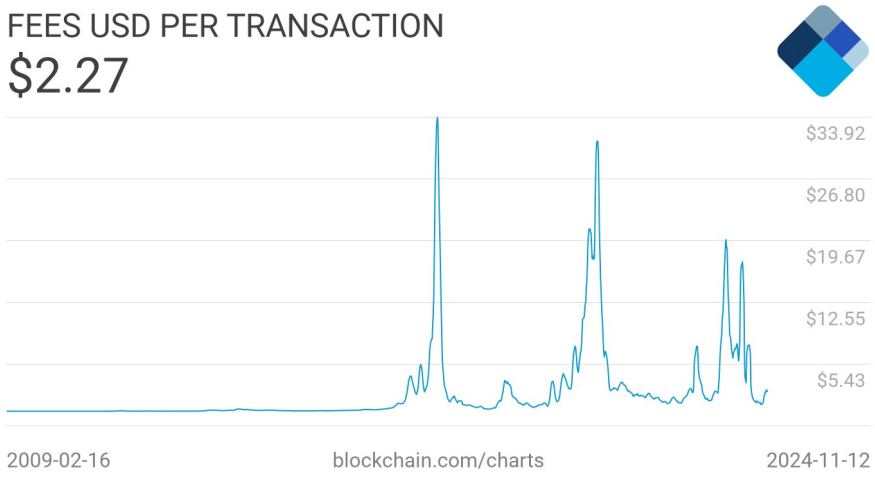
Decentralized Systems

Payment Channels

Transaction Cost



- When the number of transactions grows, transaction fees go up
- Scalability limit: fixed number of slots per second, demand may go up

First Ever Answer On Bitcoin

```
Satoshi Nakamoto wrote:
> I've been working on a new electronic cash system that's fully
> peer-to-peer, with no trusted third party.
>
> The paper is available at:
> http://www.bitcoin.org/bitcoin.pdf

We very, very much need such a system, but the way I understand your
proposal, it does not seem to scale to the required size.

For transferable proof of work tokens to have value, they must have
monetary value. To have monetary value, they must be transferred within
a very large network - for example a file trading network akin to
bittorrent.

To detect and reject a double spending event in a timely manner, one
must have most past transactions of the coins in the transaction, which,
naively implemented, requires each peer to have most past
transactions, or most past transactions that occurred recently. If
hundreds of millions of people are doing transactions, that is a lot of
bandwidth - each must know all, or a substantial part thereof.
```

Source: James A. Donald, archived by the Satoshi Nakamoto Institute

Off-Chain Transactions

- There is an inherent scalability limit on a single blockchain
  - With n users, $O(n^2)$ total storage
- Writing on a blockchain is **expensive** and it has **limited throughput**
  - We are writing **too many copies of them**
- We will look at alternatives that allow transacting **without touching the blockchain**

References

- J. Poon and T. Dryja. The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments.
- N. Narula and T. Drya, Cryptocurrency Engineering and Design. MIT OpenCourseWare, lectures 13 and 14.

The Bitcoin Lightning Network

One-Way Channels: Idea

- Incremental payment channels for **recurring payments from A to B**
- A sets aside a given amount of money in a transaction
- When they are both done, the channel is closed and each gets their remaining balance
- Key: doing it **without trust involved**

One-Way Channels: Funding

- A “channel” is just a **multi-signature** output
  - an output for Alice and Bob can only be spent by transactions signed by **both Alice and Bob**
- Alice **funds** the channel to **spend at Bob’s**

Funding transaction – on blockchain	
Input	Output
From: Alice’s output Alice’s signature	To: Alice’s and Bob’s multi-sig 10 coins

One-Way Channels: Refund

- Transactions have a **lock time:**
  - output can be spent only after the lock time (block height)
- The refund transaction is signed **before** the funding one
  - Needs segwit to work

Refund transaction – **LOCKED UNTIL December 15, held by Alice**	
Input	Output
From: funding transaction's id (txid) Bob's signature (Alice's signature not there yet)	To: Alice's address 10 coins

One-Way Channels: Full Setup

Funding transaction – on blockchain	
Input	Output
From: Alice's output Alice's signature	To: Alice's and Bob's multi-sig 10 coins

Refund transaction – **LOCKED UNTIL December 15, held by Alice**	
Input	Output
From: funding txid Bob's signature (Alice's signature not there yet)	To: Alice's address 10 coins

- Now, in the worst case, Alice can simply wait until she can get the refund
  - Alice won't lose money

One-Way Channels: Spending (1)

Funding transaction – on blockchain	
Input	Output
From: Alice's output Alice's signature	To: Alice's and Bob's multi-sig 10 coins

Spending transaction 1 **Held by Bob**	
Input	Output
From: funding txid Alice's signature (Bob's signature not there yet)	To: Alice's address 9 coins
	To: Bob's address 1 coin

- Bob can post the transaction and **close the channel**
  - He should do before the refund transaction becomes valid

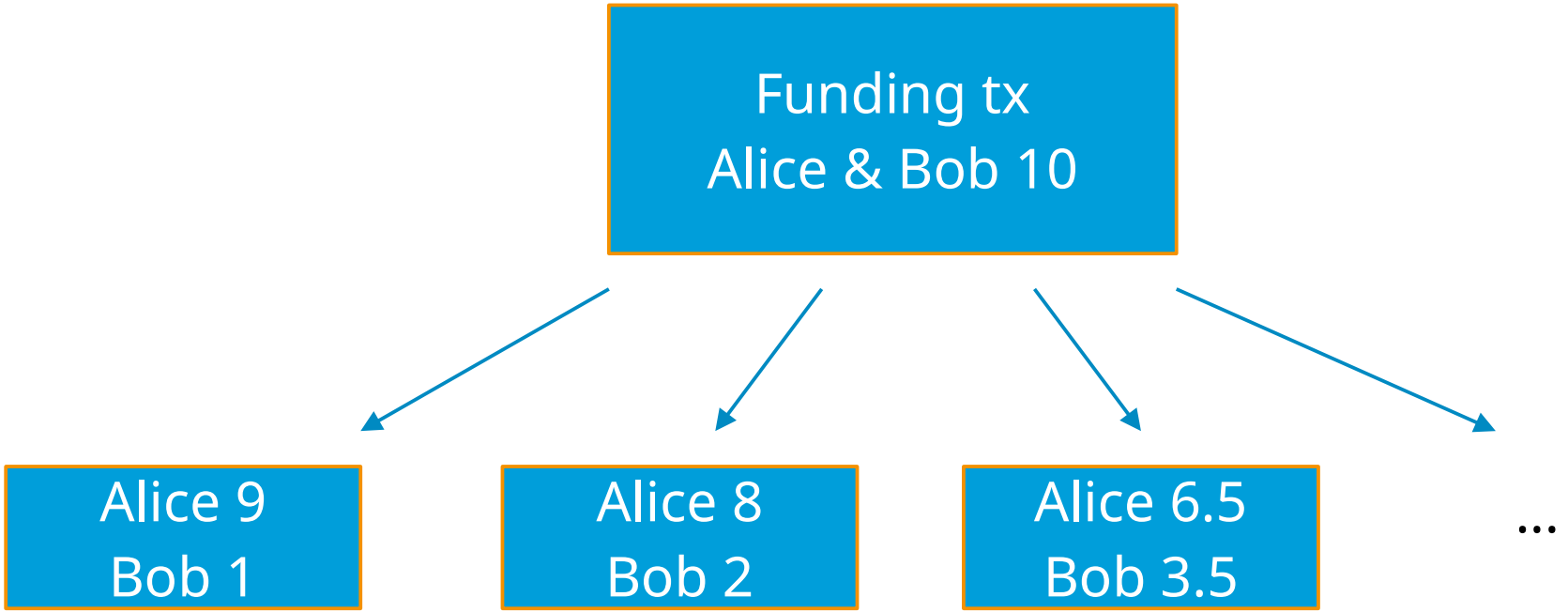
One-Way Channels: Spending (2)

