Bringing New Elements to Bitcoin

with Sidechains

SF Bitcoin Devs Meetup
June 8, 2015

Greg Maxwell

DE47 BC9E 6D2D A6B0 2DC6 10B1 AC85 9362 B041 3BFA
Bringing New Elements to Bitcoin with Sidechains

Topics
• Challenges in Advancing Bitcoin Technology
• Introducing Sidechain Elements
• Future Directions

- Deterministic Peg
- Asset Issuance
- Relative Locktime
- Segregated Witness
- Script Enhancements
- Amount under Sig
- Federated Consensus
- Confidential Transactions
Challenges in Advancing Bitcoin

• Traditional money systems require trust at all levels, this trust is costly to maintain and is unpredictably violated

• Bitcoin's solution
  – Replace most of the trust with a system of mechanically enforceable rules

• Nature protests: a copy of data is as good as the original, information doesn't have “owners”
  – Money needs controlled supply and ownership
  – People are good at ignoring rules (when they want)
Challenges in Advancing Bitcoin

**Good news:** Bitcoin employs cryptography and economics to deliver a system where rules have true force, even against popular will.

**Bad news:** This created a system where a fixed set of protocols / algorithms were in charge.

- If mankind had perfect engineering, perfect foresight, and universal values, it might be okay.
- But we don't: mistakes were made, needs change, and people sometimes earnestly have contradictory demands.
Challenges in Advancing Bitcoin

• Some have sought to create new functionality by starting brand new cryptocurrencies
• The value of a money-like good comes from acceptance – it’s practically all network effect
• A speculative race around “creating money”: bad incentives and no natural stopping point: \texttt{foocoin\rightarrow barcoin\rightarrow bazcoin\rightarrow barfcoin}
• The reboot is left with the same problem
• I wish people luck, but I don't think this is a sustainable way to build new technology
Challenges in Advancing Bitcoin

• Bitcoin was designed to embrace new uses with powerful smart contracts and extensibility

• Hobbled by bugs, but it's possible to fix and improve compatibly via soft-forks
  – Previously used to deploy P2SH, the 3- addresses

• It's hard to update a live production system esp. when it was made to be beyond influence

• Most updates are inherently much easier to do in a new network
Advancing Bitcoin with Sidechains

- Bitcoin supports verifying that a payment has happened with very small amounts of communication called SPV.
  - There is a security tradeoff with SPV: It trusts miners to verify the history, stronger non-partitioning assumption.
- What if a Bitcoin smart contract released coins only according to a Bitcoin SPV proof?
- The result is the “two-way peg” described in the sidechains whitepaper.

---

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)
Advancing Bitcoin with Sidechains

• Two-way peg freezes Bitcoins so they can only be released according to a decision by some other network
• Then they can be brought back
• Gain the freedom and agility of multiple networks without rebooting the network effect
• Put the new, risky, experimental, only-liked-by-a-few features in their own networks
• There are a lot of details to get right to make this work in a usable way
Introducing Sidechain Elements

- Project to advance the art for Bitcoin
  - “No holds barred”, exploratory technology
- With a testnet federated-peg sidechain: Alpha
  - Free Software to open new avenues for everyone
  - With many new and interesting features (elements)
  - But currently without a lot of quality assurance

It's nice to work on something without a billion-dollar economy immediately resting on it, but still have a path to production use (and not just production use in competing cryptocurrencies!)
Sidechain Elements

• Each feature in elements right now is someone's experiment
  – Though usually a few people contributed review and other assistance
• We hope that other people will find this software and approach interesting and contribute their own experiments
• I would strongly recommend against using the current code with a real with-value cryptocurrency network
Deterministic Peg

- Implementation of the two-way peg mechanism from the sidechains whitepaper
  - Allows testnet coins to be logically moved to the elements network and back again
- After 10 confirmations in testnet a move to the sidechain can be started on the sidechain side
- Funds held for 144 confirms on the sidechain, which allows someone else to prove that a longer testnet fork exists
- Fundamental insecurity of testnet is limiting here
Deterministic Peg

• “But Testnet script can't parse the return proof!”
  – Uses the Appendix A “federated peg”: a federation of oracles execute the code testnet would run (but doesn’t know how to)
  – Centralized “protocol adapter” absent native support
• Federation is an N of M threshold, plain multisig to testnet
• Participants have no discretion, and the sidechain users can mechanically detect misbehavior
• If N are compromised they can steal coins
Deterministic Peg

• Hybrid model, the other testnet → sidechain direction is verified by the sidechain
• Testnet doesn't have the commitments needed for Appendix B efficient SPV proofs
  – Could put all the testnet headers in the sidechain,
  – Instead nodes verify all they can and then RPC to a local testnet node to test chain membership

Principal Investigator: Matt Corallo
Why Issued Assets?

