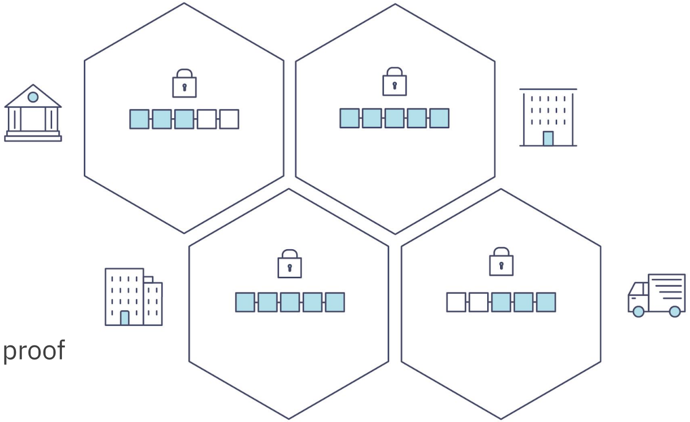
Revisiting Rollbacks on Smart Contracts in TEE-protected Private Blockchains

Systex 2024 workshop

Chen Chang Lew, ETH Zurich
Christof Ferreira Torres, ETH Zurich
Shweta Shinde, ETH Zurich
Marcus Brandenburger, IBM Research
8 July 2024

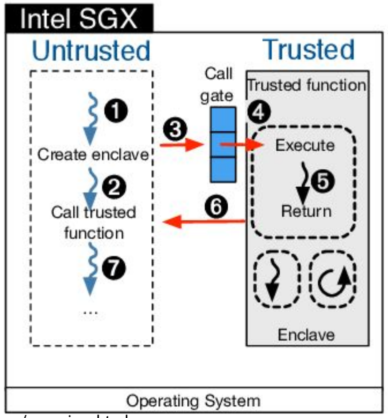
Privacy meets Blockchain

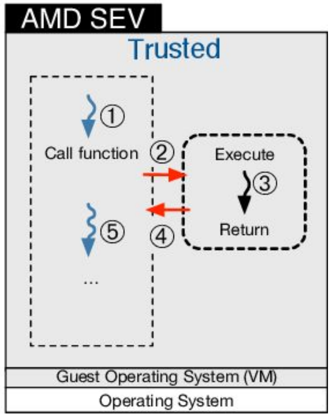
- Shared, immutable ledger
- Recording transactions and tracking assets
- All transactions and data are **visible** and **clear** to participants
- What if data are **sensitive**?
  - Hospital, clinic data
  - Research Institute
  - Company confidential data
- How can we protect data privacy?
  - Modern cryptography
    - Homomorphic encryption, multi-party computation, zero-knowledge proof
  - Hardware-based Trusted execution
    - Trusted Execution Environment (TEE)



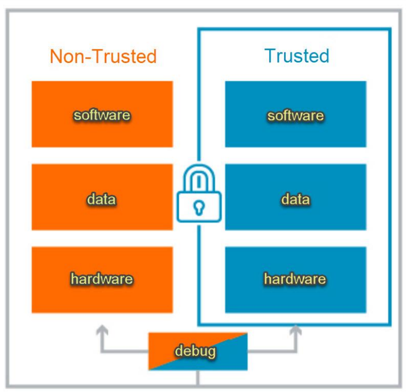
Introduction ~ Trusted Execution Environment (TEE)

- Hardware-aided Isolation.
- Which protects the code and data from unauthorized access or modifications
  - Data confidentiality
  - Execution integrity
- Protected even against a malicious high privileged software (OS)
- Remote Attestation
- Example:
  - Intel SGX[1]
  - AMD SEV[2]
  - ARM TrustZone[3]



