


Robust Account

Leveraging Account Abstraction

Hemant p 

February 2023

Submission for StarkCon Grant

Table of Contents

Motivation

Technical Proposal

Challenges

Development Roadmap

Upgrades

Additional Information

Motivation

Motivation

Ram is StarkNet power user. He would like to have

- Finer control over Account Assets aka. Customized Transaction Management
 - 2FA type functionality for NFT Transfers
 - Customized Transaction Throttling ex. Daily limit, Size limit Authorization
 - Authorised Delegated Session management
 - Not worry about MultiSig coordination and intricacies of Guardian selection
- Privacy
 - Participate in activities or have Assets that cannot be tied to his public Profile
 - Not share certain wallet/Asset/holding details with Guardians/MultiSig
- Recovery in case of Adverse event
 - Be alerted and have time to react to an attack on wallet
 - Redress or Salvage Assets to prevent complete loss
 - Use the Rollup's Settlement layer effectively

Potential/Opportunity

Account Abstraction is only limited by our imagination

- Battle Tested Secure Elements in consumer devices → Bring own 2FA functionality → Reuse existing UI/UX primitives → Behaviour composability
- Immutable Device Profiles → Customized Transaction Management → Delegated Session management
- L1 - L2 Messaging bridge → Use Settlement layer → Recovery
- Onboard Multiple Devices → MultiSig/Guardian Alternative → Privacy

Technical Proposal

Components/Terms

- Secure Element Device(SED) ex. Smart Phone, Google Titan Security Key
- Utility Client. Browser or Mobile App, CLI
- **SED** Client. App that talks to Secure Element
- Account Contract. Robust Account Implementation Contract on StarkNet
- Wallet Provider ex. Argent
- User interacting with above

High Level

1. New Account Contract deployed using Wallet Provider
2. Utility Client configures Profile and onboards **SED**
3. Assets migrated to Account Contract
4. User initiates transaction via Utility Client
5. Account Contract parses CallData, decides on further action
6. **SED** Client prompts User for action if required
7. User authorizes **SED** Client to take requisite action

Workflow contd.

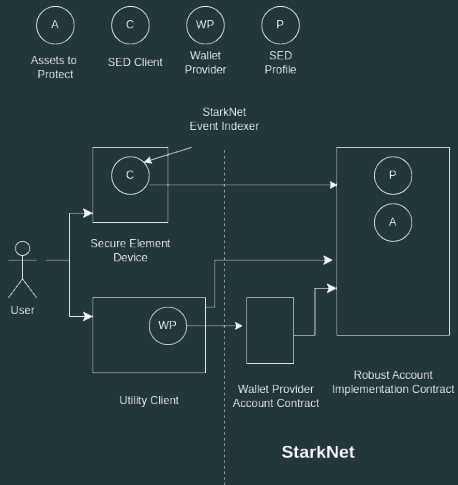


Figure 1: Component Interaction

Secure Element Device(SED) Life-Cycle

- Add/Onboard
- Relinquish
- Replace/Recover

Secure Element Device(SED) Life-Cycle: Add/Onboard

- Utility Client integration with Wallet Provider
- Preferably stand alone and self hosted, though would require Pvt. key export from Wallet Provider
- A **SED** Profile contains settings for Customized Transaction Management
- The Profile is immutable once on-boarded
- Steps
 1. **SED** onboarded using Commit-Reveal Mechanism
 2. **SED** gets white listed against specific Profile
 3. User also Commits to a Secret Note on chain

Secure Element Device(SED) Life-Cycle: Relinquish

Relinquish when **SED** to be changed

- **SED** alone can Relinquish itself
- User can only Revoke a **SED** using Secret Note

Reasoning: Damage Mitigation, Recovery when User Private Key is lost/stolen

- Attacker will first disable throttle aka **SED**
- Attacker needs access to both Wallet Private Key and Secret Note
- Time to Recover Account Contract to new address Via L1 Message
- Stolen **SED** itself useless, unless very powerful actors

Secure Element Device(SED) Life-Cycle: Replace/Recover

Replace/Recover: when **SED** is either dead or lost, updating Profile

1. Revoke using Secret Note
2. Add/Onboard

If Secret Note lost can still slowly drain to new Account Contract
Onboarding Multiple **SEDs** can provide relief

Challenges

Challenges

- NIST P-256 aka. secp256r1 Signature verification in Cairo 1.0 for StarkNet
- Modified Account Smart contract implementation, feature up-gradation and Asset migration
- **SED** Client submitting transaction to StarkNet
- Edge Cases in Life-Cycle Management
- Non Primary use case but listed for completeness
 - StarkNet App integration
 - Existing Wallet Providers don't allow for custom account contracts, if allowed almost seamless
 - Workaround is new Wallet Provider plugin, but is fraught with Security and App Integration

Development Roadmap

Development Roadmap

1. NIST P-256 aka. secp256r1 Signature verification
2. Account Smart contract for **SED** Profile management
3. **SED** Simulation
 - 3.1 CLI scripting
 - 3.2 Mimic 2FA/Simple Transfer approval
 - 3.3 Throttling tests
4. Mobile App as **SED** Client
 - 4.1 Secure element comms
 - 4.2 StarkNet Integration
 - 4.3 Functionality


Upgrades

Upgrades

- Cairo is for writing provable programs. Should be possible to replace 2 Step Commit-Reveal with 1 Step
- Integrate Settlement Layer Recovery functionality
- WebAuthn in Cairo for Session Management
- Seamless Contract Feature up-gradation
- Easy Asset migration to Robust Account
- Onboard Multiple Types of Hardware devices
- Wallet Provider Plugin for App Integration

Additional Information

Additional Information

- Relevant Experience 
 - Multi Domain Technical Hands-on experience
 - Cross Functional Exposure
 - Taken many projects/ideas to fruition
- External Expertise Required
 - Mobile App Development