Hi,

in this thread, I would like to discuss the possibility to copy utx-outputs from one plasma chain into another plasma chain.

In order to describe the technique I have in mind, I would like to talk about 3 processes:

1. How a user could withdraw all his utx-outputs from a plasma chain with one request to the root-chain.
2. How several users could copy all their utxos from one plasma chain into another one.
3. How we could allow 'trusted Exiters' to copy our outputs from one plasma chain to another very conveniently.

How a user could withdraw all his utx-outputs from a plasma chain with one request to the root-chain.

If a user detects a data unavailability or some false behavior from the chain operator, the plasma contract on the root-chain could allow him to formulate him a trueBit challenge game like this:

" If I review all my unspent, valid utxos from the plasma chain and add them up, then I get X ether and if I sort them alphabetically, put them into a Merkle tree, then I get the hash H1 and the lowest priority from all outputs is P". The plasma contract on the root chain would require the person putting up this trueBit-game to send a bond along, which could be used to reward a successful challenger. Once this withdraws request is put into the root chain anyone can inspect the plasma chain for all unspent utxos of this particular user [UTXO_1, UTXO_2, … UTXO_N], sort them alphabetically, hash them together, calculate their sum and check their lowest priority. If their hash is not H1, P is no correct or the sum is not X, everyone can challenge them:

- If the hash is not H1, one would ask for the hashes of the Merkle trees made of: [UTXO_1, UTXO_2, … UTXO_N/2] and [UTXO_N/2+1, UTXO_N/2+2, … UTXO_N]. One of them would not be correct and one could challenge them again. This proceeds until the disagreement between the challenger and the withdraw requesters is about one particular [UTXO_K]. Now the challenger would have to prove that this particular [UTXO_K] either, has already been spent, does not exist in the root chain or that there is actually another output, which should be the Kth output. All these things could be proven on the root chain.
- If the sum is not X, one would ask for the sums of the outputs of: [UTXO_1, UTXO_2, … UTXO_N/2] and [UTXO_N/2+1, UTXO_N/2+2, … UTXO_N]. One of them would not be correct and one could challenge them again. This proceeds until the disagreement between the challenger and the withdraw requesters is about one particular [UTXO_K]. Now, this output can be processed on the root chain and it could be checked whether it really has the claimed size or not.

-If P is not correct, one could just hand in an unspent output with a worse priority.

The trueBit-withdraw request would be processed with the priority P if it is not challenged. Users should not include any outputs, which they have signed once and are included in the plasma chain, but are not signed twice into the withdraw request, in order to keep things simpler.

How several users could copy all their utxos from one plasma chain into another one.

Method 1 would allow one user to withdraw all their funds into another contract by putting up this trueBit-withdraw request. Now several users could come together and put up the following trueBit-withdraw request:

" If we -the accounts [A1, A2, … A_M] - review all unspent, valid utxos from the plasma chain and add them up, then we get X ether and if we order these alphabetically, put them into a Merkle tree, then we get the hash H1 and the lowest priority from all outputs is P; this message is signed by A1, A2, …, A_M". If the request is not challenged, the plasma contract PC1 on the root chain could deposit the X Ether to another plasma chain PC2 contract. Now if an account A_i wants to withdraw one of his utxo's [UTXO_A_ui] on the 2nd plasma chain, he would post the withdraw request to the PC2 contract with a proof that [UTXO_A_ui] is legit output in the other plasma chain PC1. PC2 would now validate this claim by validating this request exactly as a withdraw request would have been validated by PC1. Also the withdraw request can be challenged by proving that this output was already spent either in plasma chain 1 or plasma chain 2.

But of course, the utxos from the first plasma chain could also be spent on the plasma chain 2 first and then only be withdrawn later.

Using this technique, we can copy many utxo's from one plasma chain into another one, without losing any information. This would be of tremendous help in case of a mass exit, as only many outputs can be processed without actually touching the root chain.

How we could allow 'trusted Exiters' to copy our outputs from one plasma chain to another very conveniently.

The technique described in 2 can be used to make mass exits from one plasma chain into another one quite convenient. A user U1 on the plasma chain could approve a "trusted Exiter" TE on the plasma chain to exit their funds in case of a data-unavailability into another predefined plasma chain. Once a trusted Exiter is approved, all funds transferred on the plasma chain from U1 are only valid, if TE signs them 2 times as well. This is needed so that a user hiding a transaction from the TE, cannot destroy the exit-request from the TE. The TE will sign all transactions of his clients immediately, once he sees them. If he does not sign them, then the TE can be unapproved by the plasma chain user.

In case of a data-availability the trusted Exiter would put up trueBit-withdraw request:

" If we review all unspent, valid utxos from the plasma chain of accounts that approved me to withdraw their funds and add them up, then we get X ether and if we sort these alphabetically, put them into a Merkle tree, then we get the hash H1 and the priority P for withdraw should be: max(priority of unspent output, priority of last spent output, which was signed two times). If this request cannot be challenged, the X ether will be sent to the predefined plasma contract PC2".

If the withdraw request is made correctly, all outputs of all accounts which approved the trusted Exiter will be sent to the new plasma chain with plasma root contract PC2. The trusted Exiter can not steal there any coins, he just transferred them into the predefined contract P2. The only trust he is getting is that he does not cause inconvenience by transferring funds into other chains although there is no data-unavailability.

Note1: the trusted Exiter would actually need to put a 2 step withdraw request:

1. First, we would put up a note that he wants to withdraw all outputs associated with his address.

2. Secondly, after 1 week when everyone revealed their withdraws request in the priority queue, he or other people can actually put the sum of outputs, hash of outputs and withdraws priorities up for a trueBit-challenge. This updating of the withdraw request needs to happen since people might wanna withdraw their outputs individually and thereby impacting the outputs of the withdraw.

Also, this would allow one user to have several trusted Exiter and then only the trusted Exiter with the highest priority would actually do the withdraws. I think the concept of a trusted exiter is very powerful since it allows copying whole output sets into another plasma chain. If there are several Plasma chains each operator could also run a trusted Exiter in the other chains and if data availability occurs in another chain, he would be able to withdraw user funds from the one plasma chain into their own plasma chain and thereby winning new customers.

Something similar could be also constructed for plasma cash.