Pettycoin: Towards 1.0?

Rusty Russell
rusty@rustcorp.com.au

Contents

- Pettycoin Background
- Massive Detour
  - *Contains Caveats and Notes!*
- Pettycoin v2?

Pettycoin

- Mining cost places lower limit on transaction fees
  - Help cut Gordian knot for bitcoin miners

Pettycoin

- Mining cost places lower limit on transaction fees
  - Help cut Gordian knot for bitcoin miners
- Fun project...

Pettycoin

- Mining cost places lower limit on transaction fees
  - Help cut Gordian knot for miners
- Fun project...

Sabbatical

Sabbatical

- 6 months off

Sabbatical

- 6 months off
  - 1 month vacation

Sabbatical

- 6 months off
  - 1 month vacation
  - 1/day week Marcus

Pettycoin Characteristics
http://pettycoin.org

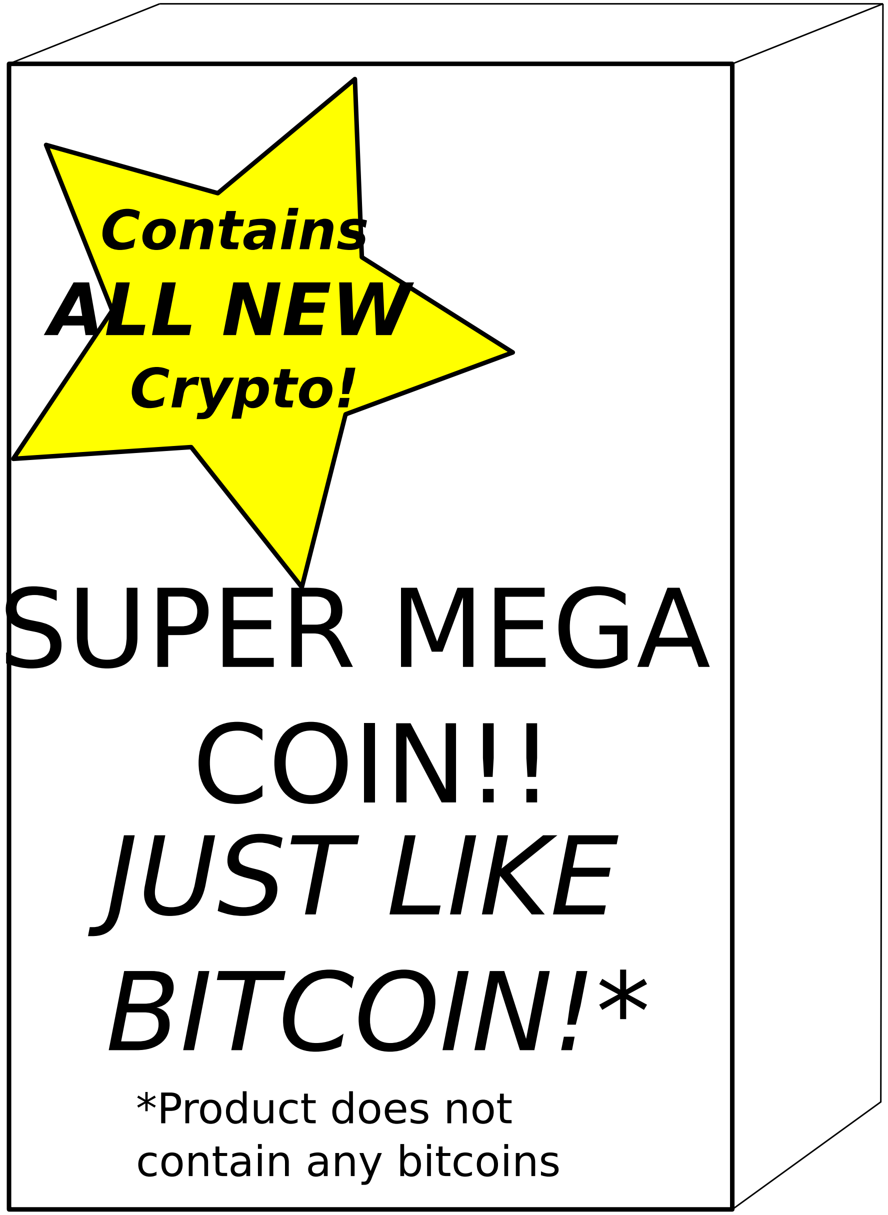
- Functionaries gateway ↔ Bitcoin network
- Limited to small amounts
- Simpler transactions
- Horizon
- Partial Knowledge
- Payback
- Fast block times

Aside: A Weird F/OSS Project

- Altcoins

Aside: A Weird F/OSS Project

- Altcoins



Aside: A Weird F/OSS Project

NOISE

Aside: A Weird F/OSS Project

2,289,384 Announcements (Altcoins)

NOISE

Aside: A Weird F/OSS Project

2,304,695 Announcements (Altcoins)

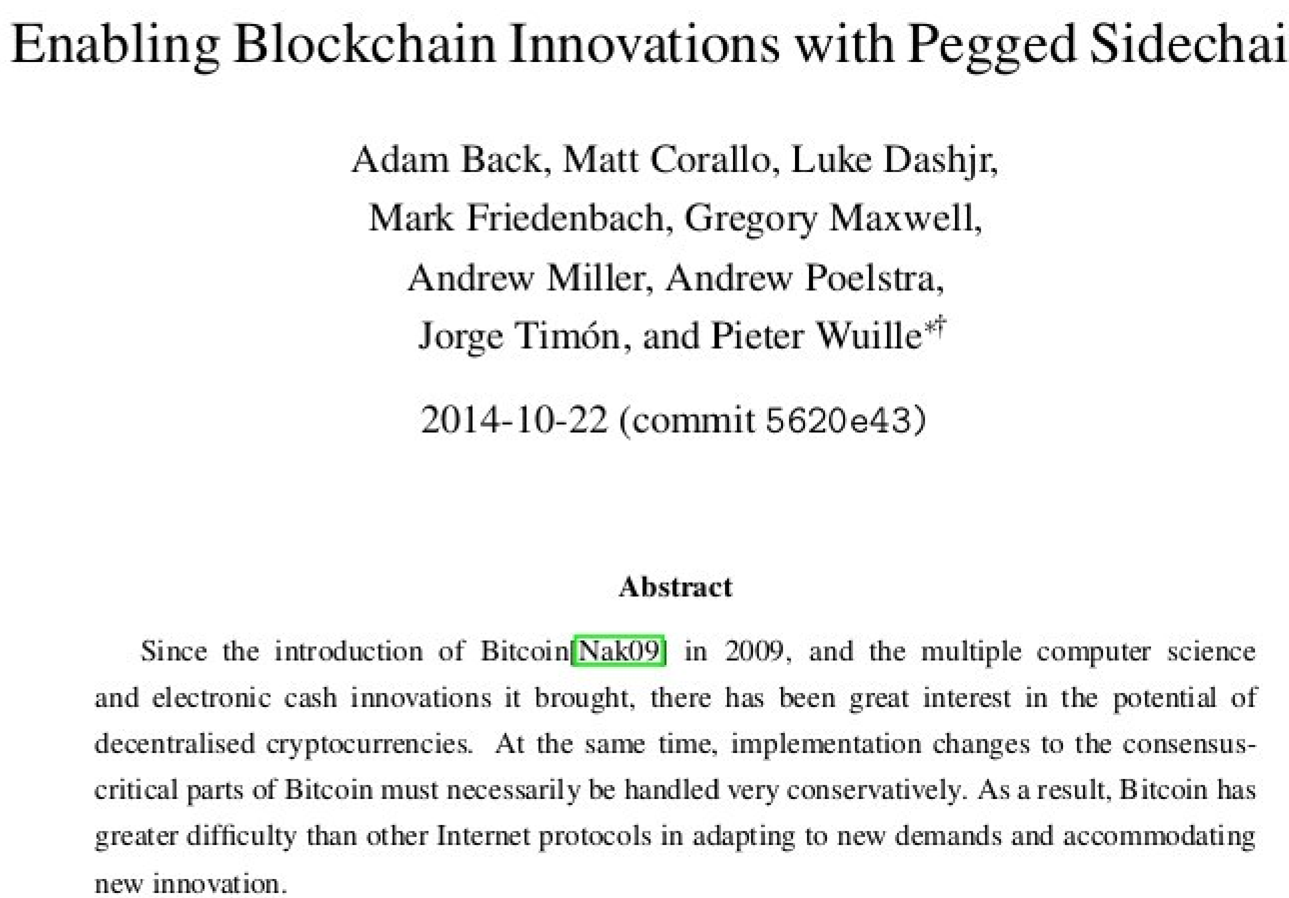
NOISE

- Hard to reach/find people genuinely interested in innovative ideas.

Meanwhile...

Sidechains

- http://blockstream.com/sidechains.pdf



Enabling Blockchain Innovations with Pegged Sidechains

Adam Back, Matt Corallo, Luke Dashjr,
Mark Friedenbach, Gregory Maxwell,
Andrew Miller, Andrew Poelstra,
Jorge Timón, and Pieter Wuille[*†]

2014-10-22 (commit 5620e43)

Abstract

Since the introduction of Bitcoin[Nak09] in 2009, and the multiple computer science and electronic cash innovations it brought, there has been great interest in the potential of decentralised cryptocurrencies. At the same time, implementation changes to the consensus-critical parts of Bitcoin must necessarily be handled very conservatively. As a result, Bitcoin has greater difficulty than other Internet protocols in adapting to new demands and accommodating new innovation.

What I Should Have Done...

What I Should Have Done...

DETOUR
AHEAD

What I Should Have Done...

- Bitcoin Basics
- How Sidechains Work
- Other Partial Knowledge Ideas

Bitcoin Basics

- Cryptographic hash functions
- Bitcoin blocks
- Bitcoin transactions

Cryptographic Hash Functions

- Cryptographic hash functions
  - Hash takes some data, produces number

- “Hi Rusty!” => 113,874,859,391,549,611,678,918,264,699,517,411,490,566,824,306,315,592,
  823,661,988,754,055,674,729,523 <= 78 digits

Cryptographic Hash Functions

- Cryptographic hash functions
  - Hash takes some data, produces number
  - No two things hash to the same value

- “Hi Rusty!” => 113,874,859,391,549,611,678,918,264,699,517,411,490,566,824,306,315,592,
  823,661,988,754,055,674,729,523
- “hi Rusty!” => 50,389,223,465,001,933,639,819,032,401,253,318,319,916,409,888,064,665,
  201,997,103,129,362,843,385,322

Cryptographic Hash Functions

- Cryptographic hash functions
  - Hash takes some data, produces number
  - No two things hash to the same value
  - No way to guess what data was except trying everything

Caveats & Notes I

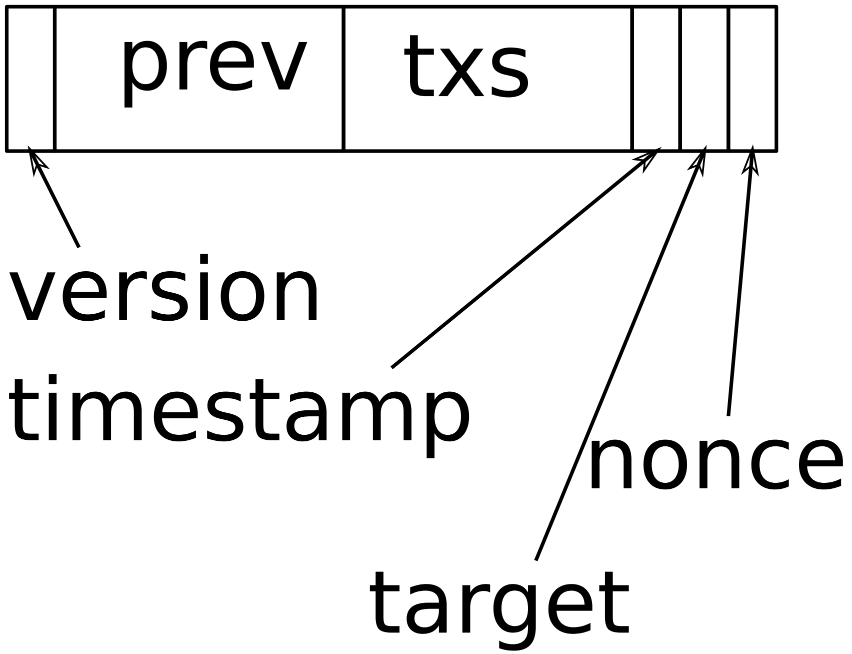
- I used SHA256. Bitcoin uses double-SHA256.
- I know “no two things hash to the same value” is impossible.
- And I know there exists no mathematical proof that it's even hard.
  - There may be an efficient way to produce duplicate hashes or calculate the reverse hash.

Bitcoin Basics

- Cryptographic hash functions √
- Bitcoin blocks
- Bitcoin transactions

Bitcoin Blocks

Bitcoin Blocks



prev

txs

version

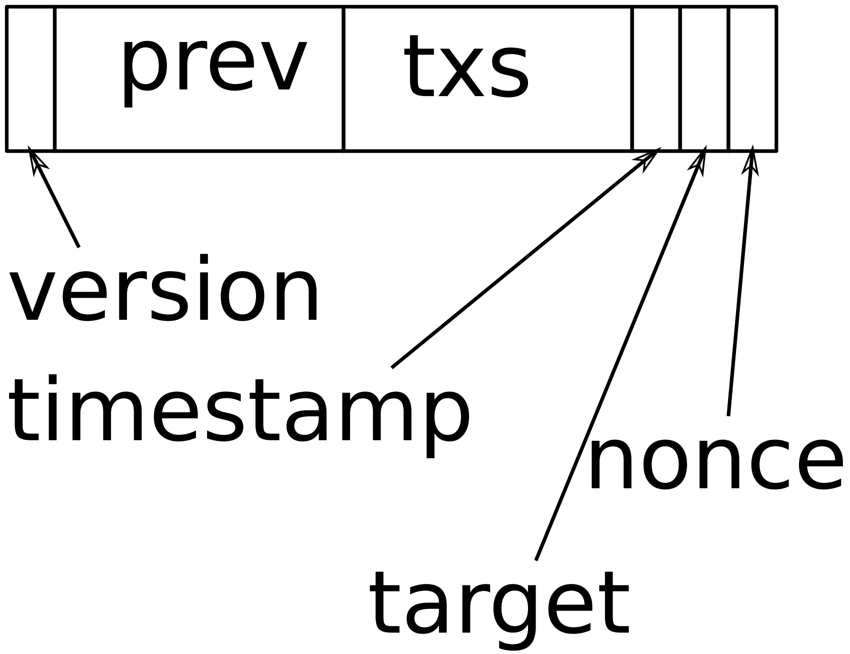
timestamp

nonce

target

Bitcoin Blocks

- Bitcoin transactions are gathered into blocks



prev

txs

version

timestamp

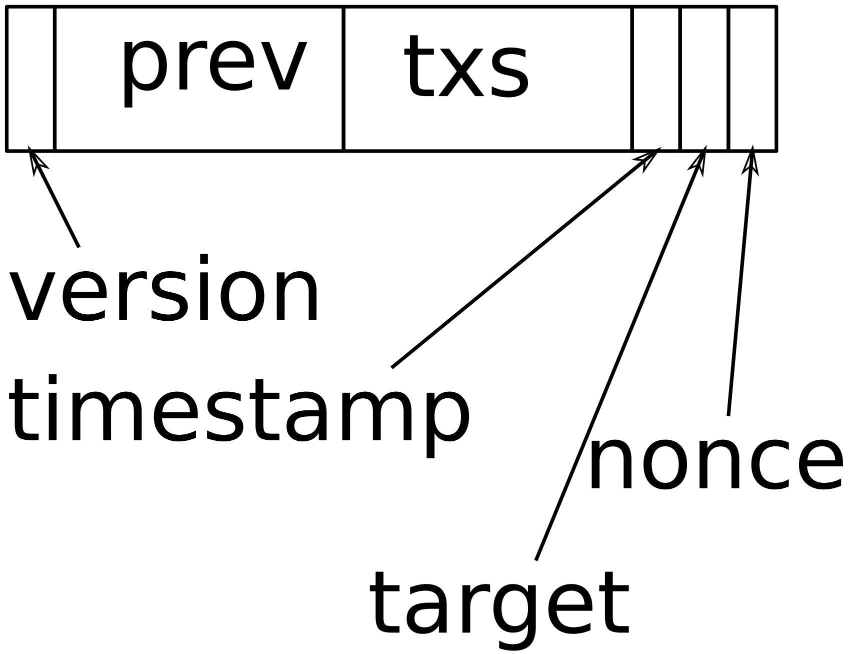
nonce

target

DETOUR

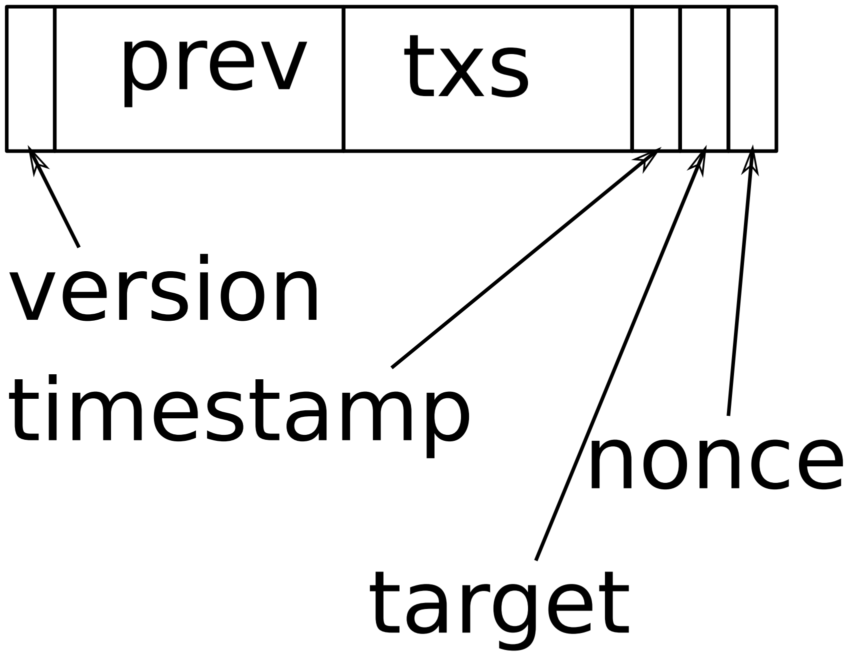
Bitcoin Blocks

- Bitcoin transactions are gathered into blocks
- Each block refers to the last one, forming a chain.



