Myself and my co-conspirator Sean (who is not on this forum) are working on a project which aims to privately trade non-fungible assets. Our project has a particular need for the ability to prove membership of transaction participants in a public whitelist in order to ensure validity of the transaction. We are looking to do so without revealing the particular identity of each participant through utilizing a zero-knowledge proof. We are both exploring zk-SNARKs to this end, and do not know of a method to do this in the prior literature. (Please point us in the right direction if there is!)

We came up with a construction of how to do this tonight and wanted to run it past the community to understand if there are any flaws in the construction. It requires:

1. The root of a Merkle Tree for this membership list that is agreed upon by all parties (public parameter)
2. A fixed depth to that tree (constant; e.g. depth of 32)
3. A Merkle proof of an identity (represented by a Public Key) in this list (private parameter)
4. The private key corresponding to the PubKey in the proof (private parameter)

Merkle Proof, for reference:

The construction of the zk-SNARK would be as follows:

1. Given H_{ABCDEFGH}

(public), prove that HASH(H_{ABCD}, H_{EFGH})

is equal

1. Assuming above holds, prove that H_{ABCD} == HASH(H_{AB}, H_{CD})
2. Repeat 2) for up to the depth of the tree (known prior to proof construction)
3. Prove the hash of the Public Key (T_D

, secret parameter) is equal to H_D

1. Finally, prove the Private Key (secret parameter) is associated with the given Public Key T_D

.

A bit of an explanation as to why this would work is that each step in the zk-SNARK provides transitive trust properties to the next step in the process, creating an overall quality guarantee of the proof that is equal to the depth times the quality guarantee of zk-SNARKs in general (which is high, but not absolute). Since the network participants all know the root hash to start with, and trust the algorithm in the zk-SNARK to evaluate the merkle proof, they can trust that each step in the process of handling the Merkle Proof validation builds upon trust of the prior and finalize to the node that is only known to the prover. The last step proves the knowledge of the secret information necessary to show that the prover is indeed in control of the key of the node required for their Merkle Proof to be valid.

Excuse my lack of precise terminology for this outline, hopefully it is clear enough what the construction is here that it is possible to evaluate its correctness. If not, I can create a more formal edit when I have more time and sleep