• Bitcoin brought us smart contracts which are enforced trustlessly by the network
• You can imagine using it to build trustless exchange, derivative assets, etc.
• But the network can only control things directly inside it*
• Building fancy synthetic assets out of smart contracts requires the network see the component assets
  – e.g., “Can be redeemed to claim one car or f(date) bitcoins”
Why Issued Assets?

• “Colored coins” have existed for years but...
  – not SPV compatible, users must trace to find the coloring; especially painful for smart property
  – Tons of tiny dusty UTXO on the network, can't have a different retention policy
  – Invisible to the colorblind network, smart contracts can’t be made asset aware; “I'll trade 1 bitcoin for 1 foo”
  – Though, they could have strong censorship resistance
Issued Assets

- Tag all coins in the network with an “asset type”
  - Immediately fixes SPV
- All accounting rules are grouped by asset type
  - e.g., sum of inputs of a type have to equal the sum of outputs of that type
- Assets issued via a new special transaction, the txid becomes the asset tag

Principal Investigator: Jorge Timón
Why (relative) check-locktime?

- Bitcoin transactions have a 'good after' date
- Many fancy contract examples need refund on timeout to prevent holdup
  - Effectively need a pubkey of A or B and Time>X
  - Add a scriptPubkey rule to check the good-after
- Relative: don't expire based on a fixed clock, instead an initial transaction starts the clock
- Absolute checklocktime (BIP65) may be available on mainnet this year
Relative check-locktime

- In Bitcoin there is a 32-bit sequence per input, increment to indicate updated transactions
- No* consensus rules in Bitcoin currently
- Insecure: miners can happily take an earlier version, no protocol rule stops them
- Soft-fork so that max-1 can spend inputs one block old or older, max-2 two blocks etc.
- Locktime relative to inputs, and sequence numbers now work with protocol enforcement

Principal Investigator: Mark Friedenbach
Why Segregated Witness?

- A Bitcoin can be spent if some input is provided which makes its assigned script returned true
- Network runs the script with the inputs to verify
- What happens if you can take a true script and modify it and get another one? “Malleability”!
  - Third parties can change transaction IDs
  - Potentially breaks multi-step contracts, or even just spending an unconfirmed transaction
  - This is virtually always true, making even boring scriptPubkeys non-malleable is hard (BIP62 tries)
Why Segregated Witness?

• A witness is a specific value that constitutes a concrete proof for an existential claim
• Bitcoin doesn't care why the scriptPubkey accepted, just that it does
• Fancy crypto can make it possible to skip sending the witnesses entirely – but not practical yet
• If you're not verifying the history, instead trusting it blindly, you don't care about the witnesses, but they are 2/3 of the data.
  – But you have to fetch it to verify transaction hashes
Segregated Witness

• Change to the transaction hashing structure
• Logically splits the transaction into two parts:
  – Witness (the scriptSig fields)
  – The transaction (everything else) TXID only covers this
• Blocks still commit to the witness:
  – $H(H(tx) || H(witness))$ in the transaction tree
• Syncing the block chain without signature checks can skip witnesses, and unwanted third-party changes are prevented

Principal Investigator: Pieter Wuille
Why Script Enhancements?

• Bitcoin Script was once much more powerful
  – A bit too powerful: crash nodes and steal coins
• Many operations “disabled” – really, removed
• Not technically hard to fix, especially in a hard-fork, but...
  catch-22: no one uses functionality that isn't there, hard to justify
  adding things people don't use
• A much more powerful system has been in the works, on and off
  for some time...
• But if experimentation is cheap, why not?
Script Enhancements

- Re-enabled: concatenate, substring, truncate right/left, shift left/right, bitwise INVERT, AND, OR, and XOR
- Plus some more e.g., a CSPRNG randrange
- Also replaced ECDSA with Schnorr
  - Efficient (non-accountable) multisig
  - Batch verification (2x speedup)
- Check signatures for arbitrary data on the stack
- No more non-verify CHECKSIG operations

Principal Investigator: Patrick Strateman
Speaking of Signatures...

