Hyperledger Ursa integration into Hyperledger Iroha
November, 2019
Hyperledger Ursa integration into Hyperledger Iroha

› Introduction
  › Name: Alexander Matson
  › Location: New York, USA
  › University: The City College of New York
  › Mentor(s): Andrei Lebedev
  › Hyperledger project: Hyperledger Iroha
Hyperledger Ursa integration into Hyperledger Iroha

**Project Description:** Hyperledger Ursa

- Ursa’s goal is to be a shared cryptographic library for the Hyperledger ecosystem
- Implemented in the Rust programming language, exposes a C interface
- Provides APIs for many cryptographic building blocks, including ed25519
Hyperledger Ursa integration into Hyperledger Iroha

**Project Description:** This project adds support for HL Ursa based cryptography to Iroha.

- Iroha is a distributed ledger technology platform, can be used for creating and managing assets, identity, and more.
- Cryptography is a fundamental component for any decentralized ledger
  - Accounts are managed by public/private keypairs
  - Transactions and blocks use cryptographic signatures
- Iroha uses the EdDSAsignature scheme (ed25519 in particular)
- Iroha core software is programmed in C/C++
Hyperledger Ursa integration into Hyperledger Iroha

Project Description: Tools & Frameworks

- Docker
- Cmake
- Gdb
- Valgrind
- Gtest
- Git/GitHub
Hyperledger Ursa integration into Hyperledger Iroha

> **Project Description**: Calling Ursa functions (Rust) from Iroha (C++)
Hyperledger Ursa integration into Hyperledger Iroha

- **Project Description**: Main challenge
  - Ursa integration was not as simple as a drop-in replacement
  - Reason: slight difference in existing Iroha-ed25519 and Ursa-ed25519
  - Iroha needs to support Iroha-crypto and Ursa-crypto for compatibility reasons

- **Solution**: Use Multiformats (used in projects IPFS & libp2p) to encode public keys and signature data. Iroha can read the encoding to determine the appropriate crypto library to use.
Hyperledger Ursa integration into Hyperledger Iroha

Project Description:

Iroha pubkey: bddd58404d1315e0eb27902c5d7c8eb0602c16238f005773df406bc191308929

Ursa pubkey: 60eb82baacbc940e710a40f21f962a3651013b90c23ece31606752f298c38d90
Hyperledger Ursa integration into Hyperledger Iroha

**Project Description:** Encode keys & sigs so Iroha knows which crypto provider to use

---

94da0320bddd58404d1315e0eb27902c5d7c8eb0602c16238f005773df406bc191308929

93da032060eb82baacbc940e710a40f21f962a3651013b90c23ece31606752f298c38d90
Hyperledger Ursa integration into Hyperledger Iroha

**Project Description:** Encode keys & sigs so Iroha knows which crypto provider to use
Project Objectives:

- Obj 1: Integrate Ursa into Iroha’s build process
- Obj 2: Interface with Ursa’s ed25519 functions
- Obj 3: Maintain support for both Iroha-crypto and Ursa-crypto
- Obj 4: Allow users to configure crypto backend with a simple option
Hyperledger Ursa integration into Hyperledger Iroha

Project Deliverables:

- Deliverable 1: Add Ursa to cmake build system (Iroha PR #126)
- Deliverable 2: FFI documentation edits & bug fix (Ursa PR #39,44)
- Deliverable 3: Integrate Ursa as an Iroha crypto provider (Iroha PR #184)
- Deliverable 4: Integrate Multihash library (Iroha PR #263)
- Deliverable 5: Add configuration file option to switch between Iroha and Ursa crypto (In progress)
Hyperledger Ursa integration into Hyperledger Iroha

Project Execution & Accomplishments:

- Deliverables 1 – 4 accomplished
- Most challenging was implementing the multihash encoding.
- Documented best practices for using the Ursa FFI from C/C++ in a memory-safe way. Added & used a missing FFI function that could have led to memory leak bugs without it.
Hyperledger Ursa integration into Hyperledger Iroha

Recommendations for future work:

- Support compiling Ursa through Iroha’s build system in non-Unix environments
- Improve documentation for using Ursa crypto provider
- Support Ursa-compatible ed25519 in Iroha’s client libraries