Hi There! I wanted to share some idea I have around implementing Account Abstraction using SUAVE, feel free to provide any feedback, especially the things that I might be mistaken about or improvements I can make to this idea, I'm kinda a newbie in this tech.

Hypothesis

It's technically possible to implement account abstraction using TEE with Suave, given that the TEE can custody your private keys and you could potentially build any authentication method on top of that using SUAVE.

Idea

Implement a simplified authentication framework based on oauth 2.0 standards using email and password as credentials.

MVP

Create an session-based authentication using a token

to identify and fetch the user's private key, this token is generated using users credentials, such as email + password and has an expiration time defined in blocks.

[

Untitled

1051×699 27.1 KB

](https://collective.flashbots.net/uploads/default/original/2X/e/ea3b749a42846abd7030cf12e11a5c52e54bf7d6.png)

Minimal implementation

1. Signup: Using email + password, we generate a private key for the user and store it as a Record in the TEE.

a. Suave checks that the email doesn't already exist.

b. Suave returns a token that expires after N blocks.

1. Sign in: Using email + password

a. Suave checks the validity of the credentials.

b. Suave returns a token that expires after N blocks

1. Send operation:

a. Suave checks the expiration of the token.

b. Suave uses the token to fetch the private key

c. Suave signs the transaction using the private key and sends it to the L1 / final chain.

1. Send recover email

a. Suave generates a new temporary password valid for N blocks

b. Suave stores email, temp pass and block validity as a new record in TEE.

c. Suave uses some email provider to send this password to the requested email.

1. Recover:

a. Users send email, temp password and new password

b. Suave checks validity of temp password

c. Suave updates user record in TEE

Example interface

// Suave chain interface Account { // creates a new private key and an associated token id in TEE. // @param password is a

private input signup(string email, string password) returns (string token);

```
// validates email and password in TEE and creates a new token
// id for using the private key
// @param password is a private input
login(string email, string password) returns (string token);

// sends any on chain operation using a token
// uses the Gateway to send operations to the final chain
// @param token is a private input
sendOperation(string token, address target, bytes memory data, uint value) returns (bytes result);

// sends the email for recovering account
// uses the Gateway to send emails off chain.
sendRecoverEmail(string email) returns (bool success);

// recovers the account
// @param tempPass is a private input
// @param newPass is a private input
recover(string email, string tempPass, string newPass) returns (string token);

// Optional. we could also allow users to find the
// public key associated to some email, this could be
// useful for funding services
getPubKey(string email) returns (string pubKey);

}
```

Open questions / challenges

- Implement temporary recover key for recovering account
- Is it possible to use some email provider to send recovery emails?

It is

- Is it possible to use some email provider to send recovery emails?

It is

- Abstract any on-chain concept in the client side, create a UX indistinguishable from any Web 2 UI.
- How Suave chain fees work? Do users need to connect a wallet? → Considering that the actual key is stored in TEE, the SUAVE transcations could be signed by a generic signer provided in the client / frontend, no need for users signatures.
- How to sponsor Suave chain transactions from the client side?
- Is this extendable to other auth providers, like github or google?

- How Suave chain fees work? Do users need to connect a wallet? → Considering that the actual key is stored in TEE, the SUAVE transcations could be signed by a generic signer provided in the client / frontend, no need for users signatures.
- How to sponsor Suave chain transactions from the client side?
- Is this extendable to other auth providers, like github or google?

Please let me know any thoughts

Thanks