Permissioned Privacy Pools

Ethereum users wish to protect their privacy while interacting with the blockchain. The only practical way for users to fund a fresh

Ethereum address is through the use of a privacy pool. In this post we start a discussion on various designs for a permissioned privacy pool

which allows users to protect their privacy, but can also allow operators to keep unwanted participants out of the pool.

There are two basic approaches to permissioned privacy pools, which can be combined. The first is to curate an “allowlist” of addresses that are allowed to deposit. The second is to curate a “blocklist” of unwanted addresses, and require a ZKP from users upon withdrawal that their deposit address is not

part the blocklisted addresses.

Allowlist

The purpose of an allowlist is to proactively

limit privacy pool participation to approved entities only. This could be token holders, NFT holders, all unique humans according to Proof-of-Humanity, or be based some other scheme.

By itself, an allowlist only provides surface level protection against unwanted deposits. If an unwanted address is mistakenly added to the allowlist, then the address will be able to deposit into a privacy pool until its access is revoked. To mitigate the potential damage from a single mistake, rate-limits can be imposed on each deposit address. More trustworthy deposit addresses can have higher rate limits. This would also help mitigate the risk of approved entities going rogue and abusing their access to a privacy pool.

Blocklist

Vitalik explains the basic outline of a blocklist-enabled privacy pool in thisclip.

The purpose of a blocklist is to reactively

restrict privacy pool withdrawals from unwanted depositors. The fundamental challenge is identifying which deposit addresses to restrict, which requires imposing a time delay (e.g. 1 week) between deposits and withdrawals. To be effective, a blocklist would need to be seeded with known unwanted addresses (e.g. addresses responsible for hacks), but also expanded to trace all addresses downstream from the initial set, including tracking cross-chain activity. Due to the computational complexity of tracking unwanted addresses, effective blocklists will likely be only queryable offchain. For onchain authentication, blocklists can publish merkle roots directly onchain periodically, or sign an offchain merkle root that can be passed along with a user’s withdrawal payload.

When attempting to withdraw from a privacy pool with a blocklist, the user would have to provide proof that:

- their deposit is valid and unspent (as usual)
- the time delay has passed (e.g. 1 week)
- their deposit is not a member of the blocklist

If a deposit is

a member of a blocklist, the withdrawal can be prevented outright, but there are other options as well:

- allow the user to withdraw, but force a withdrawal to the original deposit address
- same as above, but also charge a “blocklist exit tax”

Blocklists could also be combined with allowlists to also prevent approved depositors who go rogue from abusing the privacy pool.

To pay for the cost of managing the blocklist and querying depositor addresses, a small fee on all deposits/withdrawals would likely be necessary.

In theory, a sufficient powerful blocklist makes an allowlist unnecessary. In practice, using an allowlist to gate deposit access would mean far fewer blocklist queries—it would only be necessary to monitor for allowlist addresses being added to the blocklist.

MVP Implementation Considerations

A privacy pool using only an allowlist is relatively easy to implement. The operator could be a multisig or a DAO, depositors rate limited to 1 ETH per day. If a DAO is managing approvals, revocations could still be delegated to multisigs for expediency. With the goal of trying to keep unwanted addresses out, it may make more sense to have many smaller permissioned privacy pools instead of one really big one, with each pool better able to manage its allowlist.

Implementing a blocklist has additional challenges. Currently the only offchain lists of unwanted addresses are published by Chainalysis and TRM Labs. The lists would need to be aggregated and published in a format friendly for authentication as well as generating the ZKP for both proof-of-exclusion.

Ideally the blocklist publisher would be a separate entity than the privacy pool operator, so there may be a role for an independent community operated blocklist publisher in the future as well. Multiple privacy pools could benefit from pointing to the same blocklist.

Next Steps

This was a quick intro post to frame the problem and the solutions at a high level, but more work is needed to propose and evaluate implementation details.

We have assembled a small Privacy 2.0 R&D working group to discuss implementation - please DM me on twitter to join the discussion.