



POSE: Practical Off-chain Smart Contract Execution



Team Members:

Akshat Jain (B20AI054)

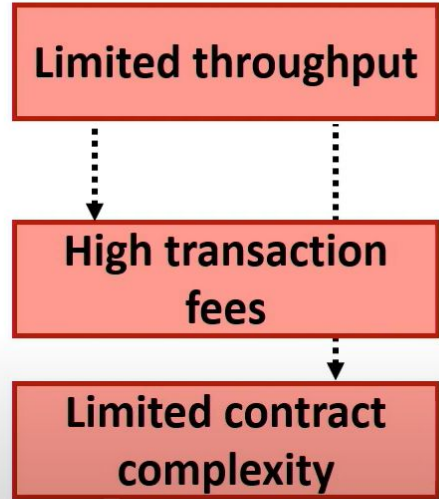
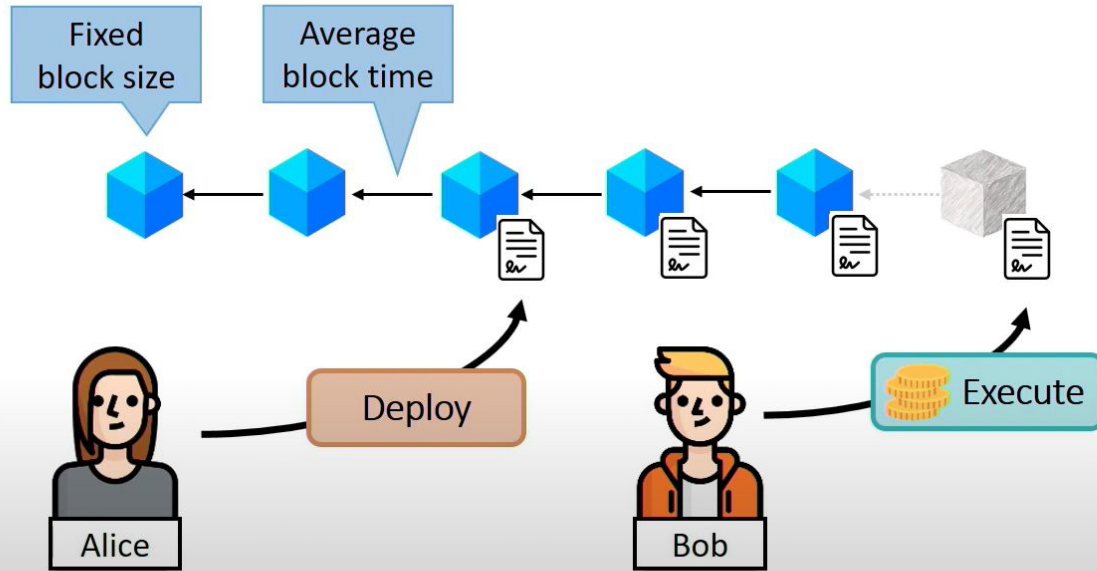
Sakshi Todi (B20EE088)



Introduction

- Smart contracts are programs that execute payments depending on complex logic on blockchains.
- Ethereum is the most popular platform for smart contracts, but it suffers from scalability and cost issues.
- Off-chain execution systems aim to move most transactions off the blockchain, while preserving security and correctness.

Blockchain Scalability Issues



Idea: Off-chain protocols!

Problem Statement

The goal of this paper is to present POSE, a practical off-chain protocol for smart contracts that overcomes the limitations of existing solutions.

All known solutions suffer from at least one of the following:

Locked collateral

Periodic on-chain tx

Fixed participants

Limited lifetime

No private state

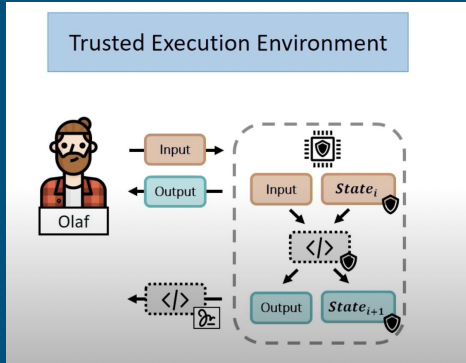
Methodology

- The paper describes the design and implementation of POSE, which involves four roles: users, operators, enclaves, and a manager smart contract.
- Users create and interact with POSE contracts by providing inputs and obtaining outputs.
- Operators own and manage the TEE-enabled systems and contribute computing power to the POSE network by creating enclaves.
- Enclaves are protected execution units that run the POSE program and perform the state transitions of the contracts.
- The manager smart contract manages the registration of enclaves and the creation of POSE contracts.
- The paper also provides a security analysis and an evaluation of POSE.

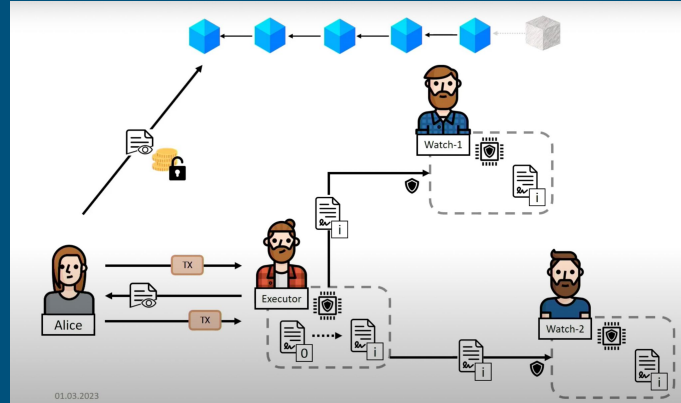
Expected Outcome

- The paper claims that POSE achieves the following properties:
 - Correctness: The contract state is consistent with the contract code and the inputs of the users.
 - State Privacy: The contract state is only known to the authorized parties and the enclaves in the execution pool.
 - Liveness: The contract execution progresses as long as at least one enclave in the execution pool is responsive.
- The paper also claims that POSE is fast and efficient, requiring no blockchain interactions in the optimistic case, and no large collaterals.
- The paper demonstrates the feasibility of POSE by implementing a prototype using ARM TrustZone and evaluating it on practical smart contracts.

