mas.s62
lecture 13
payment channels & the lightning network (pt 1)

2018-03-21

Tadge Dryja

schedule stuff

pset03 due tonight

next week: spring break (whoo)

today

payment channels

- unidirectional
- decreasing time
- lightning channels

why payment channels?

every tx going on blockchain doesn't scale. $O(n^2)$ (kind of)

First response of anyone, ever, about bitcoin:

history

Nov 2008

Satoshi: I've been working on a new electronic cash system that's fully peer-to-peer, with no trusted third party.

James A. Donald: 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.

1-way channel

initial idea: incremental payment channels

transactions have a "lock time" field

Transaction is only valid after the lock time (height) has passed

1-way channel

a "channel" is just a multisig output

2 of 2 signatures required

Alice funds to spend to Bob

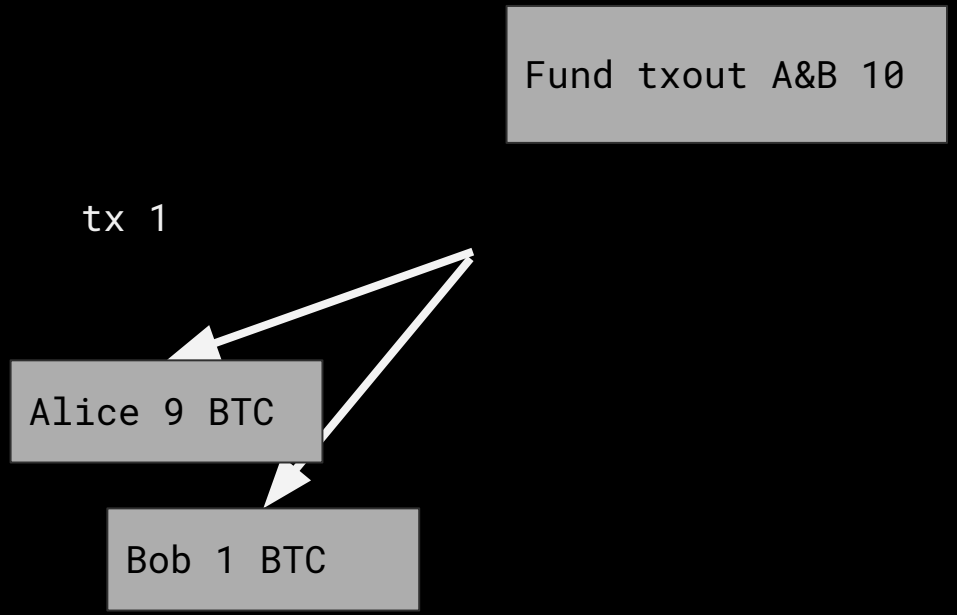
Fund Tx	
input	output
Alice's txid:index Alice's signature	Alice & Bob multisig 10 coins

1-way channel

a refund transaction is for Alice to get her money back. Lock time is set to 1 week in the future

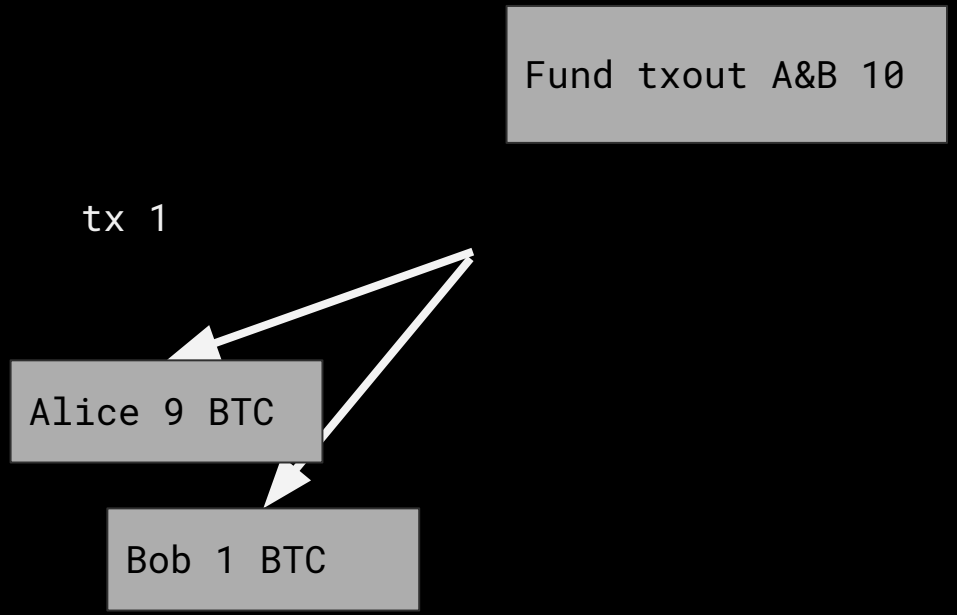
Refund Tx LOCKTIME: March 28	
input	output
fund txid Bob's signature (alice's)	Alice address 10 coins

1-way channel



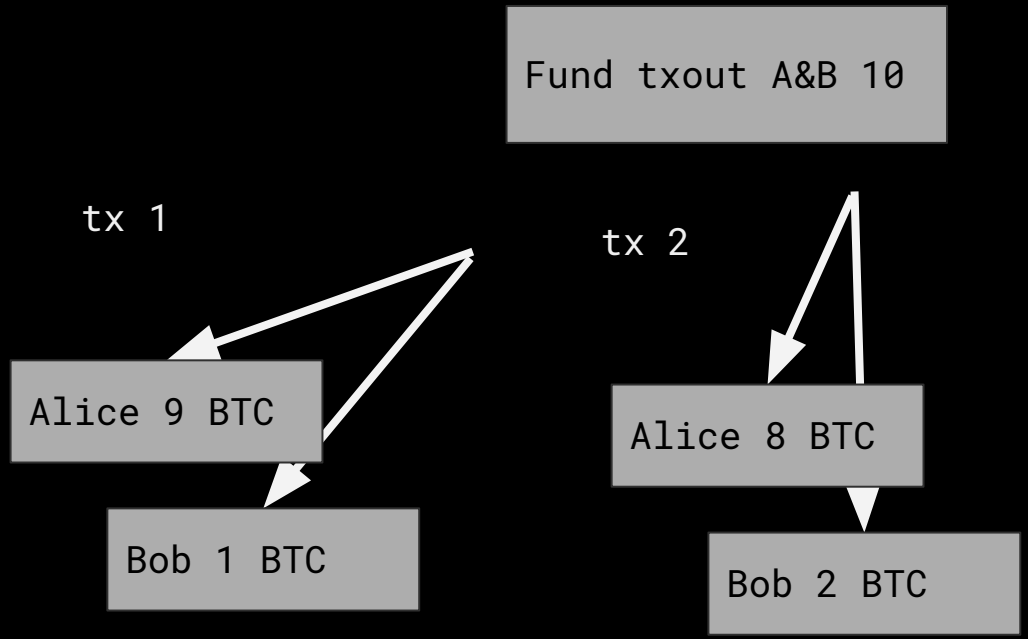
Alice signs a transaction spending the multisig output, sending 1 coin to Bob and 9 back to Alice. She sends the txc to Bob.