Spending transaction 1 **Held by Bob**	
Input	Output
From: funding txid Alice's signature (Bob's signature not there yet)	To: Alice's address 9 coins
	To: Bob's address 1 coin

Spending transaction 2 **Held by Bob**	
Input	Output
From: funding txid Alice's signature (Bob's signature not there yet)	To: Alice's address 8 coins
	To: Bob's address 2 coins

- Now Bob can forget about the first transaction: the **second is better**
- Spending transactions **aren't posted to the blockchain**
  - Until A and B want to close the channel

One-Way Channels: Spending (3)



One-Way Channels: Outcome

- Bob keeps getting half-signed transactions with more money going to him
  - **Fast** (off-chain payments) and with **no fees**
  - He **must** sign and broadcast one before the refund time!
- Currency flows **only one way**
  - From Alice to Bob, **never vice versa**
- Channels are only useful **before they expire**
  - Expiration can’t be too far away: if Bob disappears Alice’s funds are locked
- Common case: how many transactions on the blockchain?
  - Two: funding and closing (last expense by Alice)

Bidirectional Payment Channels

- A construct that, like what we have seen until now
  - Avoids paying on-chain most transactions
  - Doesn’t require trust between Alice and Bob
- But
  - Is bidirectional (money can flow in either direction)
  - Doesn’t expire
- We’ll follow the Bitcoin Lightning protocol
  - Raiden is the equivalent solution for Ethereum

Bitcoin Timing Opcodes

`OP_CHECKSEQUENCEVERIFY`

- Argument: sequence
- Requires that the input has at least n confirmations
  - i.e., it is ≥n blocks old
- Otherwise, the transaction fails

`OP_CHECKLOCKTIMEVERIFY`

- Argument: locktime
- Requires that the transactions' output has at least n confirmations to be spent
- Otherwise, the transaction fails

Using the New Opcodes

- “This transaction’s output is spendable by *[the private key corresponding to public]* key X after 100 blocks”
- “or by this other *[private key corresponding to public]* key Y”
- “or by *[…]* keys W and Z in conjunction”

Revokable Transactions

Held by **Alice**	
Input	Output
From: funding txid Bob's signature (Alice's signature not there yet)	To: Alice's key after 100 blocks OR To Bob and AliceR keys 2 coins
	To: Bob's address 8 coins

Held by **Bob**	
Input	Output
From: funding txid Alice's signature (Bob's signature not there yet)	To: Alice's address 2 coins
	To: Bob's key after 100 blocks OR To Alice and BobR keys 8 coins

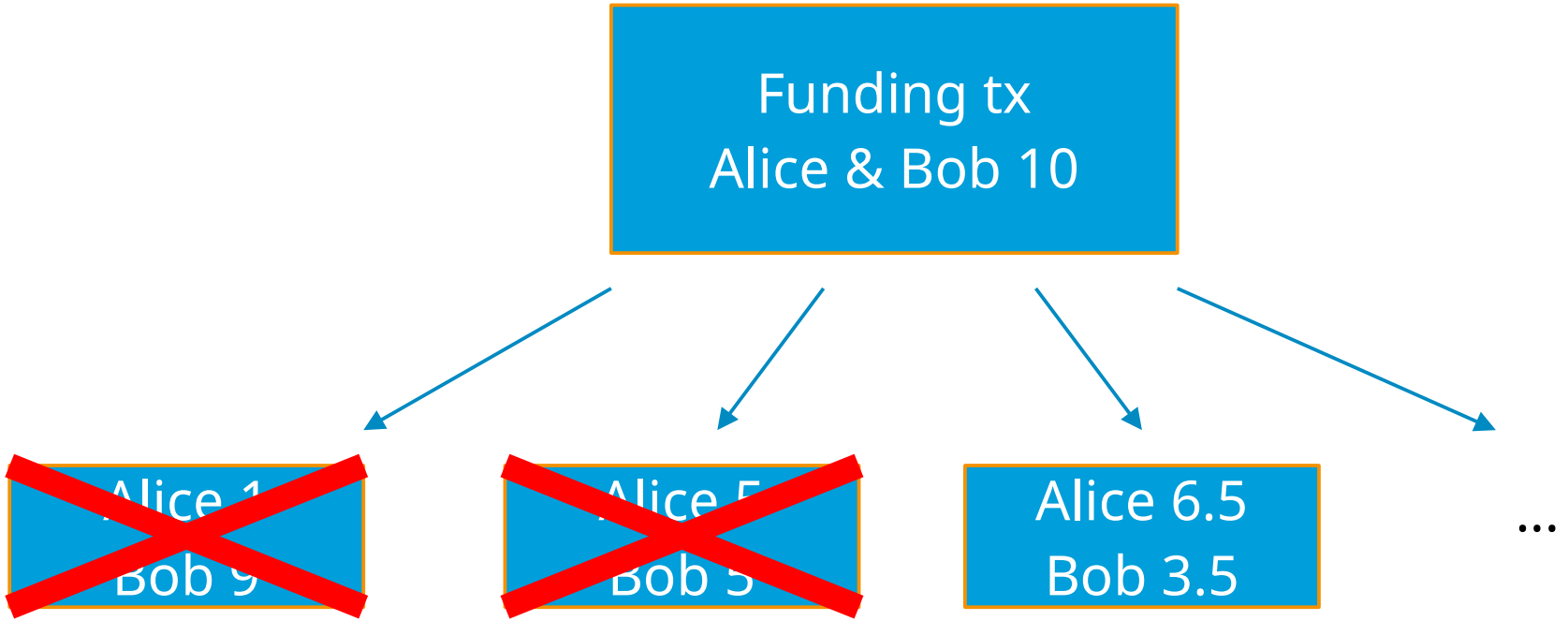
- Both Alice and Bob build new AliceR/BobR key pairs

