Payment Channels Overview

Orfeas Stefanos Thyfronitis Litos

University of Edinburgh
o.thyfronitis@ed.ac.uk

Abstract. This is an overview of the existing literature on virtual payment channels. Lightning [1], Perun [2] and TeeChan [3] are considered.

1 Introduction

Virtual payment channels are constructions that permit the secure exchange of assets between remote agents without the need for each transaction to be recorded in a global database. They are constructed in a way that either does not allow the agents to cheat (given appropriate assumptions), or give the opportunity to the cheated agents to report the latest valid state to a global database (i.e. blockchain) and reclaim their assets.

2 Perun

The basic feature of Perun is the ability to create payment channels just like Lightning. Furthermore, it uses the Turing-completeness of Ethereum contracts to provide the option of so-called multistate channels; these channels can effectively support the creation of further virtual channels on top of them. Both kinds of channels require two moments of interaction with the Ethereum (or any other Turing-complete) blockchain: one at the creation and one when closing; the intermediate states of the channel need not be recorded on the blockchain.

As an example, of a virtual payment channel on top of multistate channels, let *Alice* and *Ingrid* have a multistate channel between them and let the same hold for *Ingrid* and *Bob*. Both multistate channels correspond to on-chain contracts. Then *Alice* and *Bob* can create a virtual payment channel with which *Ingrid* has to interact only at its creation; even in case that *Ingrid* later stops cooperating and does not help the other two parties to close the channel, they can both claim their money from their underlying multistate channel and eventually from the blockchain, thus *Ingrid* is required only for the virtual channel initiation.

One practical problem of this construction is that it does not scale well to virtual channels with many intermediaries. For *Alice* to create an $Alice \leftrightarrow Bob$ virtual payment channel, when the existing multistate channels are $Alice \Leftrightarrow Ingrid \Leftrightarrow Jane \Leftrightarrow Bob$, a virtual multistate channel specially crafted to support the desired virtual payment channel must be created between *Ingrid* and *Jane*. This means that when *Alice* requests from *Ingrid* to initiate the $Alice \leftrightarrow Bob$ channel through her and *Jane*, *Ingrid* has to establish a new channel with *Jane* before forwarding *Alice*'s request to *Jane*. Only then is the message from *Jane* to *Bob* ready. For *Alice* to be sure that the virtual payment channel is open, she has to wait for another message trip from *Bob* to *Jane*, then to *Ingrid* and eventually back to her. It seems that the amount of messages needed to create a virtual payment channel of length n is $\mathcal{O}(n^2)$. This is far from the ideal requirement of *Alice* being able to add a virtual channel with no interaction with any other party than *Bob*.

References

1. Poon J., Dryja T.: The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments
2. Dziembowski S., Eckey L., Faust S., Malinowski D.: PERUN: Virtual Payment Channels over Cryptographic Currencies. IACR: Cryptology ePrint Archive (2017)
3. Lind J., Eyal I., Pietzuch P., Sirer E. G.: Teechan: Payment Channels Using Trusted Execution Environments. ArXiv preprint arXiv:1612.07766 (2016)