The following is my attempt at a two-round atomic swap protocol for Plasma Cash. Suppose an atomic swap of X going from A -> B and Y going from B -> A.

1. A generates a random key S_A

and a value h_A = hash(S_A)

. B similarly generates S_B

and h_B = hash(S_B)

. They share h_A

and h_B

.

1. A and B sign an “intent-to-transfer” message containing (i) their coin ID, (ii) the counterparty coin ID, (iii) h_A

and h_B

1. The intent-to-transfer messages get included into the Plasma Cash chain in their respective slots (that is, the “new owner” of each coin is the respective intent-to-transfer message). The Plasma Cash chain makes sure to include them in the same block.

2. A and B publish S_A

and S_B

1. A “secret publication record” containing S_A

and S_B

gets committed to the main chain in a Merkle tree (ie. the tree could contain many such records from many exchange events)

When the “current owner” of a coin is an intent to transfer message, a special exit challenge rule is added: a “secret publication event” containing S_A

and S_B

that is committed to on-chain in the block immediately after

the block containing the intent to transfer message can be used to challenge the exit to transfer the owner from the sender to the recipient, as long as this challenge is made within 7 days. If a challenge is made within 7 days on one exit, the deadline is extended to 14 days for the other exit.

These rules also apply if an intent-to-transfer owner is part of a coin’s ownership history, eg. if the ownership chain is A -> (A->B) -> C -> D, then D’s right to exit the coin would be dependent on whether or not it’s true that either (i) the (A->B) -> C transaction was signed by A

and there is no

secret publication event for (S_A, S_B)

or (ii) the (A->B) -> C transaction was signed by B

and there is

a secret publication event for (S_A, S_B)

.

At all stages of the above, if A or B or the chain fail to fulfill their duties the other parties have an emergency action they can undertake to ensure a safe outcome:

- If either A or B do not publish an h

value, the process simply terminates and fails.

- If either A or B do not publish an intent-to-transfer message, the chain can make sure the other message does not get included. If the chain misbehaves and includes only one of the two messages, or includes them in different blocks, then the honest party can exit; because they did not publish their

S

value, they are safe.

- If either A or B do not publish their S

value, the counterparty can by default wait for the next Plasma Cash block, and if nothing has happened, send a transaction to transfer their coin back to themselves; after 2 blocks it can legally get included in the Plasma chain.

- If the Plasma Cash chain withholds part of the next block, then A and B can both exit their (originally owned) coins. If either one of the exits is challenged with the (S_A, S_B)

pair, then the counterparty has the ability to do the same to the other exit within 7 days (due to the deadline lengthening rule)

Performance properties:

- A malicious party can only lock up their counterparty's coins for 2 blocks
- An honest party can only be forced to exit by a malicious operator, not a malicious counterparty
- In the simplest scheme, participants are required to watch O(N)

data (the secrets) only during the one block when they are transacting their coins, and otherwise they are required to watch O(C * log(N))

if they hold C

coins.

- The process for completing a transaction involves a round of data exchange for exchanging hashes, a Plasma confirmation for the intent to transfer, and one further round of data exchange and Plasma confirmation for the secrets, so two rounds of data exchange and Plasma confirmation in total.

The need to download the whole block containing the secrets after a transaction can be reduced further by making a 2D Merkle tree (ie. a Merkle quad tree) where (S_A, S_B)

must be included at position (A, B)

; this reduces the data requirements to O(C * log(N))

in all cases.