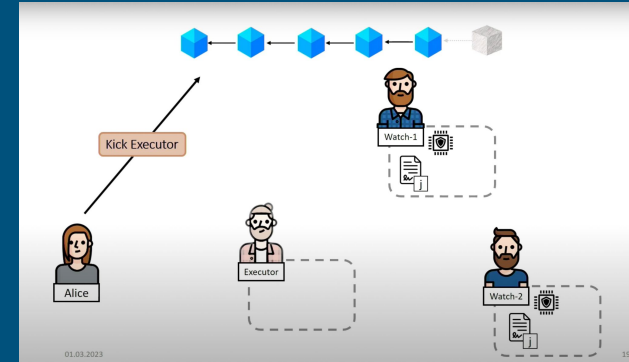
Implementation Proof of Concept



A class 'TEE' having a public and private functions, private function contains the logic and public function calls it as a black-box. (used by operators)



A Defined Pool of Operators, from which an executor, executes contract on TEE as per client instructions.



Client Has Option to Kick the executor

Implementation details

- Used remixd to connect remix with our localhost
- Used hardhat to start a local blockchain environment. This will be the environment on which we deploy our smart contract
- Compiled our code on remix platform and used it as an interface.

Logic used in the Trusted Execution Environment: The client gives a string as the input. If the count of vowels and numbers in the string is greater than the count of consonants, then the balance of the client is increased by 5, else the balance is decreased by 2. If the client enters the string “Change” , the current executor is removed and the next operator in the sequence is made the executor.

New Changes Post mid-evaluation



- The selection of the executors currently is index based from 0 to the last operator, this is now done randomly as proposed in the paper.
- We now come up with reasoning for each and every value and numerical test case.
- We also try the same for multiple test-cases (each a new one from the paper).
- We compare results of our novelty to that in the paper.


DEMO AND IMPLEMENTATION

Table

Condition	Gas Used (Off chain)	Gas Used (On Blockchain)
No conditions	0.00257	0.00274
Palindromic Check	0.00261	0.00310
Count of Vowels and Consonants	0.00259	0.00294

DETAILS DATA

 [Site suggested](#) > 


Gas *(estimated)*  0.00274157
0.00274157 ETH


Very likely in < 15 seconds **Max fee:** 0.00274157 ETH

Total 0.00274157
0.00274157 ETH

Amount + gas fee **Max amount:** 0.00274157 ETH

DETAILS DATA

 [Site suggested](#) > 



Gas *(estimated)*  0.00310151
0.00310151 ETH


Very likely in < 15 seconds **Max fee:** 0.00310151 ETH

Total 0.00310151
0.00310151 ETH

Amount + gas fee **Max amount:** 0.00310151 ETH

DETAILS DATA

 [Site suggested](#) > 

Gas *(estimated)*  0.00294285
0.00294285 ETH

Very likely in < 15 seconds **Max fee:** 0.00294285 ETH

Total 0.00294285
0.00294285 ETH

Amount + gas fee **Max amount:** 0.00294285 ETH

Novelty

- The algorithm in the paper requires all the members to be present on a common network, however we propose a newer pipeline where we remove this requirement while having a lower gas fee.
- A completely new example (not mentioned in the paper), using the algorithm proposed.
- Another which we suggest is using a POW like consensus to choose the executor instead of random selection.
- Finally, rest of the operators should also have a say in the process of choosing the executor.

Comparison with the paper

The paper uses 3 test-cases 1) Poker 2) Stone-Paper Scissors 3) Federated ML and showed their results for each operations.

We observed that our results for similar operations (deposit and withdraw) are much better.

TABLE I. COST OF EXECUTING THE METHODS OF THE *POSE* MANAGER. THE USD COSTS WERE ESTIMATED BASED ON THE PRICES (GAS TO GWEI AND ETH TO USD) ON MAY. 8, 2022 [27], [21]. *FOR COMPARISON, THESE ARE THE COSTS OF POPULAR OPERATIONS ON ETHEREUM.

Method	Cost	
	Gas	USD
registerEnclave	175 910	13.23
initCreation	198 436	14.91
finalizeCreation	79 545	5.98
deposit	37 255	2.80
withdraw	36 997	2.78
challengeExecutor	54 654	4.11
executorResponse	51 478	3.87
executorTimeout	53 327	4.01
challengeWatchdogsCreation	231 286	17.38
challengeWatchdog	131 362	9.87
watchdogResponse	36 257	2.72
watchdogTimeout	52 142	3.92
simple Ether transfer*	21 000	1.58
create CryptoKitty*	250 000	18.78

Table

Value Added	Value Subtracted	Gas Used
3	1	3810
5	2	3810
4000000	10000000	3810

Reference

- Frassetto, T., Jauernig, P., Koisser, D., Kretzler, D., Schlosser, B., Faust, S., & Sadeghi, A. R. (2023). POSE: Practical Off-chain Smart Contract Execution¹³¹⁴. In Network and Distributed System Security (NDSS) Symposium 2023.