Statechains: Off-chain Transfer of UTXO Ownership

Ruben Somsen
Email: rsomsen@gmail.com
Twitter: @SomsenRuben
What Statechains Achieve
What Statechains Achieve

- L2 scaling by avoiding on-chain transactions
What Statechains Achieve

- L2 scaling by avoiding on-chain transactions
- Advantage over Lightning: unrestricted coin movement (has synergy)
What Statechains Achieve

- L2 scaling by avoiding on-chain transactions
- Advantage over Lightning:
  unrestricted coin movement (has synergy)
- Advantage over Federated Sidechains:
  federation doesn’t have full control
What Statechains Achieve

- L2 scaling by avoiding on-chain transactions
- Advantage over Lightning:
  unrestricted coin movement (has synergy)
- Advantage over Federated Sidechains:
  federation doesn’t have full control
- Unique limitation: can only move full UTXO amount
What Statechains Build On

- Schnorr signatures
- Adaptor signatures
- Eltoo
- Graftroot
What Statechains Build On

- Schnorr signatures
- Adaptor signatures
- Eltoo
- Graftroot

Works on any cryptocurrency that supports the above
High-level Overview

1 BTC

AB

AC

AD
High-level Overview

- Change UTXO ownership off-chain
High-level Overview

- Change UTXO ownership off-chain
- Guaranteed on-chain redemption (D)
High-level Overview

- Change UTXO ownership off-chain
- Guaranteed on-chain redemption (D)
- Facilitated by “statechain entity” (A)
High-level Overview

- Change UTXO ownership off-chain
- Guaranteed on-chain redemption (D)
- Facilitated by “statechain entity” (A)
- A can collude with prior owners (B, C)
High-level Overview

- Change UTXO ownership off-chain
- Guaranteed on-chain redemption (D)
- Facilitated by “statechain entity” (A)
- A can collude with prior owners (B, C)
- Collusion/cheating always provable
Bob locks up 1 BTC with Alice...
Bob locks up 1 BTC with Alice...

...but who owns it?
Bob owns it

on-chain

B
AB

off-chain

AB
B
Bob owns it

eltoo: AB or B in 5 days

on-chain

off-chain
Can Bob transfer this off-chain to Carol?
Can Bob transfer this off-chain to Carol?

Sort of, but Carol has no control over it.
So Bob uses a transitory key X instead
So Bob uses a transitory key X instead and passes the key on to Carol.
The role of the statechain entity (A)
The role of the statechain entity (A)

- Promises to *only* cooperate with the last owner
The role of the statechain entity (A)

- Promises to *only* cooperate with the last owner
- Can be a federation

Schnorr:
A1 + A2 + A3 = A
The role of the statechain entity (A)

- Promises to *only* cooperate with the last owner
- Can be a federation
- Updates the statechain:
  - listing all UTXOs it controls (no duplicates)
  - every transfer has a signature (e.g. B to C)

Schnorr: $A_1 + A_2 + A_3 = A$
Bitcoin

1 BTC

B  AX
Bitcoin

1 BTC

B

AX

AX

B
Bitcoin

1 BTC

B  AX

↓

AX  B

Statechain

1 BTC

X
Bitcoin

1 BTC

\[B\] \[AX\]

\[\downarrow\]

\[AX\] \[B\]

Statechain

1 BTC

\[X\]

\[\downarrow\]

\[B\]
Bitcoin

1 BTC

B
AX

Statechain

1 BTC

X

B

C

eltoo
Problem: who goes first?
Statechain entity A goes first...?
User B goes first...?
Solution: Adaptor Signatures

1 BTC

B       AX

AX       B

1 BTC

X

B
Everyone shares an incomplete signature

1 BTC
B
AX

AX
B

1 BTC
X

B

AX
C

C
Completing the statechain signature...
...automatically completes the bitcoin signature
Security Model

Bitcoin: A + X
Statechain: B / C / D / E (knows X)
Owner: Prev. Last
Security Model

- Moving the coins always requires the permission of:
  a. Statechain entity A (typically a federation)
  b. a transitory key holder (who held the UTXO)

<table>
<thead>
<tr>
<th>Bitcoin:</th>
<th>A + X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statechain:</td>
<td>B/G/D/E (knows X)</td>
</tr>
<tr>
<td>Owner:</td>
<td>Prev. Last</td>
</tr>
</tbody>
</table>
- Moving the coins always requires the permission of:
  a. Statechain entity A (typically a federation)
  b. a transitory key holder (who held the UTXO)
- Entity must cooperate with LAST transitory key holder
Security Model

- Moving the coins always requires the permission of:
  a. Statechain entity A (typically a federation)
  b. a transitory key holder (who held the UTXO)
- Entity must cooperate with \textbf{LAST} transitory key holder
- Failure to do so will produce evidence of fraud

\textbf{Bitcoin:} \quad A + X \\
\textbf{Statechain:} \quad B / C / D / E \ (knows \ X) \\
\textbf{Owner:} \quad \textit{Prev.} \quad \textit{Last}
Worst case scenario
Worst case scenario

