Lightning network.
Fewer validators or bigger block size (not all can validate tx) can lead to the centralization. A network micropayment channels can solve scalability problems. SIgn tx but do not broadcast. Penalty – the honest party claim all funds. Before making bidirectional payment system users must agree on refund rules. SIGHASH_NOINUT allows to create and sign txs before it will be published in blockchain.

Commitment tx – after Funding, that shows current balance, but both users can publish commitment tx that they like more. Penalties should be included - 1 commitment tx has 2 version (A signed, B has access and Ahas access, B signed). Revocable Sequence

Maturity Contract (RSMC) - tx only spendable after N amount of blocks, that allow other party to detect cheating. Timelock on old tx, new one without locks. We are not sending new tx, to make channel available. So we are setting revocation (time) on old commitment tx(tx that holds states) and new one without it.

To protect from flood attack – timestop. Block can be overloaded (special bit 1 in version). Block is not counted as confirmed with timelock. During the flooding the honest party can not close the micropayments channel.

Payment to yourself – RSMC, payment to another user no lock. Invalidation of old txs happens using signed Breach Remedy Transaction (BR1) for penalty. Giving private key to outputs – B is publishing old state, A can change output, but A can not publish herself.
HTLC – Hashed Timelock Contract – payment between multiple nodes. A-B-C. C generates a secret and tell A only hash. Problem appears then.

 Off chain revocable HTLS – CT (Balance A, Balance B, HTLC-output) with RSMC. Keys are generated using BIP 0032 Hierarchical Deterministic Wallets. Millions of keys is generated and grouped by days. Each key – ouput. One master key – generate 1000 child keys. A send master key to B so he can delete all unused.

Bitcoin Lightning Network. If one is not available – delete from the network and send CT to blockchain. Smaller amounts – no risk for locks. Nodes with good connection can give access to channels for fees. Fees are paid inside the channel based on time and risk of refuse. Higher risk – higher fees. Only the non-responsive channels get broadcast on the blockchain, all others are settled off-chain via novation. Negative fees – give motivation to send tx backwards. To use bidirectional channel.Risks. Short HTLC

Ethereum L2 Offchain enhance saferty from ethereum, but use independent off chain solutions. Rollups – optimistic (txs are valid by default), zk proofs. State channels, sidechains, plasma, validium (like zk rollups but data does not holds in Ethereum).

Onchain Sharding – splitting blockchain into shards (discrete pieces) that will be verified by subset of validators, but now Dansksharding is used instead – use of roll ups + sharding.

What is fraud proof?

Bitcoin-Backed Altcoins и Sidechains

Make experiments without changing the original bitcoin code. User sends bitcoin to escrow, gets altcoins and after experiments burn them, so he can receive bitcoins back. To check that burn TX passed - SPV proof (bitcoin can not check what is happening in sidechains). Example, after u burn coins – generate SPV proof. Compact SPV proofs - hash < target / N

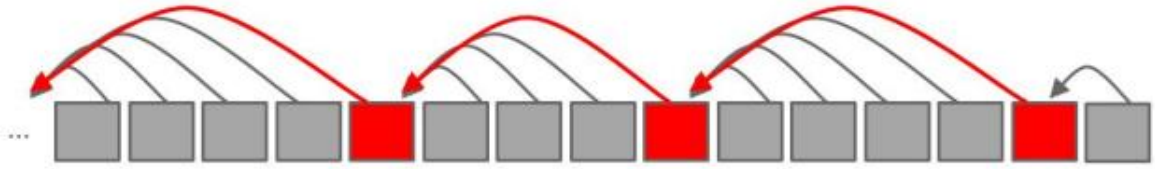


Figure 10.7: a proof-of-work skiplist. Blocks contain pointers both to the previous block and to the nearest block that satisfies *hash < target / 4*. The concept could be applied recursively, with a third level of pointers to blocks satisfying *hash < target / 16*, and so on.

This is similar to how in real life you can avoid checking every step of the work and estimate the volume based on several complex results.

Sharding

Decomposition of Sharding Blockchains

1. Node selection – set of nodes to shards
2. Epoch randomness – generate random value for config
3. Node assignment – assign nodes based on ER. Random distribution so noone can take control over the Shard. Randomness props are unpredictable, unbiased and publicly verifiable.
4. Intra Shard Consensus – consensus inside shard. Key properties are consistency and liveness. Types are eventual (one node produces block) or instant(committee) sharding
5. Cross shard transaction processing – tx that needs multiple shards. Two phase protocol – input validation and output verification. Two phase commit (2pc then every participant is ready, we can commit), transaction split, relay translation
6. Shard reconfiguration
7. Motivation Mechanism – incentives for honest based on fairness.