Bitcoin Blocks

- Bitcoin transactions are gathered into blocks
- Each block refers to the last one, forming a chain.
- Blocks are really hard to generate.

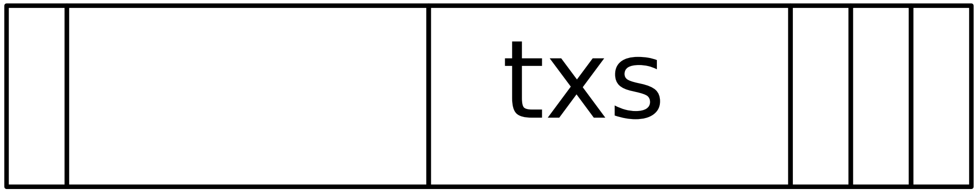


Bitcoin Basics

- Cryptographic hash functions √
- Bitcoin blocks √
- Bitcoin transactions

Bitcoin Background

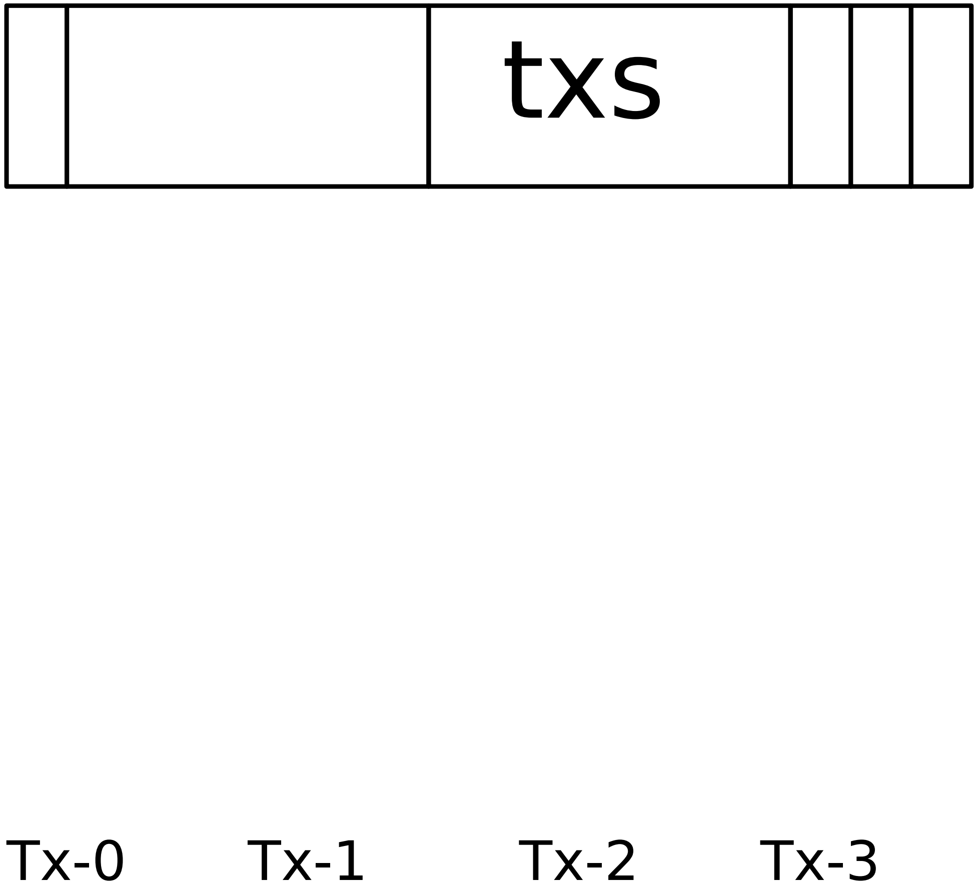
- Transactions form a tree, with root in the block header:



txs

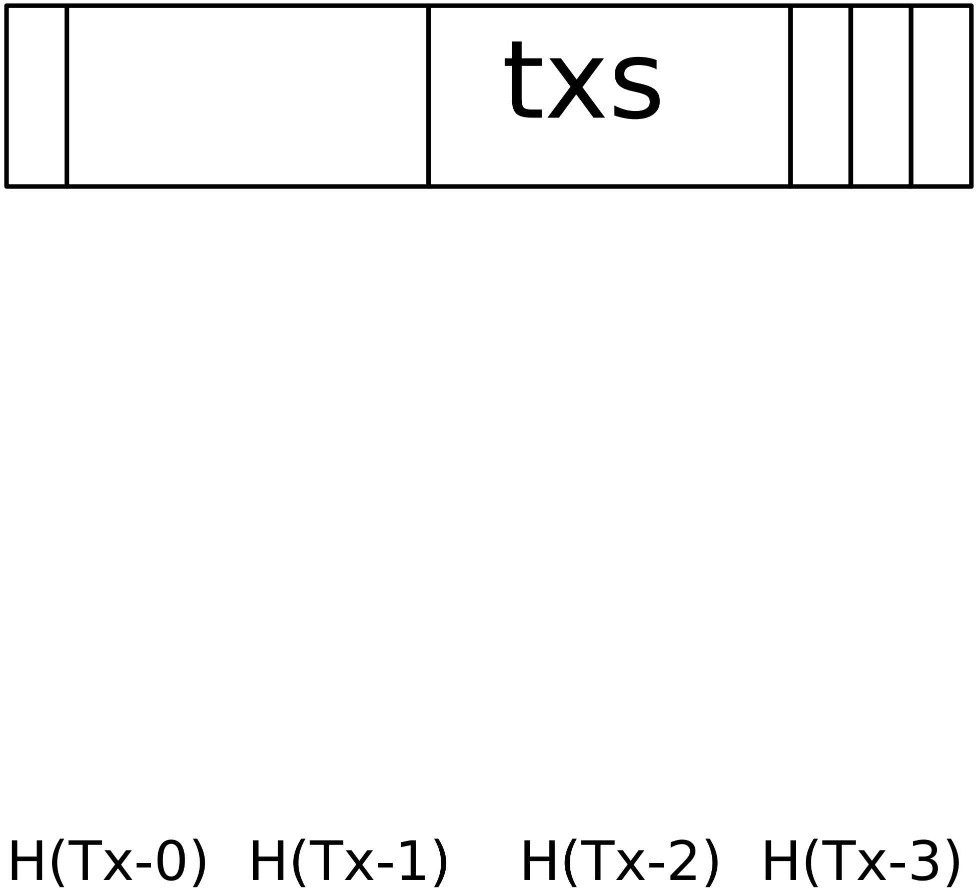
Bitcoin Background

- Transactions form a tree, with root in the block header:



Bitcoin Background

- Transactions form a tree, with root in the block header:

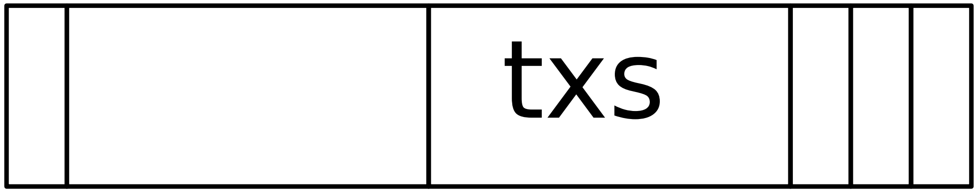


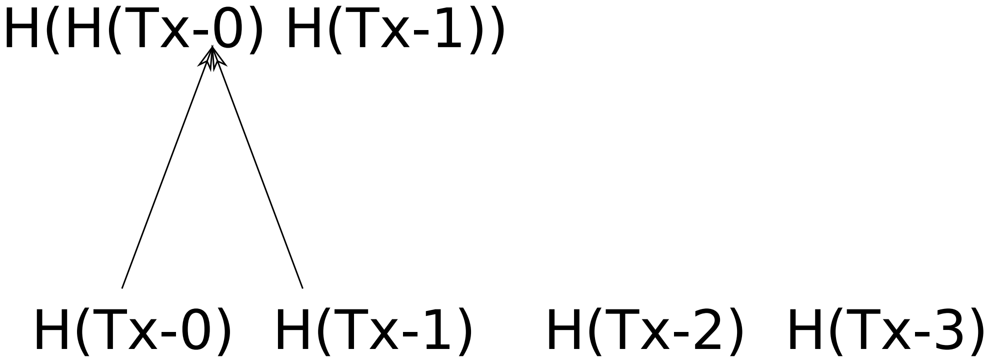
txs

H(Tx-0) H(Tx-1) H(Tx-2) H(Tx-3)

Bitcoin Background

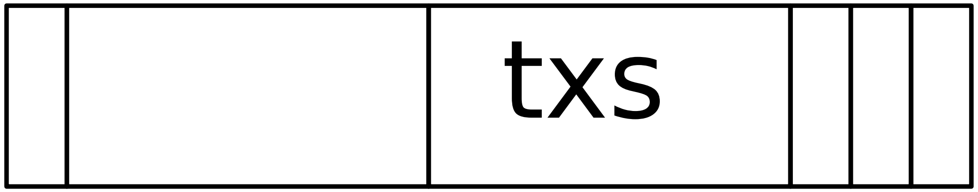
- Transactions form a tree, with root in the block header:



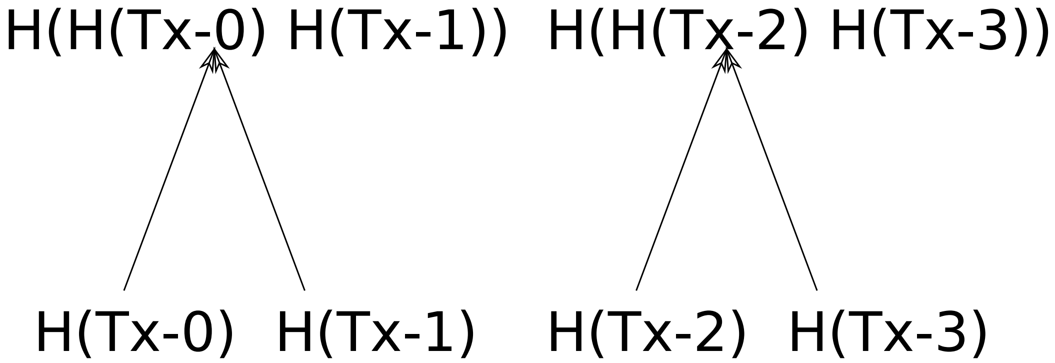


Bitcoin Background

- Transactions form a tree, with root in the block header:



txs

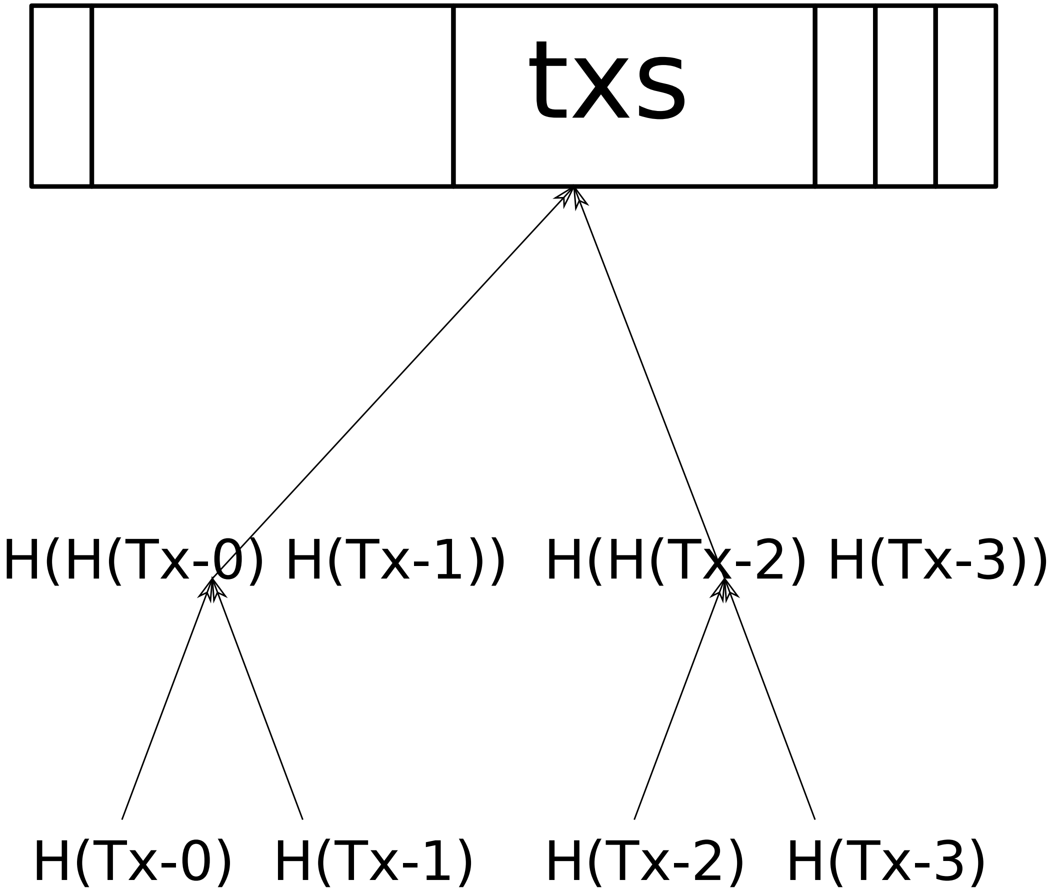


H(H(Tx-0) H(Tx-1)) H(H(Tx-2) H(Tx-3))

H(Tx-0) H(Tx-1) H(Tx-2) H(Tx-3)

Bitcoin Background

- Transactions form a tree, with root in the block header:



Merkel Tree

DETOUR

http://commons.wikimedia.org/wiki/File:Angela_Merkel_August_2014.jpg
http://commons.wikimedia.org/wiki/User:CDU_Sachsen CC BY-SA 4.0

Merkle Tree

DETOUR

http://commons.wikimedia.org/wiki/File:Ralph_Merkle.png
david.orban - http://www.flickr.com/photos/davidorban/1347574959/ CC BY 2.0

Bitcoin Transactions

Bitcoin Transactions

- Every bitcoin transaction has inputs (TxIn) and outputs (TxOut)

DETOUR

Bitcoin Transactions

- Every bitcoin transaction has inputs (TxIn) and outputs (TxOut)
  - Value of inputs >= value of outputs.
  - Each output can only be spent once.

