Off-Chain Protocols

CS 839 – Kai Mast

Blockchains Can Be Scaled at Different Layers

Layer 2: Off-Chain Protocols

State Channels

Payment Channels

Side Chains

Audit-based Systems

Layer 1: Consensus Protocols

Committe-Based Protocols

Nakamoto Consensus

Transaction Sharding

State Sharding

Layer 0: Networking

Cut-Through Routing

Gossip Protocols

Relay Networks

Block Compression

Off-Chain Protocols

Goal: Move as much data and computation off the blockchain as possible

Off-Chain in General:

- Blockchains are only involved periodically or during misbehavior
- Relies participants to check on each other (audits)
  - Usually assumes synchronous networks

Payment Channels at 10,000ft

Goal: Exchange money without the involvement of the blockchain.

- Two parties deposit money into a joint account
- When they want to transfer money, they exchange cryptographic commitments about the new balance
- Either party can close the channel and redeem their current balance at any time

The Lightning Network

- Whitepaper published by Joseph Poon and Thaddeus Dryja in 2015
- In 2016, Lightning Labs was founded by Elizabeth Stark and Olaoluwa Osuntokun, Poon and, Dryja
  - The latter two have since left
- Some people oppose the lightning network as it conflicts with "Satoshi's vision"

Most of this class is lightning specific but should apply to other payment channel solutions as well.

Recap: Smart Contracts

Bitcoin Script:

- Low-level interpreted language
- Not Turing complete; does not have loop (or jump) statements

Ethereum Virtual Machine:

- Byte code with some blockchain-specific extensions
- Supports many high-level languages such as Solidity

Payment networks can be implemented using either.

Expiring Transactions

nLockTime (absolute)

- Issuer specifies "nLockTime" in the transaction request
- Defines earliest time a transactions can be included in the chain
  - If nLockTime < blockHeight transaction cannot be included

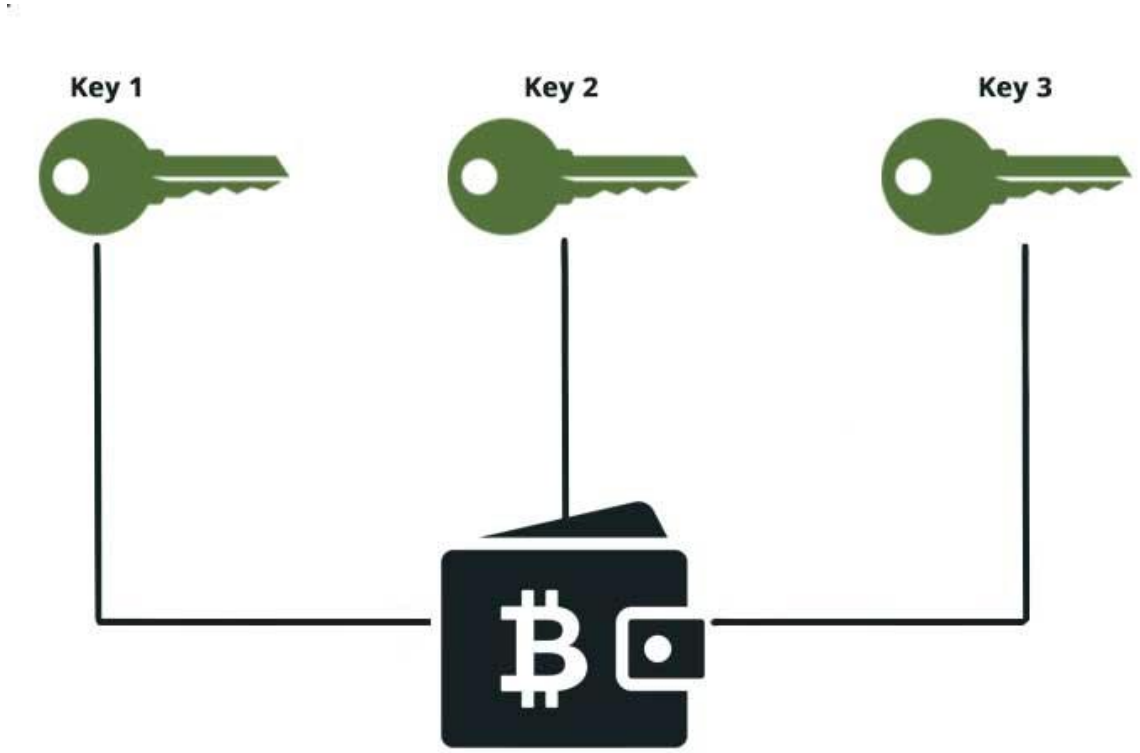
nSequenceNum (relative)

- Defines the distance (in blocks) between the transaction and its predecessor(s)
- Transaction can only be included in the chain if at least nSequenceNum blocks after the transactions which outputs it consumes

We need this primitives (+Bitcoin Script) for Lightning-style payment channels!

