Blitz: Secure Multi-Hop Payments Without Two-Phase Commits

Lukas Aumayr¹, Pedro Moreno-Sanchez², Aniket Kate³, Matteo Maffei¹

¹TU Wien, ²IMDEA Software Institute, ³Purdue University
Blitz is a new multi-hop payment paradigm for Payment Channel Networks:

- More efficient
- Reduced collateral from linear to constant
- Smaller size
- More secure

What’s in store?
Motivation and background

Blitz construction

Evaluation + comparison to current solutions

Summary
Blockchain:

- records every transaction

- Global consensus: everyone checks the whole blockchain

Bitcoin’s transaction rate: \(~10\) tx/sec
Visa’s transaction rate: \(\sim10K\) tx/sec

Exchange transactions off-chain, Blockchain for disputes
Payment Channels

1) Open
2) Update
3) Close
Payment Channels

1) Open

Alice

Lock 2 coins

Bob

2) Update

3) Close

Lock 8 coins
Payment Channels

1) Open

2) Update

3) Close

Alice

Bob

State 0

2

8
Payment Channels

1) Open
2) Update
3) Close

State 0

Alice

2
8

Bob
Payment Channels

1) Open
2) Update
3) Close

Alice

Bob

State 0

State 1
Payment Channels

1) Open
2) Update
3) Close

Alice

Bob

State 0

State 1

State 2

2 8

4 6

8 2
Payment Channels

1) Open
2) Update
3) Close

Alice

Bob

State 0
2 8

State 1
4 6

State 2
8 2

... State n
5 5
Payment Channels

1) Open

2) Update

3) Close

Alice

OR

Bob

Send state n

Send state n
Payment Channel Network (PCN)

- Infeasible to open channels with everyone
- Link channels to form a PCN
- Multi-hop payments
- e.g., Lightning Network (LN) [1]
  - 53M $ locked
  - 20k nodes
  - 46k channels

Scenario: Alice wants to pay 5 coins to Dave, via Bob and Carol
1. Dave samples $x$ and sends $y := H(x)$ to Alice
2. **Alice** sets up an HTLC with **Bob** holding 5 coins
   - **Bob** gets money if he knows $x$, s.t. $H(x) = y$
   - **Alice** gets money after timeout $3t$
Multi-hop payments in the Lightning Network

3. Bob sets up an HTLC with Carol
Multi-hop payments in the Lightning Network

4. Carol sets up an HTLC with Dave
Multi-hop payments in the Lightning Network

5. Dave redeems the HTLC with Carol by revealing $x$ and claims the 5 coins.
Multi-hop payments in the Lightning Network

6. Carol redeems the HTLC with Bob
Multi-hop payments in the Lightning Network

HTLC(Alice, Bob, 5, y, 3t)  
HTLC(Bob, Carol, 5, y, 2t)  
HTLC(Carol, Dave, 5, y, t)  

\[ y \equiv H(x) \]

7. Bob redeems the HTLC with Alice

⇒ Payment successful
Two-Phase Commit

Round 1
“Lock”

Round 2
“Release”

Two rounds of communication are required!

Round := sequential, pairwise communication from sender to receiver
Multi-hop payments in the Lightning Network

Payments happen off-chain in honest case

Staggered collateral to give enough time to claim on-chain in case of dispute
Properties & drawbacks of Lightning payments

- Scalability ✔
- “Balance Security” ✔
- Privacy ✔

Drawbacks:
- Staggered collateral lock time ✗
  - Decreases network throughput
- Takes two rounds ✗
- HTLC scripting requirements ✗
- Wormhole attack [2] ✗

Motivation and background

Blitz construction

Evaluation + comparison to current solutions

Summary
Again: Alice wants to pay 5 coins to Dave, via Bob and Carol
Alice defines a timeout T, independent of the path length.
Alice creates refund enabling transaction: $\text{tx}^{er}$
Pay-or-revoke paradigm

Before $T$:
- Alice

After $T$:
- Pay 5
  - Alice

Pay-or-revoke paradigm
Pay-or-revoke paradigm

Before $T$

Alice

After $T$

Bob

Carol

Dave

Alice

Bob

Carol

Dave
Pay-or-revoke paradigm

Alice

tx^er

Carol

Dave

Bob

before T

before T

before T

pay 5

tx^{er}

pay 5

pay 5

pay 5

before T

after T

before T

after T

before T

after T

pay 5

Alice

Bob

Carol

Dave

Alice
Pay-or-revoke paradigm

- Alice
- Bob
- Carol
- Dave

Before T:
- Alice
- Bob
- Carol
- Dave

Pay 5:
- Alice
- Bob
- Carol
- Dave

After T:
- Alice
- Bob
- Carol
- Dave

Confirmation: tx^er
Successful payment

Alice

Bob

Carol

Dave

before T

pay 5

after T

before T

pay 5

after T

before T

pay 5

after T

Confirmation: $tx^{er}$
Refund

Alice

Confirmation: tx^er

Bob

Carol

Dave

pay 5

after T

before T

pay 5

after T

pay 5

before T

pay 5

before T

pay 5

before T
More

- Fast track for instant payments

- Fast revoke for refunds without posting $tx^{er}$

- Privacy by using stealth addresses

- Check the paper for more information!
Motivation and background

Blitz construction

Evaluation + comparison to current solutions

Summary
## Comparison to current solutions

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Balance security</strong></td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Number of rounds</strong></td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1 (2 for fast track)</td>
</tr>
<tr>
<td><strong>Collateral lock time</strong></td>
<td>N/a</td>
<td>Linear</td>
<td>Linear</td>
<td>Constant</td>
</tr>
<tr>
<td><strong>Atomicity</strong></td>
<td>No</td>
<td>No (Wormhole)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Scripting capabilities</strong></td>
<td>Signatures</td>
<td>Signatures, timelocks, hashlocks¹</td>
<td>Signatures, timelocks</td>
<td>Signatures, timelocks</td>
</tr>
</tbody>
</table>

¹ Using constructions such as scriptless scripts, one could get rid of hashlocks.

Evaluation

- Blitz contract **26% smaller** than Lightning contract (HTLC)
- Can increase number of concurrent payments per channel
Evaluation

- Blitz contract **26% smaller** than Lightning contract (HTLC)
- Can increase number of concurrent payments per channel

- Simulation on Lightning Network snapshot
- Random payments, some are disrupted
- Constant (Blitz) vs. staggered (Lightning) collateral
- Depending on setting, between **4x and 33x fewer failed payments** in Blitz
Motivation and background

Blitz construction

Evaluation + comparison to current solutions

Summary
New multi-hop payment paradigm for Payment Channel Networks

- Only one round of communication
- Contract size reduced by 26%
- Reduced collateral from linear to constant
- Security against Wormhole attack

- Only requires Signatures and Timelocks
- Simulation showing practical advantage of constant collateral
- Formally modelled in UC framework and security proofs
- Compatible with the Lightning Network
Thanks!

eprint.iacr.org/2021/176.pdf

lukas.aumayr@tuwien.ac.at

@lukas_aumayr