Authors

: L4 Research Team. Special thanks to Armani Ferrante and Liam Horne for coming up with the on-chain transfer mechanism a few months ago.

Background

In a state channel network, users can interact (enter contracts) with each other even if they don't have a direct channel open with each other. A simple example of this is multi-hop payments in the lightning network, implemented via HTLCs.

There are a few constructions for this, including metachannels (section 5.9 ofhttps://counterfactual.com/statechannels), Perun's virtual channels, Sprites, Celer Network's conditional payments and others, each with different properties.

In general, however, the intermediaries that mediate an interaction must lock up some collateral for an amount of time (e.g. in the lightning network, they are locked in a HTLC). Since users won't consent to having their money locked up for an indefinite amount of time, there must be a time bound beyond which the intermediary must be able to recover their collateral. Furthermore, the intermediary is likely to charge an amount of fees proportional to the collateral lockup time (an interest rate).

Introduction

This note describes how to build metachannels for which the users (i.e., non-intermediary parties) can enter contracts beyond the length of time for which collateral is locked up.

This only describes how to do it for 2 users (I don't know how to extend this for more than two users!)

Problem

We assume that Alice and Bob wish to enter a contract S

. Alice locks an amount a_0

into the contract and Bob locks an amount b_0

into it, and intermediaries I_1, I_2, \ldots I_n

offer to lock up an amount of collateral k = a_0 + b_0

until a deadline T

.

In the old design (section 5.9 of the paper linked above), we create a proxy object for each channel, all of which observe the same counterfactually instantiated object S

(observation is denoted by dashed arrows). Let an instance of proxy object be denoted P_{ij}

where i, j

are channel participants.

[

image

1408×452 24.9 KB

](https://ethresear.ch/uploads/default/original/2X/0/0cf7f37cf9d3f5500a942f04605b2c0289b39fb7.jpg)

When S

is finalized to a value that assigns Alice an about a

, then P_{ij}

assigns a

to i

and k - a

to j

. (dashed arrows denote assignment of ownership)

[

image

1452×462 32.9 KB

](https://ethresear.ch/uploads/default/original/2X/e/ec101e8e111f466965ba7e1b4bc3f56f34c6c342.jpg)

Observe that each P_{ij}

has k

assigned to it, each I_m

recovers k

, A

recovers a

and B

recovers k-a

.

More concretely, the proxy objects implement these rules: they are parameterized by an observed object S

, a deadline T

, a left party i

, a right party j

, and a collateral amount k

.

1. If S

is not finalized and the current block height is less than T

, nothing is permitted.

1. If S

is finalized at a

, i

may withdraw an amount a

and j

may withdraw an amount k-a

This design means that A

and B

cannot enter a contract that lasts beyond T

, since after time T

, each intermediary I_m

needs to recover their collateral, hence S

must

finalize to some value (i.e., this forced finalization at T

must be enforced either in S

or in the proxy).

Solution

In the new design, if S

has not finished by the deadline (denoted by a question mark), the following happens instead (a solid arrow denotes on-chain transfer of funds):

[

image

1420×442 18.2 KB

](https://ethresear.ch/uploads/default/original/2X/9/9deb2e3273fad9cc3ecf84e0cfcb9f98a013d44d.jpg)

Once again that each P_{ij}

has k

assigned to it, each I_m

recovers k

, but now A

and B

collectively own an amount k

, locked in S

.

We see that S

must record who sent it money, in order that the proxy objects “to the left” of the sender send to the right, and proxy objects “to the right” send to the left.

Additionally, note that it is safe to allow any party to send k

at any time, not just when the deadline is over. This is desireable because parties might want to do this when gas prices are particularly low, but when the block height is not T

yet.

Hence the proxy objects now implement these rules: they are parameterized by an observed object S

, a deadline T

, a left party i

, a right party j

, and a collateral amount k

. The set of parties must be ordered, so that we can compare them; in this case, A < 1 < 2 < 3 < B

, and we require i < j

.

1. If S

has balance 0 and S

is not finalized, then either i

or j

may send an amount k

to S

.

1. If S

is not finalized, the current block height is less than T

, and S

has balance k

, nothing is permitted.

1. If S

is finalized at a

, i

may withdraw an amount a

and j

may withdraw an amount k-a

1. If S

is not finalized and the current block height is over T

, and S

has balance k

, ask S

who sent it the amount k

; call that sender is P_{i_s j_s}

. If i < i_s

, then j

may withdraw k

. If i > i_s

, then i

may withdraw k

.

These notes is also available in a version-controlled form athttps://github.com/counterfactual/t-metachannel-notes