Multisignatures

- Multiple parties can pool money into a single UTXO
- Different variations exists, e.g.,
  - 2-of-2: Two total signatures. Both must sign to spend.
  - 2-of-3: Three total signature. Two must sign to spend.

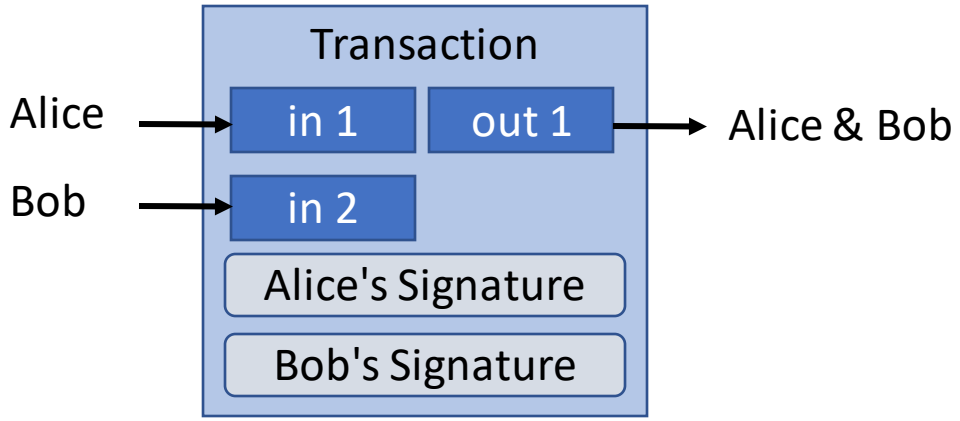


Source: coinguides.org

Payment Channels rely on Multisignatures to provide Cryptographic Commitments

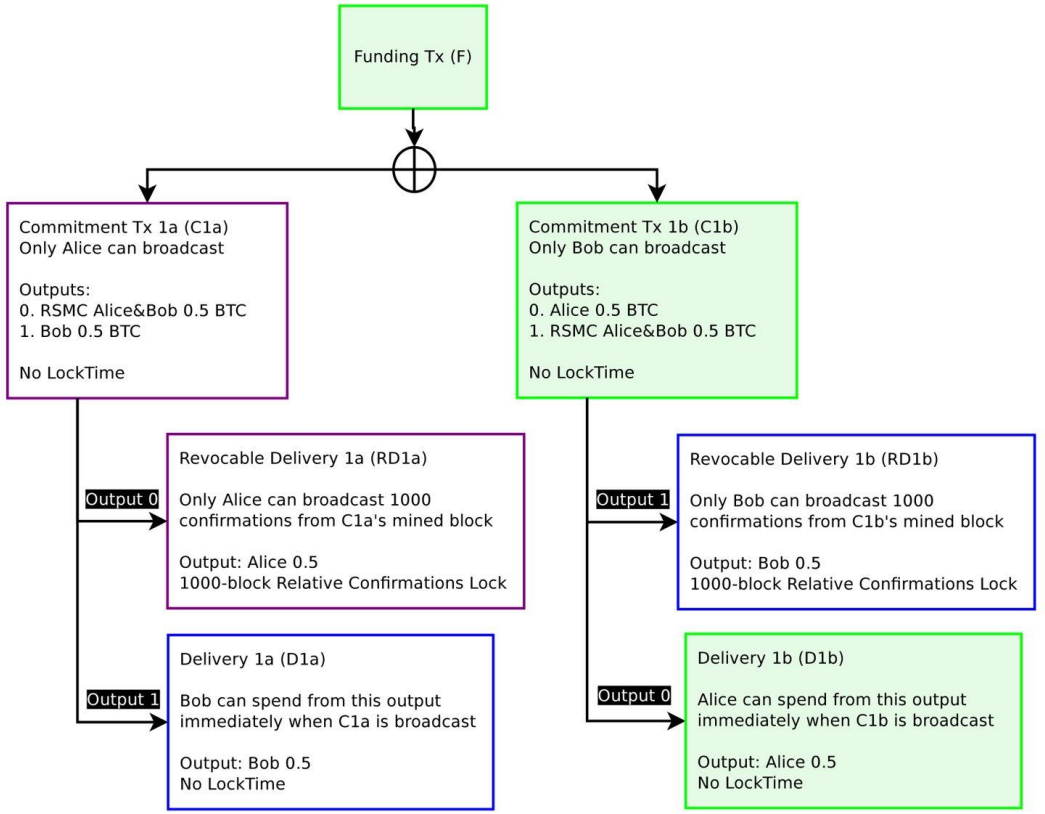
Creating a Channel

- A channel is created on-chain through a special transaction signed off by both participants
- This transaction creates a 2-of-2 multisignature UTXO
  - Needs to be signed by both parties to be spent
- Creating transaction is sent to the blockchain **after** the first commitment is generated

