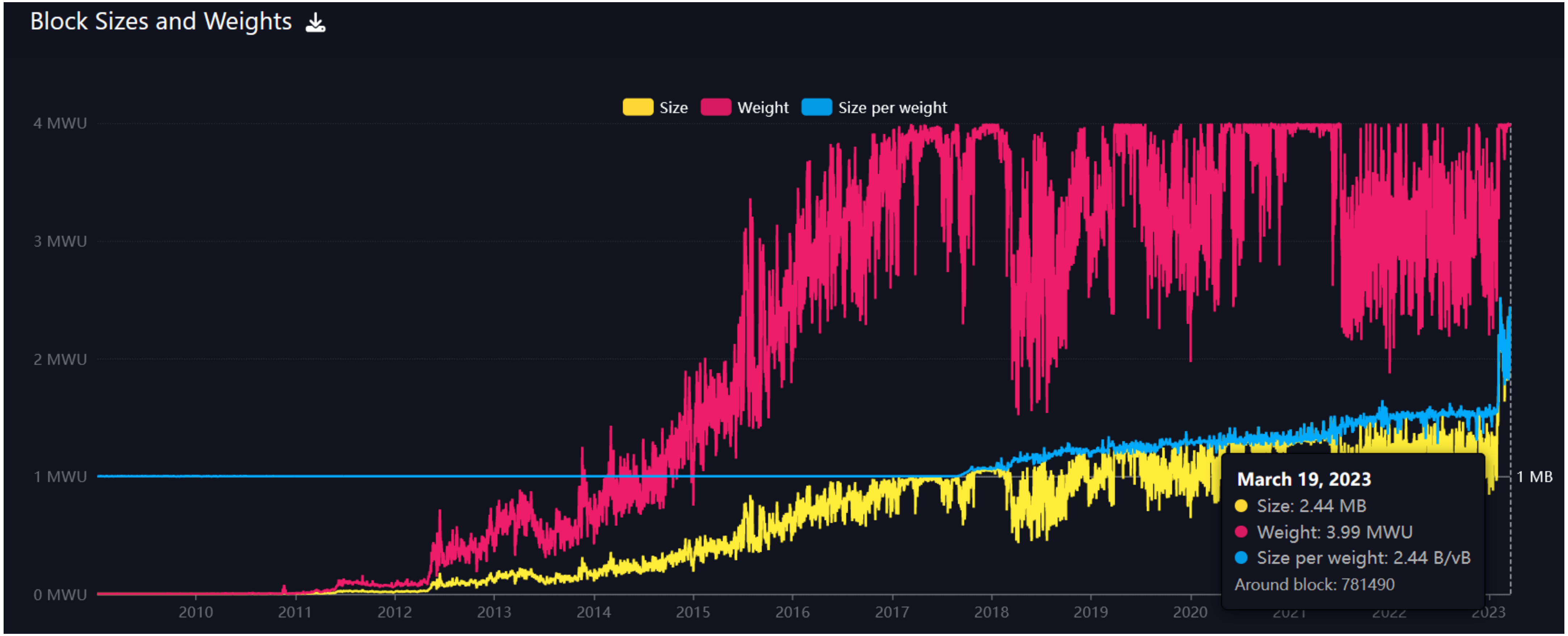
Block Size

Shows the daily average block size in bytes.

https://transactionfee.info/charts/block-size/

Ordinals/Inscription Feb 2023 (non-financial data in witness part)

Segwit activation July 2017 (blocks can be up to 4MWU)


Block Sizes and Weights
Size
Weight
Size per weight
4 MWU
3 MWU
2 MWU
1 MWU
0 MWU
1 MB
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
March 19, 2023
Size: 2.44 MB
Weight: 3.99 MWU
Size per weight: 2.44 B/vB
Around block: 781490

Case study: Ordinals/Inscriptions discussion

- Inscriptions are controversial and discussed (March 2023)
- What do you think?