Reveal to Revoke

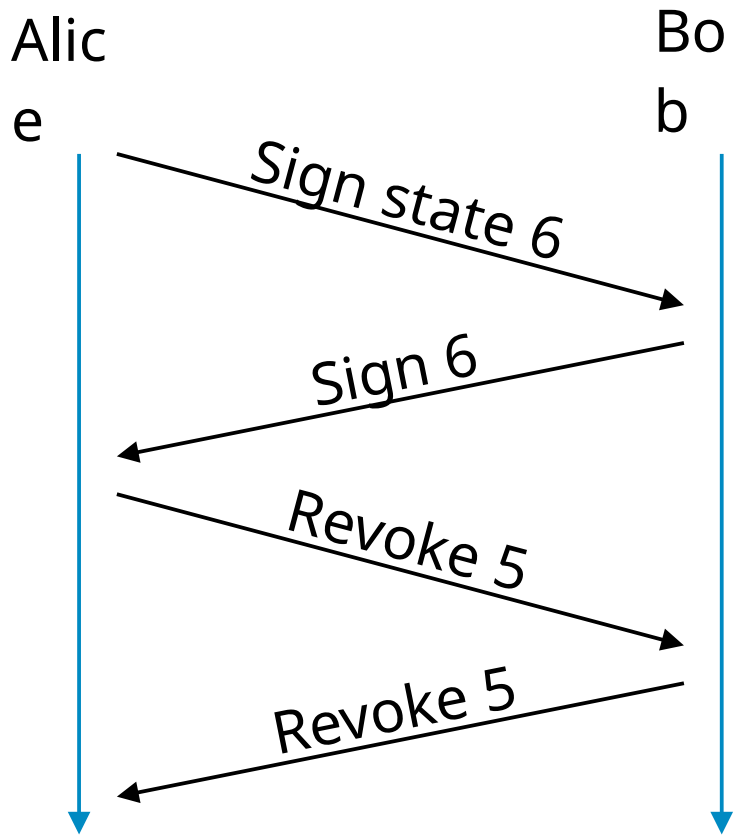
- Bob can **revoke** this transaction by sending Alice the BobR private key
- If he ever tries to broadcast this transaction, Alice can use BobR to recover **all** the funds in the channel
- Key assumption: **Alice has to remain online!**

<table>
<tr><td colspan="2">Held by Alice</td></tr>
<tr><td>Input</td><td>Output</td></tr>
<tr><td rowspan="2">From: funding txid
Bob’s signature (Alice’s signature not there yet)</td><td>To: Alice’s key after 100 blocks
OR
To Bob and AliceR keys
2 coins</td></tr>
<tr><td>To: Bob’s address
8 coins</td></tr>
</table>