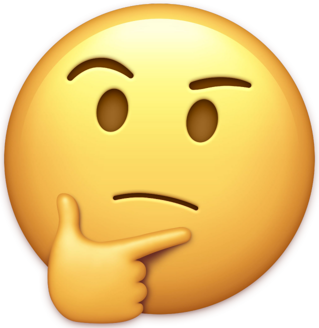


arm



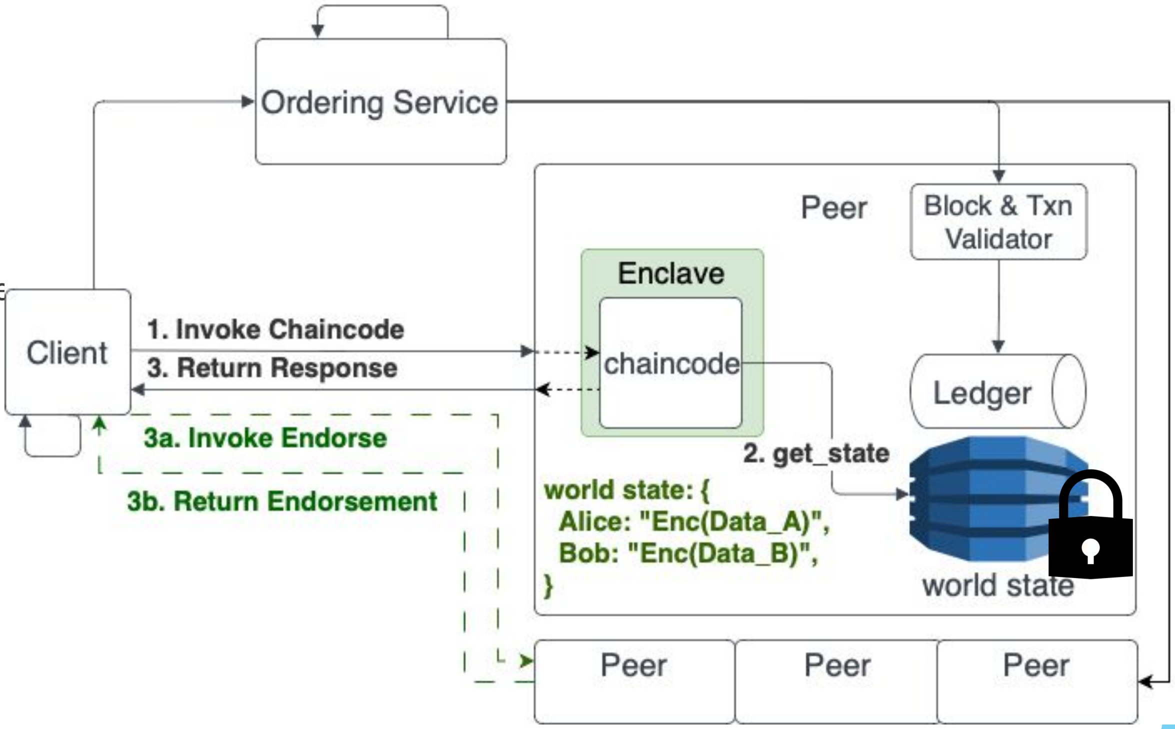
[1]https://www.intel.com/content/www/us/en/developer/tools/software-guard-extensions/overview.html
[2]https://www.amd.com/en/developer/sev.html
[3]https://www.arm.com/technologies/trustzone-for-cortex-m#:~:text=Arm%20TrustZone%20technology%20is%20used,Learn%20More

Q: Does applying TEE solves the privacy problem of blockchain?



Fabric Private Chaincode

- Hyperledger Fabric[1]
  - An open-source permissioned blockchain framework
  - support for smart contracts in the form of chaincode
- Fabric Private Chaincode (FPC)[2]
  - An extension that enables the execution of smart contracts in a secure enclave provided by TEEs



