Hyperledger Ursa integration into Hyperledger Iroha

<table>
<thead>
<tr>
<th>Title</th>
<th>Hyperledger Ursa integration into Hyperledger Iroha</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>PROJECT COMPLETED</td>
</tr>
<tr>
<td>Difficulty</td>
<td>MEDIUM</td>
</tr>
</tbody>
</table>

Description
We have an amazing new project Hyperledger Ursa, that is a secure cryptography library for different projects. As the main idea of Hyperledger is to make projects interoperable it would be great to integrate Ursa cryptography into Iroha or/and maybe include Iroha’s cryptography into Ursa.

Dictionary

Additional Information
https://www.hyperledger.org/blog/2018/12/04/welcome-hyperledger-ursa - introduction to Ursa
https://github.com/hyperledger/iroha/tree/master/shared_model/cryptography - Cryptography in Iroha

Learning Objectives
The intern will get guided by Iroha maintainers and contributors, that are ready to share their extensive knowledge obtained over years of research in cryptography and blockchain
They will be able to create a link between different projects that reflects the global tendency of cooperation and focus on interoperability including the idea of open-source development

Expected Outcome
Ursa library can be used in Iroha or Iroha cryptography could be included into Ursa libraries

Relation to Hyperledger
Hyperledger Iroha, Hyperledger Ursa

Education Level
Any.

Skills
Basic engineering skills, C++ and Rust experience.

Future plans
Continue integrating Ursa into other projects and use it in Iroha

Preferred Hours and Length of Internship
Part-time (full-time is also possible with some additional tasks).

Mentor(s) Names and Contact Info
Andrei Lebedev, andrei@soramitsu.co.jp, Soramitsu

Mentee Name and Contact Info
Alexander Matson, Alex Matson, City College of New York, alex@alexmatson.com

Project Plan

Deliverables
- Integrate Ursa with the Iroha build system
- Interface with Ursa’s ed25519 signing functions
  - Edit docs for Ursa FFI and memory bug fix
  - Wrap Ursa calls in an Iroha crypto provider (expose same interface)
- Integrate Multihash library
- Automatically switch crypto providers depending on public key value

Milestones
- First Quarter: June 3rd, 2019 - 15 Jul 2019
  - Begin researching integration plans
  - Test Ursa’s C interface over FFI
  - Update docs for Ursa’s C interface, remedying memory leak bugs
  - Add Ursa as a cmake module within Iroha’s cmake system, with build commands
- Second Quarter: July 15th, 2019 - 26 Aug 2019
  - Add a CryptoProvider for Ursa
  - Set value of DefaultCryptoAlgorithmType according to the build flag during compilation
  - Update crypto-related test cases to point to DefaultCryptoAlgorithmType
  - Update Ursa C interface with a memory-safe string destructor
  - Research approaches for multihash integration
- Third Quarter: August 26th, 2019 - 07 Oct 2019
  - Select and integrate a multihash library, plus dependencies
  - Attempt one of integrating multihash with Iroha/Ursa keys – didn’t work out
  - Attempt two of integrating multihash with Iroha/Ursa keys – simplified implementation
  - Change the CryptoVerifier to select between CryptoProviders depending on the public key’s multihash encoding
- Fourth Quarter: October 7th, 2019 - 18 Nov 2019
  - Check the loaded keypair is in the ledger
  - First pass at making block signing configurable: add a config file option
  - Second pass simplified the code changes; instead of config file option, automatically sign with the multihash type of the public key
  - Create slides for summary report presentation

Outcomes
1. The ability for Iroha to re-use a common, well-maintained crypto library.
2. Tighter integration in the Hyperledger ecosystem.
Summary Report

Hyperledger Men...son - 2019.pdf