1-way channel



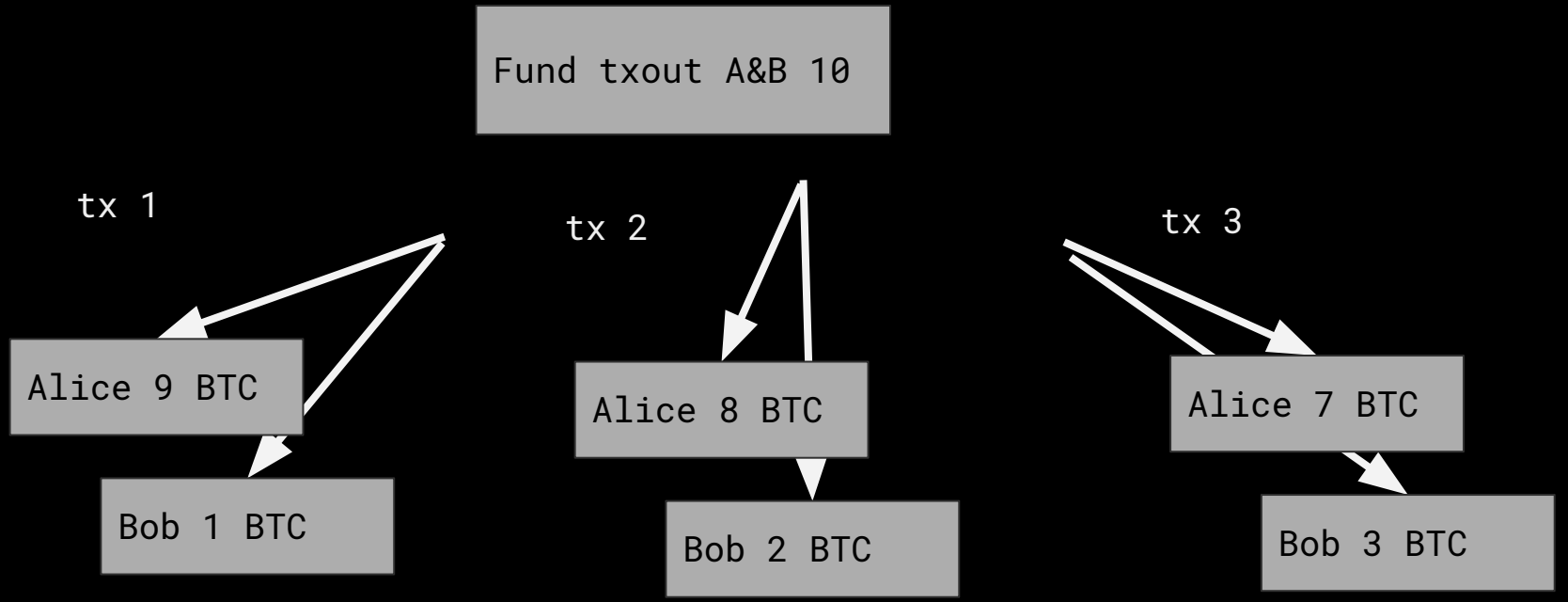
Bob DOESN'T sign his side and broadcast. Instead, he waits.

1-way channel



Alice sends a new transaction, spending the fund output, this time sending 2 coins to Bob. Again Bob waits.

1-way channel



Alice makes a new transaction, this time sending 3 coins to Bob.

1-way channel outcomes

Bob keeps getting half-signed txs with more money going to him

the old txs are useless; he can delete them

he must sign and broadcast one before next week!

1-way channel outcomes

useful, but limited

1 way: Bob can't pay Alice. Alice knows Bob retains the tx paying the most to himself

Time limit due to refund tx

Refund tx needs to be built before fund tx (malleability)

lightning channels

make a payment channel bidirectional, and indefinite duration

but how? refund tx? how to delete / revoke old txs?

timing opcodes

OP_CHECKSEQUENCEVERIFY

relative locktime opcode

require that the input have at least n confirmations to be able to spend

if not, tx fails

timing opcodes

OP_CHECKLOCKTIMEVERIFY

absolute locktime opcode

require that the transaction be confirmed in a block of at least height n

fail otherwise

revoke based on timing

keyA && keyB ||

keyC && 100 blocks

A and B together can spend any time

C can spend together, but must wait

A can grab the coins first!

revokable tx

Commit Tx (held by Alice)	
input	output
fund txid Bob's signature	Alice key & 100 blocks or AliceR & Bob key 2 coins
	Bob address 8 coins

revokable tx

Commit Tx (held by Bob)	
input	output
fund txid Alice's signature	Alice address 2 coins
	Bob key & 100 blocks or Alice & BobR key 8 coins