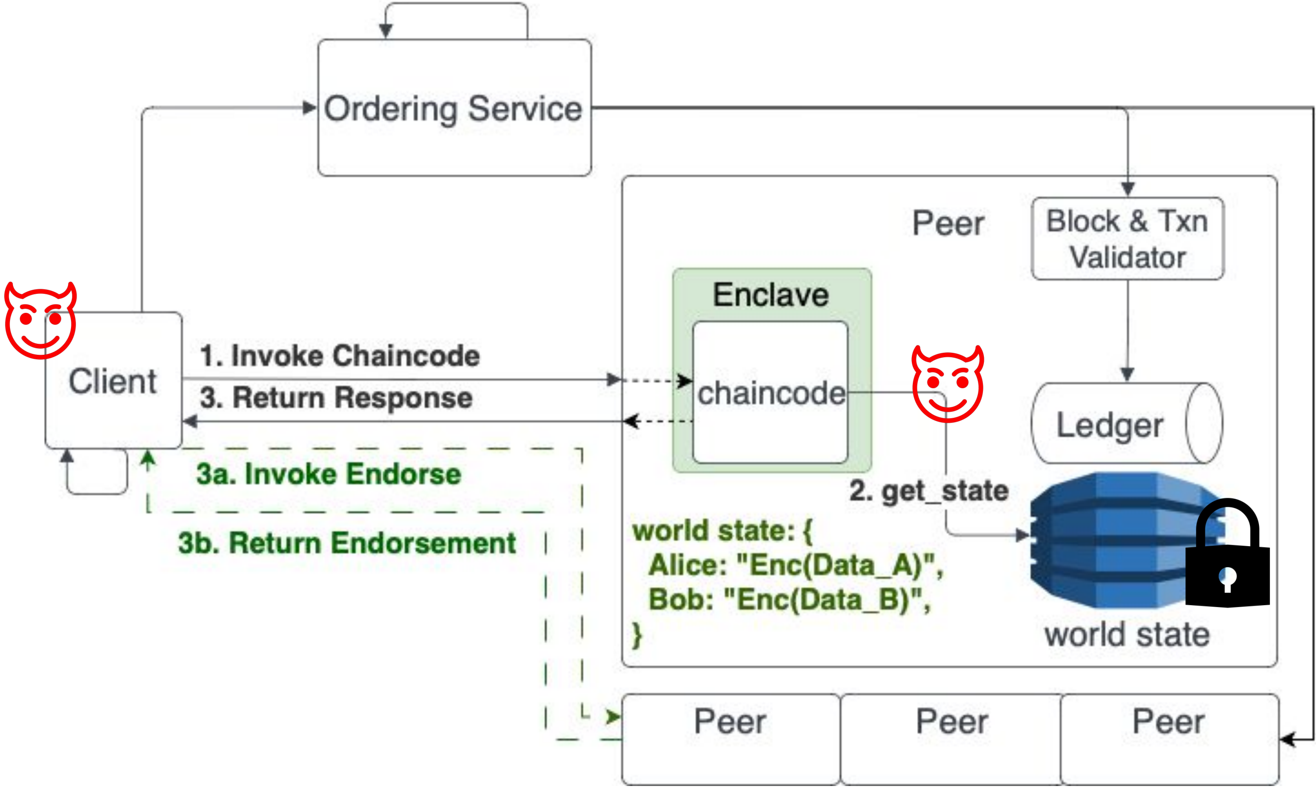
[1]https://www.hyperledger.org/use/fabric
[2]https://github.com/hyperledger/fabric-private-chaincode

Wait … maybe there is a problem …

Rollback Attack

- Malicious peer can give back the old version of the encrypted data.

- And this may break the confidentiality of the application.



Contributions

- Feasibility Analysis
- Solution Prototyping and Implementation
- Experimental Evaluation

Application: Secret Keeper Smart Contract

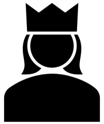
Function:
- Add User
- Remove User
- Lock New Secret
- Reveal Secret

Enclave

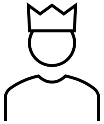
Worldstate

Authlist: [Alice]

