Unlinkable Outsourced Channel Monitoring

Thaddeus Dryja <rx@awsomnet.org>

Scaling Bitcoin Milano
2016-10-08
Channels are cool

- Cheap to make
- Cheap to break
- Update states real quick
- Link them together in a network

But...
There are risks

The price of scalability is eternal vigilance.
- Someone Famous

● Channels have to be watched
● By a full node
  (don’t use bloom filters anyway)
Get by with a little help

- Ask a friend to watch the channel, and e-mail you when it closes?
- Give them your private keys so they can grab for you?
- Give them all the txs grabbing invalid outputs?
- Give a reward for defending channel?
  - (doesn’t really work anonymously)
O(n) vs O(log n)

- Your own channels, everything can be tree-ified, and you can store log(n) data. (n = number of past states)
- Third party needs a signature for each state. No “flexible” signatures available (yet!)
How about privacy?

● Here, privacy improves safety
● Worry about people seeing your balances and transactions, less likely to outsource
● If outsourcing can be private, give it to anyone

● Minimize trust
Not really trusted third party

- Don’t trust them to:
  - Keep balance confidential
  - Keep the data you give them private
- Don’t even really trust them to monitor the channel; outsource to 10 parties, and just hope that 1 of them is paying attention
How to keep it private

- Basis: TXIDs don’t reveal the TX
- Signatures don’t reveal message
- Could encrypt data, give them txid[0:16], key is txid[16:32]
- Encryption needed for HTLCs
- Encrypted sig, script: 130 bytes, sig only: 64
- Variable size, timing; can’t be perfect
Signature storage only

- Only store 64 byte signature per state
- Observer generates pkScript when needed
- Script is:
  \[(\text{TimeoutKey AND 3 days}) \text{ OR RevokeKey}\]
- Revoke Key changes each state, hash-treeee
- TimeoutKey doesn’t need to change, but if static, can identify channel
Change revoke only

State 1
Timeoutkey: 02f8738a...
RevokeKey: 03591cb2...
<table>
<thead>
<tr>
<th>State 1</th>
<th>State 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timeoutkey: 02f8738a...</td>
<td>Timeoutkey: 02f8738a...</td>
</tr>
<tr>
<td>RevokeKey: 03591cb2...</td>
<td>RevokeKey: 02e9931b...</td>
</tr>
<tr>
<td>State</td>
<td>Timeoutkey:</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------</td>
</tr>
<tr>
<td>State 1</td>
<td><code>02f8738a...</code></td>
</tr>
<tr>
<td>State 2</td>
<td><code>02f8738a...</code></td>
</tr>
<tr>
<td>State 3</td>
<td><code>02f8738a...</code></td>
</tr>
</tbody>
</table>

Change revoke only
Change revoke only

<table>
<thead>
<tr>
<th>State 1</th>
<th>State 2</th>
<th>State 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timeoutkey: 02f8738a...</td>
<td>Timeoutkey: 02f8738a...</td>
<td>Timeoutkey: 02f8738a...</td>
</tr>
<tr>
<td>RevokeKey: 03591cb2...</td>
<td>RevokeKey: 02e9931b...</td>
<td>RevokeKey: 03aa25c1...</td>
</tr>
</tbody>
</table>

Observer knows 02f8738a... which is static
Identifying channel is trivial
Change both pubkeys each state

- TimeoutKey and RevokeKey have a base point, and a single per-state point added in
- This way both points change with each state
- Looks better, still doesn’t work though
Change both keys each state

State 1
Timeoutkey: 02f8738a...
RevokeKey: 03591cb2...
## Change both keys each state

<table>
<thead>
<tr>
<th>State 1</th>
<th>State 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Timeoutkey:</strong> 02f8738a...</td>
<td><strong>Timeoutkey:</strong> 03e4b4c7...</td>
</tr>
<tr>
<td><strong>RevokeKey:</strong> 03591cb2...</td>
<td><strong>RevokeKey:</strong> 02e9931b...</td>
</tr>
</tbody>
</table>
## Change both keys each state

<table>
<thead>
<tr>
<th>State 1</th>
<th>State 2</th>
<th>State 3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Timeoutkey:</strong> 02f8738a...</td>
<td><strong>Timeoutkey:</strong> 03e4b4c7...</td>
<td><strong>Timeoutkey:</strong> 03a7bf64...</td>
</tr>
<tr>
<td><strong>RevokeKey:</strong> 03591cb2...</td>
<td><strong>RevokeKey:</strong> 02e9931b...</td>
<td><strong>RevokeKey:</strong> 03aa25c1...</td>
</tr>
</tbody>
</table>
Change both keys each state

<table>
<thead>
<tr>
<th>State 1</th>
<th>State 2</th>
<th>State 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timeoutkey: 02f8738a...</td>
<td>Timeoutkey: 03e4b4c7...</td>
<td>Timeoutkey: 03a7bf64...</td>
</tr>
<tr>
<td>RevokeKey: 03591cb2...</td>
<td>RevokeKey: 02e9931b...</td>
<td>RevokeKey: 03aa25c1...</td>
</tr>
</tbody>
</table>

Looks harder; how to match channel state data with final script...?
Change both keys each state

- Know
  - timeBase, revBase
- Observe in final state
  - timePub = timeBase + statePoint
  - revPub = revBase + statePoint
- State point unknown, BUT:
  
  If (timePub - timeBase == revPub - revBase)
  
  Anonymity of channel is broken
Add 2 different points

- Add **2 different** points to pubkeys each state
- Both points can be HMAC derived from one parent hash, no additional storage needed

\[
\text{hash(state\_nonce, \text{"R"})} = \text{revScalar} \\
\text{hash(state\_nonce, \text{"T"})} = \text{timeoutScalar}
\]
Scalability of observer

- Observer’s DB can be much larger than the whole blockchain!
- 10K channels, 1M states each
- 10G txs, ~1TB storage
- Each in-block TXID seen, match against 10G stored partial TXIDs (doable)
Unlinkability isn’t perfect

- HTLCs. Ignore if small? Timing, add noise HTLC data to observer
- Not consensus-critical, but everyone should do the same thing! (larger set)
  - BTW everyone use BIP 66!
- Closing / deletion timing
- State update timing (add lag?)
Further ideas

• Back-propogation of decryption keys for HTLC / other data
• Group or ring signature to indicate that this is a real channel, not fake / spam
  ○ Needs known set of channel pubkeys, which you probably will need anyway for routing
• Ideally, only need 1 altruistic node to defend the whole network
Questions

● Still work-in-progress
● Looks promising; hopefully, invalid channel closes can be made close to impossible
● 1-of-N altruism seems pretty good

Thanks & Ciao!