DETOUR

Bitcoin Transactions

- Every bitcoin transaction has inputs (TxIn) and outputs (TxOut)
  - Value of inputs >= value of outputs.
  - Each output can only be spent once.
  - First tx has 1 fake input, generates coins

DETOUR

Bitcoin Transactions

- Every bitcoin transaction has inputs (TxIn) and outputs (TxOut)
  - – Value of inputs >= value of outputs.
  - – Each output can only be spent once.
  - – First tx has 1 fake input, generates coins
- Outputs have amount and a script
  - – “30 bitcoins. For a transaction signed by Alice”

Bitcoin Transactions

- Every bitcoin transaction has inputs (TxIn) and outputs (TxOut)
  - Value of inputs >= value of outputs.
  - Each output can only be spent once.
  - First tx has 1 fake input, generates coins
- Outputs have amount and a script
  - “30 bitcoins. For a transaction signed by Alice”
- Inputs have a tx hash, output number, and script
  - “Spend output N of TX X, and I, Alice, endorse this transaction”

Bitcoin Transactions

- eg. Block 300,000:

Bitcoin Transactions

- eg. Block 300,000:
(Hash: 829,998,915,579,594,092,199,999,189,
296,919,999,871,189,997,254 => 48 digits)

Bitcoin Transactions

- eg. Block 300,000:

Bitcoin Transactions

- eg. Block 300,000:

  TX 0:
  9,399,969,399,996,839,989,456,721,927,078,
  696,279,992,467,008,883,159,918,770,249,983

Bitcoin Transactions

- eg. Block 300,000:

  TX 0:
  9,399,969,399,996,839,989,456,721,927,078,
  696,279,992,467,008,883,159,918,770,249,983

  Output #0 Amount 25.0402836 BTC

Bitcoin Transactions

- eg. Block 300,000:

  TX 0:
  9,399,969,399,996,839,989,456,721,927,078,
  696,279,992,467,008,883,159,918,770,249,983

  Output #0 Amount 25.0402836 BTC

  Script: OP_DUP OP_HASH160 8,099,909,403,
  581,993,994,608,699,192,999,412,599,691
  OP_EQUALVERIFY OP_CHECKSIG

Bitcoin Background

- Was redeemed in block 300,588 in TX 1577232...

Bitcoin Background

- Was redeemed in block 300,588 in TX 1577232...

  TxIn #37:

  Tx 9,399,969,399,996,839,989,456,721,927,078, 696,279,992,467,008,883,159,918,770,249,983
  TxOut #0

Bitcoin Background

- Input script:

```
OP_PUSH<71>
3044022001005794df903dbb984f3106587a1aa848
c5067dc424f45870da9574225e85d2022017b1db57
66d1878b5076374ded3a782c9ba4b555bf8311524b
896f57aea8140201

OP_PUSH<33>
02b8c918bd169a5e669cc149549f822dd5f2c50872
eb83172a1c69172277fe378f
```

DETOUR

Bitcoin Background

- Input script:

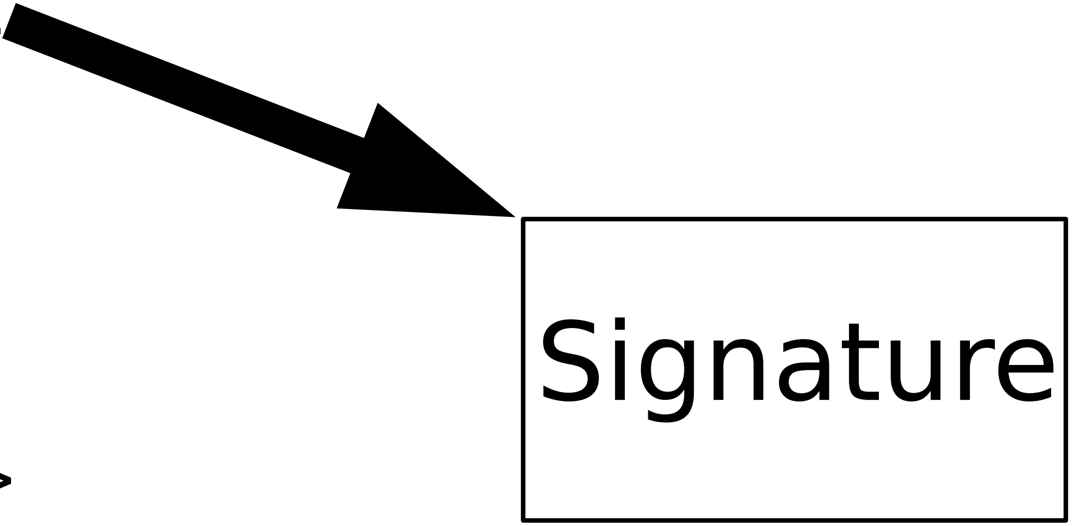
OP_PUSH<71>
<SIGNATURE>

OP_PUSH<33>
<PUBLIC KEY>

Bitcoin Background

- Input script:

OP_PUSH<71>
<SIGNATURE>



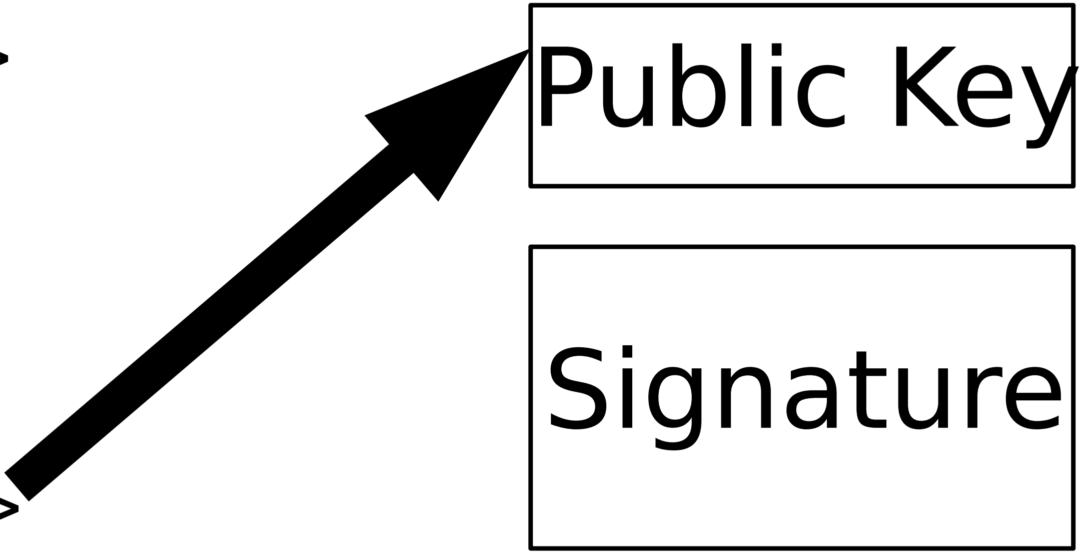
OP_PUSH<33>
<PUBLIC KEY>

Bitcoin Background

- Input script:

OP_PUSH<71>
<SIGNATURE>

OP_PUSH<33>
<PUBLIC KEY>



Bitcoin Background

Public Key

Signature

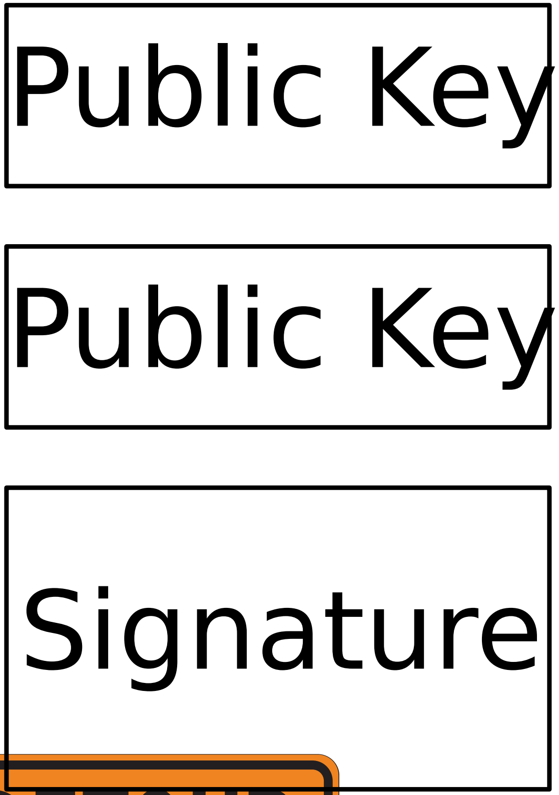
Bitcoin Background

Public Key

Signature

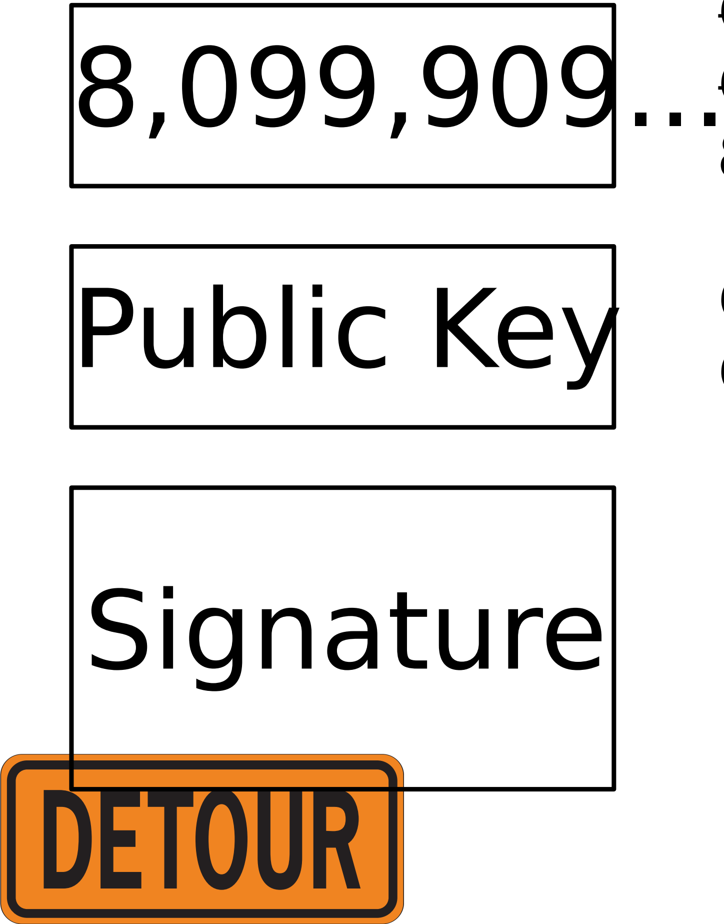
```
OP_DUP
OP_HASH160
8,099,909,403,581,993,994,608,699,192,999,
   412,599,691
OP_EQUALVERIFY
OP_CHECKSIG
```

Bitcoin Background



~~OP_DUP~~
OP_HASH160
8,099,909,403,581,993,994,608,699,192,999,
412,599,691
OP_EQUALVERIFY
OP_CHECKSIG

Bitcoin Background



~~OP_DUP~~
~~OP_HASH160~~
8,099,909,403,581,993,994,608,699,192,999,
412,599,691
OP_EQUALVERIFY
OP_CHECKSIG

Bitcoin Background

8,099,909...

8,099,909...

Public Key

Signature

~~OP_DUP~~
~~OP_HASH160~~
~~8,099,909,403,581,993,994,608,699,192,999,~~
~~412,599,691~~
OP_EQUALVERIFY
OP_CHECKSIG

Bitcoin Background

~~OP_DUP~~
~~OP_HASH160~~
~~8,099,909,403,581,993,994,608,699,192,999, 412,599,691~~
~~OP_EQUALVERIFY~~
OP_CHECKSIG

Public Key

Signature

Bitcoin Background

~~OP_DUP~~
~~OP_HASH160~~
~~8,099,909,403,581,993,994,608,699,192,999,~~
~~412,599,691~~
~~OP_EQUALVERIFY~~
~~OP_CHECKSIG~~

1

Bitcoin Background

~~OP_DUP~~
~~OP_HASH160~~
~~8,099,909,403,581,993,994,608,699,192,999, 412,599,691~~
~~OP_EQUALVERIFY~~
~~OP_CHECKSIG~~

✓

1

Caveats & Notes II

- Numbers being pushed on the stack are usually just printed; I made up OP_PUSH<> here to be explicit
- Input script is often called scriptSig
- Output script is often called scriptPubkey
- The “signature” actually has a byte appended which indicates what parts of the transaction it signed.
- The RIPEMD160 of a ECDSA secp256k1 public key is usually encoded for printing using bitcoin's base58 encoding method, and called a “bitcoin address”

Sidechains

Sidechains

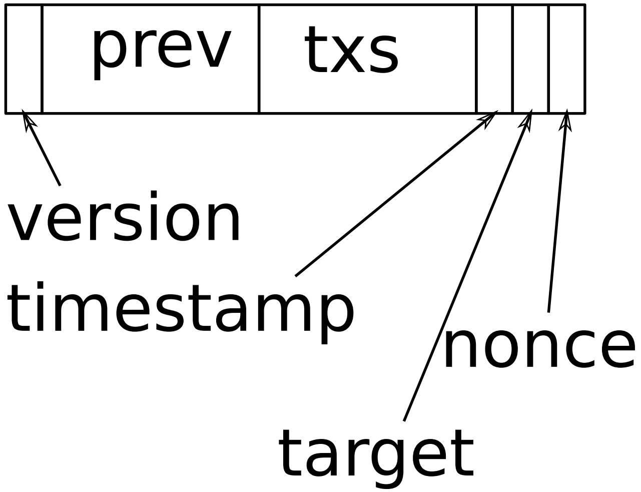
- Alternative chains which use real bitcoins
  - But may have different/experimental protocol rules

Sidechains: More Wasted Work?

Sidechains: More Wasted Work?

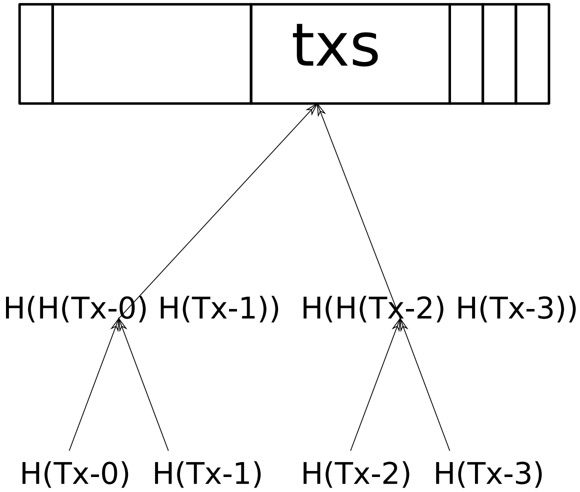
Sidechains: More Wasted Work?

- Bitcoin miners can mine other chains at the same time



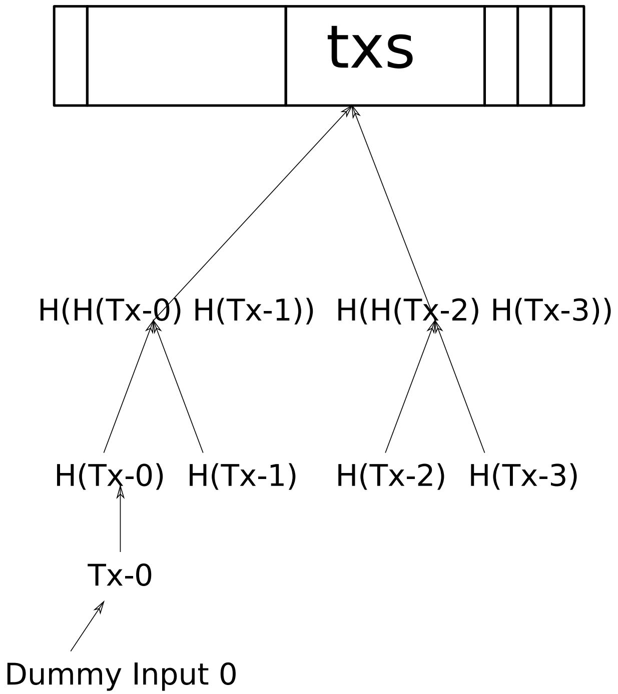
Sidechains: More Wasted Work?