reveal to revoke

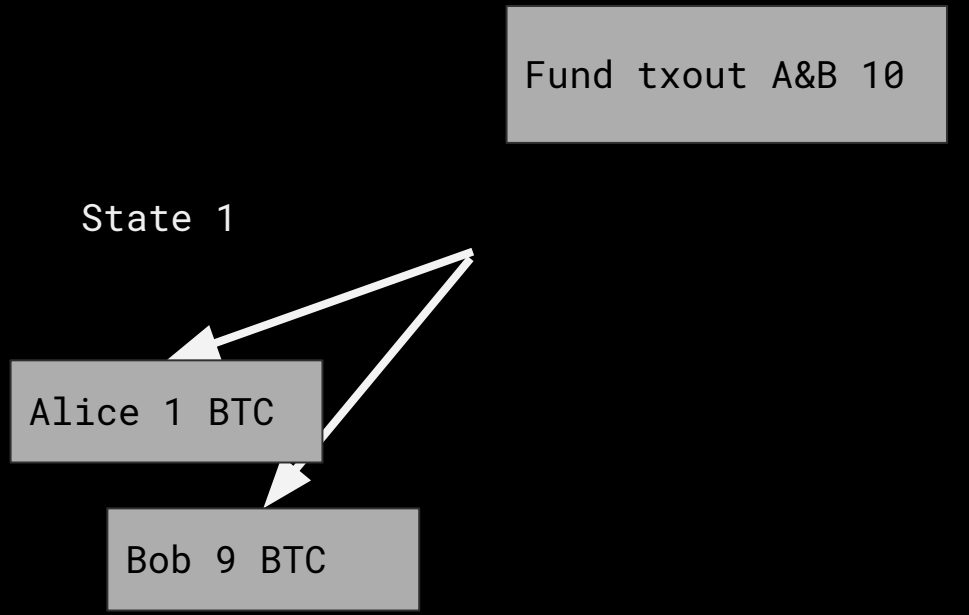
Either party broadcasts & has to wait

Alice gives Bob the AliceR privKey

Bob gives Alice the BobR privKey

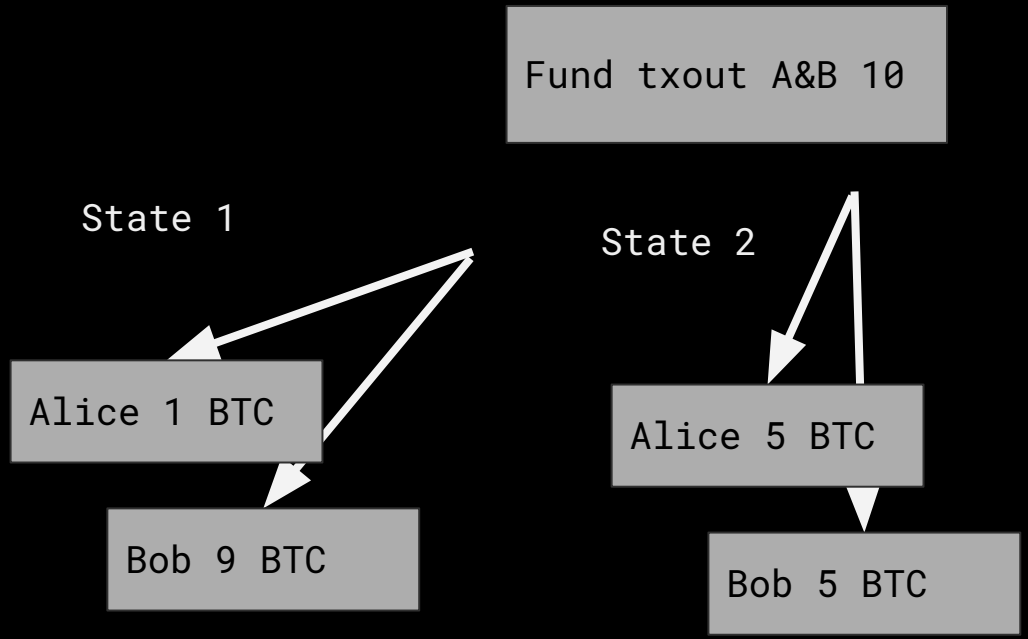
Now if they broadcast the counterparty can take all funds while they wait!

add and delete states



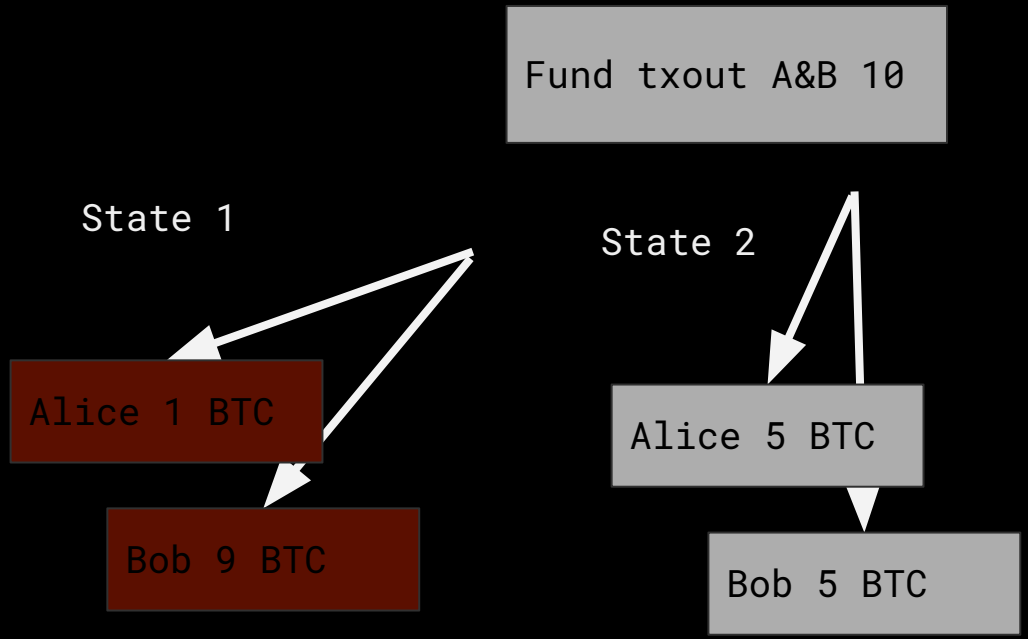
In Lightning, states are added sequentially, and validity is enforced by revealing private keys to previous states

add and delete states



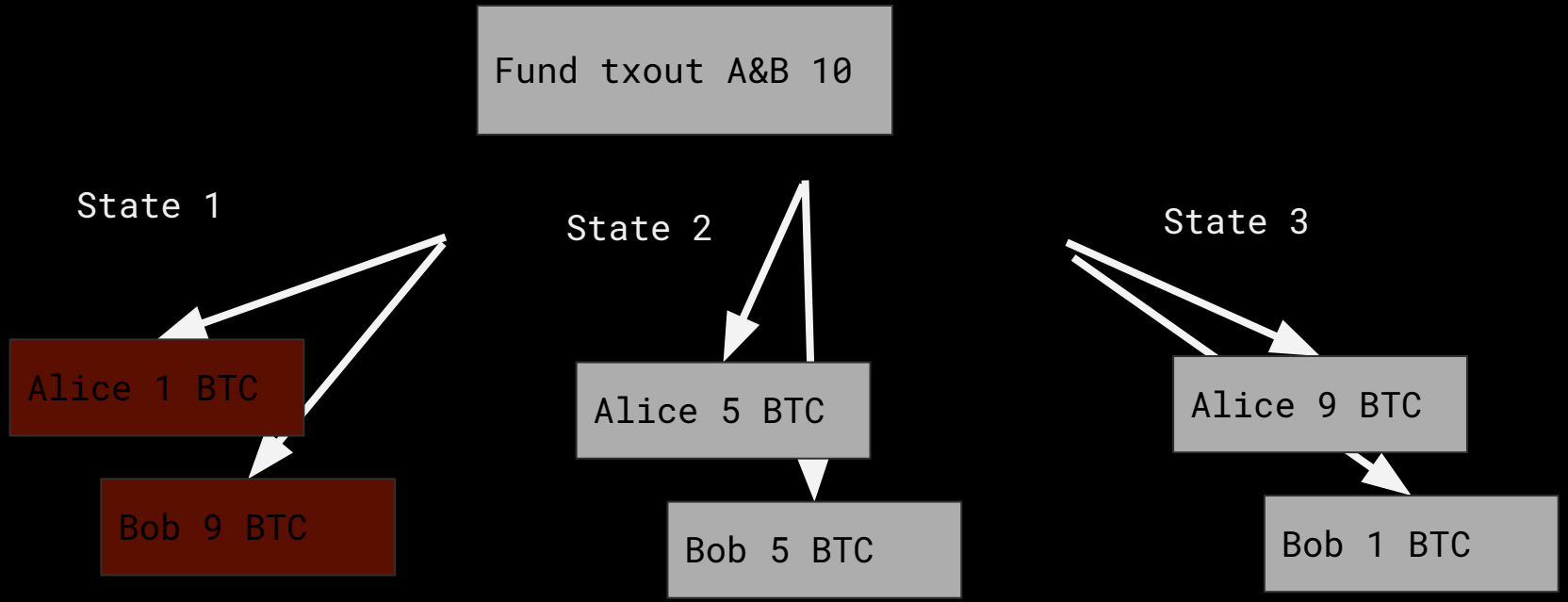
In Lightning, states are added sequentially, and validity is enforced by revealing private keys to previous states