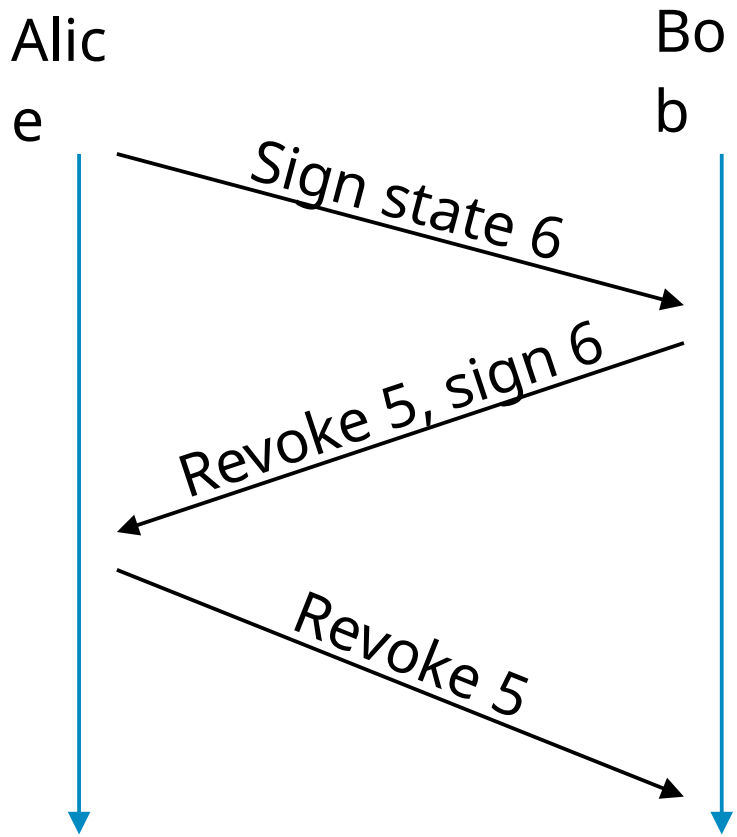
Updating States



Message Order



Optimized Message Order

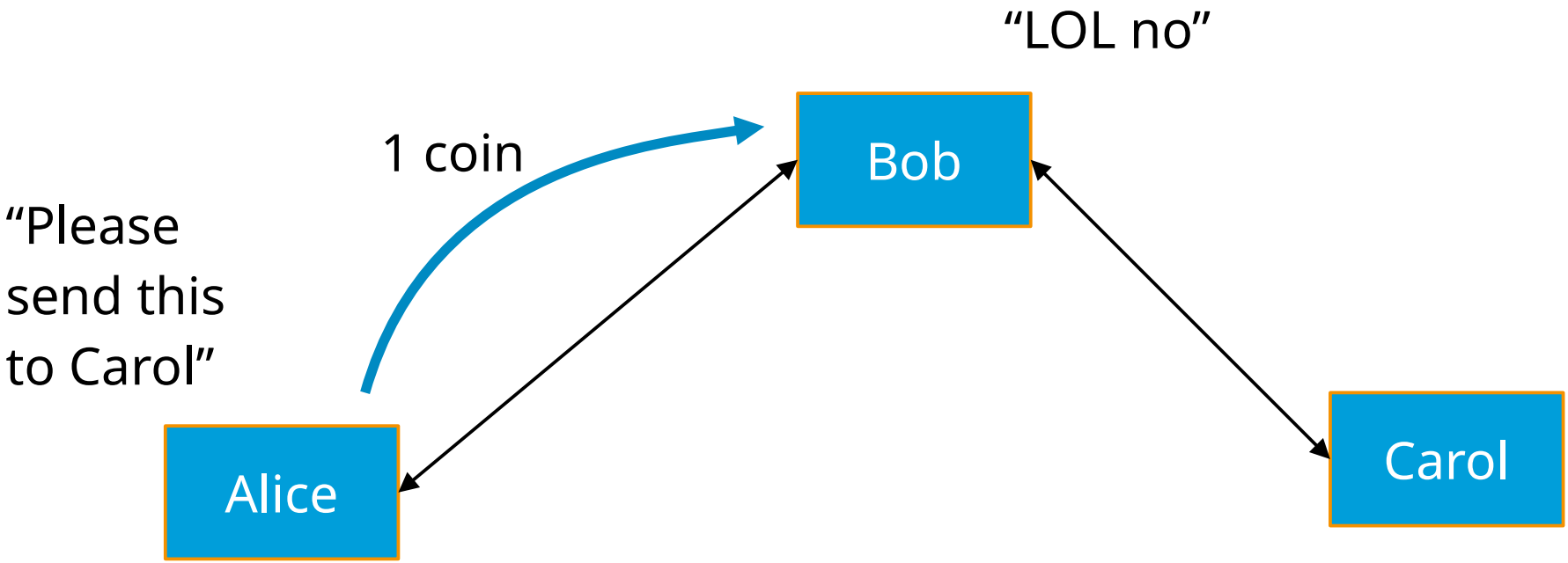


Storing Revocation Secrets

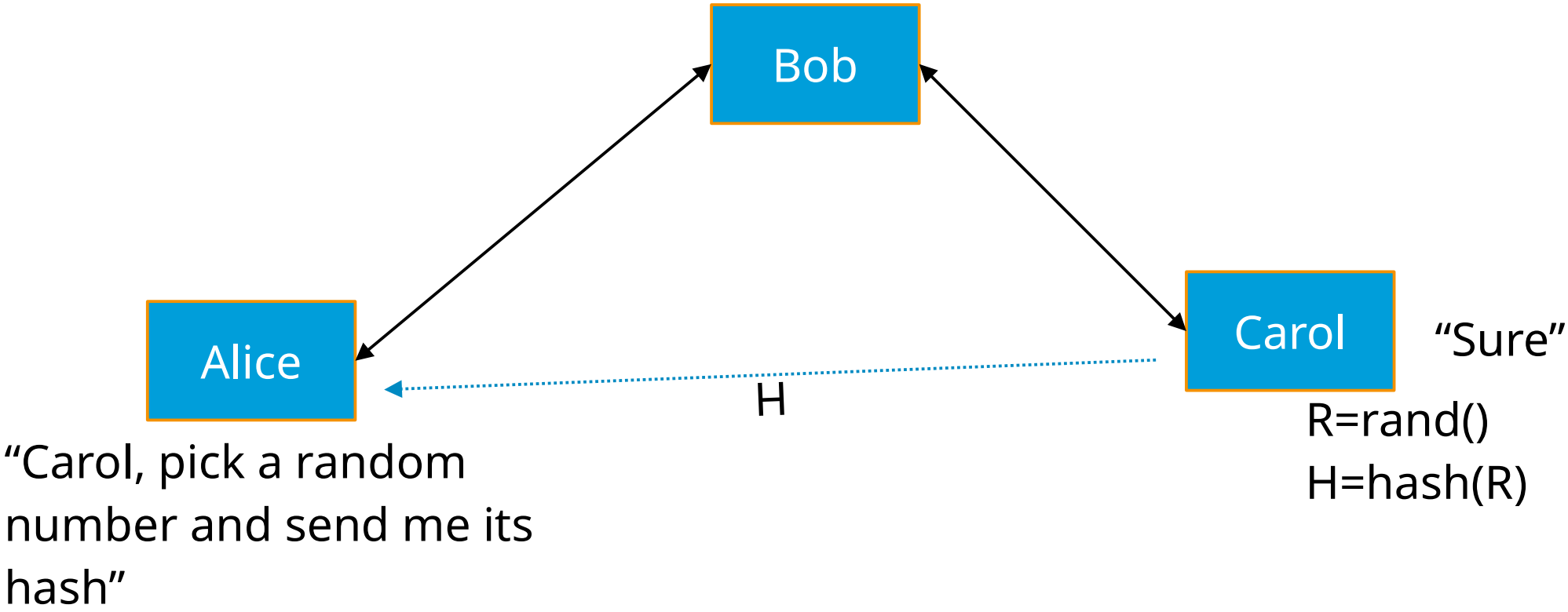
- How many revocation secrets should one store?
  - (Almost) all those in the history of the channel!
  - 32 bytes per state
- A clever optimization: Elkrem tree
- With n transactions
  - You just need to store O(log n) secrets
  - You need to perform O(log n) hashes to get to the secret