- Bitcoin miners can mine other chains at the same time



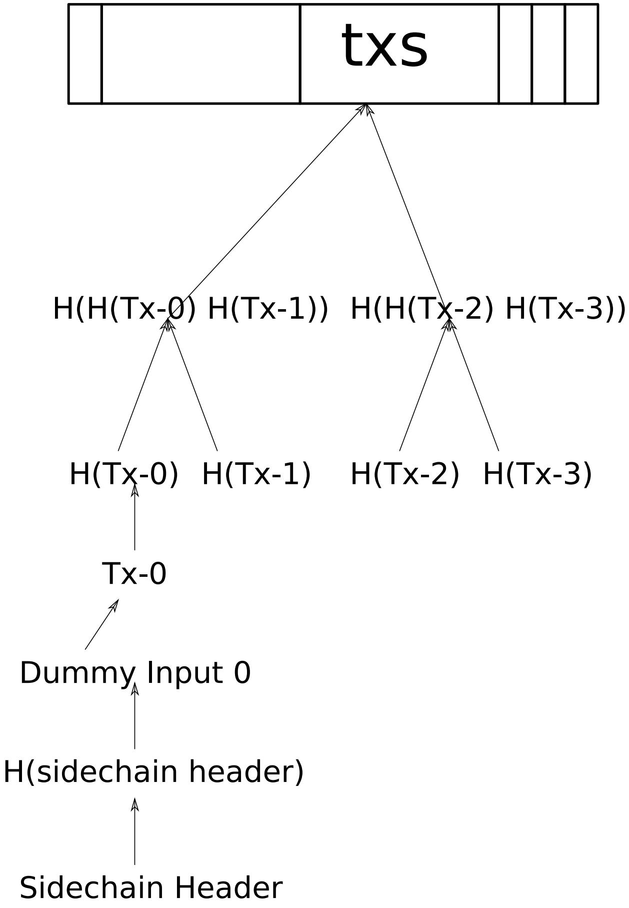
Sidechains: More Wasted Work?

- Bitcoin miners can mine other chains at the same time



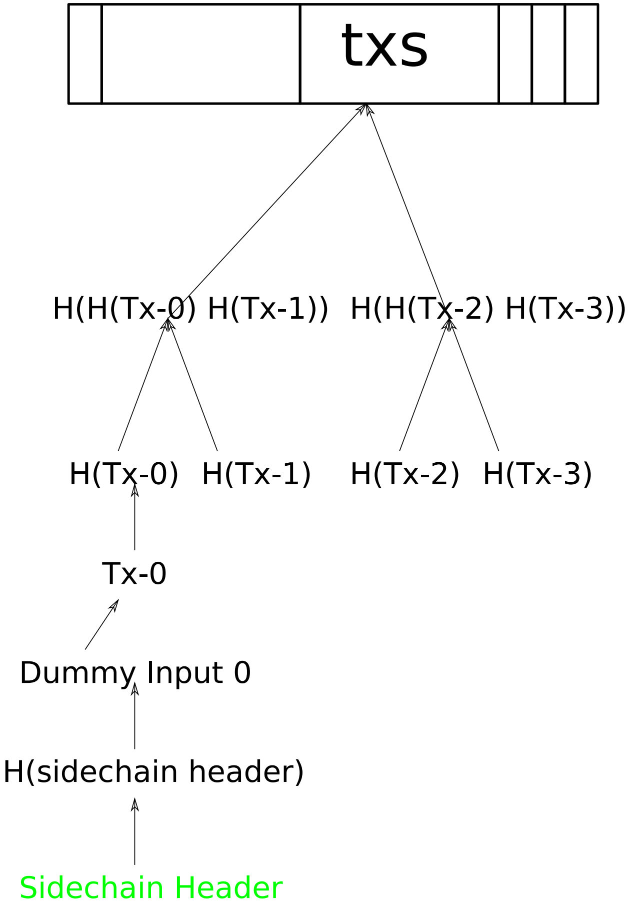
Sidechains: More Wasted Work?

- Bitcoin miners can mine other chains at the same time



Sidechains: More Wasted Work?

- Bitcoin miners can mine other chains at the same time



txs

H(H(Tx-0) H(Tx-1)) H(H(Tx-2) H(Tx-3))

H(Tx-0) H(Tx-1) H(Tx-2) H(Tx-3)

Tx-0

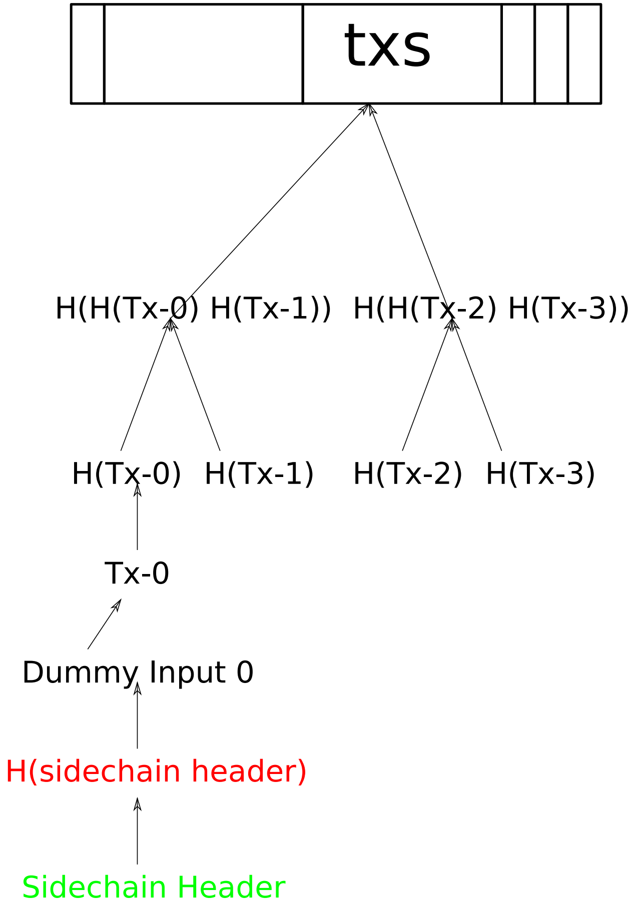
Dummy Input 0

H(sidechain header)

Sidechain Header

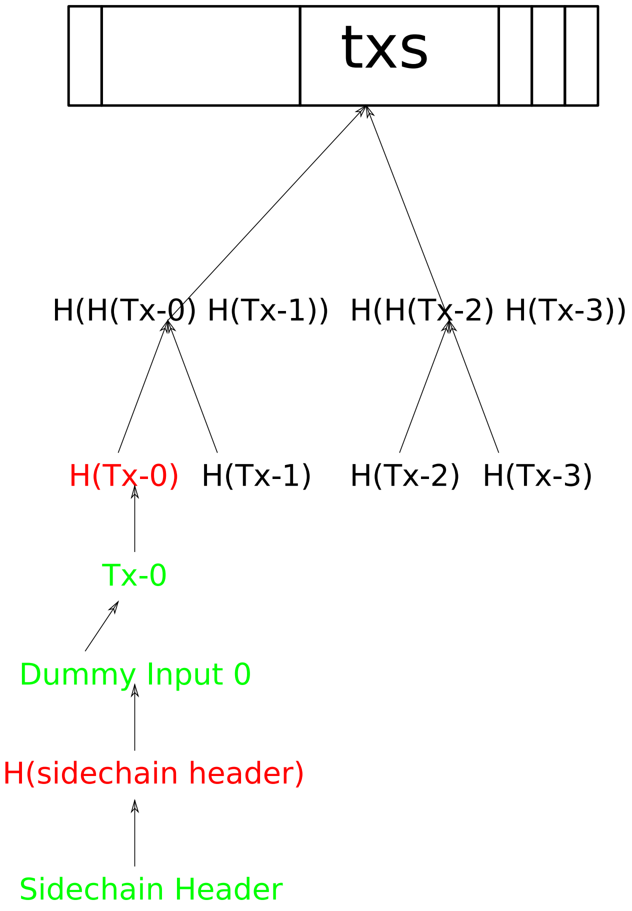
Sidechains: More Wasted Work?

- Bitcoin miners can mine other chains at the same time



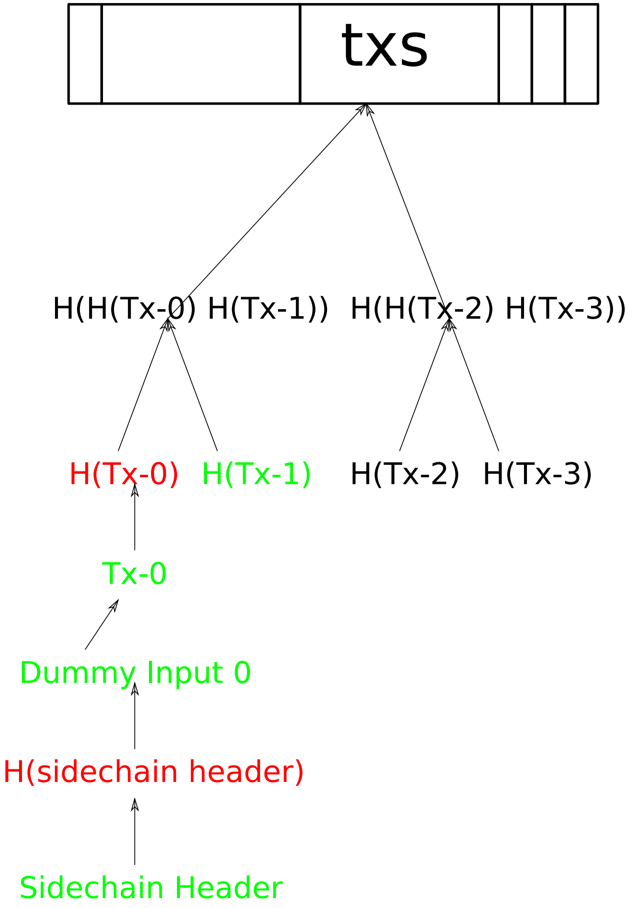
Sidechains: More Wasted Work?

- Bitcoin miners can mine other chains at the same time



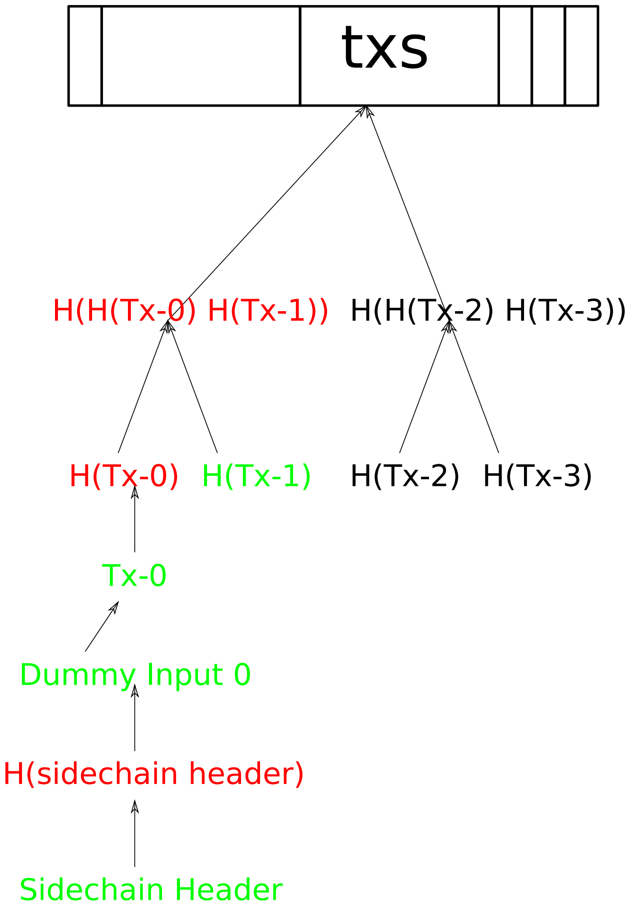
Sidechains: More Wasted Work?

- Bitcoin miners can mine other chains at the same time



Sidechains: More Wasted Work?

- Bitcoin miners can mine other chains at the same time



txs

H(H(Tx-0) H(Tx-1))

H(H(Tx-2) H(Tx-3))

H(Tx-0)

H(Tx-1)

H(Tx-2)

H(Tx-3)

Tx-0

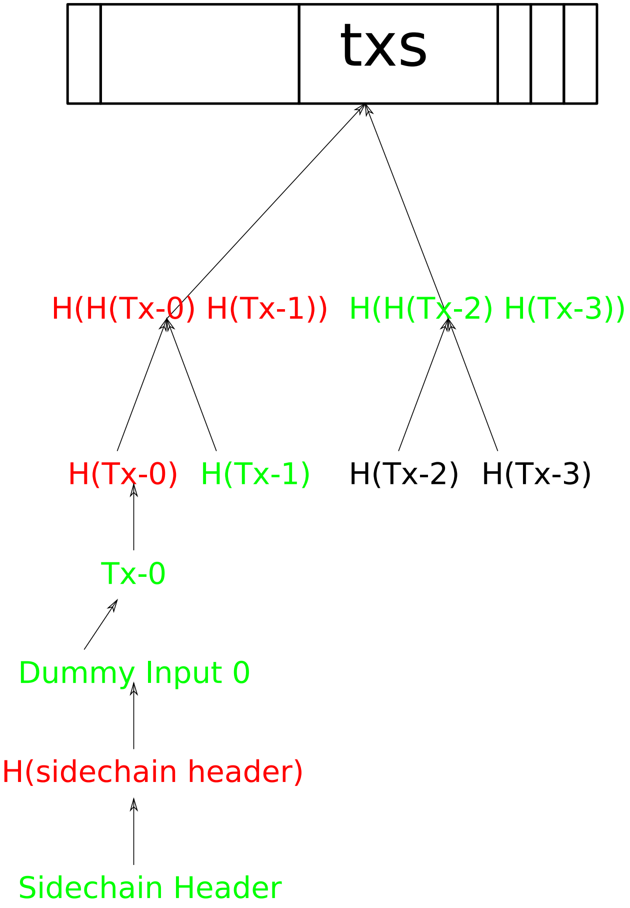
Dummy Input 0

H(sidechain header)

Sidechain Header

Sidechains: More Wasted Work?

- Bitcoin miners can mine other chains at the same time


txs

H(H(Tx-0) H(Tx-1)) H(H(Tx-2) H(Tx-3))

H(Tx-0) H(Tx-1) H(Tx-2) H(Tx-3)

Tx-0

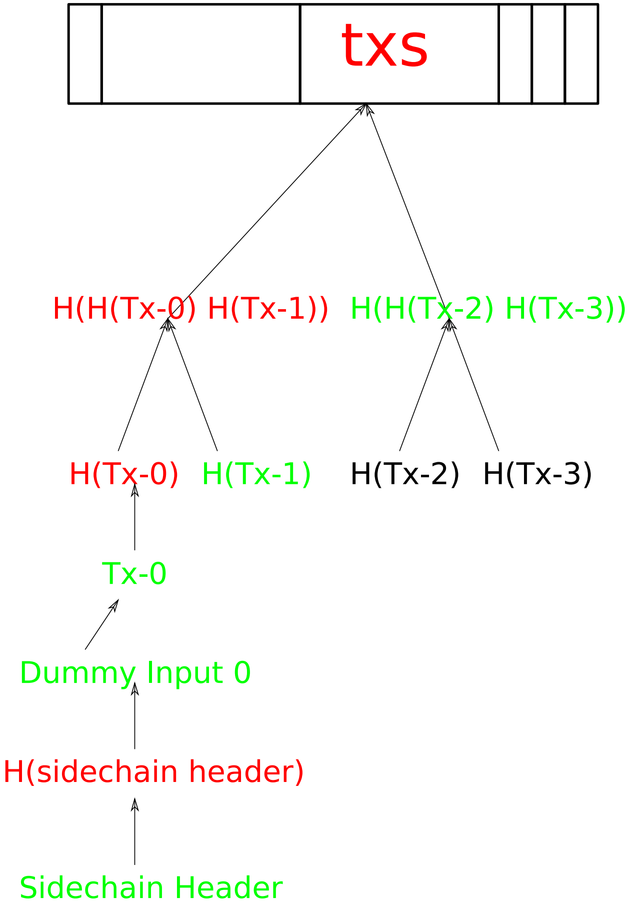
Dummy Input 0

H(sidechain header)

Sidechain Header

Sidechains: More Wasted Work?