add and delete states



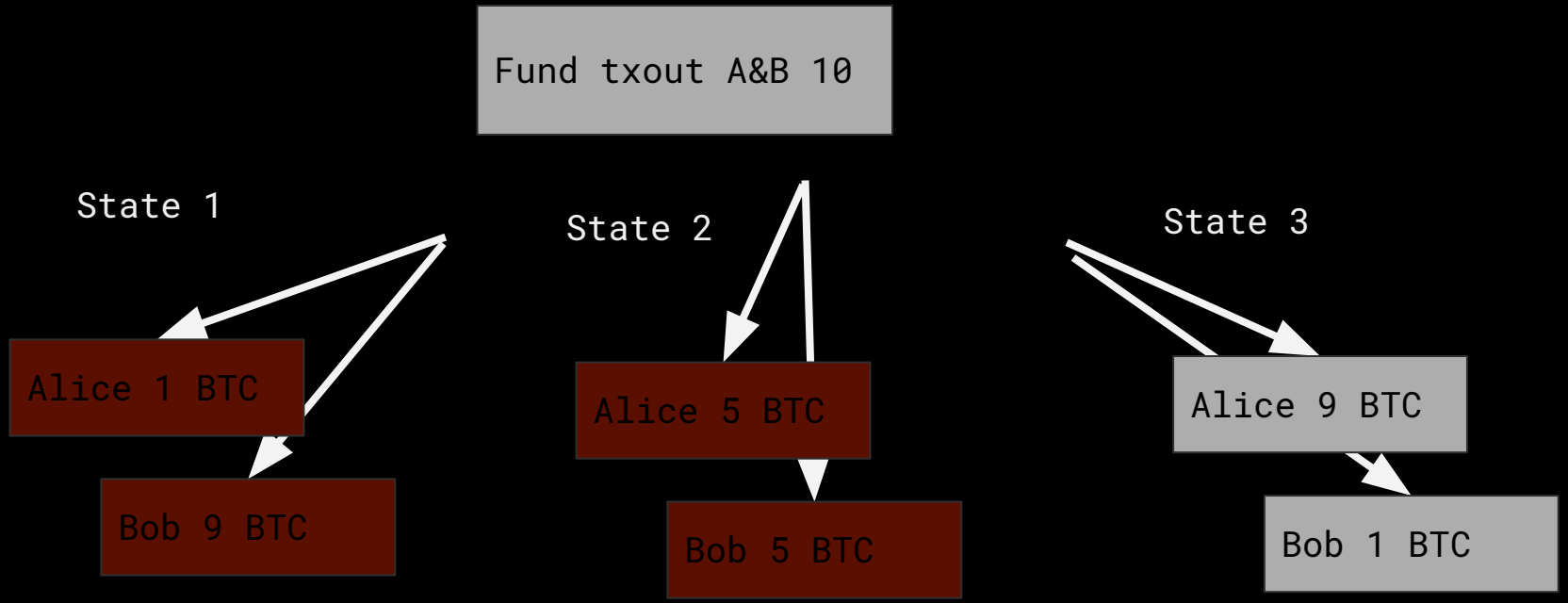
In Lightning, states are added sequentially, and validity is enforced by revealing private keys to previous states

add and delete states



In Lightning, states are added sequentially, and validity is enforced by revealing private keys to previous states

add and delete states



In Lightning, states are added sequentially, and validity is enforced by revealing private keys to previous states

2 party, indefinite

Still need to create channel to pay

1 tx to open, 1 tx to close channel

potentially 2 txs to close (rare)

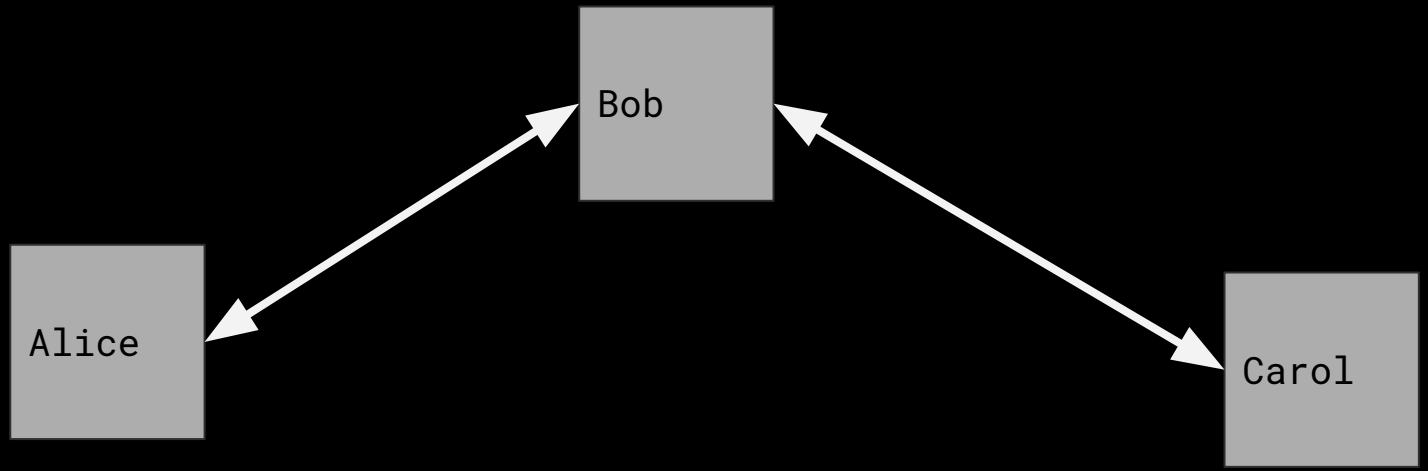
(broadcast commit tx, sweep)

multiple party channels

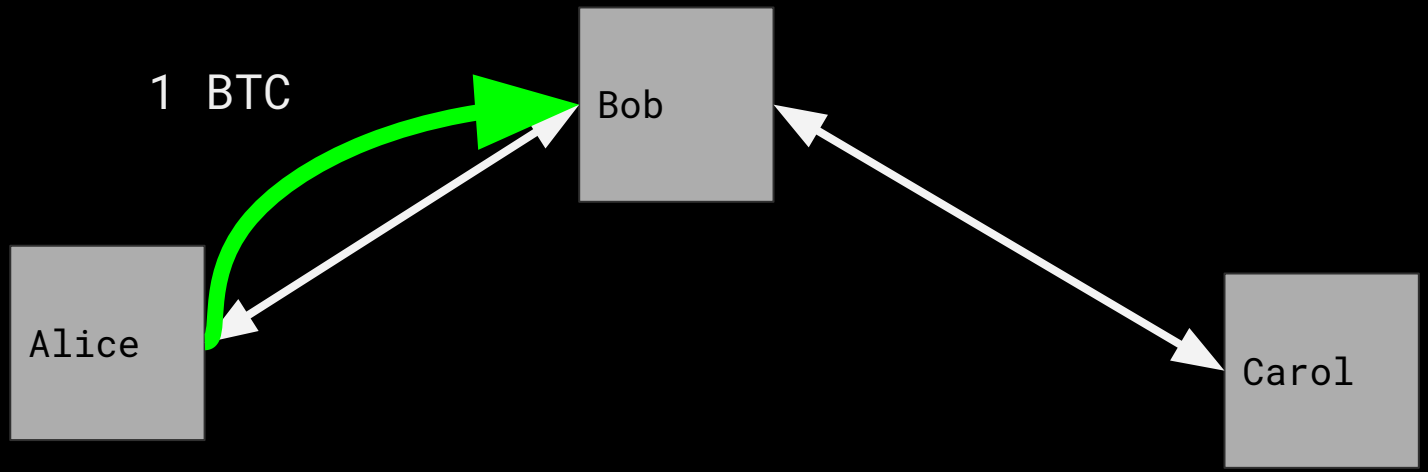
single channel with 3+ users gets really complicated

what about a forwarding network of point to point channels?

