Settling Payments Fast and Private: Efficient Decentralized Routing for Path-Based Transactions

Stefanie Roos
Pedro Moreno-Sanchez
Aniket Kate
Ian Goldberg
Limitations of Blockchains

- **Scalability**  
  (order of) 7 transactions/s in Bitcoin

- **Privacy**

  Traceability
Off-chain Transactions

A can send up to X funds to B

A sends Y funds to B via link with X funds

OR
Path-Based Transactions (PBTs)

Send 5 from S to R
Path-Based Transactions

Send 5 from S to R: 1 path
Path-Based Transactions

Send 6 from S to R: 2 paths
Algorithms in PBT Networks

- Routing

- Payment

- Accountability
Routing Algorithm Goals

- Privacy
  - Value privacy

- Sender/Receiver Privacy

Send ? from ? to ?

non-malicious
malicious

E sender? B receiver?
Routing Algorithm Goals (2)

- Scalability
- Effectiveness
- Efficiency
Related Work

- Canal/PrivPay: central server
- Max-Flow Algorithms: inefficient
- Flare: issues with network dynamics
SilentWhispers: Setup

• Landmark-based routing

Spanning Tree T1

Periodically rebuild spanning trees
SilentWhispers: Routing

1. For each landmark:
   Path = Sender to Landmark + Landmark to Receiver

2. Fund distribution:
   Get minimum funds on each path
   Sender assigns funds accordingly

For each landmark:
Path = Sender to Landmark + Landmark to Receiver

Fund distribution:
Get minimum funds on each path
Sender assigns funds accordingly

(c1 + c2 = c)
SpeedyMurmurs: Setup

- Key idea: network embeddings

Coordinate = Parent Coordinate + Enumeration Index

Dynamic+local reconstruction on join/leave
Network Embedding Routing

\[ \text{dist}(u,v) = |u| + |v| - 2 \text{cpl}(u,v) \]

Vector length
Common prefix length

<table>
<thead>
<tr>
<th>Neighbor ( u )</th>
<th>( \text{dist}(u,(1,1)) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>()</td>
<td>2</td>
</tr>
<tr>
<td>(1)</td>
<td>1</td>
</tr>
<tr>
<td>(1,2)</td>
<td>2</td>
</tr>
</tbody>
</table>
SpeedyMurmurs: Routes

Neighbor $u$ | $\text{dist}(u,(1,1))$
---|---
() | 2
(1) | 1
(1,2) | 2
SpeedyMurmurs: Routes

1. Divide funds randomly before determining the paths

\[ \sum c_i = c \]

2. Select neighbor such that
   1) Neighbor is closer to receiver
   2) Link has at least \( c_i \) funds

\( c_1 = 5 \Rightarrow \) forward to ( )

\( c_1 = 5 \Rightarrow \) forward to ( )
Privacy

- Value $c$ hidden from nodes not on paths
- Nodes on paths can estimate $c$
  
  5 landmarks
  
  Expected $c$: 10

- Sender/Receiver Privacy: obfuscated coordinates (Roos et al., Infocom 2016)
Static Evaluation: Results

Results for 3 landmarks (highest degree)

SW – SilentWhispers
SM – SpeedyMurmurs
FF – Ford-Fulkerson

Success Ratio
Path Length
Messages

SW  SM  FF
SW  SM
SW  SM  FF
49500
Dynamic Evaluation: Results

[Graphs showing the relationship between epoch number and count/stabilization for different scenarios labeled as Transactions and Set Link on the left, and SilentWhispers and SpeedyMurmurs on the right.]
Summary

- SpeedyMurmurs
  - Embedding-based routing
  - Dynamic maintenance
  - Concurrency-aware routing
- Effective, efficient, scalable, privacy-preserving
- Data sets and simulation framework:
  https://crysp.uwaterloo.ca/software/speedymurmurs/
Settling Payments Fast and Private

Stefanie Roos

Security

- Confidentiality
  
  ![Image of secure channel]

- Integrity
  
  ![Image of integrity verification]

  Verify c1, c2, ...

- Availability
  
  ![Image of availability]

  c1
Concurrent Transactions

• Routing for Y funds but payment not yet done

  ![Diagram showing routing for Y funds with payment not yet done]

• What about a second request?

  ![Diagram showing routing for Y funds with a question mark indicating uncertainty about a second request]

• Don’t route second request if X-Y-Y2 < 0
Evaluation: Data Set

Transactions Link changes 2013-2016

To USD

Ripple

Ripple

Static

Dynamic

Snapshot Nov 2016
>60k nodes
~100k links
~400k transactions

Starting with Jan 2013
(~9000 links)
~800k link changes
~700k transactions

Stefanie Roos

Settling Payments Fast and Private