Multi-Hop Transfers

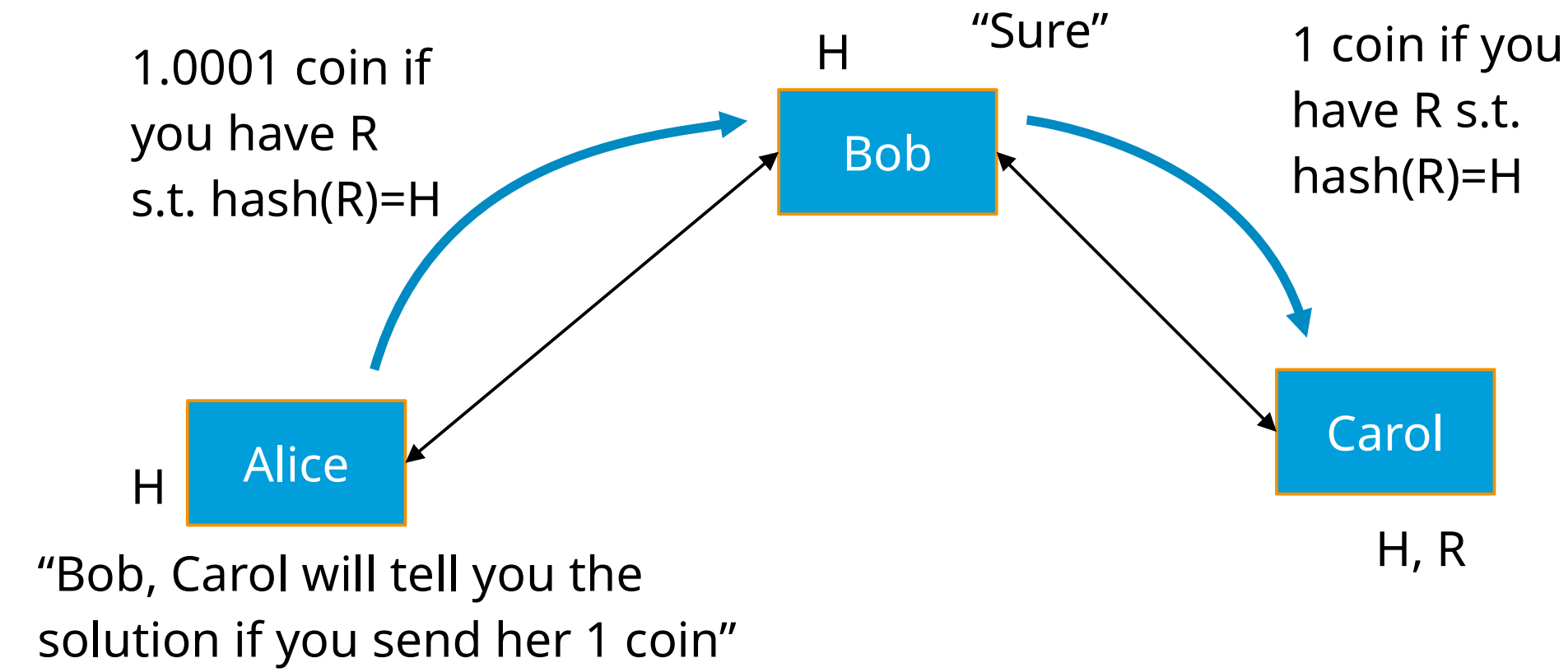
Trust Issues



The Method (1)


Bob
Alice
Carol
“Sure”
H
R=rand()
H=hash(R)
“Carol, pick a random number and send me its hash”

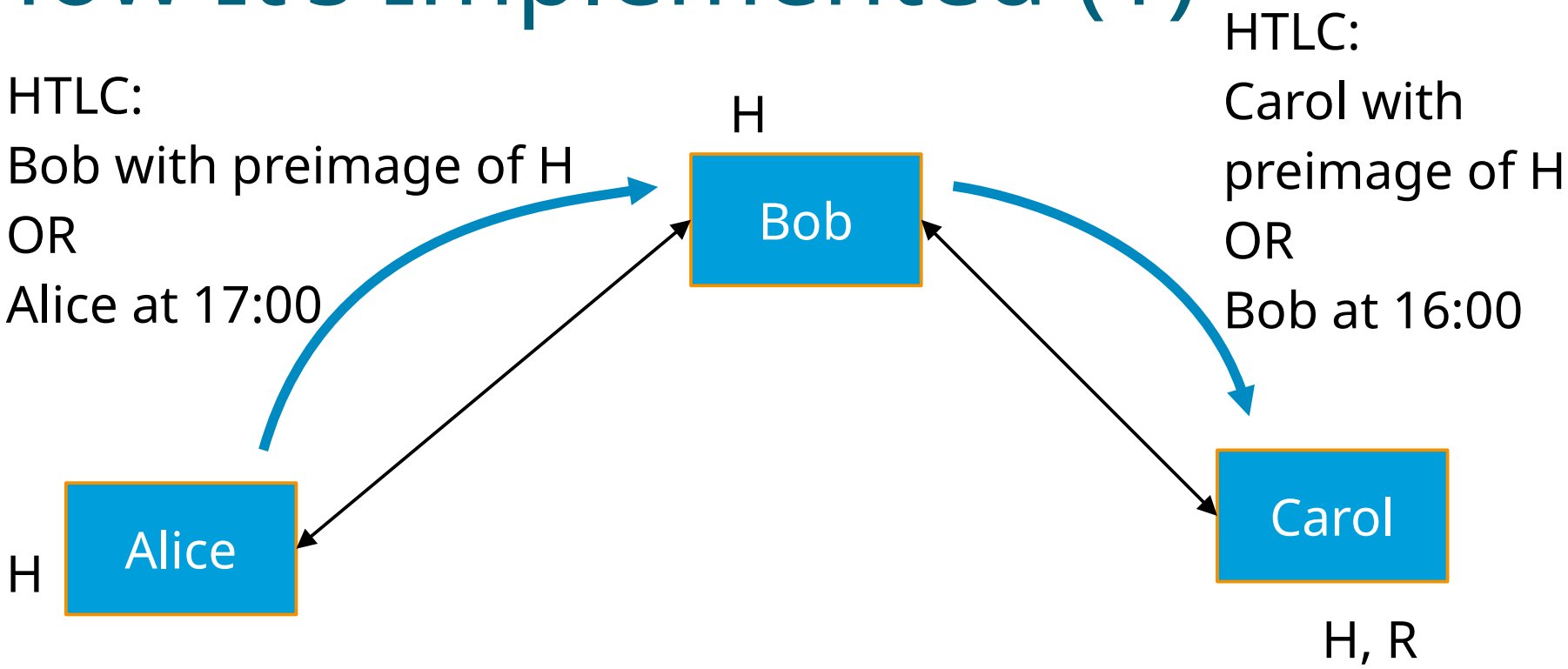
The Method (2)


1.0001 coin if you have R s.t. hash(R)=H
H
"Sure"
1 coin if you have R s.t. hash(R)=H
Bob
Alice
H
Carol
H, R
"Bob, Carol will tell you the solution if you send her 1 coin"

