High level problem

1. Having msg.sender

be a private contract address damages anonymity

1. Allowing msg.sender

to be obscured makes writing public functions unintuitive from the perspective of an Ethereum developer

What we desire

When a user is interacting with a public protocol, they should be able to seamlessly perform the following pattern or obtain anonymity benefits equivalent to this pattern

1. Unshield tokens into a random single-use address
2. Perform a defi interaction via the single-use address
3. Take the proceeds of the defi interaction and shield them into their main address

Possible solution

1. Public function calls have an additional boolean status flag from_shielded_address
2. For public->public calls, from_shielded_address = true
3. For private->public calls, from_shielded_address

value defined by the private function making the call

If from_shielded_address == true

, `msg.sender = -1

Pros:

- Maximum anonymity

Cons:

- Hard to write public functions using this paradigm
- Does not solve the DevEx issue of having to create single-use accounts/addresses for DeFi interactions that require persistent state that is linked to the user (e.g. a collateralised debt position or a share in a liquidity pool)