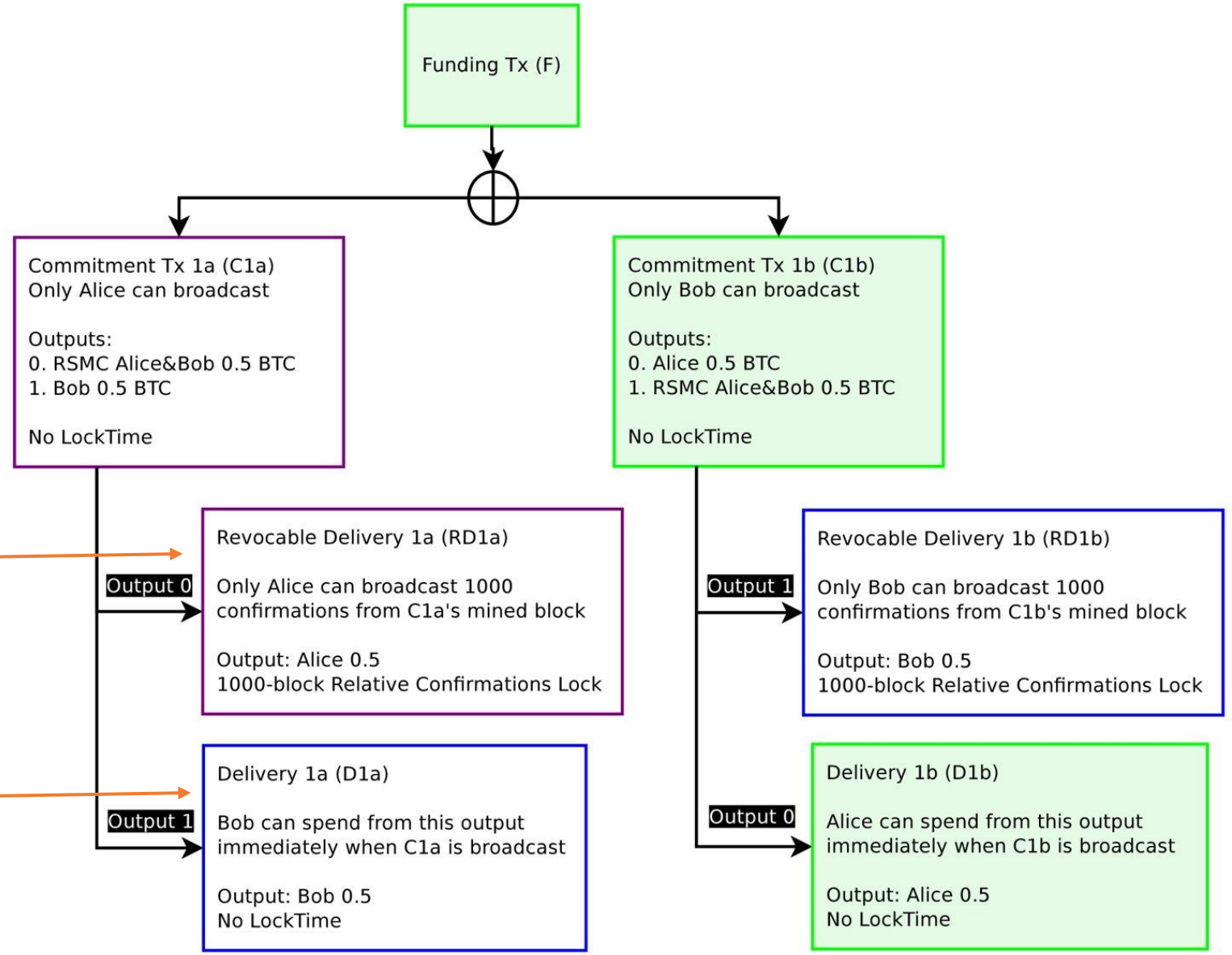


Closing Channels (One-Sided)

- Send the most recent commitment to the blockchain to transfer funds
- Other party has some time to object
  - In Lightning the timeout is 1000 blocks
- After that funds are unlocked and can be used
- There are two sets of transactions in each commitment
  - Only one can be included in the blockchain
- The closing party's funds are initially locked, while the other party can spend immediately



Closing Channels (One-Sided)



