Proof of Achievement

As the first milestone, we tasked ourselves with the following:

- Modules for arithmetic circuit generation for a JSON pattern match can be found in the repo
- Modules for checking signatures on HTTPS responses and circuit generation from those checks can be found in the repo

For context, in the Dapp, we allow users to withdraw funds provided that they submit the data from a JSON file of a particular structure together with the signature from our fintech partner.

The particular data structure of interest (task 1) is implemented in https://github.com/zkFold/p2p-onramp/blob/main/src/ZkFold/P2P/Contract.hs. This module also contains an implementation of the signature verification protocol (task 2). Finally, the module contains an early prototype of our P2P on-ramp smart contract.

These artifacts heavily rely on our arithmetic circuit compiler which is split across many modules. We refer the reader to the folder https://github.com/zkFold/zkfold-base/blob/main/src/ZkFold/Symbolic/Compiler/. The compiler performs arithmetic circuit generation from a high-level code of a smart contract.

Together, these modules give a complete solution to Milestone 1, as it is possible to generate the circuit performing task 1 and task 2 using this code.

For more information we refer the reader to the following pieces of documentation. User documentation for smart contract developers: https://docs.zkfold.io/introduction/what-is-zkfold-symbolic. Developer documentation for contributors to zkFold Symbolic: https://hackage.haskell.org/package/zkfold-base-0.1.0.0/candidate.