- Bitcoin miners can mine other chains at the same time



txs

H(H(Tx-0) H(Tx-1)) H(H(Tx-2) H(Tx-3))

H(Tx-0) H(Tx-1) H(Tx-2) H(Tx-3)

Tx-0

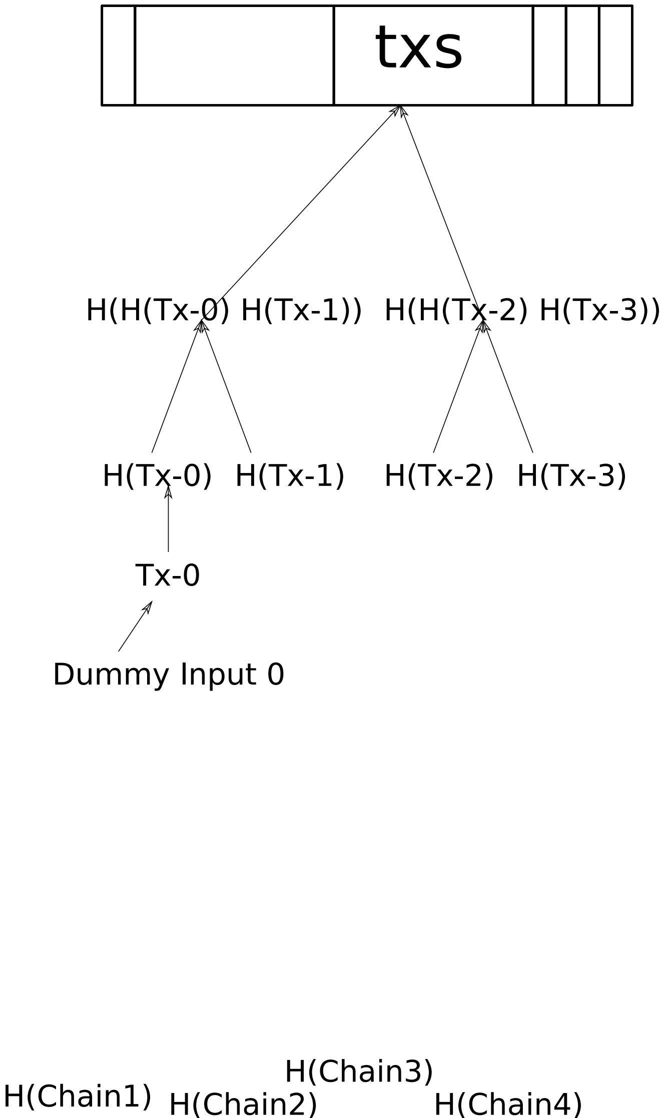
Dummy Input 0

H(sidechain header)

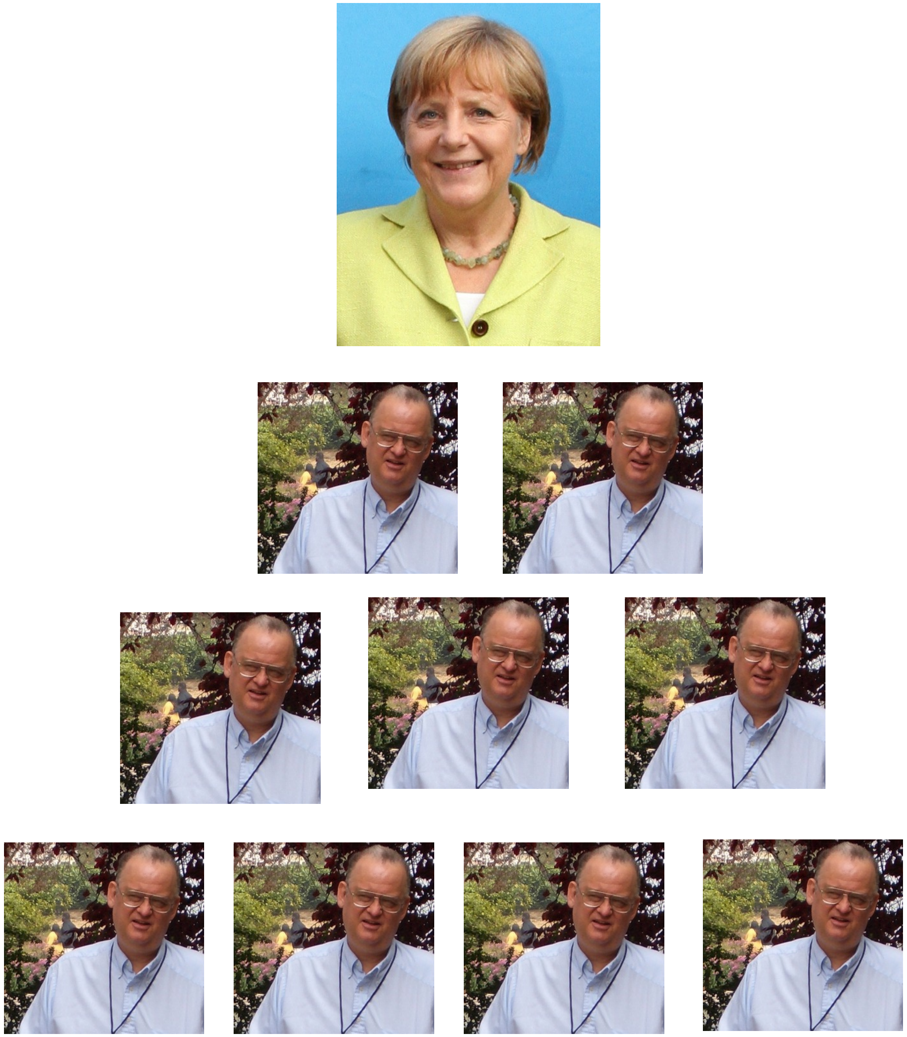
Sidechain Header

Sidechains: More Wasted Work?

- Bitcoin miners can mine other chains at the same time



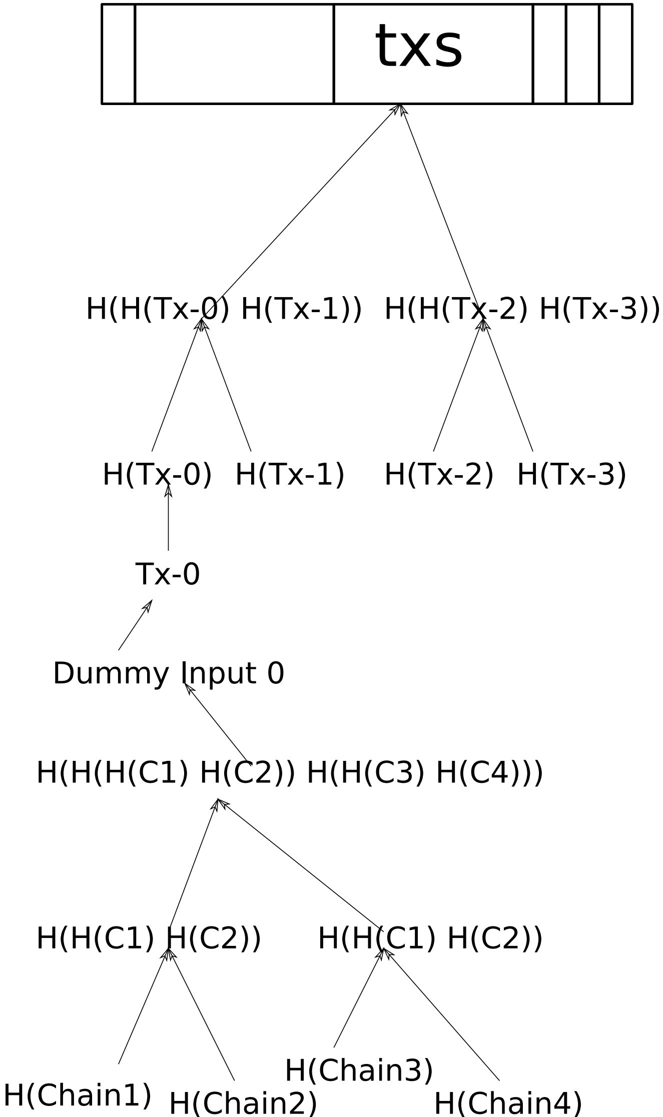
Sidechains: More Wasted Work?



DETOUR

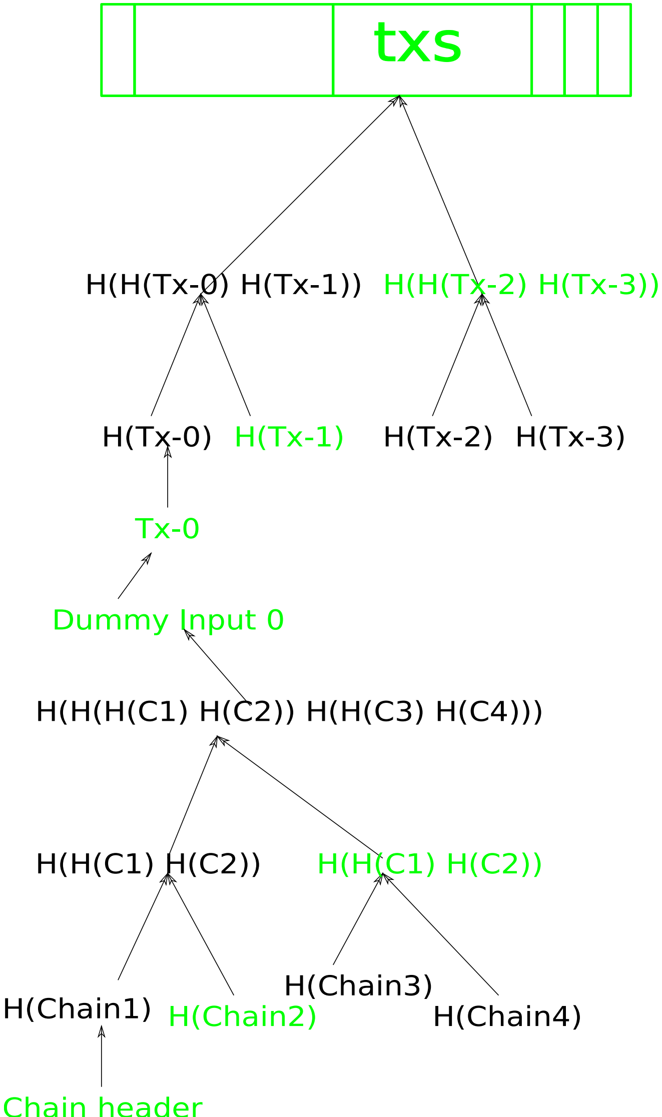
Sidechains: More Wasted Work?

- Bitcoin miners can mine other chains at the same time



Sidechains: More Wasted Work?

- Bitcoin miners can mine other chains at the same time



Sidechains

- Alternative chains which use real bitcoins
  - But may have different/experimental protocol rules

Sidechains

- Alternative chains which use real bitcoins
  - But may have different/experimental protocol rules
- Special bitcoin transactions send to the sidechain.

Sidechains

- Alternative chains which use real bitcoins
  - But may have different/experimental protocol rules
- Special bitcoin transactions send to the sidechain.
- Special sidechain transactions return bitcoins to bitcoin.

Sidechains

- Alternative chains which use real bitcoins
  - But may have different/experimental protocol rules
- Special bitcoin transactions send to the sidechain.
- Special sidechain transactions return bitcoins to bitcoin.
- Prove to the bitcoin network that the return happened in the sidechain, and bitcoin will let you spend those bitcoins again.

To Sidechain

- A bitcoin transaction output script would “send” bitcoins to the sidechain:
  - <hash-of-sidechain-block> OP_SIDECHAINPROOFVERIFY

On the Sidechain...

- Hey, a new OP_SIDECHAINPROOFVERIFY bitcoin output for us!

On the Sidechain...

- Hey, a new OP_SIDECHAINPROOFVERIFY bitcoin output for us!

  … some time later...

On the Sidechain...

- Hey, a new OP_SIDECHAINPROOFVERIFY bitcoin output for us!

  … some time later...

- That can now be spent like any other unspent transaction output.

On the Sidechain...

- Hey, a new OP_SIDECHAINPROOFVERIFY bitcoin output for us!

  … some time later...

- That can now be spent like any other unspent transaction output.

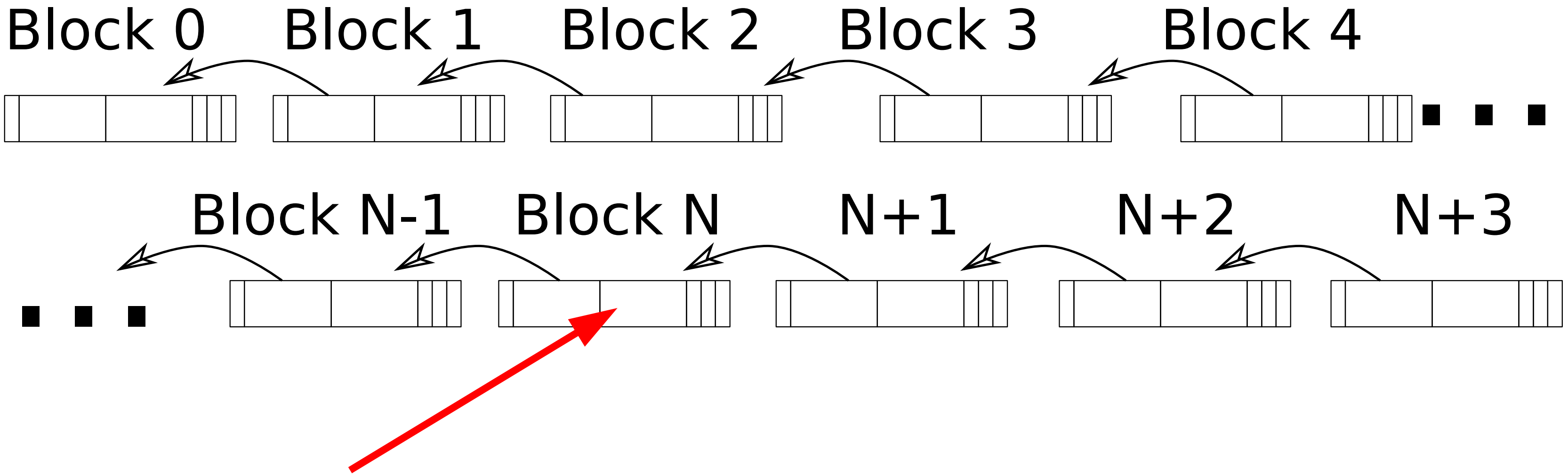
  … coins move around sidechain...

- A special unspendable output script returns the funds to the bitcoin network.