Secret: Null

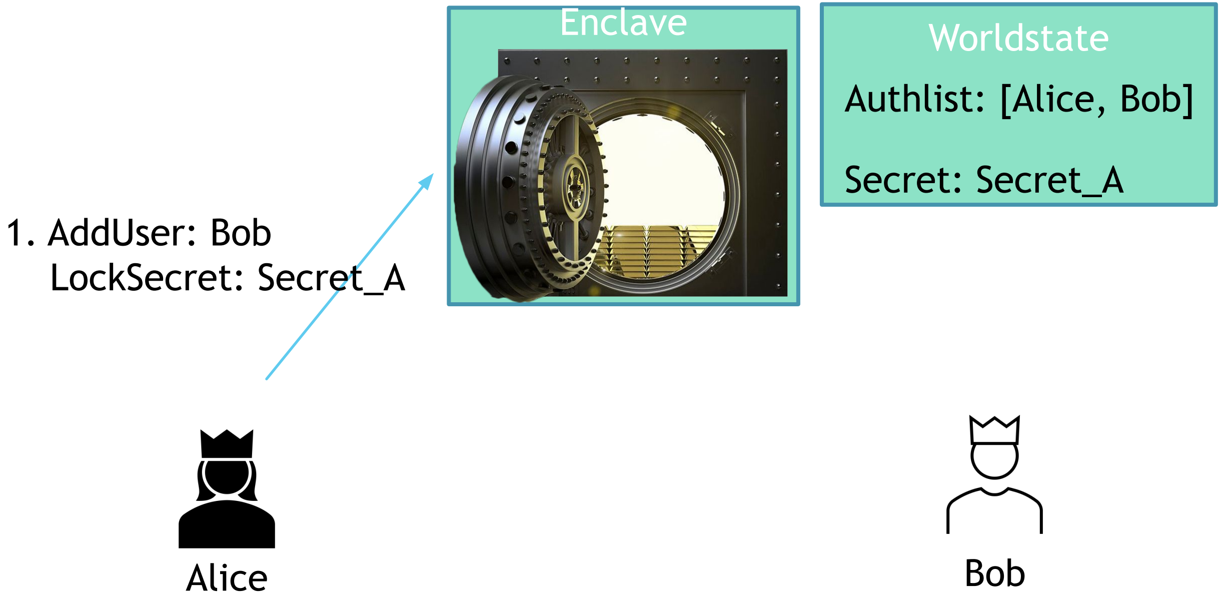


Alice

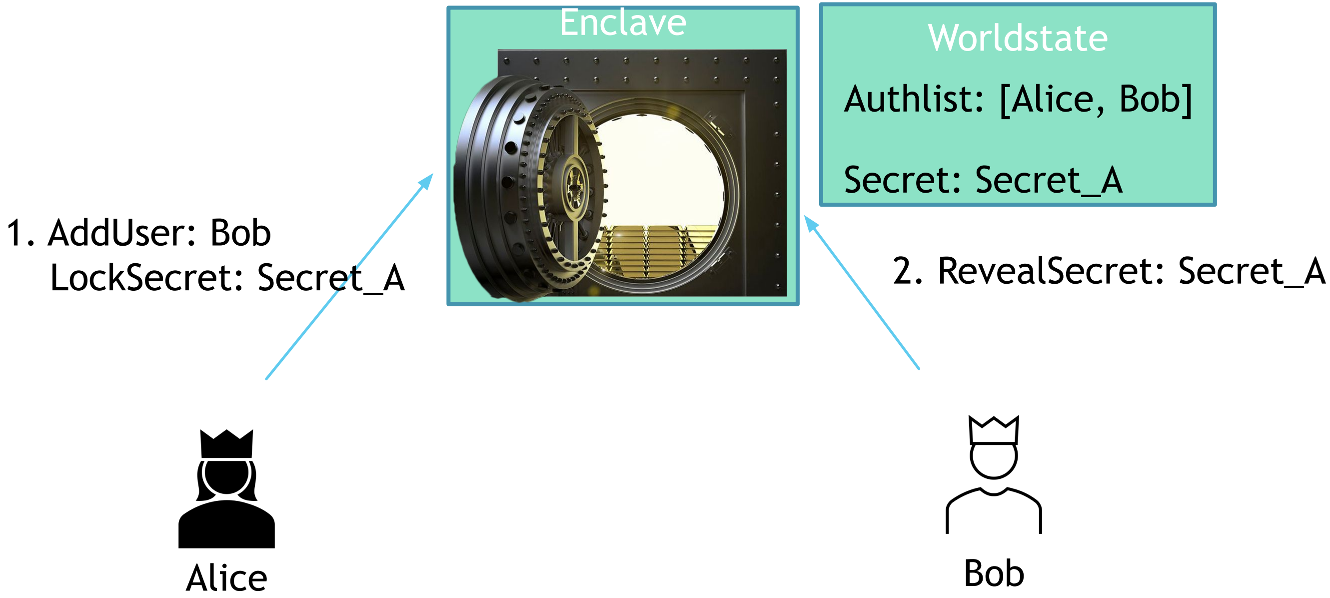


Bob

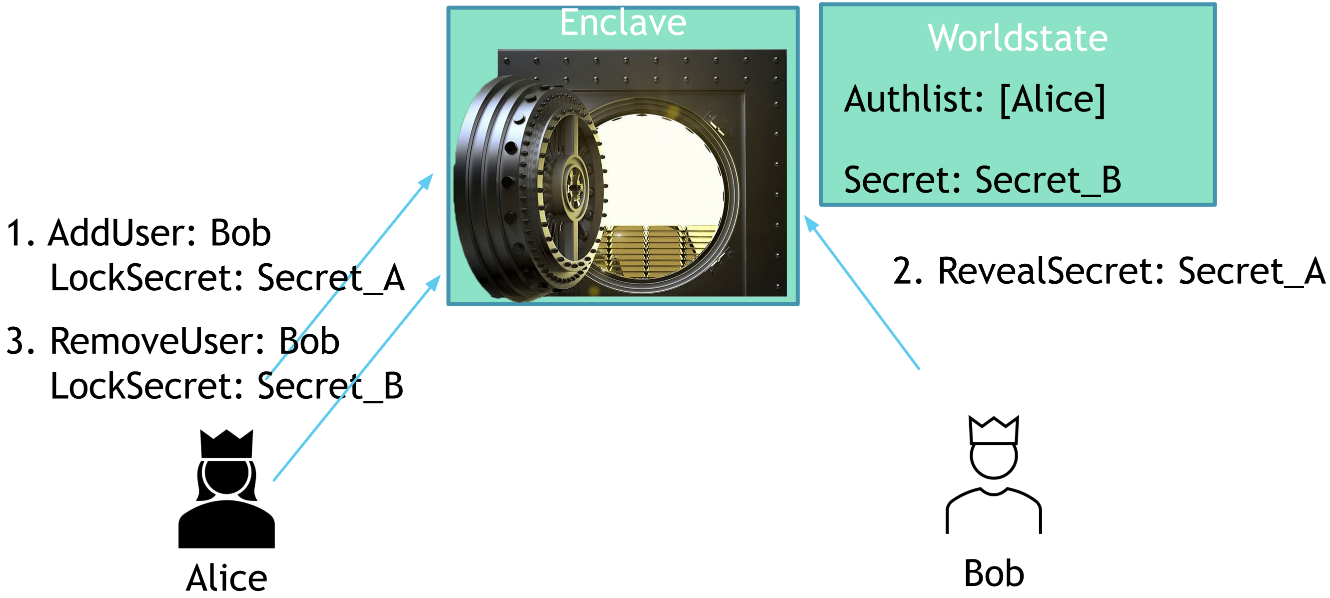