Alice needs to wait, because Bob might have a more recent commitment

Bob can immediately spend if Alice closes the channel

Revoking Deliveries

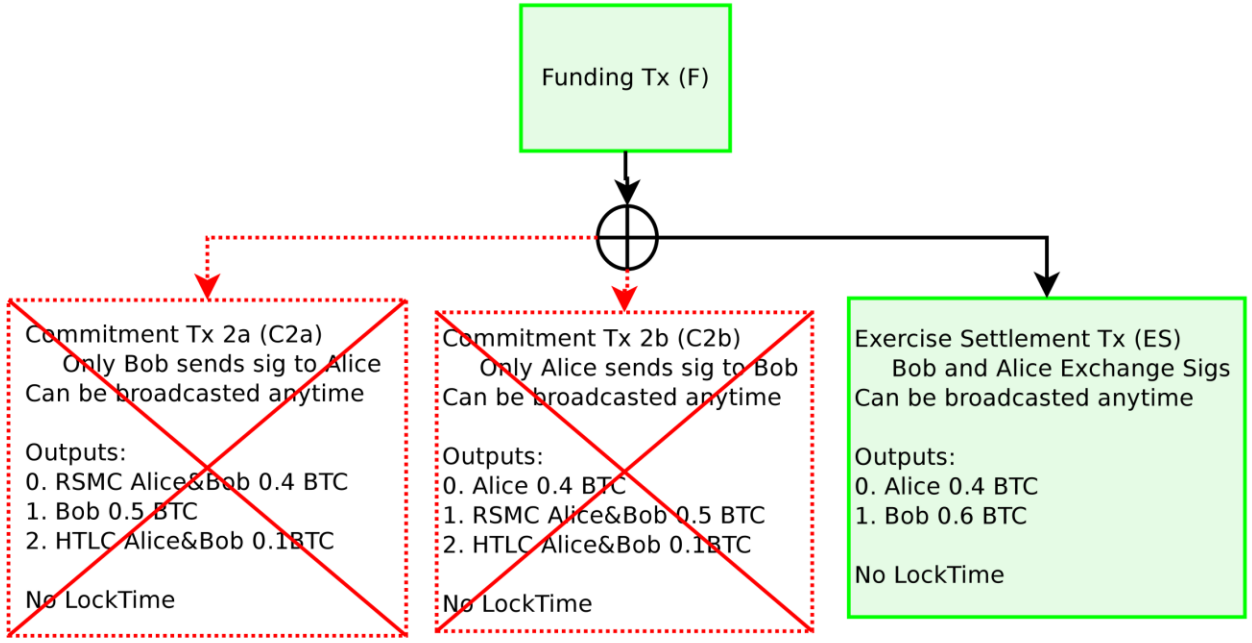
Goal: Punish participants that publish an outdated commitment on the blockchain

- A commitment contains a "breach remedy" transaction that allows overwriting the revokable delivery
- In this case, the other party gets all the money

Closing Channels (Collaborative)

Goal: Close the channel and make funds available *immediately* to both parties

- Only requires a single on chain transaction to close.
- One-sided requires at least two on-chain transactions
- After settlement no commitment transactions can be included on chain anymore

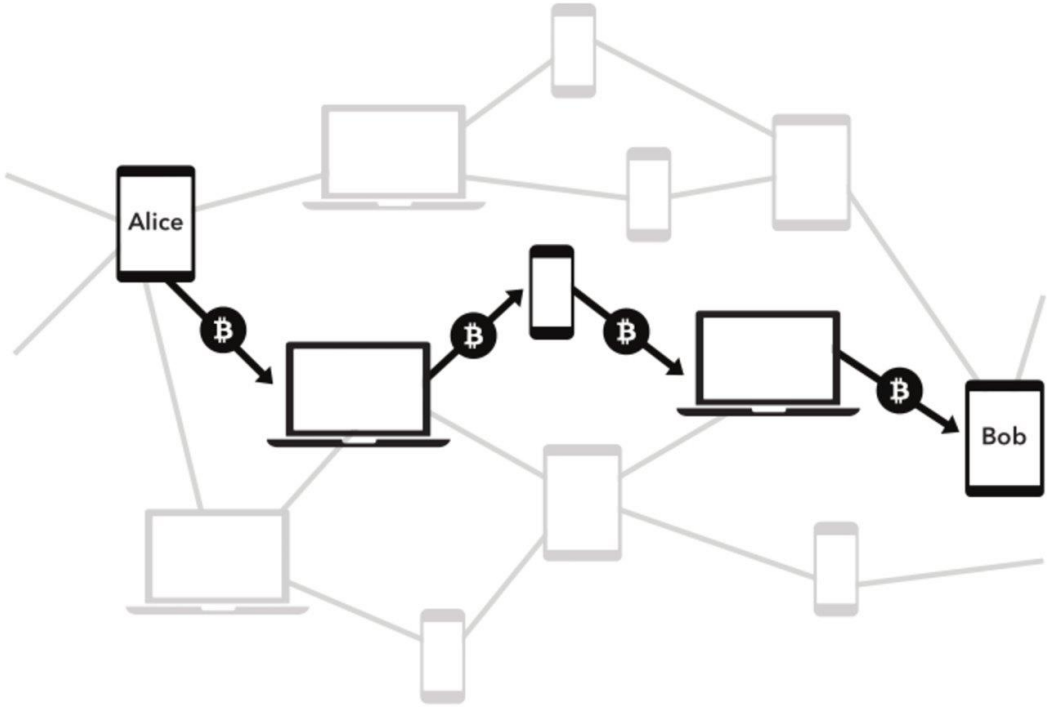


Payment Networks at 10,000ft

Goal: Route payments through a network of payment channels

Why:

- Reduces the number of payment channels required
  - Allows scaling to many users
- Payment channels are only useful if more than two payments between two parties
  - They require at least one transaction to close and one to settle

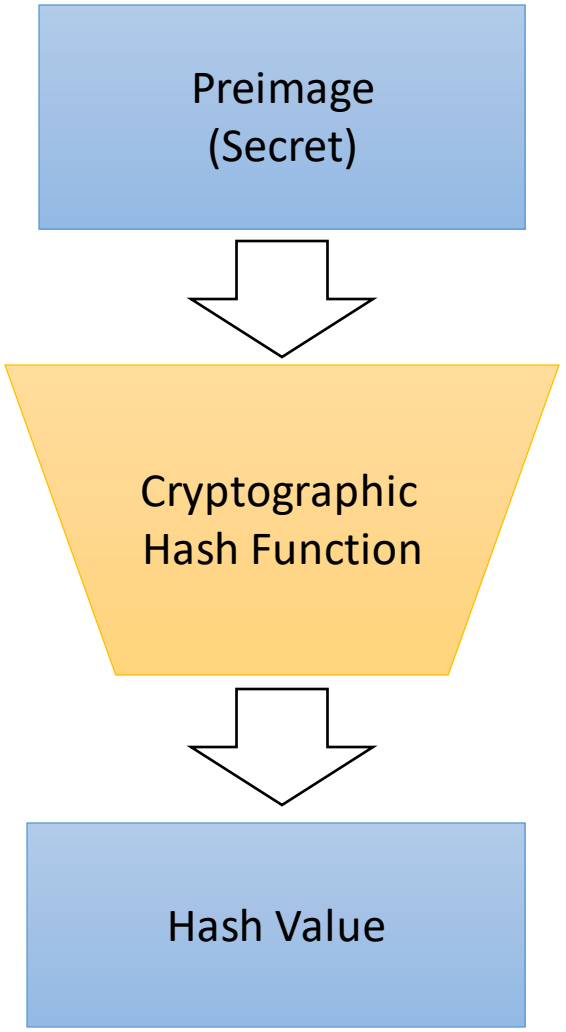


Hash Time Locked Contracts (abstractly)

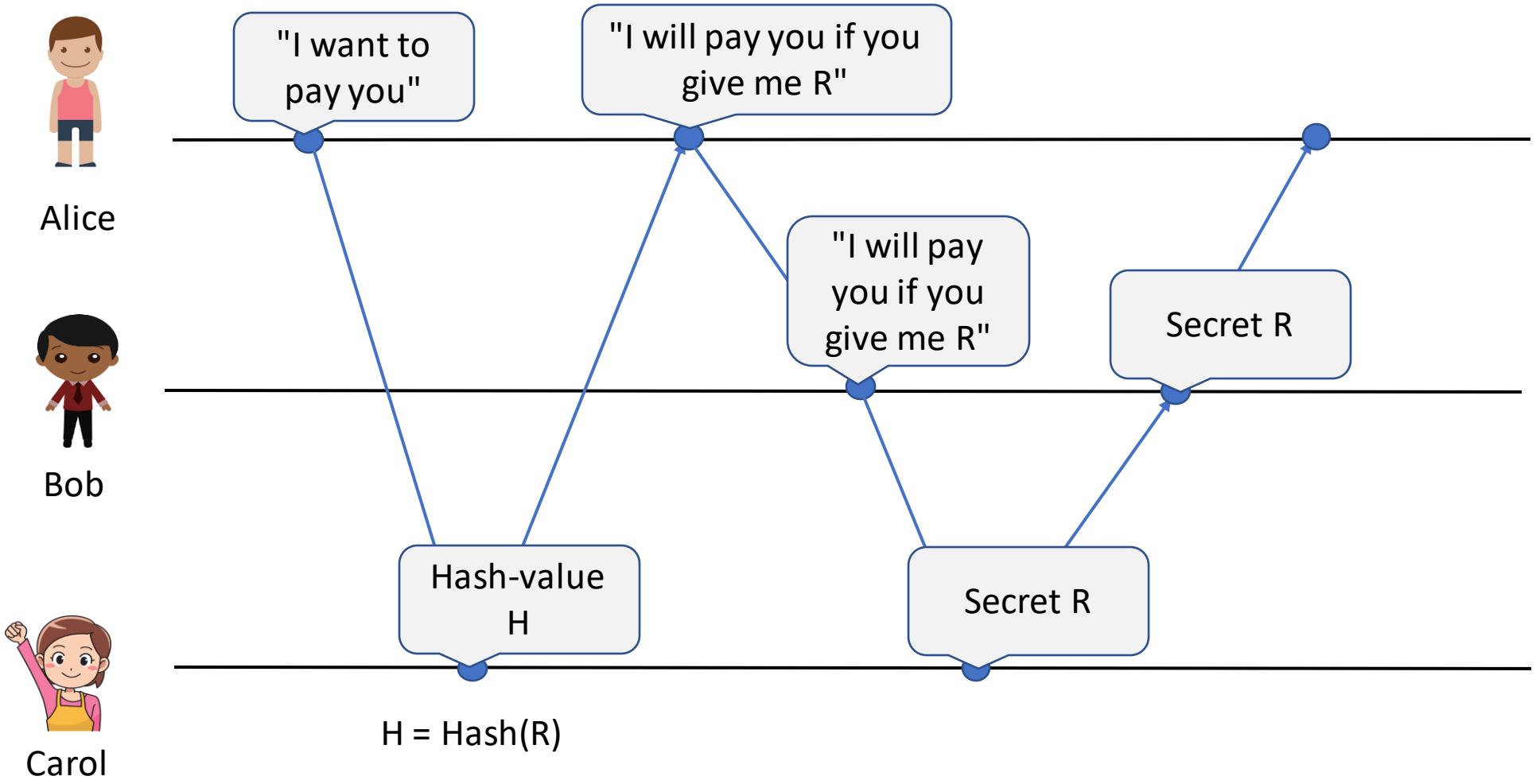
- Participants reveal a special transactions that contains a cryptographic hash value
- The transaction can only be redeemed using the secret that generates the hash
- We can chain these promises to build payment networks