Caveats & Notes III

- In practice, would use proofs for bitcoin → sidechain (as we'll see for the other way)
- There's no BIP yet describing this, but the unspendable output could be as simple as OP_RETURN.
- My guess is:
  - <bitcoin-genesis>
    OP_RETURN
    <extra-script-to-be-evaluated-on-bitcoin-side>

On The Sidechain



Block 0

Block 1

Block 2

Block 3

Block 4

Block N-1

Block N

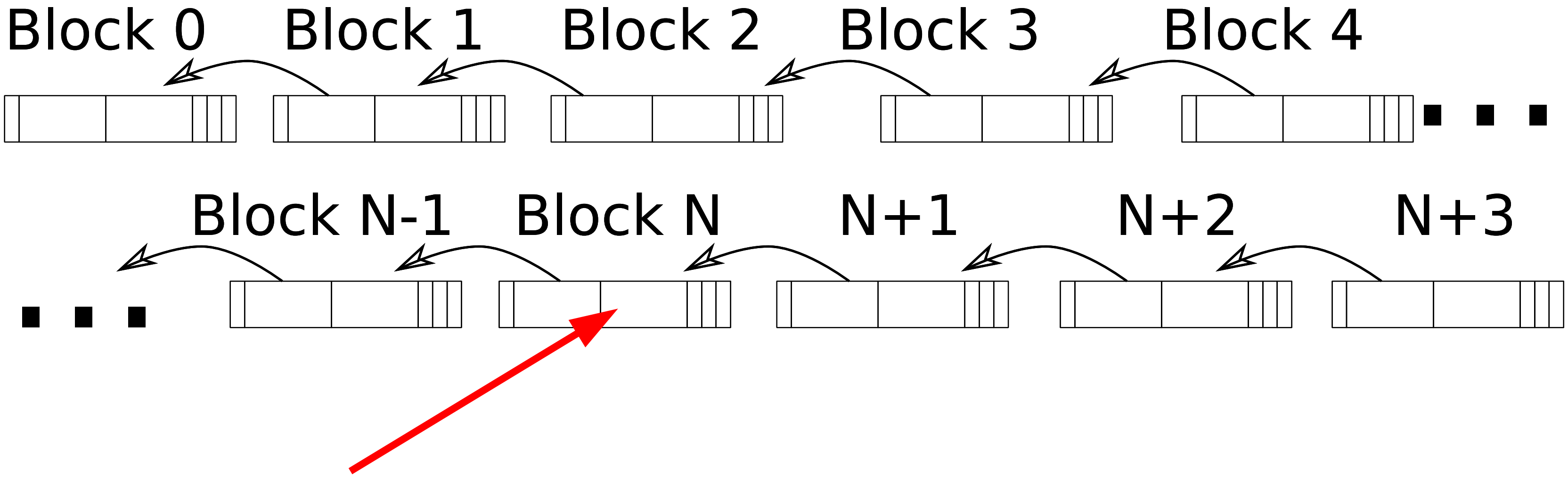
N+1

N+2

N+3

Return-to-bitcoin tx output

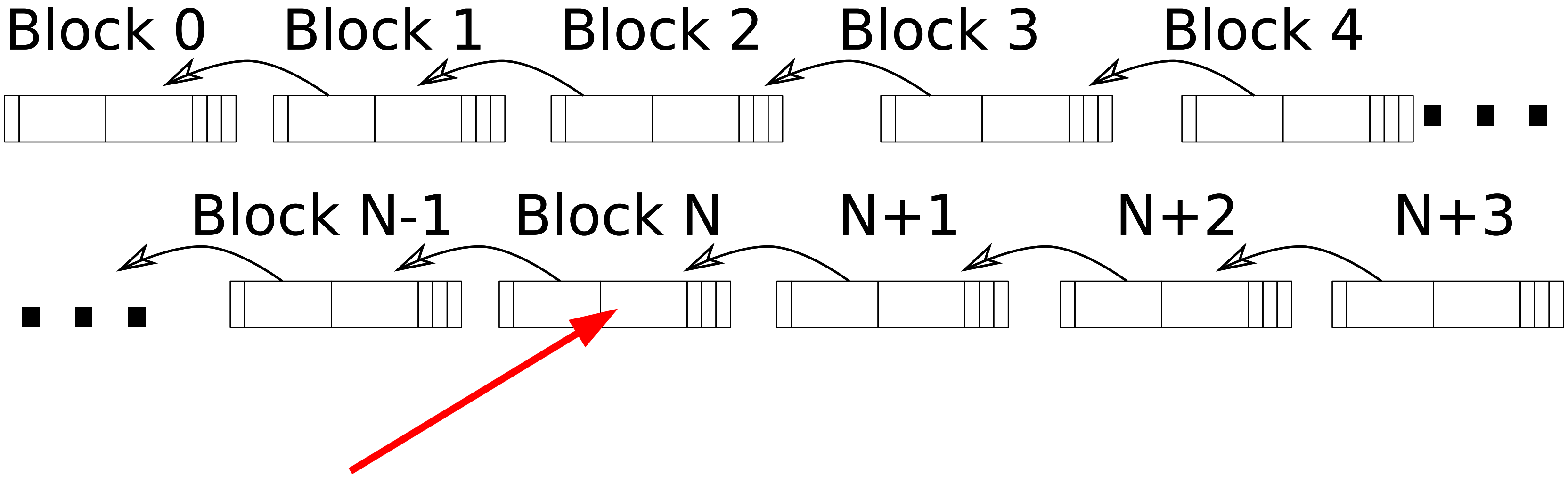
… Back To Bitcoin



Return-to-bitcoin tx output

- To spent the bitcoin OP_SIDECHAINPROOFVERIFY output
  - *Prove* the return-to-bitcoin tx is in the sidechain

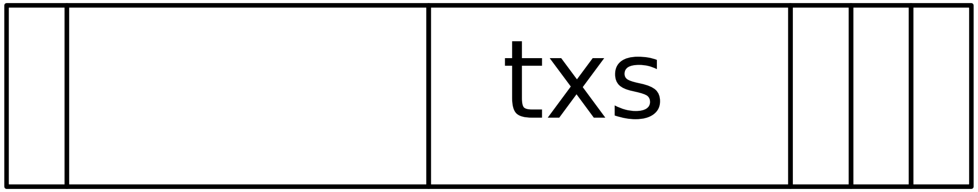
… Back To Bitcoin



Return-to-bitcoin tx output

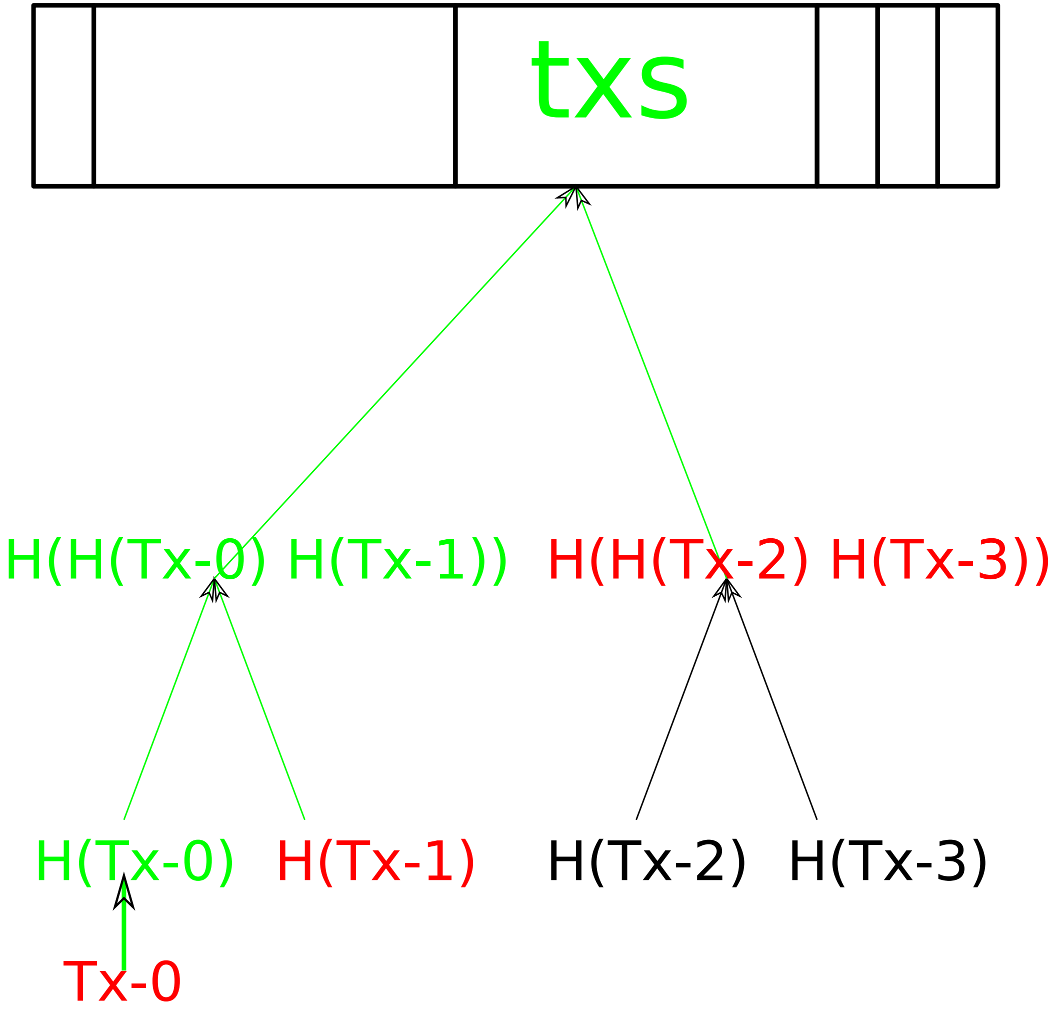
- Prove the tx is in block N
- Prove block N is in sidechain.

Prove TX in Block



Tx-0

Prove TX in Block



Prove Block in Sidechain

Prove Block in Sidechain

- Provide every block back to genesis?

Compact SPV Proofs

Compact SPV Proofs

- Since every block has to hash below some target value...

Compact SPV Proofs

- Since every block has to hash below some target value...
  - ½ the blocks will be ½ the target or less.
  - ⅓ the blocks will be ⅓ the target or less.
  - 1/100 will be 1/100 of the target...

Compact SPV Proofs

- Since every block has to hash below some target value...
  - ½ the blocks will be ½ the target or less.
  - ⅓ the blocks will be ⅓ the target or less.
  - 1/100 will be 1/100 of the target...
- You may skip back N if your hash is <= target/N.

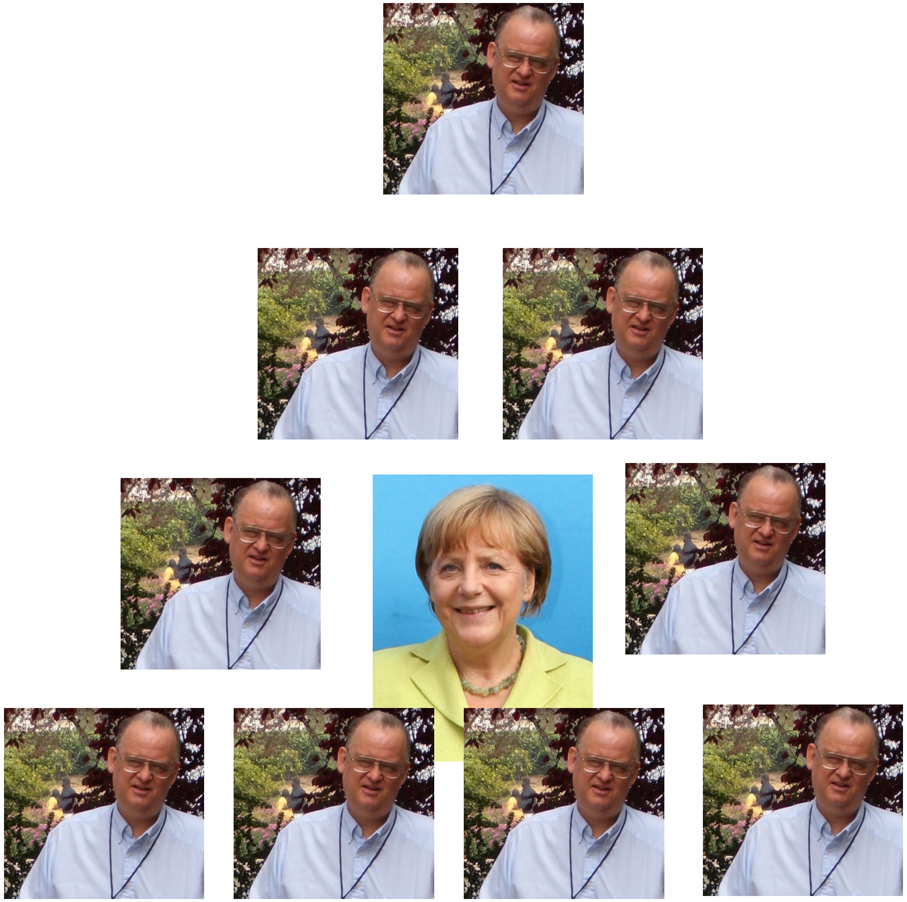
  => log(N) steps to get back to genesis.

Compact SPV Proofs

- How do we put all the previous block hashes in the block header?

Compact SPV Proofs

- How do we put all the previous block hashes in the block header?



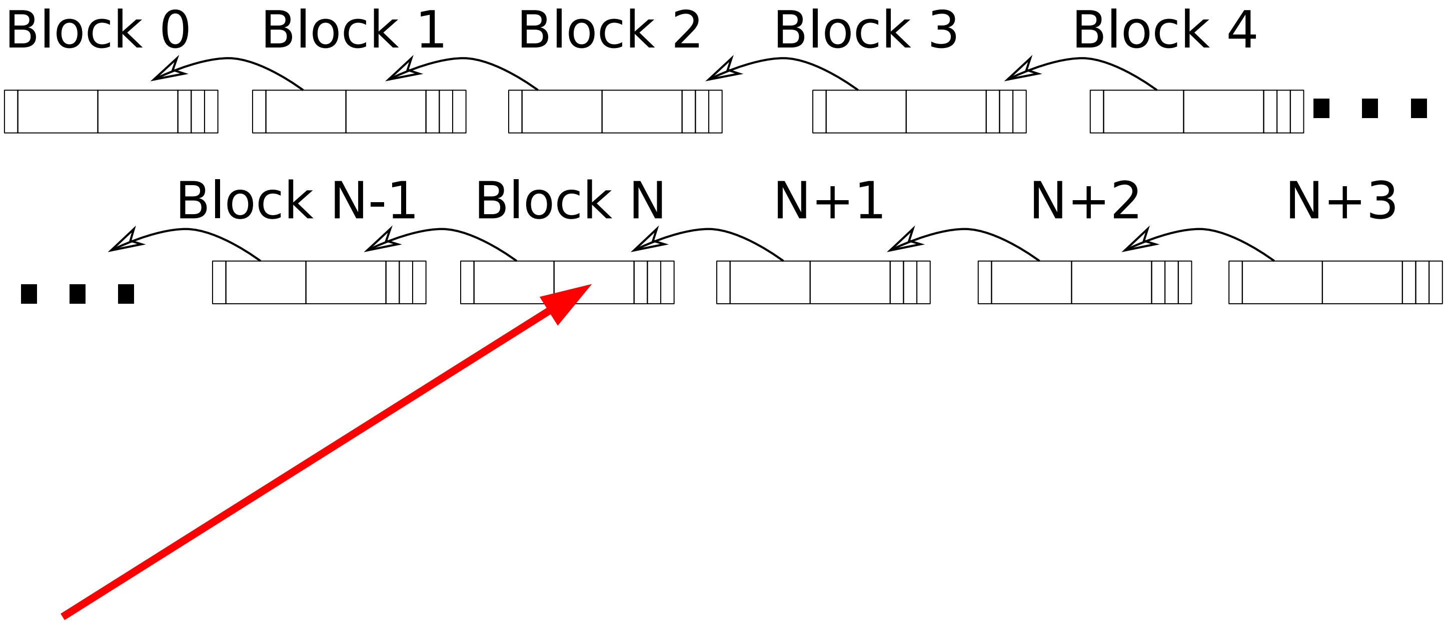
Compact SPV Proofs

- How do we put all the previous block hashes in the block header?
  - Merkle Tree!
- For 1M blocks, ~60 block headers and ~550 merkle proof hashes

Caveats & Notes IV

- Number of hashes is very sensitive to topology of merkle tree. See rustyjunk on github (WIP)
- Your path from N+<number> to genesis must include N, so it won't be quite this good.
- Target changes, so you need to include the actual distance in difficulty steps in your tree.
- CSPV proofs do not ratchet like normal blockchain: a 10% attacker has 10% chance of producing a valid-looking winner.