Application: Secret Keeper Smart Contract



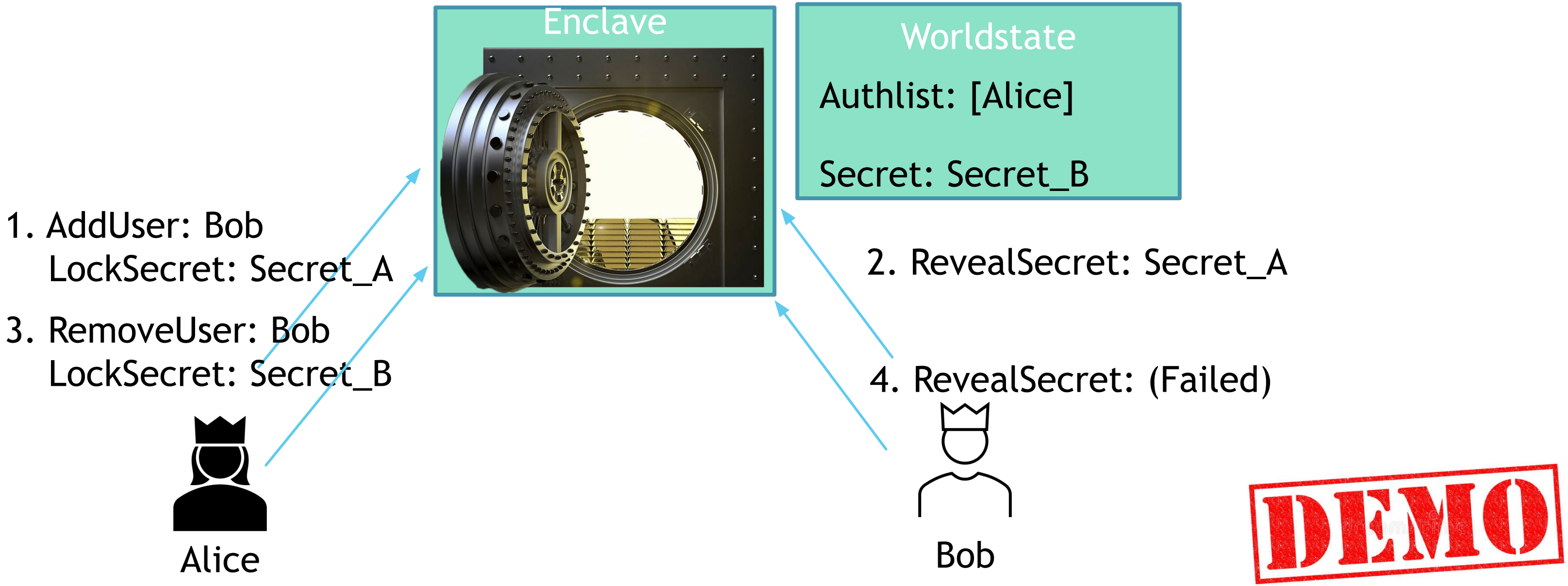
Application: Secret Keeper Smart Contract