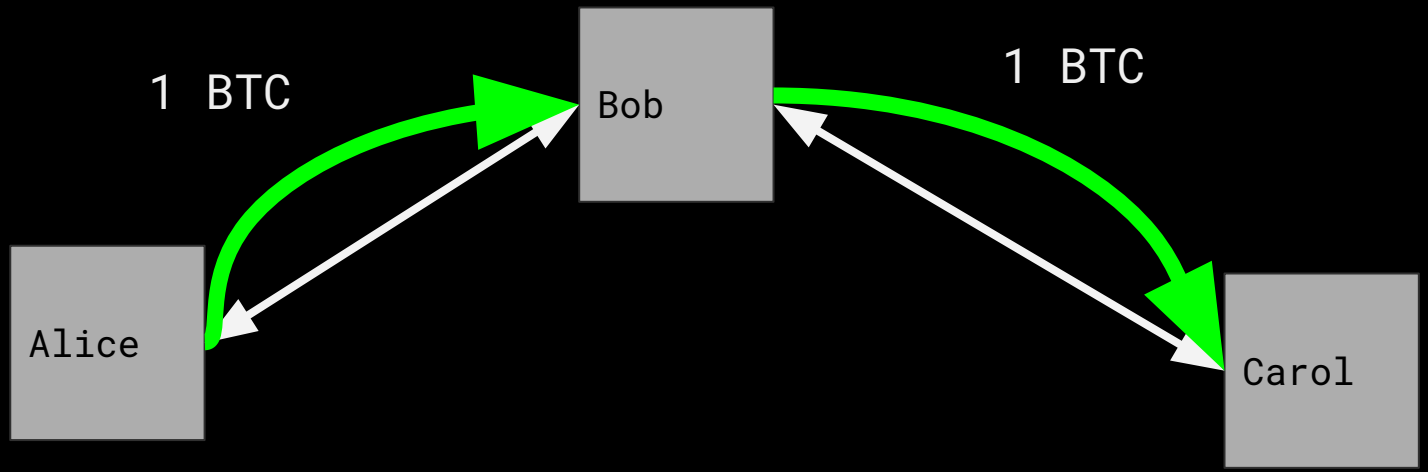
multiple party - optimistic



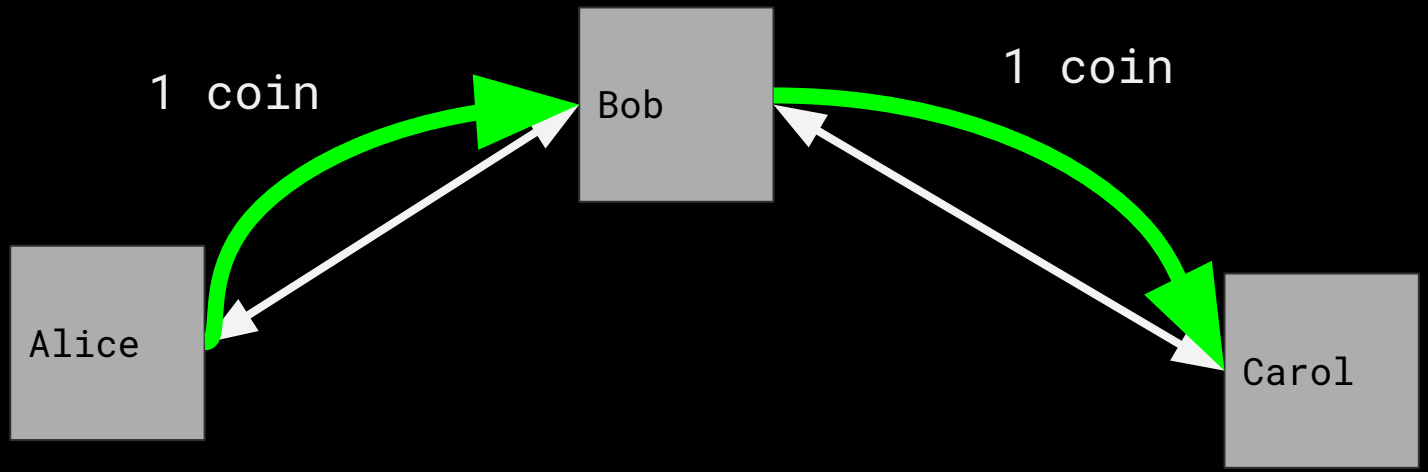
multiple party - optimistic



multiple party - optimistic

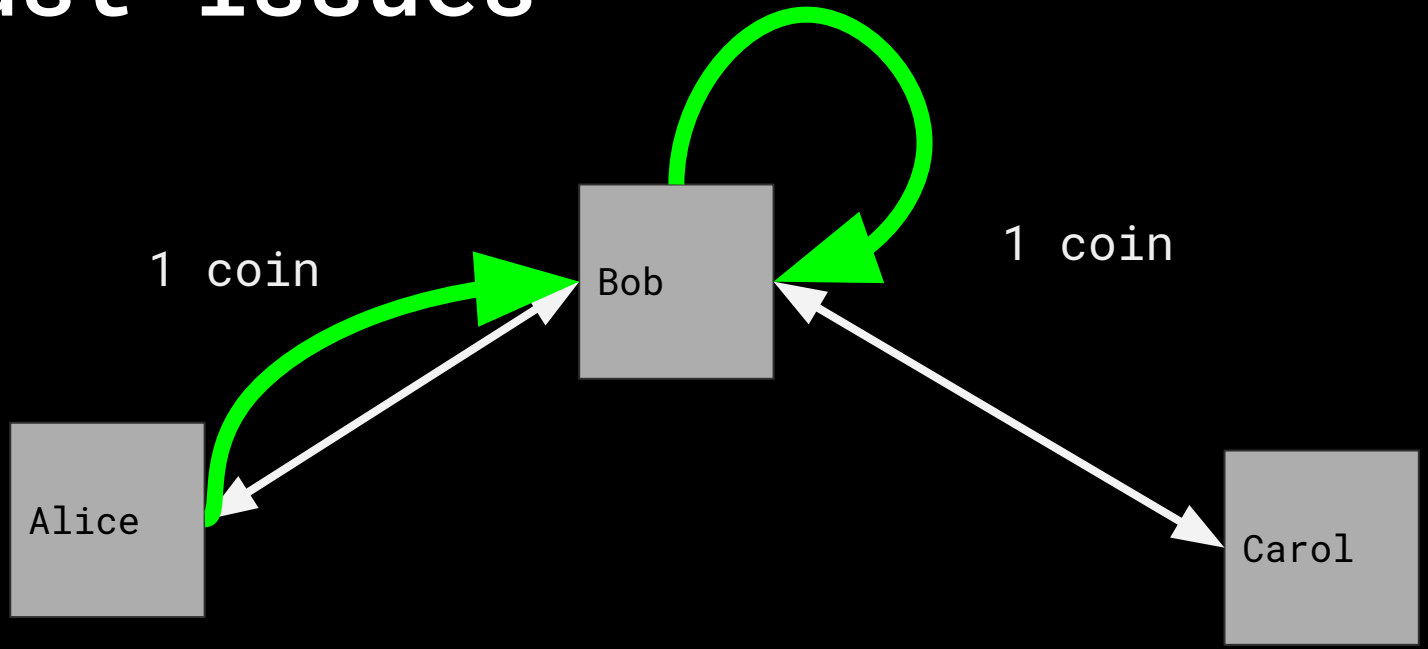


multiple party - optimistic



Alice pays Bob 1 coin, and Bob pays Carol 1 coin

trust issues



Bob keeps the money. Thanks Alice

Preimage determines who spends

New output script type: HTLC

Hash/Time Locked Contract

KeyA && preimageR ||

KeyB && OP_CLTV

Preimage determines who spends

New output script type: HTLC

