TEE Prover

TEE-Prover improves prover diversity with the first implementation of Intel SGX in a multi-prover system. There are many reasons why a multi-prover system, and specifically the use of trusted execution environments - such as Intel SGX - as a secondary prover, is desirable:

- Multi-prover rollups encourages greater resilience and decentralization
- Distributing trust across different proof constructions mitigate systemic vulnerabilities
- Natural intuition for TEE Prover to scale security across Layer 2s with negligible overheads and hardware-grade isolation
-

TEE Prover is developed in collaboration with Scroll. It has successfully validated all blocks on the Scroll Sepolia testnet.

Design of TEE Prover

?

There are two main components to the architecture of the TEE Prover with Scroll:

1. SGX Prover. An off-chain component that checks that the post-state root matches the existing state root after block execution within the secure enclave, and submits the Proof of Execution (PoE) to the SGX Verifier.
2. SGX Verifier. An L1 contract that confirms the correctness of state transition proposed by the SGX Prover. It also verifies the attestation report submitted by the Intel SGX enclave to ensure prover integrity.
3.

On-chain verification of Intel SGX

Remote attestation allows the properties and integrity of the Intel SGX enclave to be programmatically verified. This is a critical process for establishing and ensuring that any computations or data processing it performs are trustworthy.

?

- Using a smart contract as a remote party creates a public and transparent on-chain anchor that enables trust composability. Other smart contracts can also rely on computations carried out within the secure enclave.
-

We have successfully developed a Solidity version of DCAP attestation that allows for the full verification of attestation reports from enclaves to take place on-chain. * The attestation report contains the cryptographic measurement of the execution environment, including hardware, software, and custom data, which is fundamental for: * * Integrity, ensuring that the SGX Prover operates the anticipated software version within a verifiable TEE that is impervious to forgery or alteration, even by the infrastructure operator * * Authenticity, wherein the SGX Prover possesses a keypair securely confined within the TEE. The public key from this pair is embedded in the attestation report, allowing external verification of the message's source through the report's authentication. * * *

Open-source implementation

Visit the code repository for SGX prover and SGX Verifier here . The mono-repository is licensed under the Apache 2.0 agreement:

?

Previous Reproducible Build Next TEE Builder Last updated24 days ago On this page * Design of TEE Prover * On-chain verification of Intel SGX * Open-source implementation

Was this helpful?