Case study: Ordinals/Inscriptions discussion

- Inscriptions are controversial and discussed (March 2023)
- Discussion points (Do your own research!)
  - Blockspace used for non-financial data, needs to be downloaded/stored by all
    - Segwit part, only download, fast verification (OP_FALSE), no UTXO set bloat
  - Legal implications (3d printed guns, child abuse material…)
  - NFT getting discount price, spam
    - Segwit data bytes are discounted, but ordinary tx is significantly more dense => shall outprice Inscriptions
  - Pricing out people wanting to do on-chain transactions for small value
    - Mainnet not meant for small tx longer (Lighting, sidechains)
    - Is now increasing rewards for miners => more blockchain security
  - Impacting fungibility of Bitcoin, push for more smaller transactions (UTXO set)

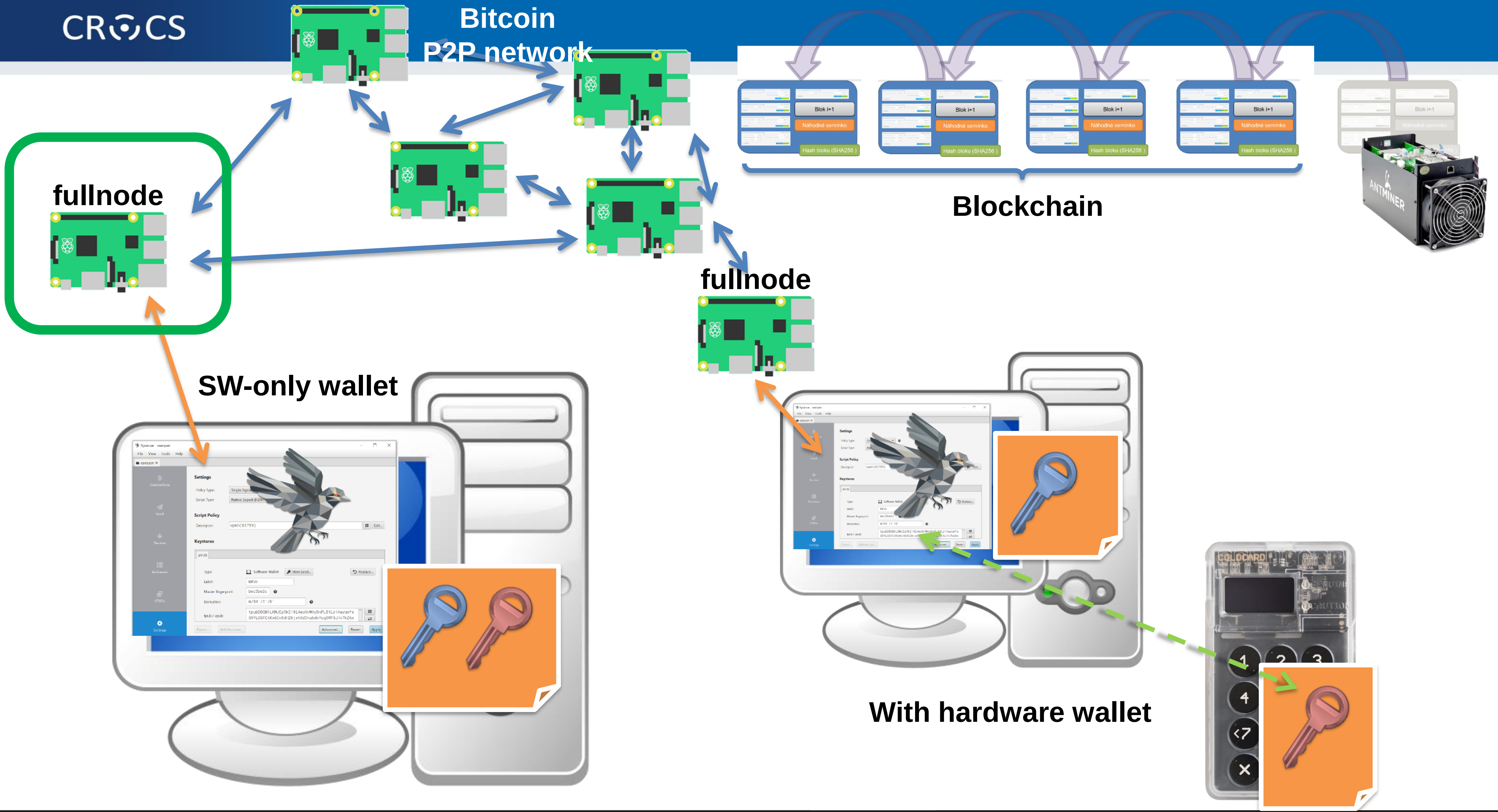
Q&A

slido

Audience Q&A Session

ⓘ Start presenting to display the audience questions on this slide.