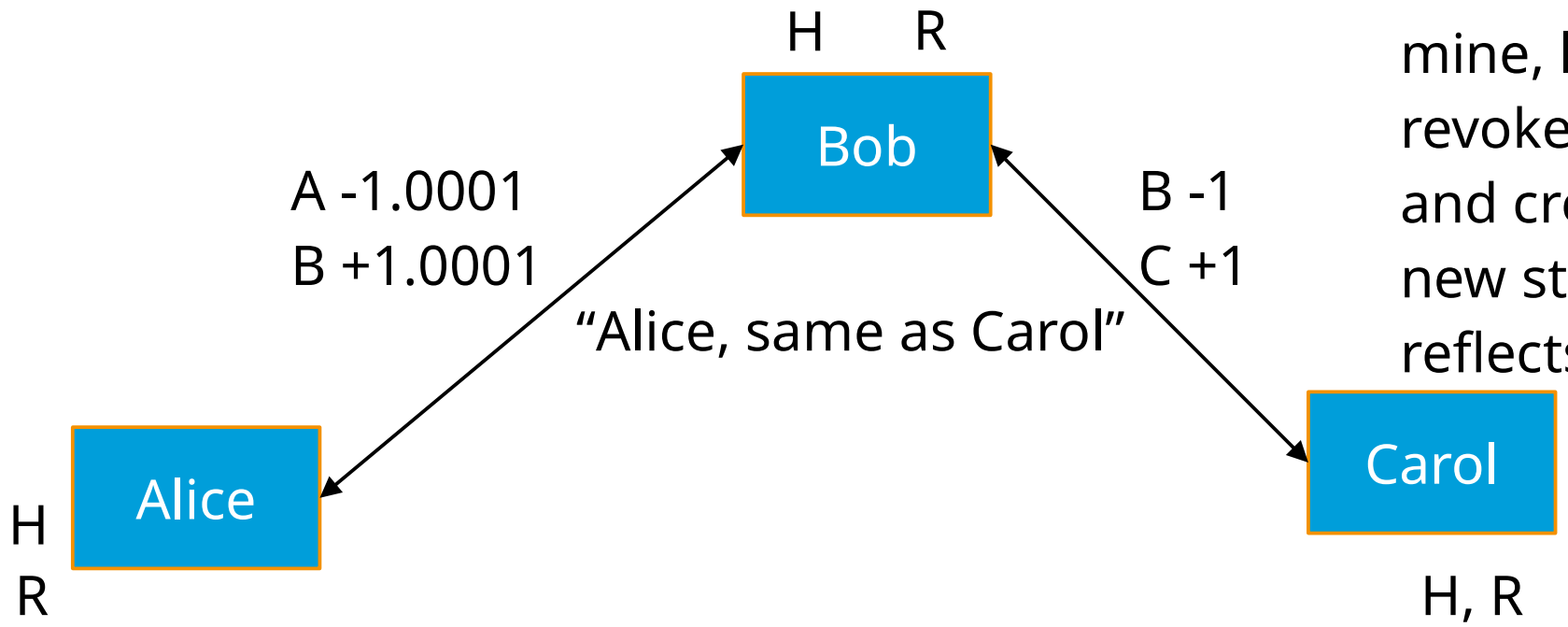
Hash/Time Locked Contract (HTLC)

<table>
<tr><td colspan="2">Held by Bob</td></tr>
<tr><td>Input</td><td>Output</td></tr>
<tr><td rowspan="3">From: funding txid
Alice’s signature
(Bob’s signature not there yet)</td><td>To: Alice’s address
0.9999 coin</td></tr>
<tr><td>To Bob’s key after 100 blocks
OR
To Alice and BobR keys
8 coins</td></tr>
<tr><td>To Bob if he knows the preimage to H
OR
To Alice at 17:00 (uses block height)
1.0001 coin</td></tr>
</table>

How It’s Implemented (1)



How It's Implemented (2)



"Bob, here's R, the money is mine, let's revoke the HTLC and create a new state that reflects that"

Implementation

Revocation Keys as Hash Preimages

Held by **Alice**	
Input	Output
From: funding txid Bob's signature (Alice's signature not there yet)	To: Alice's key after 100 blocks OR To Bob and AliceR keys 2 coins
	To: Bob's address 8 coins

Held by **Alice**	
Input	Output
From: funding txid Bob's signature (Alice's signature not there yet)	To: Alice's key after 100 blocks OR To Bob **if he knows P s.t. H(P)=AliceR** 2 coins
	To: Bob's address 8 coins

- Less storage on blockchain (~20B vs ~80B), same concept

An Elliptic Curve-Based Trick

- Remember the slides on elliptic curves? Say we have two private/public key pairs (b, B) and (c, C)
- (b+c, B+C) is a working key pair because
  - bG=B, cG=C → (b+c)G = B+C

ECC: scalar multiplication 

- Given a known **base point G**, it can be multiplied by a private key **Sk** to find the corresponding public key **Pk**

Even Better: Just Sum Keys

Held by **Alice**	
Input	Output
From: funding txid Bob's signature (Alice's signature not there yet)	To: Alice's key after 100 blocks OR To Bob and AliceR keys 2 coins
	To: Bob's address 8 coins

Held by **Alice**	
Input	Output
From: funding txid Bob's signature (Alice's signature not there yet)	To Alice's key after 100 blocks OR To **KeyR=Bob+AliceR** 2 coins
	To: Bob's address 8 coins

- Even shorter script
- Compactness matters this much on the blockchain!

The Reduced Script

```
OP_IF KeyR
OP_ELSE
   <delay>
   OP_CHECKSEQUENCEVERIFY
   OP_DROP
   KeyA
OP_ENDIF
OP_CHECKSIG
```

- Stack-based language
- Working inputs on stack:
  - 1 SigR
  - 0 SigA (after the delay is passed)
- OP_CSV doesn’t consume the stack for the soft fork backwards compatibility

The Lightning P2P Network

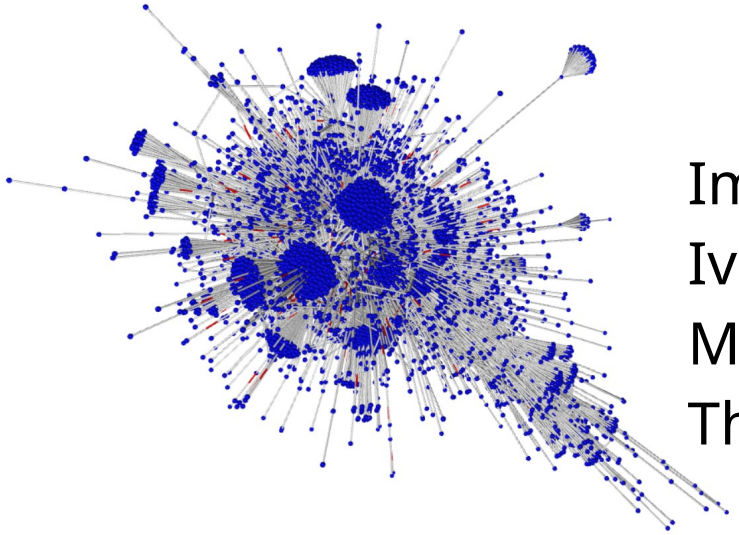
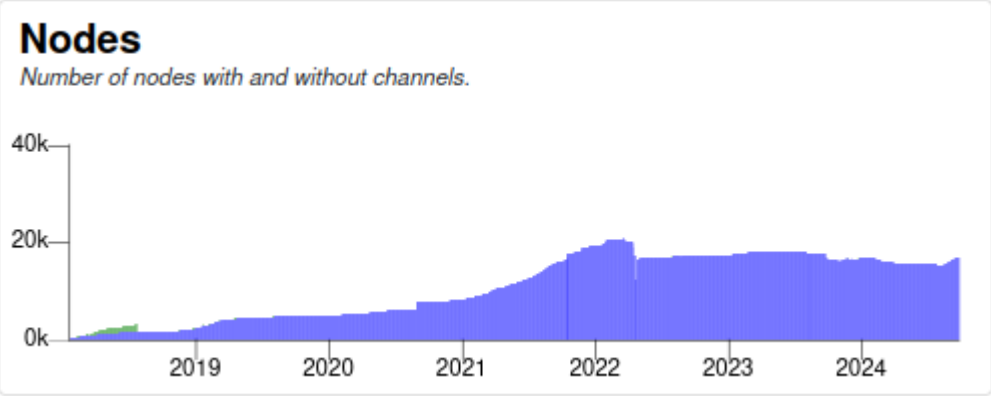