...Back To Bitcoin

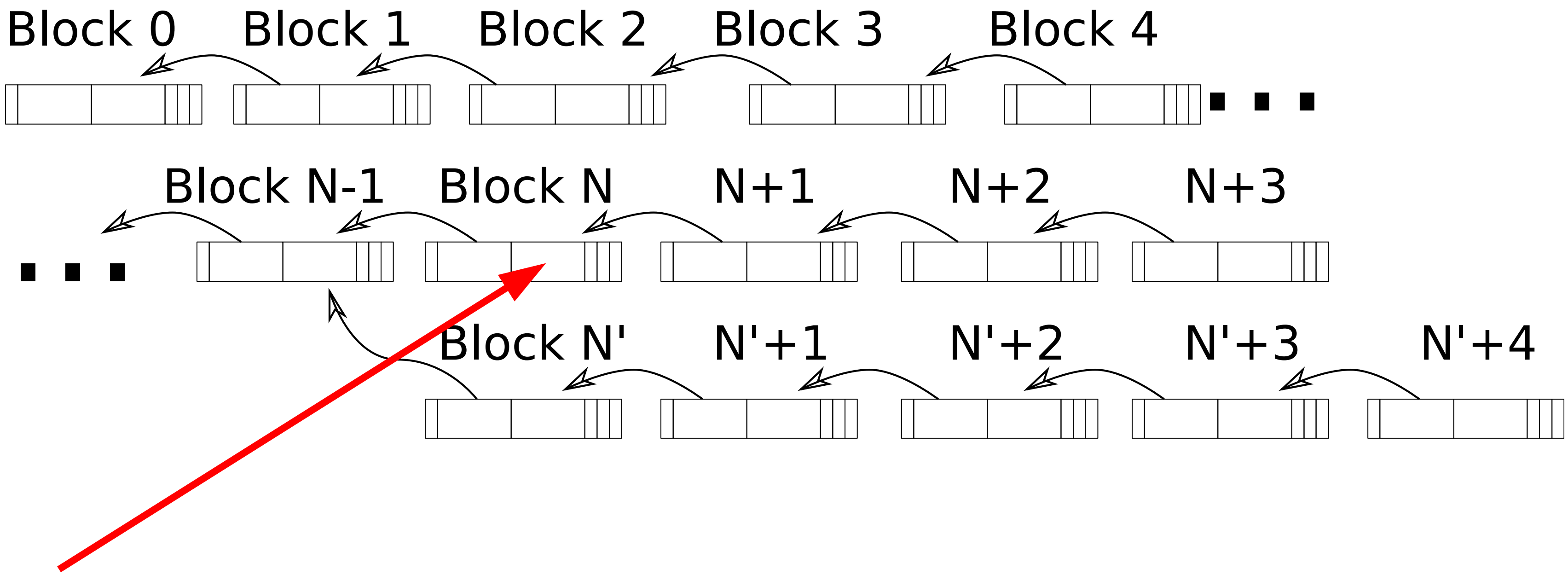


Block 0 Block 1 Block 2 Block 3 Block 4 . . .

. . . Block N-1 Block N N+1 N+2 N+3

Return-to-bitcoin tx output

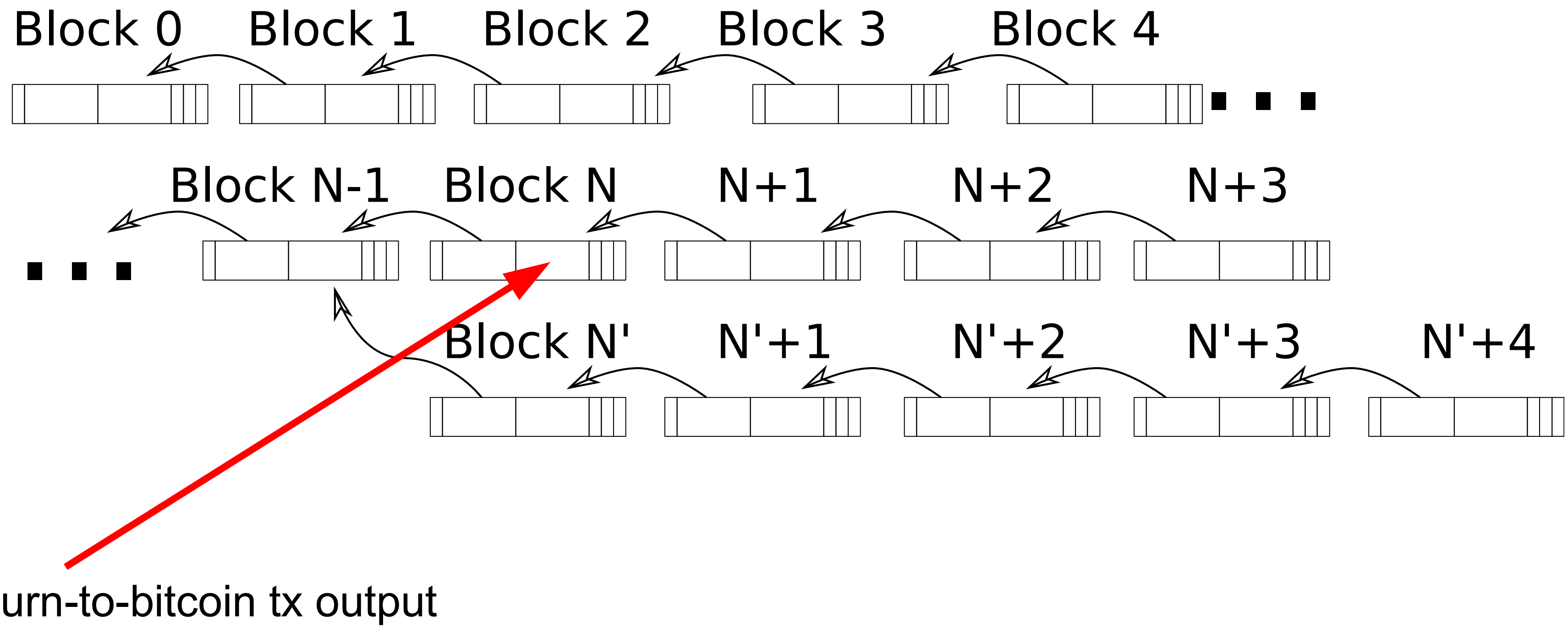
...Back To Bitcoin



Return-to-bitcoin tx output

...Back To Bitcoin

- We need to wait for some contest period to allow “reorganization proofs”.



DETOUR

Caveats & Notes V

- Reorganization proofs will presumably “invalidate” by consuming transaction outputs and producing a new OP_SIDECHAINPROOFVERIFY output.
- Gregory Maxwell suggests that transactions which simply consume OP_SIDECHAINPROOFVERIFY outputs to combine them into a single OP_SIDECHAINPROOFVERIFY output could be done without proofs, to make return txs smaller.

OP_SIDECHAINPROOFVERIFY

OP_SIDECHAINPROOFVERIFY

- Adding a new script opcode to bitcoin forks the protocol.

OP_SIDECHAINPROOFVERIFY

- Adding a new script opcode to bitcoin forks the protocol.
  - Old clients must still see the transaction as valid.
  - eg. rename OP_NOP3.

Lansinoh
mOmma
Fork
Rocking helps develop motor skills!
6m+
Ergonomic for baby's hands
Extendable handle
Portable, hygienic travel cover
BPA FREE
CONTAINS: 1 Fork with Extendable Handle

DETOUR

http://www.vitacost.com/momma-toddler-soft-fork-orange-1-piece

OP_SIDECHAINPROOFVERIFY

- Expensive

OP_SIDECHAINPROOFVERIFY

- Expensive:
  - Block headers of merge-mined sidechains are about 500 bytes.
  - Hashes are 32 bytes.
  - => Block 1M == 60*500 + 550*32 == 48k.

OP_SIDECHAINPROOFVERIFY

- Expensive:
  - Block headers of merge-mined sidechains are about 500 bytes.
  - Hashes are 32 bytes.
  - => Block 1M == 60*500 + 550*32 == 48k.
- Slow:
  - Maybe 1 day confirmation requirement, 1 day contest period.

Atomic Swaps

Atomic Swaps

- Alice has 1 pettycoin. Bob has 1 bitcoin.

Atomic Swaps

- Alice: “*To redeem this 1 pettycoin you need to present the value that hashes to X, and Bob's signature*”

DETOUR

Atomic Swaps

- Alice: “*To redeem this 1 pettycoin you need to present the value that hashes to X, and Bob's signature*” OR “*Alice can have it after 48 hours*”

Atomic Swaps

- Alice: “*To redeem this 1 pettycoin you need to present the value that hashes to X, and Bob's signature*” OR “*Alice can have it after 48 hours*”
- Bob: “*To redeem this 1 bitcoin to need to present the value that hashes to X, and Alice's signature*” OR “*Bob can have it after 24 hours*”

Atomic Swaps

- Alice: “*To redeem this 1 pettycoin you need to present the value that hashes to X, and Bob's signature*” OR “*Alice can have it after 48 hours*”
- Bob: “*To redeem this 1 bitcoin to need to present the value that hashes to X, and Alice's signature*” OR “*Bob can have it after 24 hours*”
- Alice uses the 1 bitcoin output, revealing the value that hashes to X.

Atomic Swaps

- Alice: “*To redeem this 1 pettycoin you need to present the value that hashes to X, and Bob's signature*” OR “*Alice can have it after 48 hours*”
- Bob: “*To redeem this 1 bitcoin to need to present the value that hashes to X, and Alice's signature*” OR “*Bob can have it after 24 hours*”
- Alice uses the 1 bitcoin output, revealing the value that hashes to X.
- Bob can now use the 1 pettycoin.

Caveats & Notes VI

- Requires transaction malleability to be resolved (BIP 62) or OP_CHECKTIMELOCKVERIFY (BIP 65) (better!)

Sidechains Technology

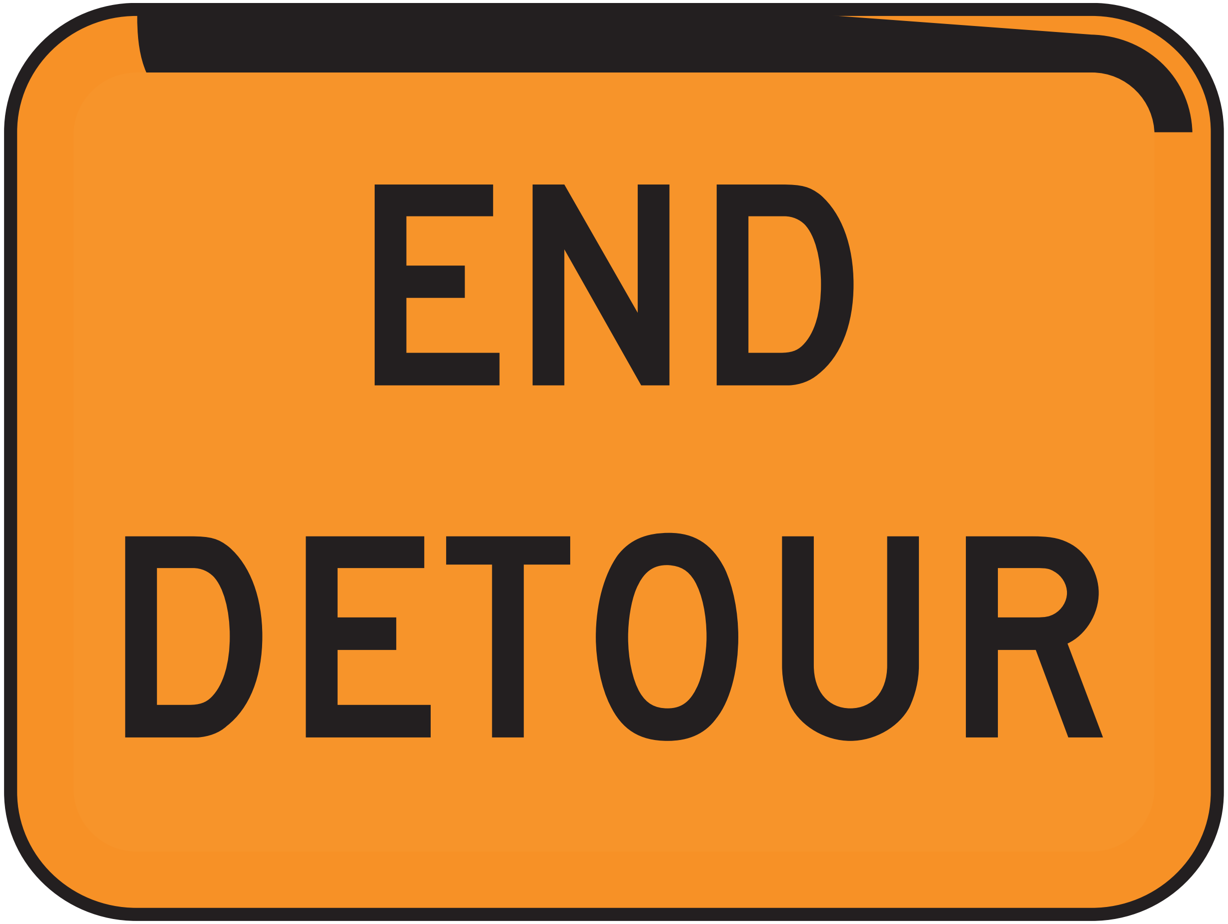
- Merkle trees
- Merge mining
- Sophisticated scripting language
- Soft fork
- Compact SPV proofs
- Atomic swaps

Sidechains Technology

- Merkle trees[1]
- Merge mining[2]
- Sophisticated scripting language[3]
- Soft fork[4]
- Compact SPV proof[5]
- Atomic swaps[6]

[1] 2008: S Nakamoto Bitcoin: A Peer-to-Peer Electronic Cash System
[2] 2009?
[3] 2014: BIP 65 https://github.com/bitcoin/bips/blob/master/bip-0065.mediawiki
[4] 2012: BIP 16, BIP 30, BIP34
[5] 2012: The High Value Hash Highway https://bitcointalk.org/index.php?topic=98986.0

[6] 2013: T. Nolan, Re: Alt chains and atomic transfers, https://bitcointalk.org/index.php?topic=193281.msg2224949


END
DETOUR

Sidechains Paper Side Effect

Exposure To Other Ideas

Exposure To Other Ideas

- Funding protocol bootstrap
- Proving Flaws using Partial Knowledge
- Calculating Fees with Partial Knowledge
- Proving Non-existent TX Inputs
- Proving Double Spends
- Proving All Block Information Is Available

Exposure To Other Ideas

- Funding protocol bootstrap
- Proving Flaws using Partial Knowledge
- Calculating Fees with Partial Knowledge
- Proving Non-existent TX Inputs
- Proving Double Spends
- Proving All Block Information Is Available

Exposure To Other Ideas

- Funding protocol bootstrap
- Proving Flaws using Partial Knowledge
- Calculating Fees with Partial Knowledge
- Proving Non-existent TX Inputs
- Proving Double Spends
- Proving All Block Information Is Available

See https://en.bitcoin.it/wiki/User:Gmaxwell/features#Proofs
And http://rustyrussell.github.io/pettycoin/ Pettycoin Revisited parts 1-7.

Partial Knowledge

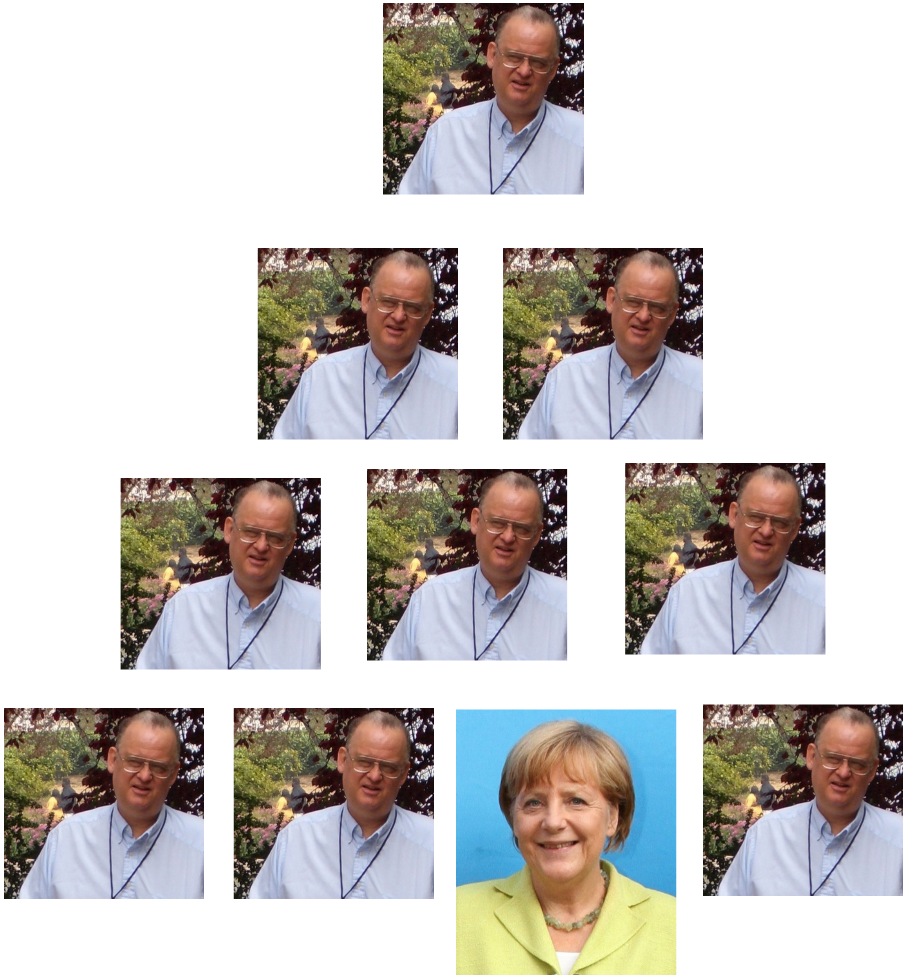
- Is the miner collecting fair rewards?