RUNNING OWN FULL NODE



Bitcoin P2P network

Blockchain

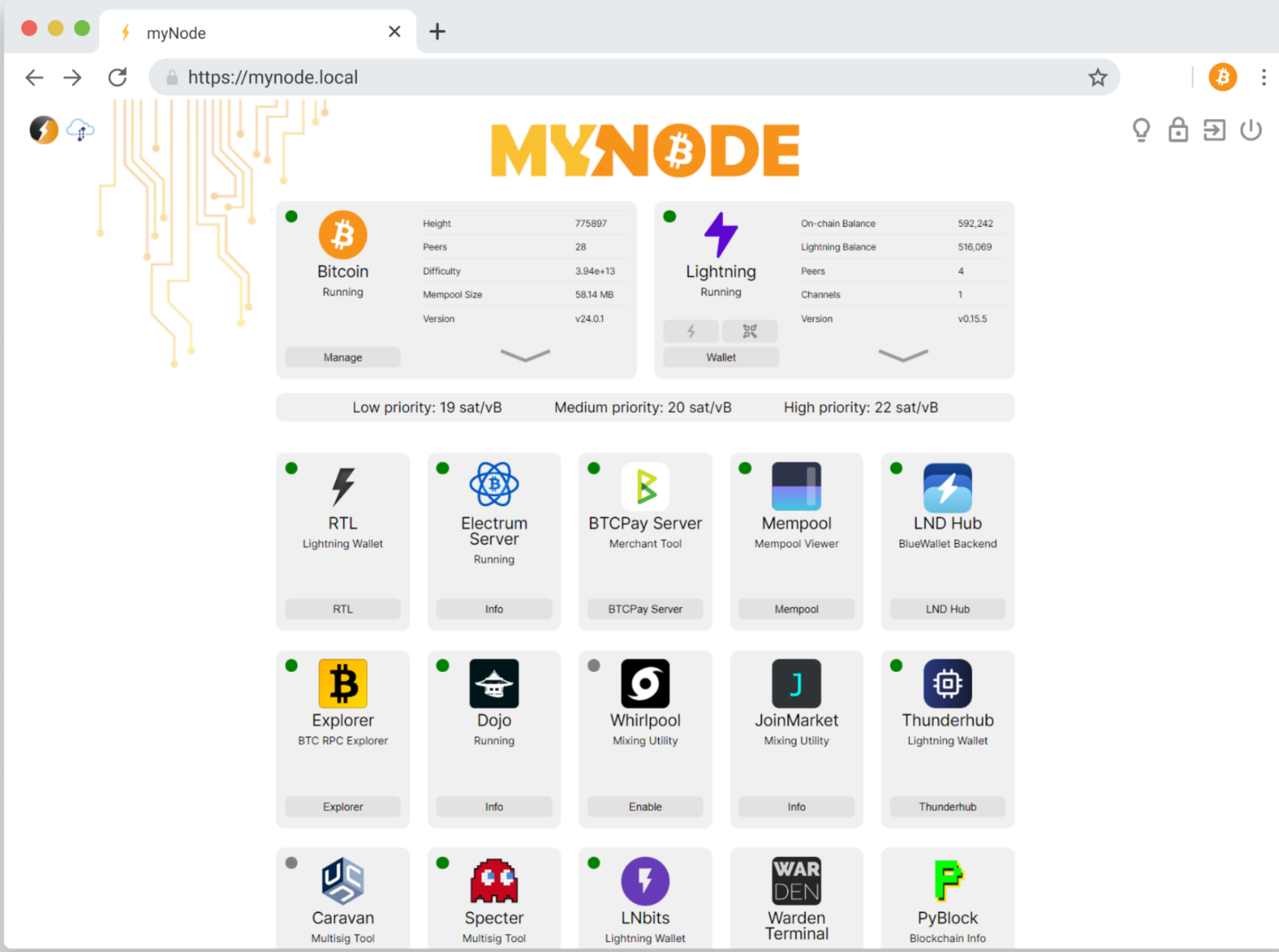
fullnode

fullnode

SW-only wallet

With hardware wallet

Running own full node



myNode

https://mynode.local

MYNODE

Bitcoin
Running

Height	775897
Peers	28
Difficulty	3.94e+13
Mempool Size	58.14 MB
Version	v24.0.1

Manage

Lightning
Running

On-chain Balance	592,242
Lightning Balance	516,069
Peers	4
Channels	1
Version	v0.15.5

Wallet

Low priority: 19 sat/vB Medium priority: 20 sat/vB High priority: 22 sat/vB

RTL
Lightning Wallet
RTL

Electrum Server
Running
Info

BTCPay Server
Merchant Tool
BTCPay Server

Mempool
Mempool Viewer
Mempool

LND Hub
BlueWallet Backend
LND Hub

Explorer
BTC RPC Explorer
Explorer

Dojo
Running
Info

Whirlpool
Mixing Utility
Enable

JoinMarket
Mixing Utility
Info

Thunderhub
Lightning Wallet
Thunderhub

Caravan
Multisig Tool

Specter
Multisig Tool

LNbits
Lightning Wallet

Warden Terminal

PyBlock
Blockchain Info

Additional software to run on “full node”

- Bitcoin Core basic client (bitcoind, bitcoin-qt)
  - Many additional software packages possible
- Lighting network software (LND, c-Lighting, Eclaire, RTL, LNbits…)
- Payment servers (BTCPay server)
- Blockchain explorers / indexers (Electrum, mempool.space, Explorer…)
- CoinJoin clients (Whirpool, JoinMarket…)
- Multisignature coordinators (DoJo, Specter, CKBunker…)
- Pre-prepared fullnode distributions (software above included)
  - MyNode, Umbrel, RaspiBlitz…

Summary

WHERE SHOULD WE FOCUS THIS YEAR?
"BLOCKCHAIN"
IT WILL CHANGE EVERYTHING.
EVERYBODY IS TALKING ABOUT IT.
THE POTENTIAL APPLICATIONS ARE ENDLESS.
WE DON'T WANT TO BE LEFT BEHIND.
WHAT EXACTLY IS BLOCKCHAIN?
ALSO, "ARTIFICIAL INTELLIGENCE"
TOM FISH BURNE
© marketoonist.com

