Sputnik: Fully Homomorphic Smart Contracts

Compute on encrypted data on the blockchain

Agenda

Overview

Language

Demo

Overview

Private smart contracts are *still* an open problem. Our intention is to eventually solve this problem with Fully Homomorphic Encryption.

Fully Homomorphic Encryption allows a person to compute any arbitrary logical circuit on encrypted data.

Recent concerns have been with performance, but NuCypher has accelerated FHE with the NuFHE library by 100x to drop gate performance to .13ms (7,000 gates/sec).

Language

EXEC

Program entrance is declared by calling `EXEC` and passing your entrance variables

VARIABLES

We have variable creation and control via the contract STATE and the PUSH keyword, e.g.: PUSH my_var STATE

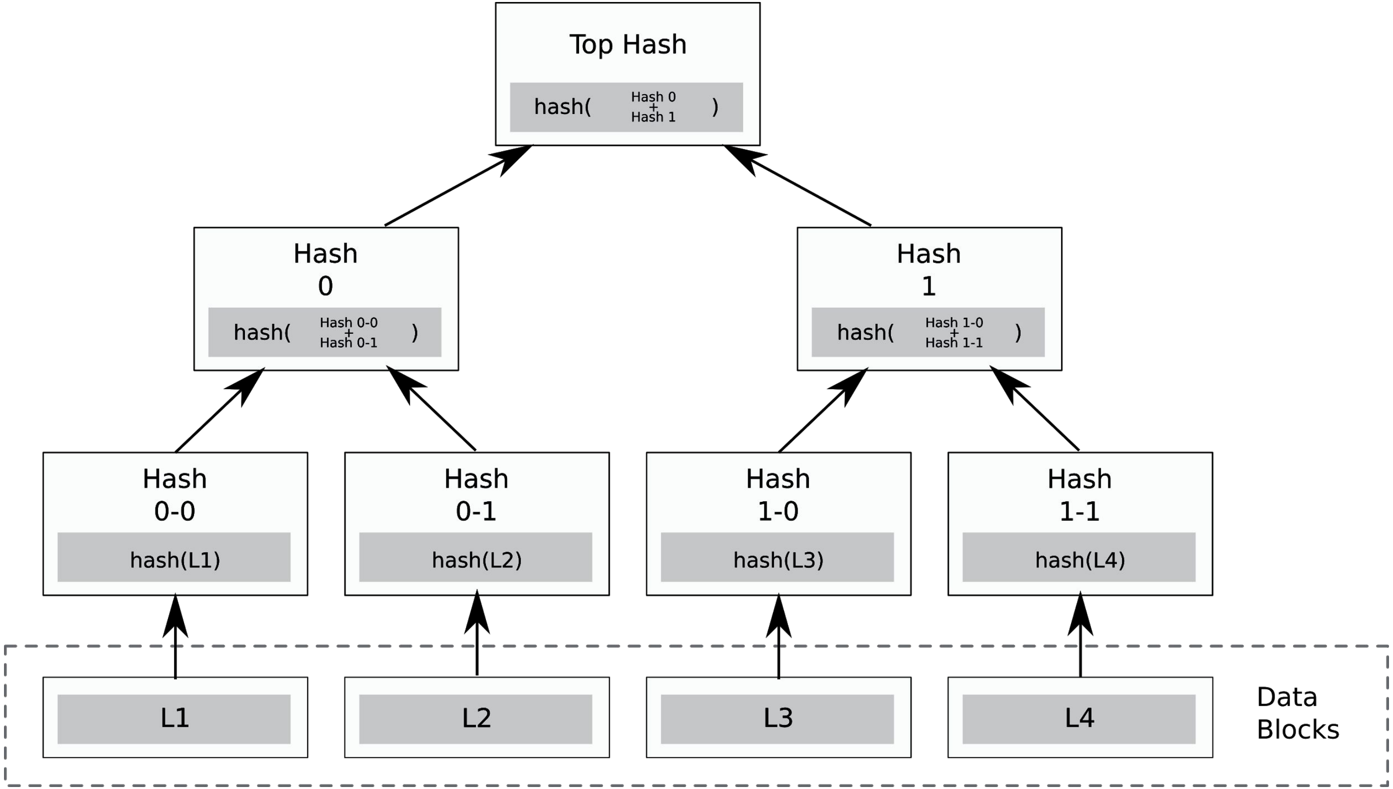
GATES

Because we use Fully Homomorphic Encryption, we are able to perform any arbitrary logic gate, e.g: XOR, AND, NAND, NOT, NOR, XNOR, etc

EXIT

When the EXIT keyword is called, the contract STATE is returned.

Let’s Talk Merkle Trees



DEMO

We will be demoing a fully homomorphic one time pad.

To be clear, this is performing the one time pad while it’s encrypted and the computing party has no knowledge of the data being operated on.

This is the first Fully Homomorphic smart contract.

Questions?