Partial Knowledge

- Is the miner collecting fair rewards?
  - Pettycoin uses a lottery, “random” transaction chosen and multiplied.

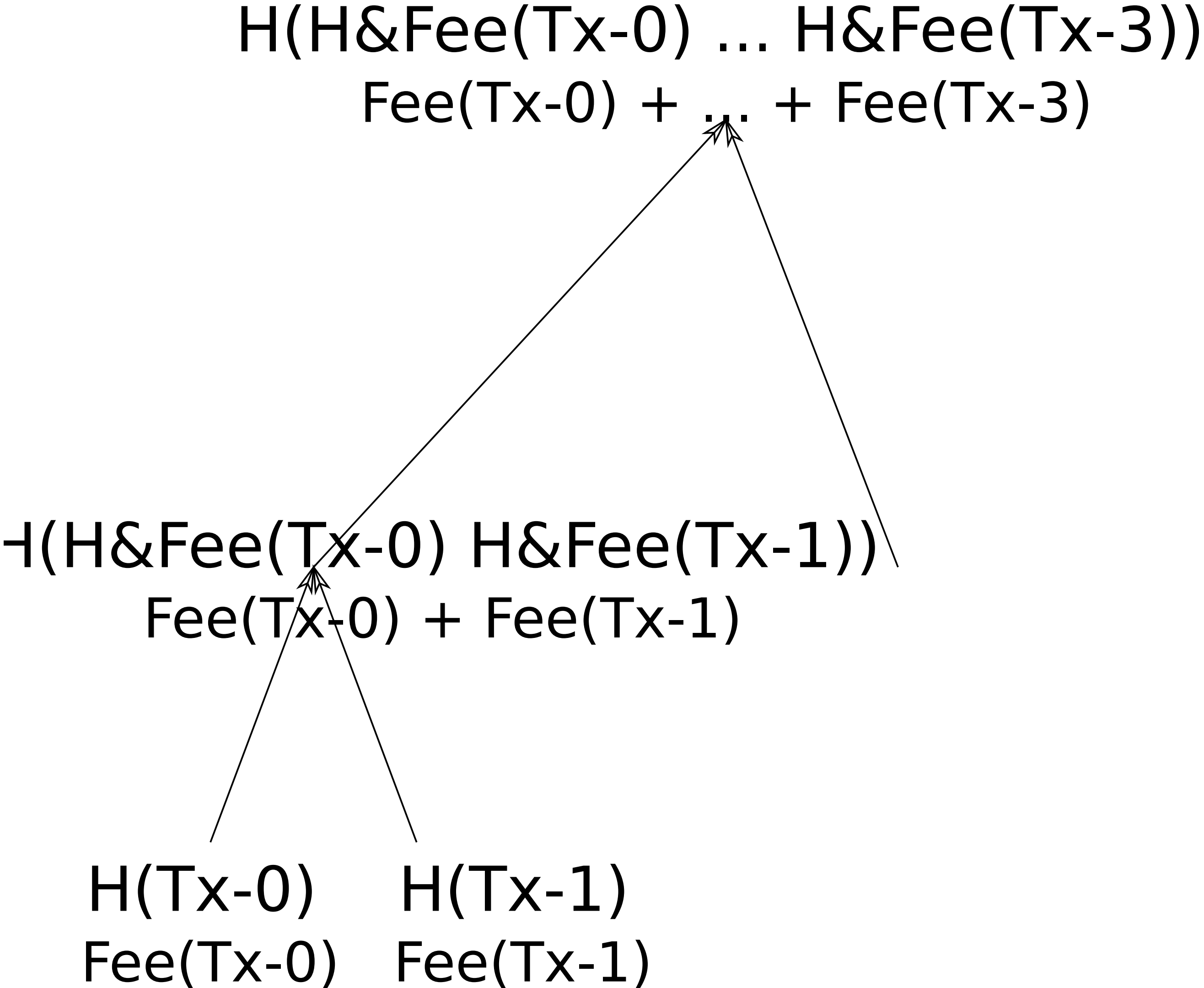
Partial Knowledge

- Is the miner collecting fair rewards?



Partial Knowledge

- Is the miner collecting fair rewards?



H(H&Fee(Tx-0) ... H&Fee(Tx-3))
Fee(Tx-0) + ... + Fee(Tx-3)

H(H&Fee(Tx-0) H&Fee(Tx-1))
Fee(Tx-0) + Fee(Tx-1)

H(Tx-0)
Fee(Tx-0)

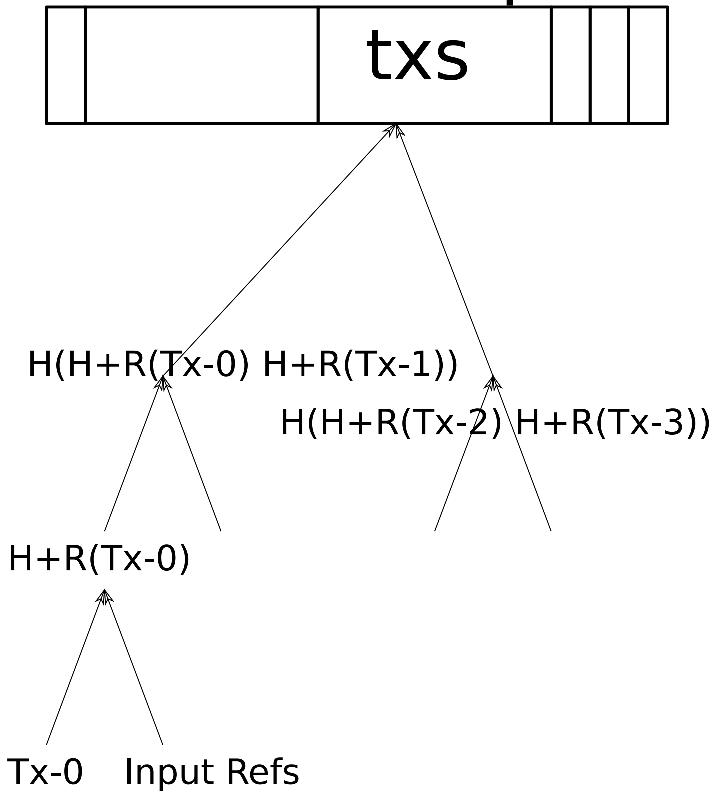
H(Tx-1)
Fee(Tx-1)

Non-existent Inputs

- Block N contains TX1 which spend output from TX <made-up-hash>?

Non-existent Inputs

- Block N contains TX1 which spend output from TX <made-up-hash>?
  - Pettycoin miners attach backrefs which say where in chain you can find the input transactions:



Non-existent Inputs

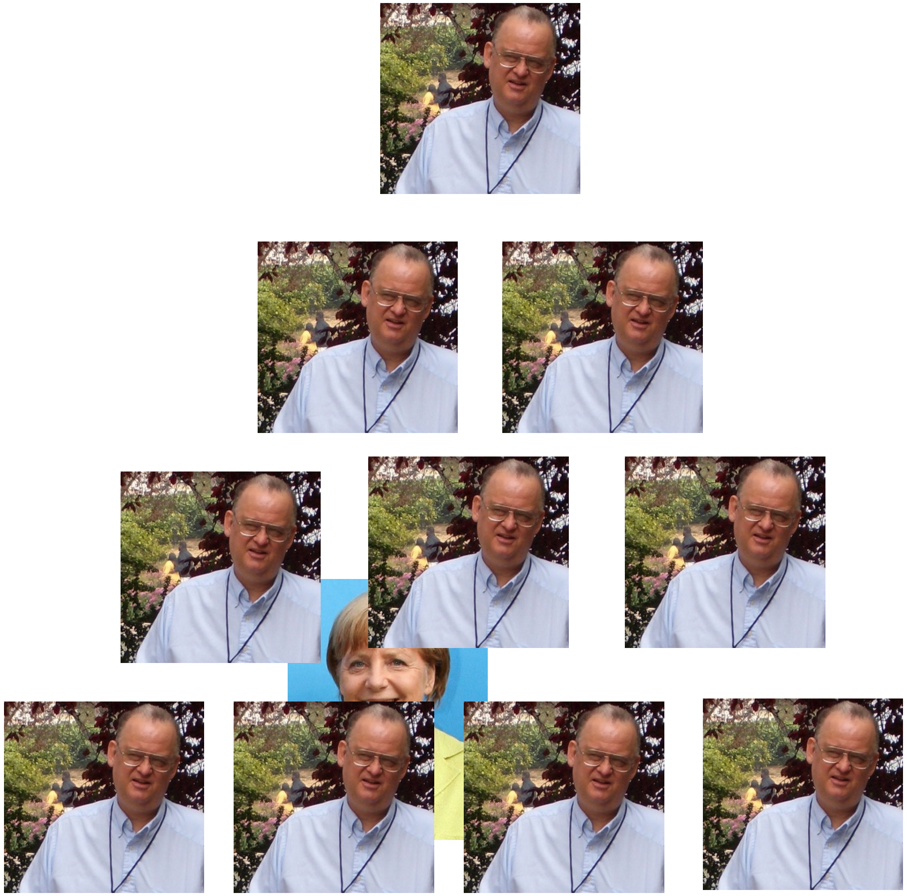
- Block N contains TX1 which spend output from TX <made-up-hash>?
  - UTXO commitments.

UTXO Commitments

- Include every Unspent Transaction Output in the header.

UTXO Commitments

- Include every Unspent Transaction Output in the header.

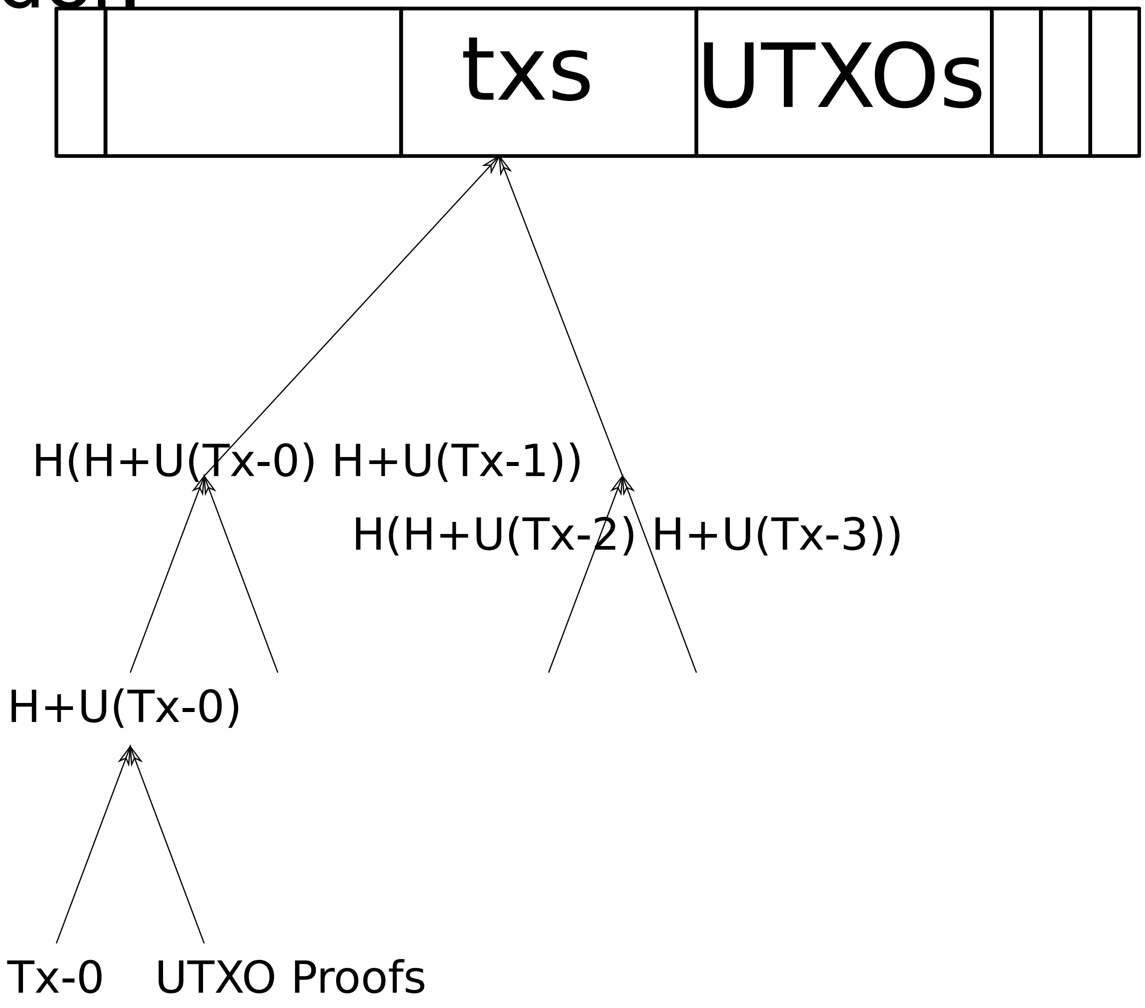


UTXO Commitments

- Include every Unspent Transaction Output in the header.
  - For each input, attach proof that it was in UTXO tree.
  - For each output, attach proof showing where it goes in (updated) UTXO tree.

UTXO Commitments

- Include every Unspent Transaction Output in the header.



Caveats & Notes VII

- A patricia trie is usually suggested for this structure.
- If it's keyed by Txid then output, it's fairly trivial to group output insertion into a single proof.

Proving Double Spends

Proving Double Spends

- Pettycoin relied on someone reporting (with proof) that a TX output was used before.
- UTXO commitments make this impossible anyway.

Fast Block Times

Fast Block Times

- 10 second blocks.

Fast Block Times

- 10 second blocks.
- 1% of blocks take over 46 seconds.
- Accept “easy” block after 20 seconds passed, with a modified heuristic to determine which easy block wins.[1]

[1] http://rustyrussell.github.io/pettycoin/2014/10/30/More-Regular-Block-Times.html

Caveats and Notes VIII

- Convergence difficult unless propagation time >> block time.
  - GHOST helps here[1]
  - 10 seconds is probably close to lower bound.
- Bitcoin's testnet does this horribly using timestamps: don't copy!

[1] Accelerating Bitcoin's Transaction Processing Y Sompolinsky, A Zohar https://eprint.iacr.org/2013/881.pdf

What Does This Mean for Pettycoin?

What Does This Mean for Pettycoin?

What Does This Mean for Pettycoin?

- Need to be more bitcoin-like.
  => Just use the bitcoin reference code.

  (But there may be many sidechains to copy)

What Does This Mean for Pettycoin?

- Need to be more bitcoin-like.
  => Just use the bitcoin reference code.

  (But there may be many sidechains to copy)
- We now have a name for what we built.
  - Pettychain?

What Does This Mean for Pettycoin?

- Need to be more bitcoin-like.
  => Just use the bitcoin reference code.

  (But there may be many sidechains to copy)
- We now have a name for what we built.
  - Pettychain?
- Fastchain should be a separate sidechain experiment.

Thanks

- My family.
- Robert Collins
- Bitcoin wizards, esp. Gregory Maxwell.
- IBM

Thanks

- My family.
- Robert Collins
- Bitcoin wizards, esp. Gregory Maxwell.
- IBM

Thanks

- My family.
- Robert Collins
- Bitcoin wizards, esp. Gregory Maxwell.
- IBM

Questions?