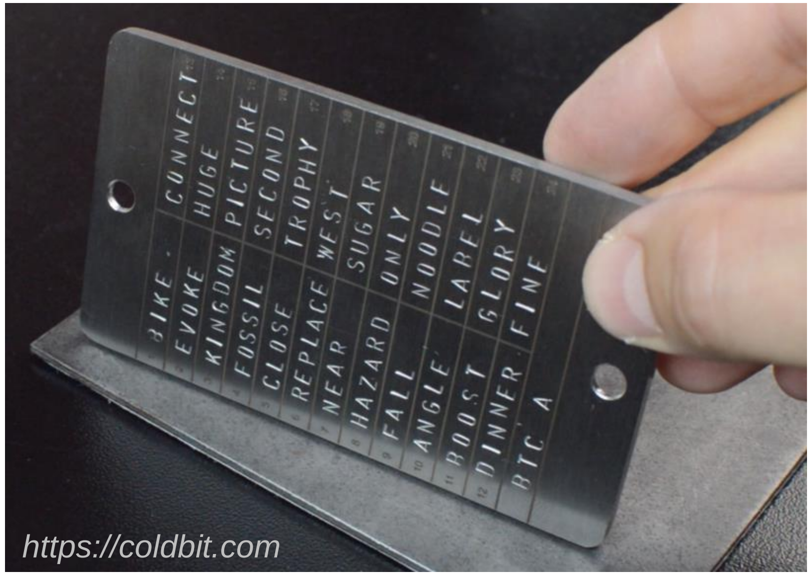
WALLET TOPICS

Generating new "wallet"

- A "wallet" is key management software controlling your private and public keys (ECDSA, Schnorr)
- The most important part of wallet is random number called root seed (128 or 256 bits)
- Root seed is used to deterministically generate practically unlimited number of keypairs
  - Specified in BIP32, "root seed" and "derivation path" used to derive next private key => next public key => next address
- Clever construction allowing to compute future public keys (and only public keys) for specified derivation path without the need for root seed (aka xpub or extended public key)
  - Knowledge of xpub allows to compute all future public keys, but not private keys
  - Owner of root seed can compute all future private keys and their corresponding public keys
  - xpub allows to pay someone to fresh addresses noninteractively (no interaction with owner of root seed required), receiver will only later compute candidate private keys and their public keys to check for total balance (== set of UTXOs)
- Wallet software is monitoring blockchain for addresses corresponding to stored root seed (or xpub)
- Root seed can be stored:
  1. Directly in software wallet (file on harddisk, optionally encrypted) == aka hot wallet, least secure against malware
  2. Loaded every time before use (e.g., from QR code), still vulnerable to malware during use
  3. On external hardware signing device called hardware wallet (the most secure option)

Backing up entropy (“master seed”)

- 128 or 256 bits of entropy (12 or 24 words)
- How to store securely?
  - Write on paper, punch into metal plate, carve into stone…
  - How to prevent human typing error (bits → mnemonics, BIP39)
  - Do not write somewhere digitally (malware may steal it)
- How to prevent single point of failure?
  - Make two copies (=> more robust against accidental loss)
  - Make (threshold) parts Shamir (=> more robust against intentional theft and loss - threshold)
  - Require multiple signatures (multisig, MPC)