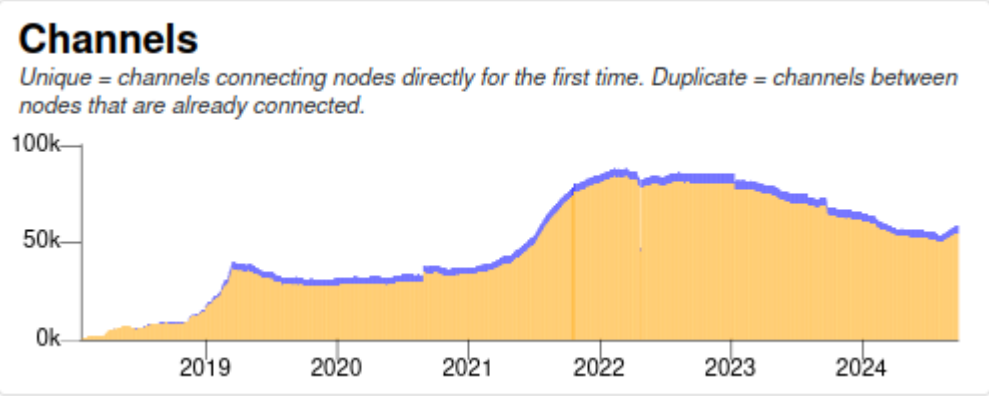
Image from Ivan Gallo's Master Thesis

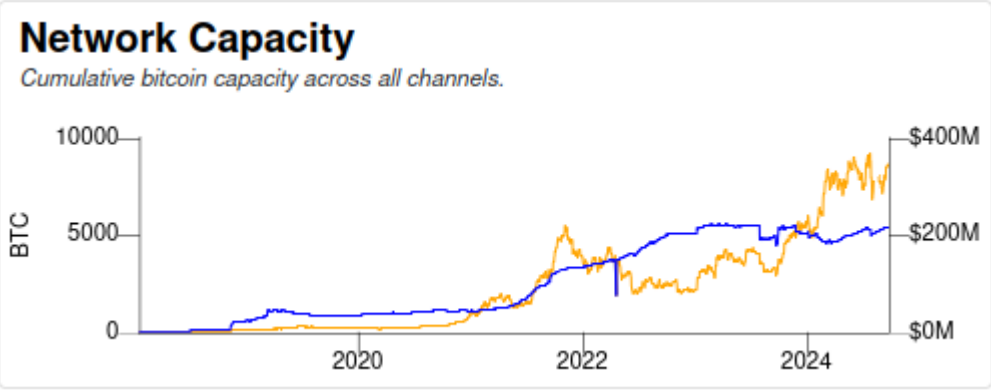
The Network

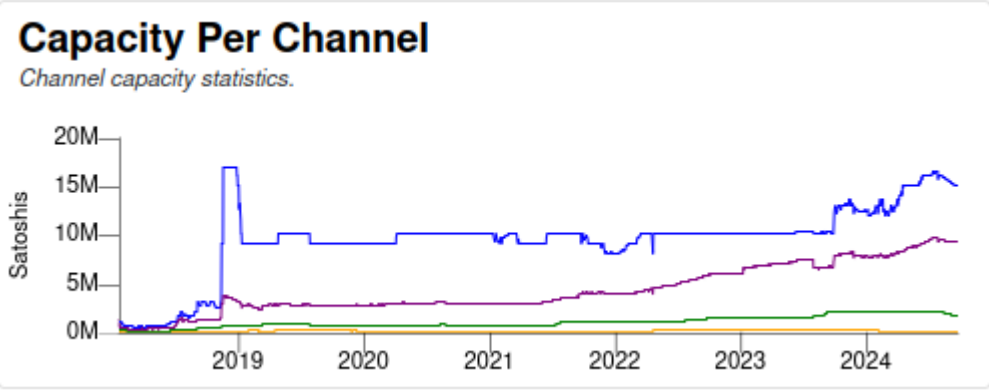
- Nodes use **gossiping** to exchange information about the available channels
- Finding a way to send money from A to B is a graph problem: find paths from A to B
  - Easy if you have all the network
  - Some research on doing it with privacy about existing channels
- Not very big, but appears to have grown fast
  - +1212% in 2021-2023 (source: river.com)

Some Stats









Source: Bitcoin Visuals

Cross-Chain Swaps

The Tragedy of Centralized Exchanges

Mt. Gox

11 languages

Article Talk

From Wikipedi

Mt. Gox was a
handling over
ceased operati
thousands of b

In February 20
filed for bankr
proceedings.[1
disappearance
New evidence
that "most or a
cryptocurrency

FTX

20 languages

Article Talk

From Wikipedia, the free encyclopedia

For other uses, see FTX (disambiguation).

FTX Trading Ltd., commonly known as **FTX** (short for "Futures Ex
company that formerly operated a fraud-ridden cryptocurrency exc
fund.[6][7] The exchange was founded in 2019 by Sam Bankman-Fri
peak in July 2021, the company had over one million users and was
cryptocurrency exchange by volume.[8][9] FTX is incorporated in An
headquartered in the Bahamas.[10] FTX is closely associated with F
available to US residents.[11]

Since November 11, 2022, FTX has been in Chapter 11 bankruptcy
system.[12][13][14][15] Public concern began with rumors of unethica
company transfers of client funds. In November 2022 CoinDesk als
that FTX's partner firm Alameda Research held a significant portion

