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

<table>
<thead>
<tr>
<th>Fund Tx</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>input</td>
<td>output</td>
</tr>
<tr>
<td>Alice's txid:index</td>
<td>Alice &amp; Bob multisig</td>
</tr>
<tr>
<td>Alice's signature</td>
<td>10 coins</td>
</tr>
</tbody>
</table>
1-way channel

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

<table>
<thead>
<tr>
<th>Refund Tx</th>
<th>LOCKTIME: March 28</th>
</tr>
</thead>
<tbody>
<tr>
<td>input</td>
<td>output</td>
</tr>
<tr>
<td>fund txid</td>
<td>Bob's signature (alice's)</td>
</tr>
<tr>
<td></td>
<td>Alice address</td>
</tr>
<tr>
<td></td>
<td>10 coins</td>
</tr>
</tbody>
</table>
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

Fund txout A&B 10

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

<table>
<thead>
<tr>
<th>Commit Tx (held by Alice)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>input</strong></td>
<td><strong>output</strong></td>
</tr>
<tr>
<td>fund txid</td>
<td>Alice key &amp; 100 blocks</td>
</tr>
<tr>
<td>Bob's signature</td>
<td>or AliceR &amp; Bob key</td>
</tr>
<tr>
<td></td>
<td>2 coins</td>
</tr>
<tr>
<td></td>
<td>Bob address</td>
</tr>
<tr>
<td></td>
<td>8 coins</td>
</tr>
</tbody>
</table>

18
### revokable tx

<table>
<thead>
<tr>
<th>Commit Tx (held by Bob)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>input</strong></td>
</tr>
<tr>
<td>fund txid</td>
</tr>
<tr>
<td>Alice's signature</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
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.

State 1
- Alice 1 BTC
- Bob 9 BTC

State 2
- Alice 5 BTC
- Bob 5 BTC

Fund txout A&B 10
add and delete states

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

1 BTC

Alice -> Bob

Bob -> Carol
multiple party - optimistic

Alice

Bob

1 BTC

Carol

1 BTC

2 BTC
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**

<table>
<thead>
<tr>
<th>Commit Tx (held by Bob)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>input</strong></td>
</tr>
<tr>
<td>fund txid</td>
</tr>
<tr>
<td>Alice's signature</td>
</tr>
<tr>
<td>Bob key &amp; 100 blocks</td>
</tr>
<tr>
<td>or Alice &amp; BobR key</td>
</tr>
</tbody>
</table>
Preimage determines who spends

<table>
<thead>
<tr>
<th>Commit Tx (held by Bob)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>input</strong></td>
<td><strong>output</strong></td>
</tr>
<tr>
<td>fund txid</td>
<td>Alice address: 2 coins</td>
</tr>
<tr>
<td>Alice's signature</td>
<td>Bob key &amp;&amp; 100 blocks</td>
</tr>
<tr>
<td></td>
<td>7 coins</td>
</tr>
<tr>
<td></td>
<td>HTLC Alice &amp;&amp; R</td>
</tr>
<tr>
<td></td>
<td>1 coin</td>
</tr>
</tbody>
</table>
multiple party - adversarial
multiple party - adversarial

Alice

Bob

Carol

\[ H = \text{hash}(R) \]
multiple party - adversarial
multiple party - adversarial

HTLC:
Bob \&\& R \|\
Alice \&\& 17:00
multiple party - adversarial

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

HTLC:
Carol && R ||
Bob && 16:00

H

H

H, R

Alice

Bob

Carol
multiple party - adversarial

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

reveal R, clear HTLC
multiple party - adversarial

reveal R, clear HTLC

Alice

Bob

H, R

reveal R, clear HTLC

Carol

H, R
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!
MAS.S62 Cryptocurrency Engineering and Design
Spring 2018

For information about citing these materials or our Terms of Use, visit: https://ocw.mit.edu/terms.