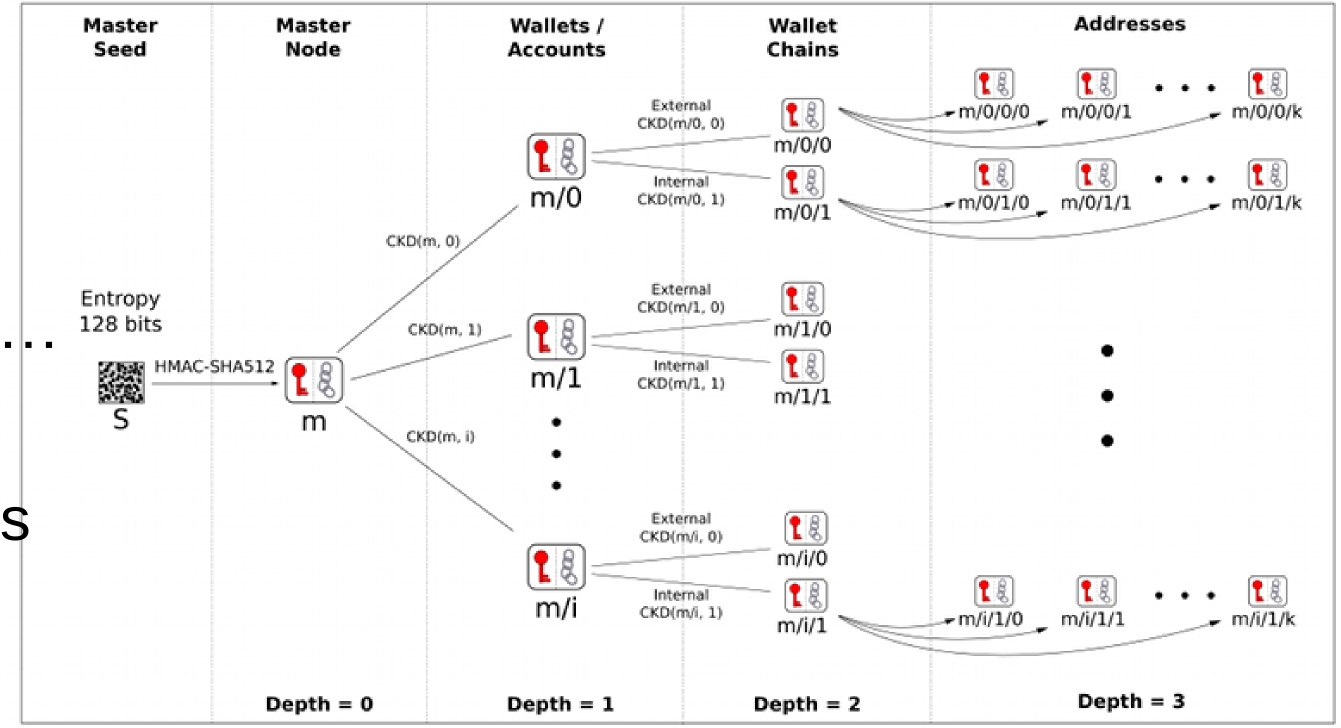


https://coldbit.com

Making fresh private keys (with backup) BIP32, BIP44…

- Deterministic derivation from:
  - master seed (key)
  - derivation path (data)
    - m/purpose/coin/account/receive…
- Single master seed allows:
  - Generate many distinct private keys
  - Sharing sub-tree value allows:
    - Generate keys in sub-trees
    - Cannot generate keys from other trees
- Deterministic generation, Master Seed enough to recover whole tree

BIP 32 - Hierarchical Deterministic Wallets



Child Key Derivation Function ~ CKD(x,n) = HMAC-SHA512(x_{Chain} , x_{PubKey} || n)

Receiving (testnet) bitcoins

- You generate new "address"
  - deterministically derived from your root seed and fresh derivation path (path + counter) => new ECDSA keypair [BIP32]
  - public key X is pasted into locking script ("who can sign with private key verifiable with X can move bitcoin further") and hashed => "address" [P2SH/P2WSH] (Pay to witness script hash)
- Service coinfaucet.eu owns multiple tBTC
  - Service is providing limited number of test bitcoins (tBTC) for free
  - Service owns UTXOs => someone previously locked some tBTC to their keypair(s)
  - Service creates new transaction with some tBTC locked to your "address"
  - New transaction is broadcasted to Bitcoin P2P network and stored in mempools (set of unconfirmed transactions)
- Miners will eventually include this transaction into new block (head of blockchain)
  - Confirmed and removed from mempools
  - Your Sparrow wallet is monitoring both mempool and blockchain (instant notification about pending transaction)

Blockchain explorers

- Everybody with access to Bitcoin P2P network can analyze blockchain
  - Everybody running Bitcoin fullnode
  - All past transactions, human-readable visualizations, search for address…
  - Convenient quick check of funds send
- Third parties are operating public explorers (convenient, but privacy risk)
  - It is very important to use Tor Browser when accessing public block explorers
  - Explorer operator may log your IP address and transactions you are searching for and later sell it (chain surveillance companies)
    - Heuristic assumption that you are the owner of funds for searched transaction
- Ideally use your own full node with your own blockchain explorer
- Sparrow wallet allows you to visualize your transactions
  - Inputs, outputs, feed paid