Brick
Asynchronous Payment Channels

Zeta Avarikioti
Eleftherios Kokoris-Kogias & Roger Wattenhofer

Fundamentals of Channels
Fundamentals of Channels
Fundamentals of Channels

Funding transaction
Fundamentals of Channels

Commitment transaction

5  4
Fundamentals of Channels

Commitment transaction

5  4

2  7

8  1
Funding

Commitment

Dispute period

Fundamentals of Channels
Watchtowers

Funding

Commitment

Dispute period
Attack the Liveness of the Blockchain

Funding
Commitment
Dispute period
Censorship Congestion
Time = CryptoMoney!
Time = CryptoMoney!

Asynchronous channels?
Be proactive, not reactive

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

Signatures of Alice & Bob
OR
Signatures of WT & (Alice or Bob)
Watchtower Committee

Committee
n = 3f+1
f Byzantine

Signatures of Alice & Bob
OR
Signatures of 2f+1 WTs & (Alice or Bob)
Challenges

1) Consensus is costly
2) Privacy is important
3) Incentives are critical
Consistent Broadcast

- $O(n)$ communication complexity for state updates
- Verification of consensus between Alice & Bob
- No guarantees, if Alice & Bob both misbehave
Encrypted State

- Privacy preserving
- Alice/Bob cannot publish a previous transaction
Brick Architecture

(3) Execute

H(     )
(1) Update

H(     )
(2) Consistent Broadcast

H(     )
(2) Consistent Broadcast

(3) Execute

Close: max state of 2f+1 submitted states.
Safety
A channel will only close in the freshest committed state

Brick Security Analysis

f slow honest WTs

2f+1 WT closing state (previous committed state)

2f+1 WT freshest committed state
Brick Security Analysis

Liveness
Any valid operation (close, update) will eventually be committed

Not committed = Invalid operation (failed verification)
Challenges

1) Consensus is costly
2) Privacy is important
3) Incentives are critical
Why be a Watchtower?
Repeated game lifts the fair-exchange impossibility
Per-update fees

Watchtower paid while channel is alive!
Incentives to close?
Why assist to close honestly?

Collateral
Why assist to close honestly?

Collateral

Asynchronous channels?
Collateral

Fraud proofs
  two signed conflicting states

Party claims the collateral
Collateral

Fraud proofs
two signed conflicting states

Party claims the collateral

channel value \( v \)

claimed collateral \( v/f \cdot (f+1) \)
Collateral

Where do we close?
when >f fraud proofs are submitted

all channel value→ counterparty
Collateral

Where do we close?
when \( \leq f \) fraud proofs are submitted

run close again without the malicious \( \rightarrow \) max state of 2f+1
Collateral

Profit =
channel balance $(c)$ + fraud proofs $(v/f)$ - bribes $(v/f + \varepsilon)$

1. **0 FPs**: profit $= c \leq v$

2. **$> f$ FPs**: profit $\leq v + y^*\frac{v}{f} - y^*(\frac{v}{f} - \varepsilon) = v - \varepsilon$

3. **$f$ FPs and “correct” close**: profit $= c + v$

4. **$f$ FPs and “incorrect” close**: profit $= v - \frac{v}{f} - \varepsilon$

$v = \text{channel value}$
$f = \text{Byzantine watchtowers}$
y = bribed watchtowers
WTs collude → Hostage situations

Why assist to close?

Closing fees
prisoner’s dilemma
Why request close?

Parties collude $\rightarrow$ Hostage situations

Committee size > 7
richest party loses more
Committee size

The more (WTs) the merrier!

↑ robustness
↓ collateral per WT
≈ cost for parties
Brick Advantages

- Privacy
- Incentive-compatible
- Good performance
- Asynchronous
  - censorship
  - congestion
  - liveness attacks
Limitations, Extensions & Future Work

- Minimum collateral
- Update fees via one-way channel
Limitations, Extensions & Future Work

- Minimum collateral
- Update fees via one-way channel
- Watchtower replacement
- Auditability
- Consensus → fork resilient
Limitations, Extensions & **Future Work**

- Minimum collateral
- Update fees via one-way channel
- Watchtower replacement
- Auditability
- Consensus → fork resilient
- Multiple parties
Thank you!

Questions?