Rollup, Privacy and Mixers

Benedikt Bünz
Recap: Rollup

Today: every miner must verify every posted Tx

verify all Tx \Rightarrow \text{short proof } \pi

Coordinator

\text{summary, } \pi

verifying proof is much easier than verifying 10K Tx
Referee Delegation

Coordinator and Validator run interactive binary search

Checks whether
$S_{n/2} = S'_{n/2}$
If yes disagreement in first half
Otherwise in second
Problem: Checks take a long time

- $\log_2(n)$ messages (1 hash per message)
- 1 Verification step on smart contract
- If either party timeouts declares winner
- Looser gets *slashed*, Winner rewarded
- Problem: $\log_2(n) \times$timeout
- No incentive to cheat
- But: Long wait till finalization!
Pipelined Assertions

- If prior state invalid, all subsequent bonds are slashed
- Coordinators can build on states before timeouts
Pipelined Assertions

Bond $i$

Rollup state $i$

State $i$ valid

Bond $i+1$

Rollup state $i+1$

State $i+1$ not valid

Bond $i+2$

Rollup state $i+2$

Coordinators can claim prior state not valid and continue given this.

Rollup state $i+1'$

If no successful fraud proof then reward gets slashed
Multiple Rollup Coordinators

• Rollup coordinator (in either scheme) is not trusted for security
• It can reasonably be a single coordinator
• But it is trusted for liveness
  • Censorship resistance
  • Progress of rollup state
• Multiple Coordinators?
Multiple Rollup Coordinators

• Rotating coordinators
• Random coordinator (using Beacon)
• Race to submit new rollup state (usually same party wins)
• One solution is using classical consensus between fixed set of coordinators
  • At least $2/3^{rd}$ of coordinators sign roll up
  • If trusted instant finality
Multi Coordinator Insurance

- Get insurance signature from $\frac{2}{3}^\text{rd}$ of coordinators
- If next block does not include transaction post signature
- Slash reward from intersection of insurer and rollup block signers
  - At least $\frac{1}{3}^\text{rd}$ of the coordinators
## Comparison SNARK vs Optimistic Rollup

<table>
<thead>
<tr>
<th>Optimistic Rollup</th>
<th>zkRollup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Higher TPS</td>
<td>Lower TPS</td>
</tr>
<tr>
<td>Arbitrary Smart complex</td>
<td>Only simple transfers</td>
</tr>
<tr>
<td>Slow finality (hours/days)</td>
<td>Faster Finality (minutes)</td>
</tr>
<tr>
<td>Instant finality with insurance</td>
<td>Instant finality with insurance</td>
</tr>
<tr>
<td>Trust that someone verifies</td>
<td>No trust required</td>
</tr>
</tbody>
</table>
Privacy
Privacy for Cryptocurrencies

What information might a user want to hide?

Identity (anonymity):
- Who they are
- Who they pay
- Who pays them

Amounts:
- How much they are paying
- How much are they receiving
- E.g. salary

Metadata:
- Script Sig, e.g. multisig threshold
- Smart contract
Anonymity

Weak Anonymity (Pseudonymity):
One consistent Pseudonym (e.g. reddit)
Pros: Reputation
Cons: Linkable posts, one post linked to you -> all posts linked to you
Writing style, topics of interest may link you

Strong Anonymity:
Cons: No Reputation
Who needs privacy for payments

Companies
Ford does not want to reveal cost of tires
Salaries of employees
Hedge funds want to keep investments private
Who needs privacy for payments

Consumers
Salaries, Rent, Purchasing things online, Donations
Who needs privacy for payments

- Criminals
- Stolen funds (WannaCry), buying/selling drugs, tax evasion
Who needs privacy for payments

• Applications
Privacy can prevent frontrunning
Exchanges may want to keep orderbook private
Sealed bid auction
Privacy of Digital Payments

Payments publicly visible/linkable

Payments only visible to bank/venmo. Optionally sender/receiver public

Unlinkable private payments

Less private  More private
Privacy in Ethereum

Weak Pseudonymity

Account public

Values public

Mostly one account per user

Some accounts known (Binance)
## Summary

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>1110 (bytes)</td>
</tr>
<tr>
<td>Fee Rate</td>
<td>0.0016173243243243244 BTC per kB</td>
</tr>
<tr>
<td>Received Time</td>
<td>Apr 10, 2017 12:38:00 AM</td>
</tr>
<tr>
<td>Mined Time</td>
<td>Apr 10, 2017 12:38:00 AM</td>
</tr>
<tr>
<td>Included in Block</td>
<td>000000000000000000000001f0115cca5856466832b337404032c88539ce2995e799e5c</td>
</tr>
</tbody>
</table>