- In Bitcoin signatures only sign the amount of the coin they're spending by signing its txid.
- To prove to a hardware wallet what its signing you have to stream all the input transactions to the device.
  - Otherwise it can't tell how much it's spending.
- In a contrived case you could make this be as much as a gigabyte of data in Bitcoin today.
- Just include the amount directly in the signing hash and the transaction is invalid if you lie to the device.

Principal Investigator: Glenn Willen
Why Federated Consensus?

• Decentralized consensus is essential for upholding the Bitcoin ethos in a public system.
• But what does that mean for private systems?
• Is decentralized consensus even possible for experimental, low value, or small systems?
• How can you safely bootstrap mining?
• Lots of other consensus models exist...
• Sidechains paper describes Bitcoin's mining consensus as a dynamic-membership multiparty-signature
Federated Consensus

- Replace mining DMMS with a plain multiparty-signature:
  Yields a *centralized* security model
- But (arbitrarily) better than “trust one party”
  - Real-time audited by all participants
  - Most dishonest behavior machine decidable
  - Arbitrary multisig policy (A & 5-of-8) | (8-of-8)
- No human discretion required: can implement on tamper-resistant hardware
- Some applications need trust: if you have it, why not use it?

*Principal Investigator:* Jorge Timón
Why Confidential Transactions?

• Traditional transaction systems provide privacy
  – Essential for both commercial and personal use
  – Absent it, thieves can target selectively; negotiating positions undermined; fungibility lost
• Public consensus needs public verification
  – Surprisingly: compatible with complete privacy

Consider digital signatures: your secret key is secret, but you prove you know it ...
Why Confidential Transactions?

- Bitcoin uses pseudonymity
  - Fragile at best. Paying someone usually leaks your identity and financial information, addr reuse leaks it to everyone
- Lack of privacy oft-cited as a concern by institutions
- Transparency is a powerful feature, but it cuts both ways if not controlled by its users
  - Exacerbates existing power imbalances
  - Besides, raw information isn't meaningful transparency
- Yet harmful uses still have privacy, it's just expensive
Why Confidential Transactions?

• Many involved long-term in the development of Internet protocols regret that we lack ubiquitous encryption today
• There was always a reason: “It's complex”, “It's slow”, “It’s incompatible” … technically true, but mostly insignificant in hindsight. The failure to make crypto default only gets harder to fix

If Bitcoin displaced other systems of money, would I want to live in that world?

Not without major improvements on this issue
Why Confidential Transactions?

• Past proposals to improve Bitcoin privacy:
  – Compatible: CoinJoin, CoinSwap, centralized servers, ...
  – Cryptographic solutions: Zerocoin, OWAS, Traceable ring signatures (bcn/xmr), Zerocash, ...

• Compatible solutions mostly suffer from loss of privacy due to transaction amount tracing

• So far, cryptographic solutions break pruning and often need new strong assumptions, have very poor performance, and/or just aren't implemented
Confidential Transactions

• Prior work focuses on the transaction graph...
  – What if you make transaction amounts private?
• Amounts are usually more important to keep private
• 8-byte amounts become 33-byte commitments – like a hash
• The blinded commitment preserves addition
  – Thus the network can verify that the amounts add up
• Originally proposed by Adam Back in 2013: “bitcoins with homomorphic value” on bitcointalk
Confidential Transactions

• Must prevent negative values when splitting: \((1 + 2) = (-10 + 13)\)
  – needs zero-knowledge range-proof, linear in size
• I invented a generalization of ring signatures and other optimizations to make the range proofs more efficient
  – 2.5KB for 32bits: up to 42, 429, 4294, ... BTC depending on exponent
• Then came up with a way to use 80% of their size to communicate a private message to the payee
• Compatible with watching wallets: share a scanning key to allow watching without spending, or share a blinding value to prove a payment amount to anyone

Principal Investigator: Greg Maxwell
Future Direction

• I'm looking forward to watching alpha network explode in *interesting ways*
  – Testnet itself has been under some interesting attacks lately...
• Continuing refinement of these elements and creating more
• Will the potential to introduce new technology for Bitcoin without seeking permission or rebooting the adoption result in more development beyond ours?
  – I don't know, let's find out together
Thanks for your time.

Greg Maxwell