HTLCs have two potential outcomes:

1. Secret is revealed, and the payment will be fulfilled
2. Transaction times out and funds are returned



Payment Networks using HTLCs



Hash Time Locked Contracts in Detail

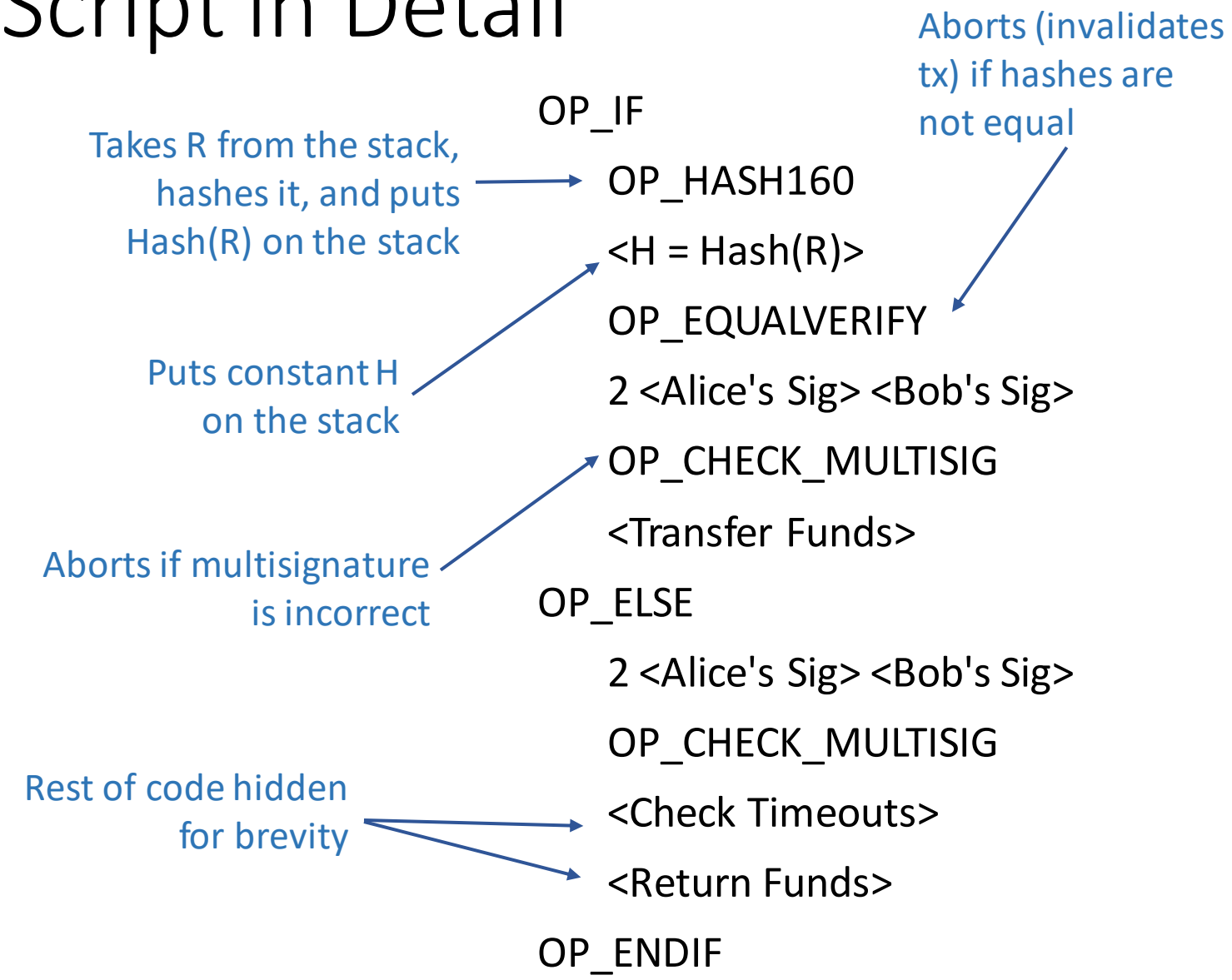
- Implemented in Bitcoin Script
- In Bitcoin, HTLCs are applied to the output of a transaction and defines how that output can be spent
- The spending transaction calls the HTLC using a set of arguments, which are put on the initial stack