Hash/Time Locked Contract

KeyA && preimageR ||

KeyB && OP_CLTV

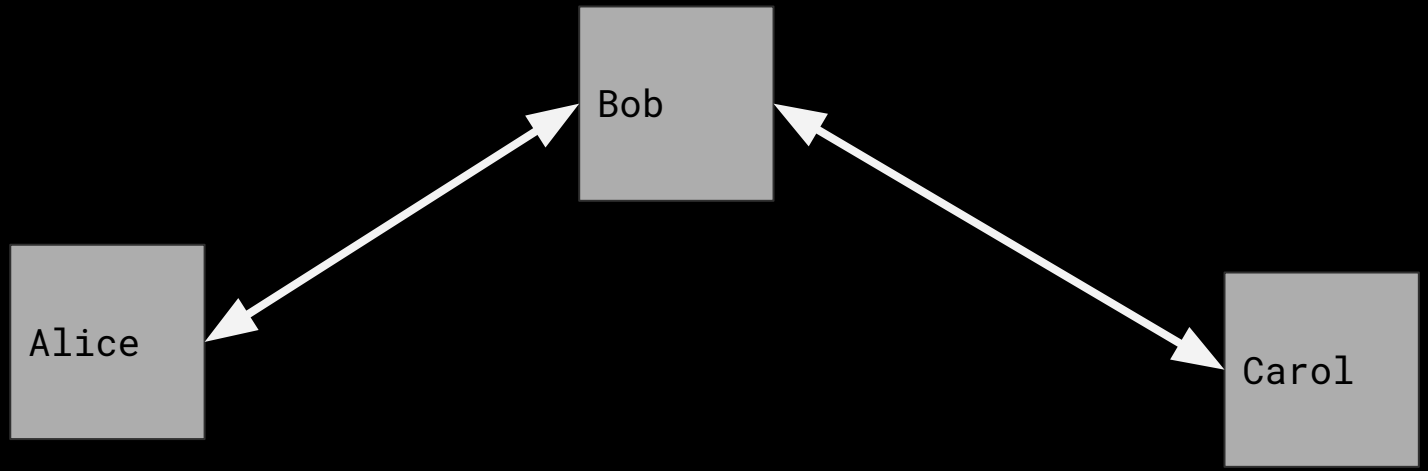
revokable tx

Commit Tx (held by Bob)	
input	output
fund txid Alice's signature	Alice address 2 coins
	Bob key & 100 blocks or Alice & BobR key 8 coins

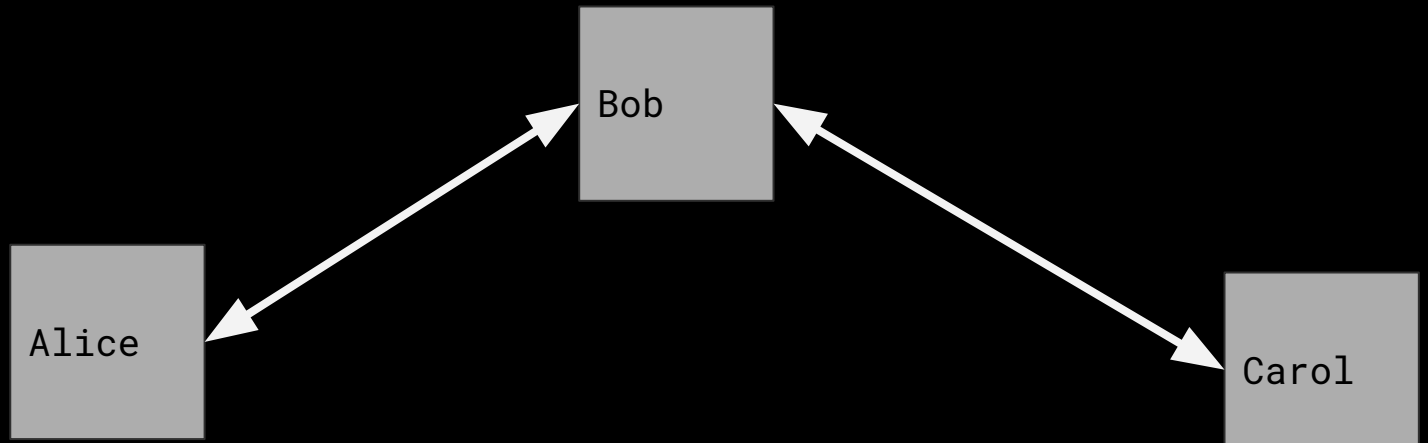
Preimage determines who spends

Commit Tx (held by Bob)	
input	output
fund txid Alice's signature	Alice address: 2 coins
	Bob key && 100 blocks || Alice && BobR key 7 coins
	HTLC Alice && R || Bob && height 500000 1 coin