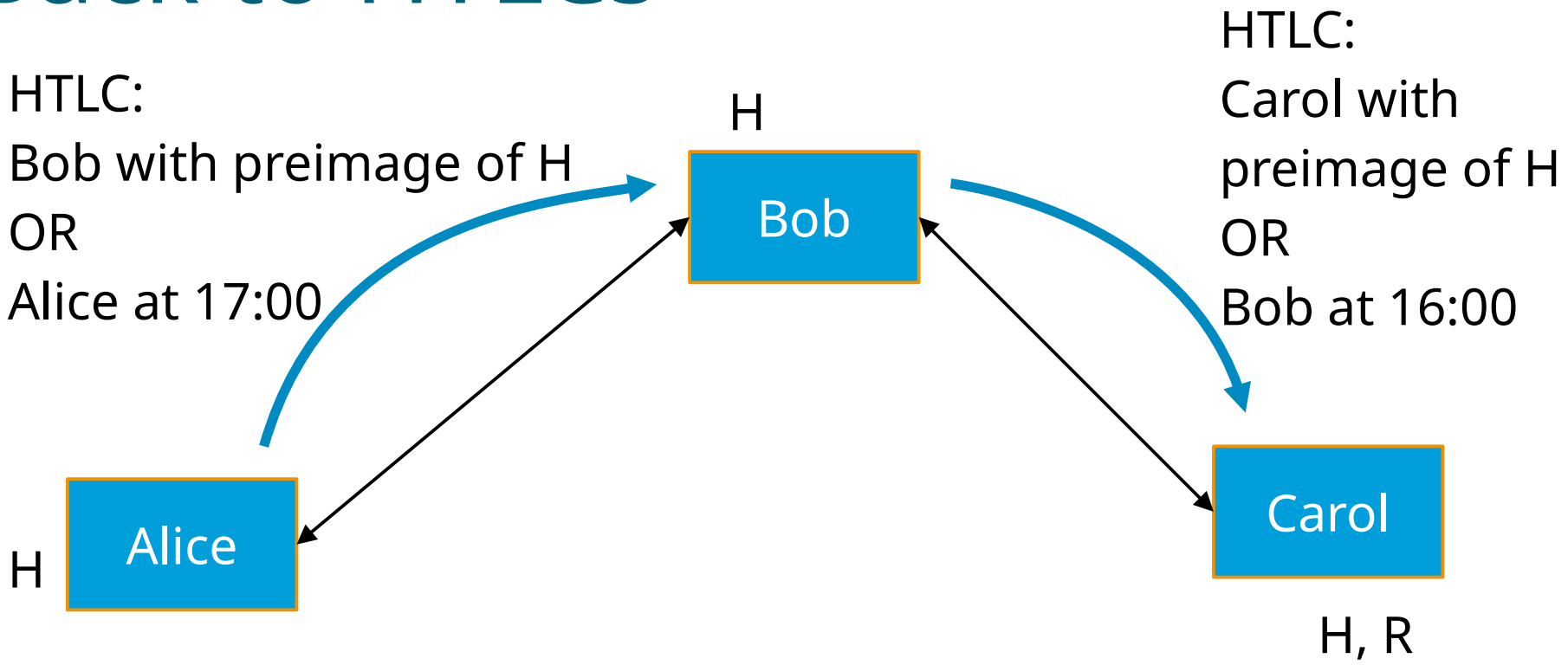
Crypto giant Binance admits to money laundering and agrees to pay $4.3bn

CEO Changpeng Zhao will resign and pay a $50m individual fine as part of a plea deal with the US Justice Department

Trustless Cross-Chain Trading

- Requirement:
  - A sends X COIN1 to B
  - B sends Y COIN2 to A
- A and B agree on the exchange rate
  - But don’t trust each other!
- Idea: use payment channels for this!

Back to HTLCs

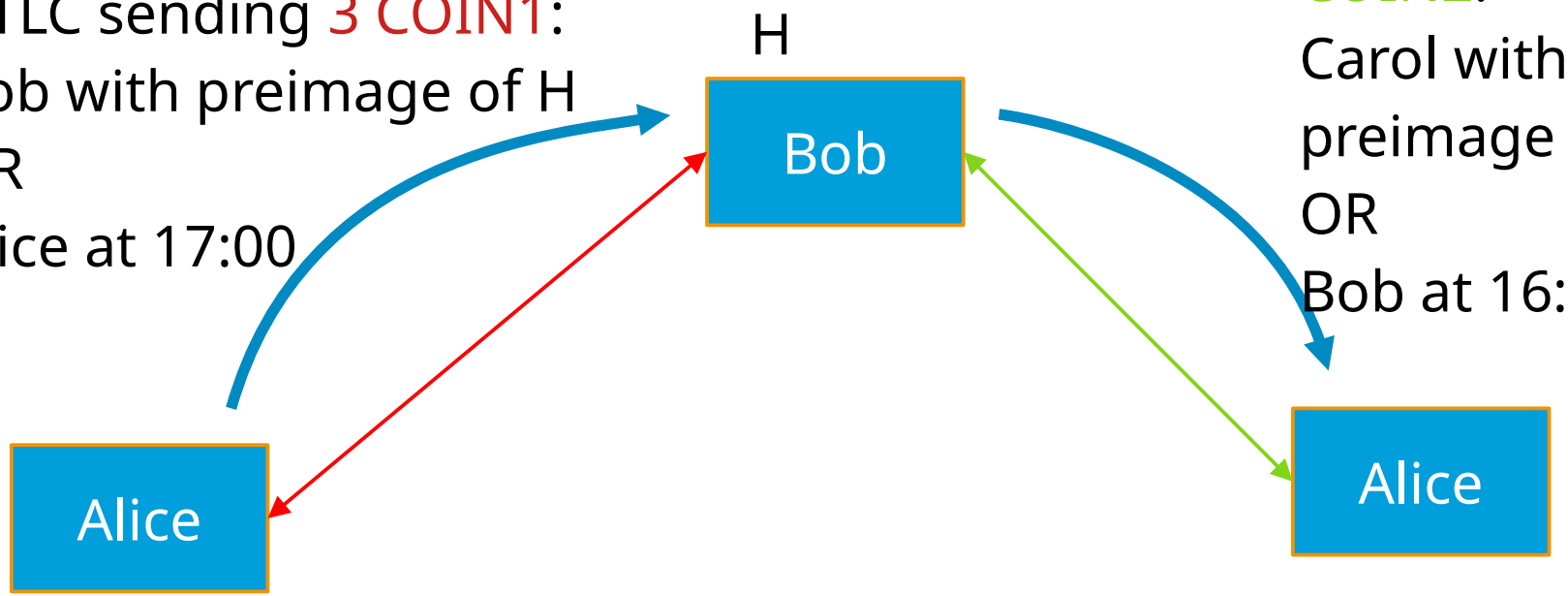


This works even if the two channels are on **different blockchains**!

Cross-Chain Swaps

HTLC sending 3 COIN1:
Bob with preimage of H
OR
Alice at 17:00

H



HTLC sending 2 COIN2:
Carol with preimage of H
OR
Bob at 16:00

H,R

H, R

Here, Alice is exchanging 3 COIN1 for 2 COIN2 with Bob