HTLCs and Bitcoin Script in Detail

- An operation may put a new value on top of the stack
- An operation may also consume one or more values from the top of the stack
- Statements in <> are abbreviated

Initial values on the stack:

- For transfer: [1, R]
- For refund: [0]



State Channels?

- So far, we only looked at payment channels
- It is hard to "channelize" arbitrary applications
  - Channels only work well for application involving two (or a low number) of participants
- But approaches exist:
  - Sprites [Miller et al.; 2016]
  - Perun [Dziembowski et al.; 2018]

Perun: Virtual State Channels

Authors: Stefan Dziembowski (TU Darmstadt) and others

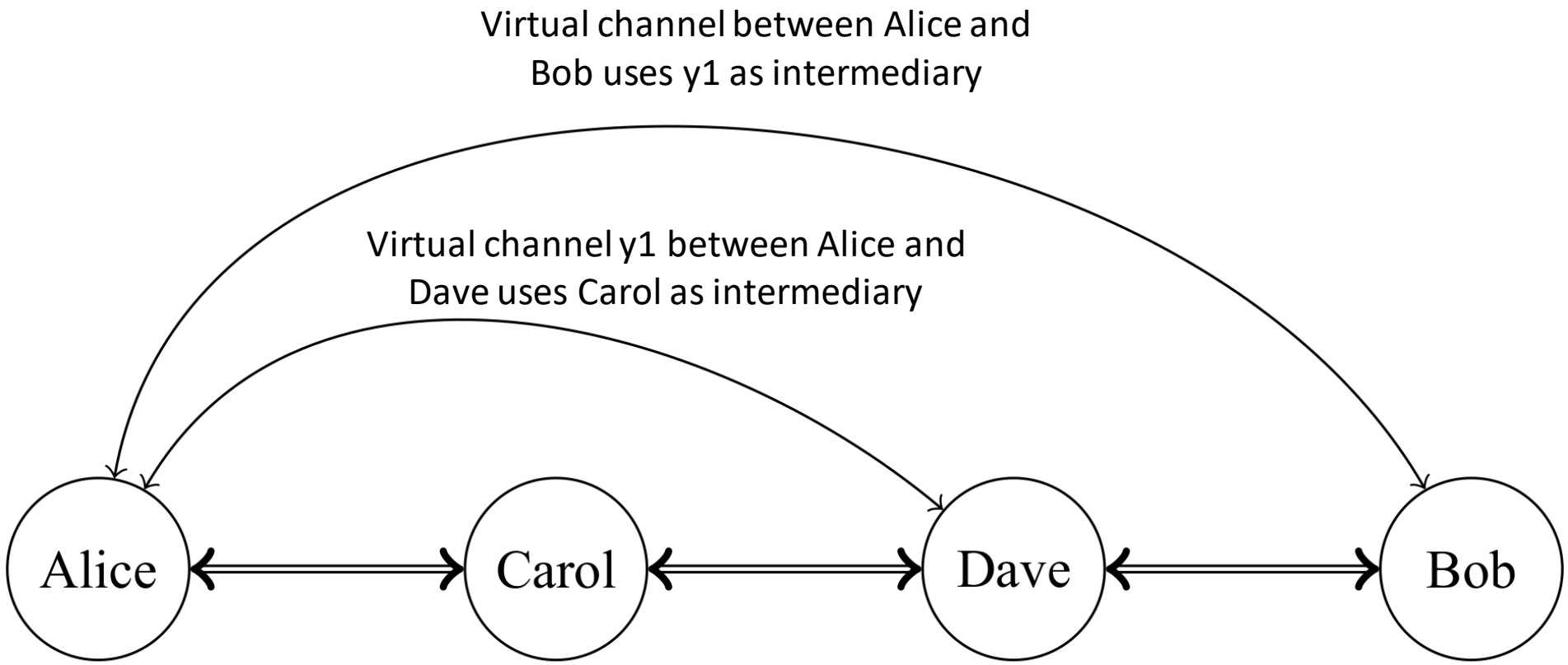
Goals:

- Don't involve intermediary for every transaction
- No need to always involve blockchain for channel setup/teardown
- Support arbitrary state, not just micropayments

Approach:

- Establish a virtual channel where, instead of the blockchain, a third party takes the role
- If third party misbehaves, we can still resort to the blockchain

Perun's Virtual Channels

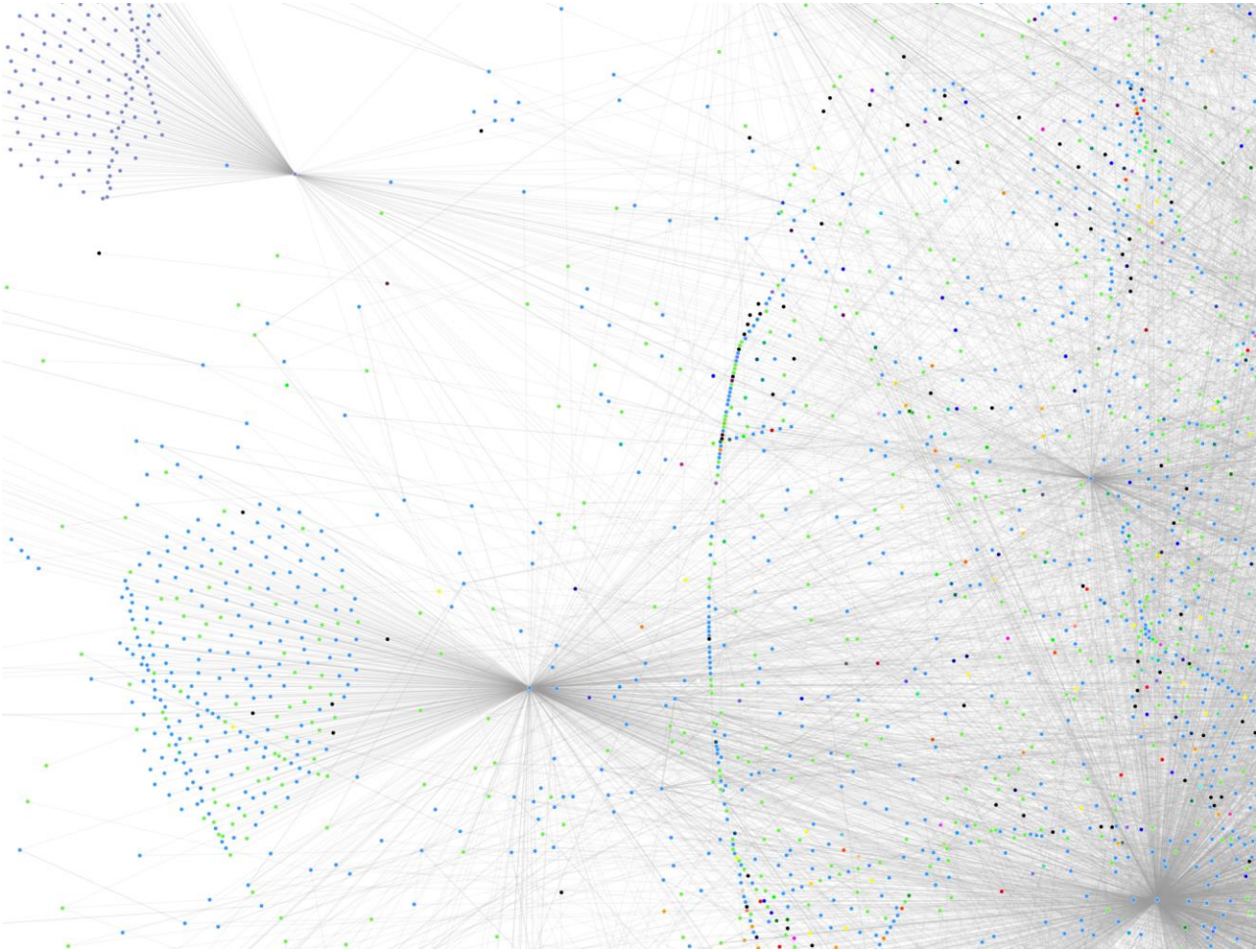


Regular Payment channels exist between Alice and Caro, Carol and Dav, and Dave and Bob

Virtual channels can be created recursively!

Centralization in Payment Channels

- Routing is hard in general
- Larger nodes might be more reliable and have more funds
- But centralization hurts the core promise of blockchains



Full graph at graph.lndexplorer.com

That's all for today

- Next time: ZCash and Private Cryptocurrencies
  - You don't need to understand all the cryptography in this paper!

- Lots of stuff we did not talk about
  - Routing
  - Watchtowers
  - Seggregated Witnesses (SegWit)
  - Multi-Path Payments
  - …