Thanks to David Knott and Ben Jones for conversations that led to this post, and to Vi for review

Background

We previously introduced a mechanism for fast withdrawals in Plasma. Our design basically allowed users to tokenize and quickly sell withdrawals to liquidity providers. This decreases the wait time for the end user and generally improves user experience.

However, we designed our mechanism in the context of a functioning Plasma chain. Part of that design relied on users first sending funds to a special address on the Plasma chain. If the consensus mechanism stops producing blocks, then it's not possible for users to send their funds to this address. We still want to enable fast withdrawals in all cases, so we were motivated to find a construction that can operate in this worst case.

Construction

This design relies on the operator locking funds up for individual users in exchange for some fee. Users can then point to these bonds when they want to execute a fast withdrawal. Users tokenize these withdrawals through a smart contract and market them to third-party liquidity providers (probably at a discount). When liquidity providers want to purchase an exit, they'll first verify that the UTXO was unspent at the time of exit (plus a few blocks buffer time to avoid race conditions). If the UTXO is valid, providers can safely purchase the exit.

If there are no challenges, then the provider is simply paid out by the exit. If the UTXO was actually spent before the end of the buffer period (exit time + a few blocks), then a challenge will block the exit and the liquidity provider will not receive funds. This can only happen in the case that the liquidity provider did not correctly validate the UTXO before purchasing. However, if the UTXO is ever spent after the buffer period, then the operator cheated (included a double spend) and the provider cannot be at fault. A challenge that shows this is the case will block the exit but pay out the provider from the operator's bond

. This way, the provider can be sure that they'll always be paid out.

Note that the value of the withdrawal should be less than or equal to the value of the user's rented bond. Otherwise, the bond wouldn't fully cover the provider. Users could be allowed to point multiple UTXOs at the same bond, as long as the total value pointed at the bond is less than the value of the bond. The same bond can be reused for new exits as the original exits process.

If the Plasma contract allows users to specify an address to which exiting funds are paid, then we can simply implement this as an additional trustless contract on top of the Plasma contract.

Considerations

Operator Strategy

Users can be certain that they'll be able to use fast withdrawals whenever they have a bond with the operator (with value up to the value of the bond). It's likely that an operator would therefore attempt to limit the total outstanding bond value before trying to cheat. Although this is unavoidable, it does act as a strong canary for operator malfeasance and should be taken into account. Users who must be sure that their funds can always be withdrawn quickly may wish to exit if they're ever unable to secure a bond.

PoS Validator Stake

It may also be possible to make dual use of existing stake by allowing validators to use their stake as a bond. Fast withdrawal bonds are paid out in the case that an exit is blocked by a later spend. Conveniently, this is also a slashing condition. If the bond is several times larger than the exit value, we can pay out the exit value and slash the remainder. This may have some weird economic effects and might not play nice with the broader PoS mechanism, so it probably merits further exploration.

Capital Lock-up Costs

It's usually pretty expensive to lock up funds. We can assume that the operator will only lock up funds for these bonds if it's profitable to do so. The operator may have some sort of extra-protocol incentive to provide these bonds at a lower-than-borrowing rate, but it's something to consider. Users may be comfortable paying these fees if framed as insurance on their funds.

The existence of capital lock-up costs may also limit the ability for the operator to provide these bonds. This may decrease the impact of low bond issuance as a canary for bad behavior.

In the end, it's always possible for users to sell withdrawals to liquidity providers in a trustful way by promising not to spend their UTXOs. This sort of arrangement is much cheaper (no lock-up costs) but requires some sort of out out-of-band (legal) settlement if the users misbehave.

Further Questions

- How can we reduce the required capital lock-up?
- Can multiple users share the same bond?
- How do we efficiently represent these bonds?

Notes

As always, feedback is much appreciated. Let me know if you spot any problems with this design or if you’d like to add anything!