multiple party - adversarial

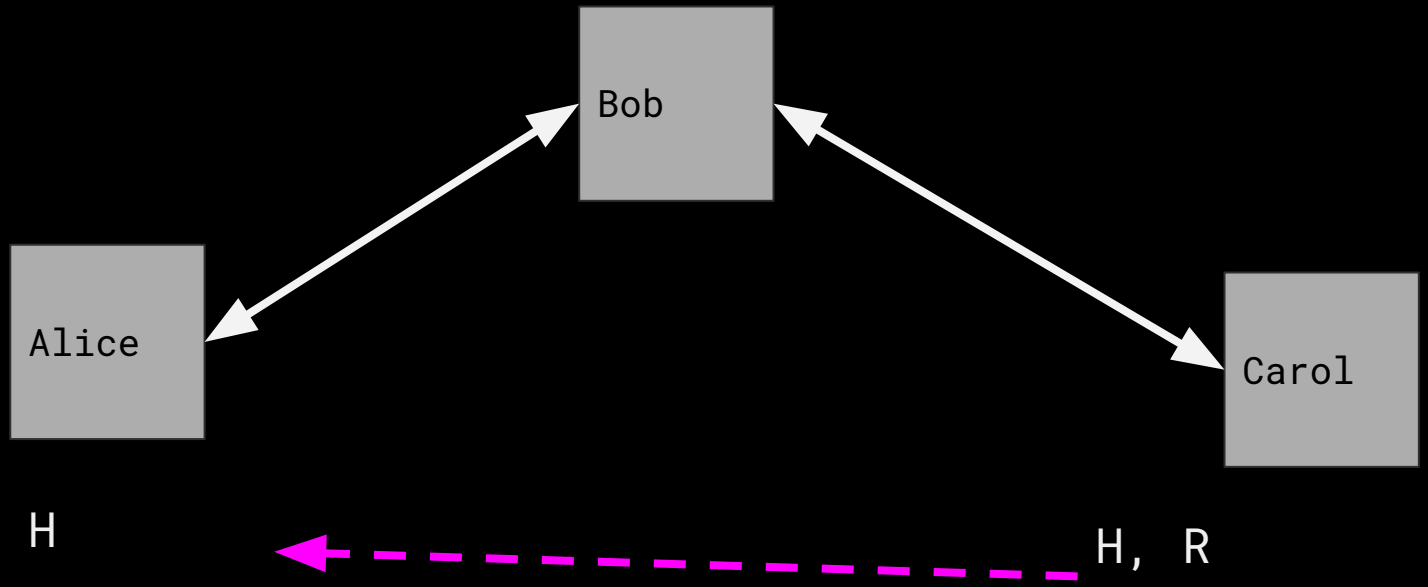


multiple party - adversarial



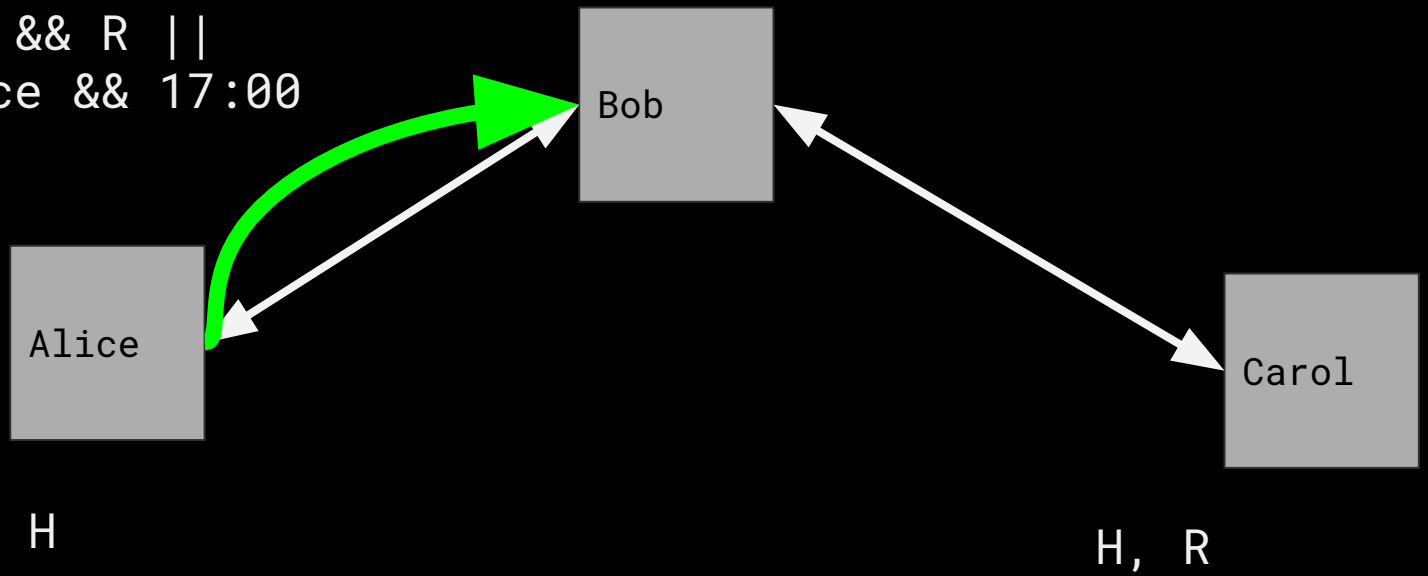
H = hash(R)

