Intro and context

Context:

- Former protocol design: A protocol to Air-drop shielded Namada native token to Zcash Shielded Assets holders - HackMD
- Chris Zcash post: RFC: Proposal for a strategic alliance between Namada and Zcash - Ecosystem Updates - Zcash Community Forum
- Techincal RFC: Technical RFC: ZEC-NAM shielded airdrop mechanism - Ecosystem Updates - Zcash Community Forum
- Repo: GitHub - anoma/namada-shielded-airdrop

In brief, the first iteration of the protocol (which is worth reading to have a good understanding of the problem space) constructed a transaction the linked together an input zcash note and an output MASP note on a 1:1 ratio using the convert circuit.

To do so, we where creating a spend circuit on zcash, which was never going to be posted on the zcash chain, and would have acted as a proof of holding shielded assets. After a conversation with ZCash cryptographers, we realized that this approach was publishing the nullifier of a zcash note before said note had been spent: a likability issue between note commitment and its nullifier will arise when the note will be spent.

To come around this problem, a claim circuit

is introduced. The claim circuit, seen on the github page, acts as a spend circuit, but without ever revealing the nullifier of the zcash note used for the claim. It works with a non-membership proof, that proves that a private witness nullifier is not among the public set of relieved nullifiers.

A transaction is then made of:

- Claim description
- Covnert description
- Output description

Similarly on how a transaction with using the convert circuit happens.

Challenges

- Protocol Implementation: the implementation of the protocol would be complex, as it involves writing novel circuits, transaction structures and signatures which have to be written from scratch. On the other hand there are libraries that can be used as blueprint model (zcash sapling, MSAP).
- Protocol integration: once the protocol is implemented and a working API has been set in place, a pipeline that involves zcash wallets, namada validity predicates (which could be on-chain or off-chain) would need to put in place. It would define where the airdrop transaction is created and how it gets validated.
- Orchard: a lot of shielded assets are kept in the Orchard pool, which is written in halo2. I am currently researching the feasibility of having a convert circuit that can bundle together in a transaction an Orchard note with a MASP note. In my current understanding it would involve writing circuits for JubJub curve (the sapling curve) operations in halo2.The challenge lies mostly in the value balance check, as the pederson commitments for the value commitment are not homomorphic when they are computed on different curves. On the other hand, we could have a validity predicate that host verifiers in both groth-16 (masp) and halo2 (orchard).
- MASP transparent balance: talking to Marco we realized that the shielded airdrop increases the value of the MASP pool without increasing its transparent balance, which can cause funds to get stuck in the MASP. A naive solution to this problem would be to increase the MASP transparent balance by the total amount of the balance of the zcash pool. Different solutions should be researched, keeping in mind the the value of the claim airdrop can not be leaked.

First to-dos (2-3 weeks)

- Research Orchard-MASP compatibility and evaluate complexity of the problem. I think this should be the first to do, because it can have an impact on how we abstract the protocol specification and modularization of the problem, vs a monolithic spec that is Groth16 through and through.
- Write a fully descriptive specs of the protocol.

Other usage outside airdrop

It would be interesting to see if this protocol can open up to further applications other than a shielded airdrop that can increase the utility of the MASP and Namada chain in general for zcash holders.