Light Clients for Building UTxO Ledger Transactions

Pyrros Chaidos[1 [0000−1111−2222−3333]], Aggelos Kiayias[1 [1111−2222−3333−4444]], Marc Roeschlin[1 [0000−1111−2222−3333]], and Polina Vinogradova[1 [0000−0003−3271−3841]]

Input Output, Global firstname.lastname@iohk.io iohk.io

Abstract. The abstract should briefly summarize the contents of the paper in 150–250 words.

Keywords: First keyword · Second keyword · Another keyword.

1 Introduction

What sets us apart?

- Atomicity of payment+service
- Model for (trustless) 2-party transaction construction rather than proving things about chain/ledger state
- Do not require establishing a relationship with SP or any other set-up
- inherent timeliness of transaction construction incentivized by SPs desired to earn their tip. This is in contrast with the possibility of stale info provided from old Mithril snapshots in other LC models

We claim that our work can be used by any UTxO or EUTxO blockchain (with some adjustments to the details of intent specification). We use blind signatures [1]

2 Related Work

Compare our approach with :

- "Free" websites monitoring the chain – mention they lack long-term sustainability.
- Bridges (trustless and trusted)
- Payment channels
- LCs that operate on single-prover model (eg. with an established relationship via deposit)
- LCs that operate on multi-prover model
- LC SoK
- Solver networks

2.1 Technical Background

3 Ledger Model

We take as our ledger model the UTxOma model, which is UTxO with multi-assets. Explain why we chose this over the basic UTxO and the full EUTxO models.

4 Light Client Specification

What is a light client?

- User - LC interface
- LC characteristics/ limitations
  - What are the capabilities of a light client?
  - Does it remember all addresses it has been paid at (tx history)?
  - Do we assume that viewing keys exist? Can we assume LC can generate first x addresses from its private key in a deterministic way?
  - Is light client allowed to maintain state, and what state can they maintain if so? Secret key is the minimum state.
  - Sanity test: if we dismiss all requirements we should recover a full client.
- LC - Full Node interactions
  - Protocols to support user requests
  - Security reqs (protecting integrity/ privacy/ SPO revenue?)

Can we describe the differences between light wallets, bridges and light nodes in our framework? (Probably yes).

How can we formalize the intent of a light client without revealing secret key?

Can we have viewing keys?

5 Threat Model

- Client finds out too much from SP answer and can submit tx without payment to SP (that's why the inputs must be hidden)
- SP lies to client about the UTxOs being spent by the tx, and tricks them into doing something they didn't want to do (that's why 0-knowledge proof that outputs were correctly specified)
- Suboptimal transactions possibility : Do we want to let users specify what they want optimized for in their intents (e.g. minimize fee or find best price on exchange offer)? Can we assume that competition across SPs will enable client to get a better response?

6 Intent Specification

Intent (DSL or predicate to describe intent of the client, ie what they want to do to the ledger state).

7 Light Client Protocols

8 Analysis

9 Conclusion

Acknowledgments. A bold run-in heading in small font size at the end of the paper is used for general acknowledgments, for example: This study was funded by X (grant number Y).

Disclosure of Interests. It is now necessary to declare any competing interests or to specifically state that the authors have no competing interests. Please place the statement with a bold run-in heading in small font size beneath the (optional) acknowledgments[1], for example: The authors have no competing interests to declare that are relevant to the content of this article. Or: Author A has received research grants from Company W. Author B has received a speaker honorarium from Company X and owns stock in Company Y. Author C is a member of committee Z.

References

1. Fuchsbauer, G., Wolf, M.: Concurrently secure blind schnorr signatures. In: Joye, M., Leander, G. (eds.) Advances in Cryptology – EUROCRYPT 2024. pp. 124–160. Springer Nature Switzerland, Cham (2024)

[1] If EquinOCS, our proceedings submission system, is used, then the disclaimer can be provided directly in the system.