Metamask Integration Delivery Report

During the event, we were able to implement and complete all the milestones described in the proposal with some interesting and unexpected challenges.

We believe that this project can significantly improve the user experience for cross-chain activities such as atomic swaps and bridging by allowing the user to interact with both chains using a single wallet.

Delivers

- Metamask Snap integration module with HD key derivation and transaction signing capabilities exposed through an RPC API which allows for dApp interactions similarly to EIP-12 API (dApp Connector).
- Wallet management UI.

Challenges

- **Prover implementation**: During the implementation, we realized that we couldn't use libraries such as sigma-rust and Sigma.JS due to the restrictions imposed by the Metamak plugin execution environment. To overcome this, we had to implement the Ergo's Schnorr signature scheme in pure JavaScript, which was successfully accomplished and gave us a better understanding of how Schnorr signatures work.
- **Privacy and blocking concerns**: Community discussions about Metamask's questionable privacy and transaction blocking practices have led us to change the integration module to act solely as an offline key deriver and signer module (something closer to what hardware wallets do), thus improving privacy and preventing transaction blocking by avoiding internet connections. This comes with a drawback though, all the work of fetching inputs and broadcasting transactions must be done on the dApp side, to address this we are producing a library to make it as easy to use as the EIP-12 API.

Next Steps

- Improve the user experience on the UI
- Produce a dApp <> Wallet integration library that abstracts away all the complexities of the RPC protocol.