Built a fun hack during ETH Global London 2024 - thought it would be interesting to share here.

Architecture

The motivation is to leverage SUAVE's trusted execution environment to coordinate threshold secret sharing, where a user's signing key is split into multiple shares and stored in different SUAVE nodes (using SUAVE's confidential compute request). Users can then submit transaction calldata, where the signing key is reconstructed inside SUAVE to sign the transaction calldata. The resulting signed transaction can be submitted to other chains. As such, the user does not need to manually manage multiple keys on different chains.

[

architecture

1172×593 27 KB

](https://collective.flashbots.net/uploads/default/original/2X/4/460d08f7bf1c576a6d48b65b2f14308374485d6e.png)

Future Work

There are a few areas to make it a more comprehensive solution:

1. Threshold signature: currently the private keys are reconstructed before signing the transaction calldata. A better alternative would be for each key share to separately sign the transaction before aggregating the signature.
2. Cross chain call: in the current implementation, we haven't yet implemented the cross chain RPC call. We can just use SUAVE's API to do so.
3. Kettle isolation: we should store the each key share in different kettles, this will increase the security in case of compromise.

Acknowledgements

Thanks to the Flashbots team for explanining and debugging various integrations, along with help on the secret sharing library.

Here's the repository for source code:GitHub - eerkaijun/suave-mpc