- Entity obtains a bunch of transitory keys (X, Y, Z...)
Worst case scenario

- Entity obtains a bunch of transitory keys (X, Y, Z…)
- Proceeds to (provably) steal the coins
Worst case scenario

- Entity obtains a bunch of transitory keys (X, Y, Z...)
- Proceeds to (provably) steal the coins
- Uncompromised transitory keys withdraw on-chain
Worst case scenario

- Entity obtains a bunch of transitory keys (X, Y, Z...)
- Proceeds to (provably) steal the coins
- Uncompromised transitory keys withdraw on-chain

Harmless without transitory keys (weak assumption):
- The statechain entity gets hacked
Worst case scenario

- Entity obtains a bunch of transitory keys (X, Y, Z...)
- Proceeds to (provably) steal the coins
- Uncompromised transitory keys withdraw on-chain

Harmless without transitory keys (weak assumption):
- The statechain entity gets hacked
- Court order to freeze/confiscate coins
Swapping to smaller amounts

1 BTC  1 BTC  2 BTC
X      Y      Z
B      B      C
Swapping to smaller amounts

1 BTC 1 BTC 2 BTC

X Y Z

B B C

C C B
Possible with other coins

1 BTC  1 BTC  200 LTC

X  Y  Z

B  B  C

C  C  B
Microtransactions

“Anything smaller than an economically viable UTXO”
Microtransactions

“Anything smaller than an economically viable UTXO”

- Required if Statechain entity wants to charge fees
Microtransactions

“Anything smaller than an economically viable UTXO”

- Required if Statechain entity wants to charge fees
- Needed when swapping between multiple currencies
Microtransactions

“Anything smaller than an economically viable UTXO”

- Required if Statechain entity wants to charge fees
- Needed when swapping between multiple currencies
- Ideally solved without trusting the statechain entity
  (important legal reason: no custody over ANY coins)
Lightning Channel Creation

1 BTC

[Diagram showing the creation of a Lightning Channel with nodes A and X and 1 BTC on each side.]
Lightning Channel Creation

1 BTC

B  AX

AX  B

AX  BC

B

BC
Lightning Channel Creation

1 BTC

B   AX

AX   B

AX   BC

BC   B   C

0.9 BTC

0.1 BTC

1 BTC

X

B

BC
Lightning on Statechains
Lightning on Statechains

- Channel updated together with multi atomic swap
Lightning on Statechains

- Channel updated together with multi atomic swap
- Small channels: up to the amount of the smallest UTXO
Lightning on Statechains

- Channel updated together with multi atomic swap
- Small channels: up to the amount of the smallest UTXO
- Uncooperative close similar to regular Eltoo
Lightning on Statechains

- Channel updated together with multi atomic swap
- Small channels: up to the amount of the smallest UTXO
- Uncooperative close similar to regular Eltoo
- Close/reopen channel low-friction: it’s all off-chain! (e.g. adding/removing bitcoins)
Potential Use Cases
Potential Use Cases

- Off-chain value transfer
Potential Use Cases

- Off-chain value transfer
- Platform for Lightning channels
Potential Use Cases

- Off-chain value transfer
- Platform for Lightning channels
- Betting channels (multisig, Discreet Log Contracts)
Potential Use Cases

- Off-chain value transfer
- Platform for Lightning channels
- Betting channels (multisig, Discreet Log Contracts)
- Fork-agnostic ETF (UTXOs don’t move)
Further topics
Further topics

- Non-interactive version
Further topics

- Non-interactive version
- Use HSM to transfer transitory key (attestation)
Further topics

- Non-interactive version
- Use HSM to transfer transitory key (attestation)
- Graftroot withdrawal (allows redeeming forks)
Further topics

- Non-interactive version
- Use HSM to transfer transitory key (attestation)
- Graftroot withdrawal (allows redeeming forks)
- Succinctly store and relay statechain (per UTXO)
Further topics

- Non-interactive version
- Use HSM to transfer transitory key (attestation)
- Graftroot withdrawal (allows redeeming forks)
- Succinctly store and relay statechain (per UTXO)
- Variant using blind signatures:
  
  Entity unaware which UTXOs it holds (unblind p2p)
Thank You

Paper: goo.gl/RWQ4ue

Email: rsomsen@gmail.com

Twitter: @SomsenRuben
Thank You

Paper: goo.gl/RWQ4ue

Email: rsomsen@gmail.com

Twitter: @SomsenRuben
Bitcoin

on-chain state

prior state

AX or B*

*timelock

new state
Bitcoin

on-chain state

prior state

AX

AX or B*

*timelock

new state

AX

AX or C*

Statechain

“AX”

B

C

last owner
AX or B

on-chain state

AX or B*

prior state

*timelock

AX or BC*

new state

AX or BC

Lightning

BC

0.9 BTC

B

0.1 BTC