## Details

<table>
<thead>
<tr>
<th>Transaction ID</th>
<th>Amount</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>c2561b292ed4878bb28478a8cafd1f99a01faeb9c5a906715faa595cac0e8d1d8</td>
<td>0.53333328 BTC</td>
<td></td>
</tr>
<tr>
<td>16k4365RzdeCPKGGwJDNNBEkXj696MbChwx</td>
<td>0.01031593 BTC (U)</td>
<td></td>
</tr>
<tr>
<td>1Bsh4KD9JT4dJcoo7S5uS1jvtmtVmREb7</td>
<td>1.47877788 BTC</td>
<td></td>
</tr>
<tr>
<td>1JgVBpwSTDMTROzXg9XpPDQRRHtNb5CsPA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1AFLhD4Eg2uZmFxfdXCyGUnqCqD5887u</td>
<td>2 BTC (S)</td>
<td></td>
</tr>
</tbody>
</table>

**Fee:** 0.00179523 BTC
Alice can have many addresses (creating address is free)

Alice’s addresses

Bob’s address

Change address

Ins: A1: 4 A2: 5  out: B: 6, A3: 3
• Buying book from merchant
  • Alice learns one of merchant’s addresses (B)
  • Merchant learns three of Alice’s addresses
• Alice uses an exchange BTC ↔ $  
  • KYC (Know your customer)
• Money serving business collect and verify IDs
Linking Addresses to Identities

- Buying book from merchant
  - Alice learns one of merchant’s addresses (B)
  - Merchant learns three of Alice’s addresses
- Alice uses an exchange \( \text{BTC} \leftrightarrow \$ \)
  - KYC (Know your customer)
  - Money serving business collect and verify IDs
- Exchange learns real ID
Donating to Wikileaks

Wikileaks had one address -> Easy to see who donates
Is Bitcoin Anonymous?

No!

It is possible to:

• Link all addresses of a single entity:
  • Determine total assets
• Given two TX A->B, C->D, Are B&C the same
  • If D knows C, can unmask B
  • Trace stolen funds, find tax evasion
  • Oppressive governments (Venezuela, North Korea)
• Test if Alice ever paid Bob (Wikileaks)

Often answer is yes for all 3. How?
Network Anonymity

end users

$\text{sk}_A$

$\text{sk}_B$

$\text{sk}_C$

Signed Tx

Can learn Alice’s IP address

Bitcoin P2P network

Solution:

Tor
Light client network anonymity

SPV client

All addresses and transactions

Full node

Fully linkable!
Heuristic 1:

Two addresses are input to same TX (and not multisig script)

-> both addresses are controlled by same entity
Idioms of use

Heuristic 2:
Change address is controlled by same user as input address
Which is change address: Used to be first address
Heuristic: Only new address, Non round, Less than inputs
Example tracing

output

transaction

Coinbase knows entity!
• Use Heuristic 1 and 2 -> 3.3M clusters
• ID 1070 addresses by interacting with merchants
  • Coinbase, Bitpay, ...
• Learn ID of 2200 clusters
  • 1.8M address
  • 15% of total value
• Track multiple thefts
• Learn total assets for each cluster
Another example

Ins: A1: 1. out: EC1 1
Ins: EC1: 1 out: S: 0.8, EC2: 0.2

Alice and Subcontractor learn EC’s profit margin. How can we prevent this?
Another example

Ins: A1: 1. out: EC1 1

Ins: EC1: 1 out: S: 0.8, EC2: 0.2

EC has many customers. Mix payments -> use some to pay sub
Making Cryptocurrencies anonymous

Mixing

Anonymous cryptocurrencies
Mixing

A1 -> M: 1

B1 -> M: 1

C1 -> M: 1

A2

B2

C2

Mixer

TLS

Ins: M: 3 Outs: B2: 1, A2: 1, C2: 1
Mixing Analysis

• Outside observer who is A2?
  • \( A2 \in \{Alice, Bob, Carol\} \)

• For Bob
  • \( A2 \in \{Alice, Bob, Carol\} \)

• The more the better mixing
Mixer Problems

• Mixer can deanonymize
• All outputs MUST have same value
  • If not you can match inputs and outputs
• Mixer takes transaction fees
• Mixer can steal funds
• ScriptPK for all outputs must be the same
  • Otherwise linkable on spend
CoinJoin (Mixing without Mixer)

CoinJoin TX

Ins: :A1: 5, B1: 3, C1: 2
Outs: B2: 2, A2: 2, C2: 2
Change (not private): A3: 3, B3: 1
Signed: Multisig A1, B1, C1

Out value = min of inputs

Usually ~40 inputs
CoinJoin

A1: 5, A3 (change)
A2 (over Tor)

Same

Add Signatures

Publish Transaction

Online Forum

PASTEBIN

A1: 5, A3
B1: 3, B3
C1: 2, C3
B2, A2, C2

What if A1 is spent?
End of lecture

Next lecture:
Zero-knowledge SNARKs