Proof of Achievement

Bindings for FFT and multi-scalar multiplication imported from Rust into Haskell.

The relevant bindings are located here:

https://github.com/zkFold/zkfold-base/tree/initial-setup-cargo-cabal-bindings-for-rust/rust-ffi

You can find the E2E tests here:

https://github.com/zkFold/zkfold-base/tree/initial-setup-cargo-cabal-bindings-for-rust/rust-ffi-test

FFT and MSM are the key performance-critical operations in the most widely used proving algorithms, including Plonk. This part of the project ensures that our solution is competitive in terms of the proving speed.

Plonk prover algorithm. Relevant modules:

https://github.com/zkFold/zkfold-base/blob/main/src/ZkFold/Base/Protocol/ARK/Plonk.hs

https://github.com/zkFold/zkfold-base/blob/main/src/ZkFold/Base/Protocol/ARK/Plonk/Internal.hs

A Haskell implementation of the Plonk prover. It is a well-known and a widely used zero knowledge proof protocol. This implementation allows us to implement recursive ZKP protocols in the future which would be of great benefit to DApp developers, especially those who use ZK in the data protection context.