Concurrency and Privacy with Payment-Channel Networks

Alejandro García Castañeda
Bitcoin’s throughput

Bitcoin transaction rate: **3 to 7 transactions per second.**

Visa’s peak transaction rate: **up to 47,000 transactions per second.**

**Scalability issue!**
Payment channel
Payment channel

Bob will get: 0.05BTC
Coffee shop will get: 0 BTC

Opening payment channel
Payment channel
Payment channel
Payment channel
Payment channel network (PCN)

Example: The Lightning Network

How can we guarantee that Bob is paying Alice?
Hash Time-Lock Contract (HTLC)

**Atomicity:** either the capacity of all channels in the path is updated or none of the channels is changed

HTLC (Alice, Bob, y, x, t):
1. If Bob produces the condition $R^*$ such that $H(R^*) = y$ before $t$ days, Alice pays Bob $x$ bitcoins.
2. If $t$ days elapse, Alice gets back $x$ bitcoins.
2 main issues

- Payment Privacy
- Concurrent payments
Privacy issue: Easy to see that contracts belonging to the same transactions are *linkable* among each other.

Use crypto? Cannot be verified by current implementation of Bitcoin (hash function, computing signatures)
Multi-Hop HTLC

**Goal:** The HTLCs of a single transaction are independent from each other to provide payment privacy.

Alice

Independent strings: $(x_1, … , x_n)$

\[ y_3 = H(x_3) \]
\[ y_2 = H(x_3 \oplus x_2) \]
\[ y_1 = H(x_3 \oplus x_2 \oplus x_1) \]
Concurrent Payments

What happens if two or more simultaneous payments share a payment channel in their paths?
Concurrent Payments

Two approaches for concurrent payments:

- **Blocking**: A payment is aborted if channel does not have enough credit => Possible deadlocks
- **Non-blocking**: At least one payment is guaranteed to terminate
Blocking Payments

- Let both payments fail.
- Receiver would not disclose the release condition for the locked payment channels.
- Payment channels would get unlocked only after timeout and bitcoins are sent back to the owner.
- The sender of an aborted payment can then randomly choose a waiting period to reissue the payment.
Non-blocking Payments (Rayo)

Assumes that there exists a global ordering of payments.

**Txid**: Transaction id field attached to each of the payments.

Users can queue payments with higher identifier than the current one “in-flight”, and abort payments with lower identifiers.

How about Privacy?
Concurrency vs Privacy

There is an inherent tradeoff between how to handle concurrent payments (blocking or non-blocking) and the anonymity guarantees.

Global identifiers leak the transaction ID of intermediate users!
5 hops payment:

- Lightning Network: 609 ms
- Multi-Hop HTLC: 1929 ms
Conclusion

• Efficient method to perform anonymous payments in PCNs.
• An inherent tradeoff between privacy and concurrency
• A system fully compatible with Bitcoin

Open Problems:

• Routing in PCNs
Thank you.
Any further questions?