Payment Channels
Designing Secure Watchtowers
Why be a Watchtower?

Assuming rational parties and watchtowers…

- Will a party commit fraud?  
  - No

- Will a watchtower get paid?  
  - No

- Will a party commit fraud?  
  - Yes

- Will a watchtower get paid?  
  - Yes

- Will a party commit fraud?  
  - No

- Will a party commit fraud?  
  - No
Why be a Watchtower?

<table>
<thead>
<tr>
<th>Watchtowers →</th>
<th>Active</th>
<th>Inactive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parties ↓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fraud</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No Fraud</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Why be a Watchtower?

<table>
<thead>
<tr>
<th>Watchtowers →</th>
<th>Active</th>
<th>Inactive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parties ↓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fraud</td>
<td>![Diagram]</td>
<td>![Diagram]</td>
</tr>
<tr>
<td>No Fraud</td>
<td>![Diagram]</td>
<td>![Diagram]</td>
</tr>
</tbody>
</table>
Why be an active Watchtower?

Collateral
Lightning Channels

\[ \sigma^A \land \Delta t \lor \sigma^A \lor \sigma^{AB} \]

Funding
On-chain

Commitment
(1)
Published by A

Commitment
(i)
Published by A

Commitment
(i+1)
Published by A

Revocation
Published by B, W

\[ \#\sigma^A \]

\[ \#\sigma^B \]

\[ \sigma^A \]

\[ \sigma^B \]

\[ a \]

\[ b \]
Cerberus Channels

- **Funding**
  - On-chain
  - Input: \#σ_A, a, \#σ_B, b
  - Output: σ_{AB}, a+b

- **Commitment (1)**
  - Published by A
  - Input: (σ_A \wedge Δt) ∨ σ_{AW}
  - Output: a, σ_{BW}, b

- **Commitment (i)**
  - Published by A
  - Input: (σ_A \wedge Δt) ∨ σ_{AW}
  - Output: a_i, σ_{BW}, b_i

- **Revocation**
  - Published by B, W
  - Input: σ_{AW}
  - Output: σ_B, a_i+b_i

- **Commitment (i+1)**
  - Published by A
  - Input: (σ_A \wedge Δt) ∨ σ_{AW}
  - Output: a_{i+1}, σ_{BW}, b_{i+1}

- **Penalty 1**
  - Published by B
  - Input: σ_{BW}, c+b_i
  - Output: σ_B

- **Collateral**
  - On-chain
  - Input: \#σ_W, c
  - Output: σ_{BW}, c

- **Reclaim**
  - Published by W
  - Input: σ_{BW}
  - Output: c
Cerberus Channels
Fundamentals of Channels
Fundamentals of Channels
Attacks

- Eclipse
- Censor
- Congestion

Funding
Commitment
Dispute period
Time = CryptoMoney!
Time = CryptoMoney!
Be proactive, not reactive

I believe in a better way.
Be proactive, not reactive

Funding

Close

Signatures of Alice & Bob

OR

Signatures of ⅔ WT & (Alice or Bob)
Challenges

1) Consensus is costly
2) Privacy is important
3) Incentives are critical
O(n) communication complexity for state updates

Verification of consensus between Alice & Bob

No liveness guarantees, if Alice & Bob both misbehave

Consensus needed only for closing, if there is a dispute
Encrypted State

Privacy preserving

Alice/Bob cannot publish a previous transaction
Brick Architecture

(3) Execute

(1) Update

(2) Consistent Broadcast

(3) Execute

(2) Consistent Broadcast
Incentives

- Unilateral channel for fees:
  Repeated game lifts fair exchange impossibility

- Fees for closing the channels:
  Only payable in dispute → Incentive to agree

- Collateral for anti-bribing:
  Reduction to fair-exchange
  WT Committee size ↑ → per WT collateral ↓
Brick Advantages

- Asynchronous channels
- Security even under L1 failure
- Privacy
- Incentive-compatible
- Embarrassingly parallel
- Linear communication

[Avarikioti et al. Brick: Asynchronous State Channels.]