Bounty Proposal: Privacy & ZKP Curated Bounty

Proponent: ZKValidator, 5FsYTPecuFY6HGsqmg5yi9hdxj5TcQp8YNK9x1MYPpyMaDuE
Date: 15.11.2020
Requested allocation: 800 KSM
Proposed curator reward: 10% - 80 KSM
Total allocation: 880 KSM
Short description: We propose 2 Privacy and ZKP related technical bounties

---

Context

We propose two bounties/installments to privacy/zkp related projects. In addition, ZKValidator (ZKV) would like to put itself forward for the role as a curator of this bounty. ZKVs aim is to promote privacy and the use of zero knowledge proof systems on the networks we work on, therefore this bounty aligns very well with our mandate.

This past summer, we kicked off the ZKValidator Polkadot and Kusama Working Group in which community representatives from Parity Technologies and Web3 Foundation participated to tease out concrete steps ZKV could take. You can read a summary of the workshop here. The relevant high level points of how the ZKV can add value to Polkadot/Kusama were:

1. Identify relevant teams that can build zk related substrate pallets.
2. Curate follow up workshops and events to further develop use-cases for zkps
3. Support teams in getting grants or other funding, taking on the role of privacy curator.

This bounty focusses on point 1.

Bounty Request: 880 KSM
General Category: Privacy & ZKPs
Deposit : 44 KSM (5%)
Curator: ZKValidator. *Note: We will be asking a few key advisors to unofficially aid us in the evaluation of the grants. However we are also in the midst of growing a more formal advisory board which for subsequent grants will hold a key to the curator multisig.*

This bounty asks for proposals to two projects, we describe each separately below. If one of the two proposals is rejected in discussion we will remove it and reduce the bounty amount accordingly.

Problem statement:

What is a STARK and how does it compare to other proof systems? A STARK is a Zero-Knowledge Proof (ZKP) system, with the “S” standing for scalable and with the “T” standing for "transparent". STARKs resolve one of the primary weaknesses of SNARKs, its reliance on a "trusted setup". STARKs come with much simpler cryptographic assumptions, avoiding the need for elliptic curves, pairings and the knowledge-of-exponent assumption and instead relying purely on hashes and information theory; this means that they are secure even against attackers with quantum computers. However, there is a tradeoff: the size of a proof goes up from ~300 bytes to a few hundred kilobytes. Depending on the need for trust minimization, proofsize, and one's assumption if elliptic curves will break and emergence of quantum computing, STARKs can be the better choice over SNARKs. See here and here for more detail.

What can you do with a STARK? In general Zero Knowledge proofs can be used for privacy and scaling of blockchain applications and infrastructure. Current examples of how STARKs are used in practice include layer- 2 scaling like DeversiFi using StarkEx and Verifiable Delay Functions.

What’s the status of STARKs on Kusama? Recently, Starks Network started to bring zero-knowledge proof technology to the Polkadot/Kusama ecosystems. It’s goal is to bring proof generation/verification capability to general purpose computations. The Starks Networks node is built on Substrate and it uses the Distaff VM, a zk-STARK virtual machine, for STARK proof generation and verification. In the future, Starks Network wants to become a parachain/parathread in Polkadot/Kusama and serve other chains in the network via cross-chain communications.

What is the problem? There is currently no Substrate pallet which allows embedding STARK verification into a parachain.

Both bounties below have been drafted in collaboration with Xiao Zhang of STARKs Network to serve the immediate needs of this project. ZKValidator seeks to support experimentation with ZKPs and wants to make open source pallets available for the Kusama community as useful building blocks.

Proposal #1 STARK proof verification pallet

Right now Stark Networks has a STARK proof verification module implemented using Distaff VM using Rust standard libraries. This bounty is to rewrite the proof verification module as a WASM pallet in Substrate without using the Rust standard library. As such it can be embedded into parachains much more easily.

Budget: The workload is estimated to be around 3 person-months 480 KSM

Milestones: Detailed technical milestones should be provided to the curator after the design phase.

Proposal #2 Off-chain worker pallet

Design/implement an off-chain worker and a WASM pallet to access the Distaff VM module via the Substrate runtime_interface.

The Starks Network project aims to bring zk-STARK proof generation and verification function to the Polkadot/Kusama network. A user can generate a STARK proof with a user client (currently via command line, later maybe by an app) but the proof will not be stored on-chain as its size is around 100KB-200KB—too large for on-chain storage. This proof will be sent to an off-chain storage pool (implemented as a public data store)—periodically checked and fetched by an off-chain worker. The off-chain worker needs to verify any new proof sent to the proof pool and produce verification results as output.To perform the proof verification work, the off-chain worker (in WASM runtime) needs to access the Distaff VM native runtime module via the Substrate runtime_interface. After the verification result is generated, the off-chain worker can update some on-chain data store (e.g. a smart contract) to record the result.

The entire work flow of proof collection/verification and the use of verification results to update the data stored in an on-chain smart contract (e.g. a simple list/hash table) should be demoed. The smart contract pallet will be added to the chain in its simplest form, such as shown in a Substrate tutorial.

Distaff Vm is an open source zk-STARK VM project. You can find the repo here.

Budget: The workload is estimated to be around 2 person-months 320 KSM

Milestones: Detailed technical milestones should be provided to the curator after the design phase.