multiple party - adversarial

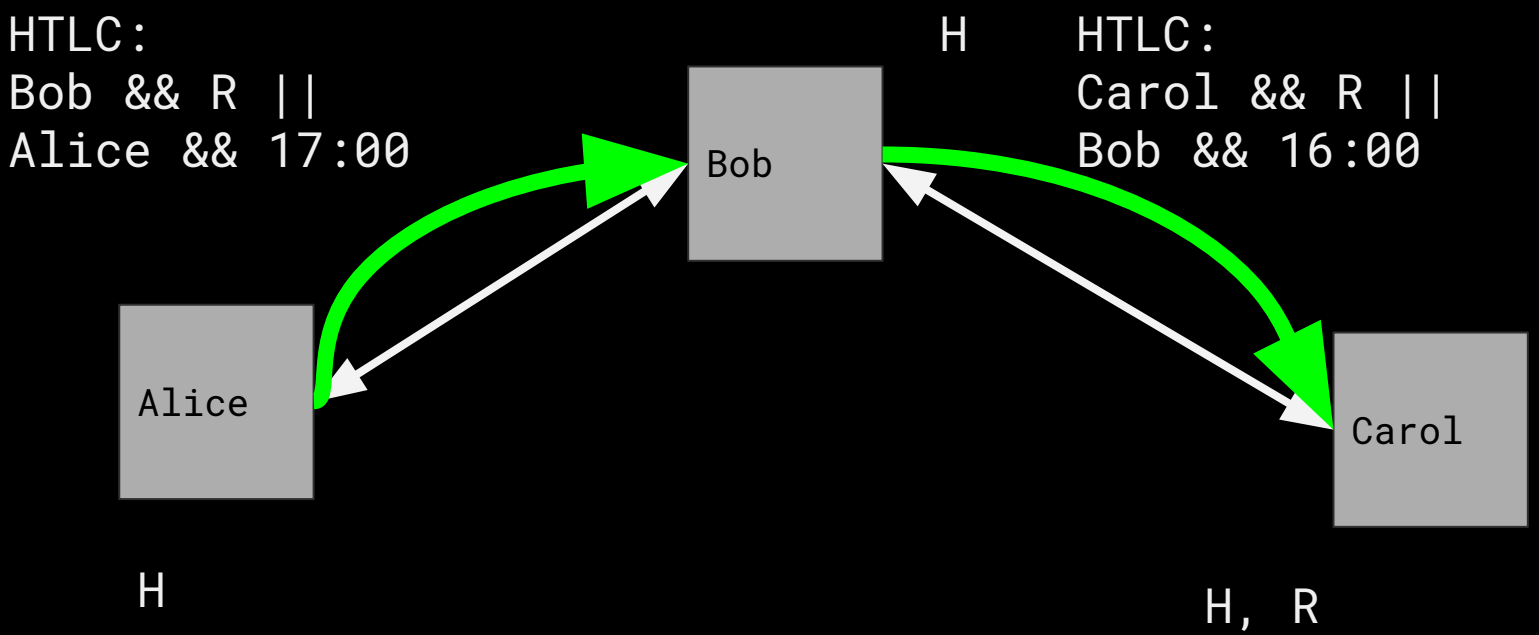


multiple party - adversarial

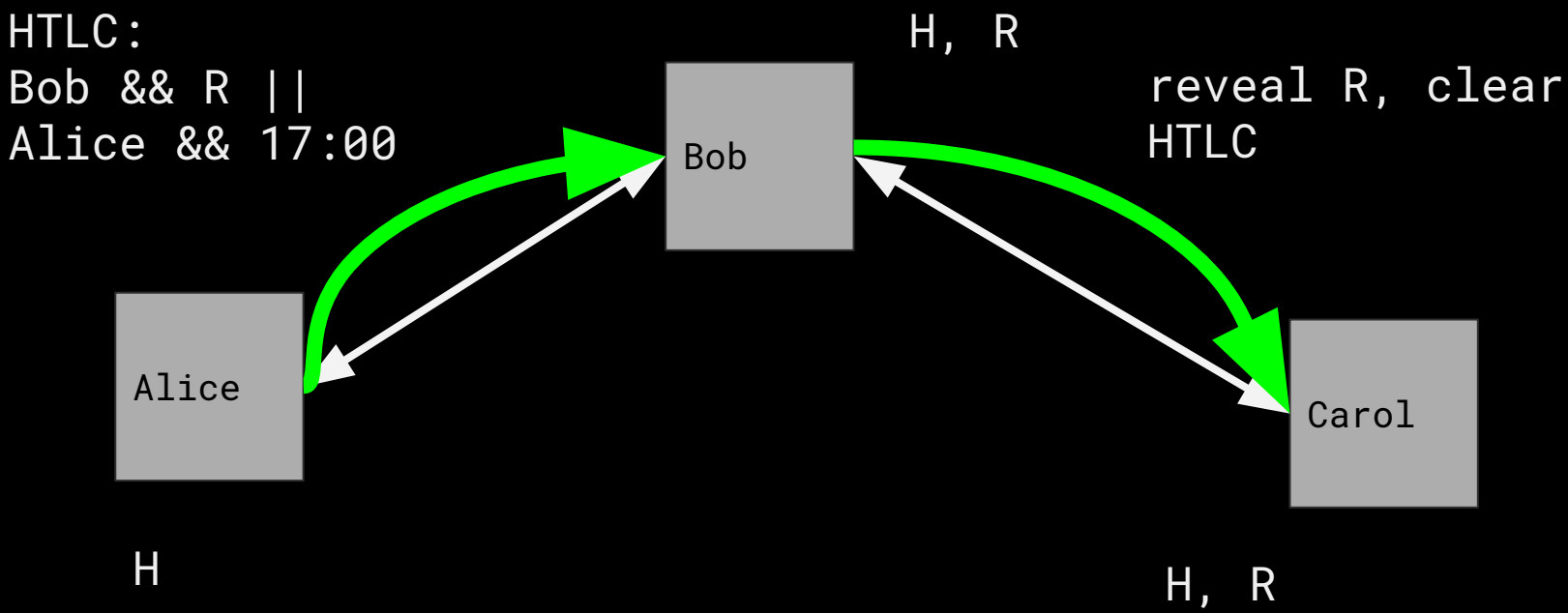
HTLC:
Bob && R ||
Alice && 17:00



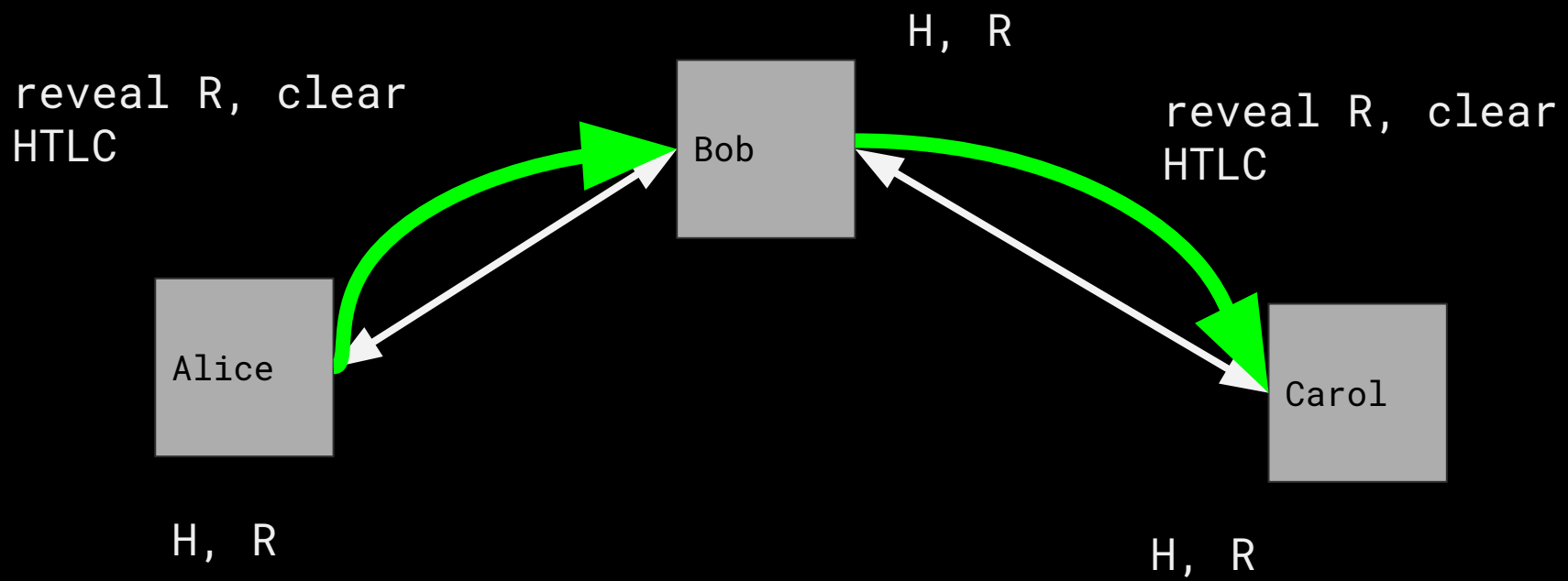
multiple party - adversarial



multiple party - adversarial



multiple party - adversarial



lightning network

lots of nodes with channels connecting, forming a graph

request payment routing via HTLC outputs

open few channels, able to pay many users on the network

lightning network

cross chain swaps

security: monitoring, outsourcing

stuck HTLCs, dust, fees

lots more you can do - will go into detail next time!