Application: Secret Keeper Smart Contract



Application: Secret Keeper Smart Contract



Demo: https://www.loom.com/share/e3dca62f8df849229e2c6414fd374289?sid=473acda1-92ac-4ad6-89c4-ddb2f5786dd5

Rollback Attack! Secret Keeper

Action		AuthList	Secret
• Alice (Lock)	▫ Secret A	• [Alice, Bob]	Secret A

Rollback Attack! Secret Keeper

Action		AuthList	Secret
• Alice (Lock)	¤ Secret A	• [Alice, Bob]	Secret A
• Bob (Reveal)	¤ (Success)	• [Alice, Bob]	Secret A

Rollback Attack! Secret Keeper

Action		AuthList	Secret
• Alice (Lock)	◻ Secret A	• [Alice, Bob]	Secret A
• Bob (Reveal)	◻ (Success)	• [Alice, Bob]	Secret A
• Alice (Remove)	◻ Bob	• [Alice]	Secret A

Rollback Attack! Secret Keeper

Action		AuthList	Secret
• Alice (Lock)	◻ Secret A	• [Alice, Bob]	Secret A
• Bob (Reveal)	◻ (Success)	• [Alice, Bob]	Secret A
• Alice (Remove)	◻ Bob	• [Alice]	Secret A
• Alice (Lock)	◻ Secret B	• [Alice]	Secret B

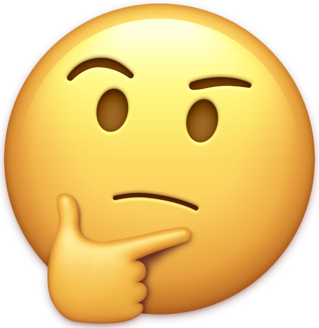
Rollback Attack! Secret Keeper

Action		AuthList	Secret
Alice (Lock)	Secret A	[Alice, Bob]	Secret A
Bob (Reveal)	(Success)	[Alice, Bob]	Secret A
Alice (Remove)	Bob	[Alice]	Secret A
Alice (Lock)	Secret B	[Alice]	Secret B
Bob (Reveal)	(Success)	[Alice, Bob]	Secret B

DEMO

Demo: https://www.loom.com/share/e540bf6395f94ab8ba547bd43942d063?sid=b32de7ea-e5d7-43ef-9e06-bf11b3cfc6ff

Q: How can we overcome this issue?



Related Work & Analysis

- Solution for rollback attack on TEE
  - Monotonic Counter[1]
  - ROTE[2]
  - Enclave DB[3]

Not Suitable In FPC

[1] https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjb8I3dr8D_AhVe_7sIHTj3BhgQFnoECAcQAQ&url=https%3A%2F%2Fcdrdv2-public.intel.com%2F671564%2Fintel-sgx-platform-services.pdf&usg=AOvVaw3Cbr31cwfEXIgDGYZoXsgr

[2] Sinisa Matetic et al. “ROTE: Rollback Protection for Trusted Execution”. In: Pro- ceedings of the 26th USENIX Conference on Security Symposium. SEC’17. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/matetic

[3] C. Priebe, K. Vaswani, and M. Costa, “EnclaveDB: A Secure Database Using SGX,” in Proc. IEEE Symposium on Security and Privacy (SP), 2018. https://ieeexplore.ieee.org/document/8418608

Strawman approach: Single Key Value Storage (SKVS)

- We only store a single key in KVS
- Advantages:
  - Naive to Implement.
- Disadvantages:
  - Performance drop as the application state gets larger
  - Concurrent transactions result in conflicts (which may impact performance badly)

Instead of doing this,

Data in worldstate:

```
{
    “Alice”: “Enc Alice’s data”,
    “Bob”: “Enc Bob’s data”,
}
```

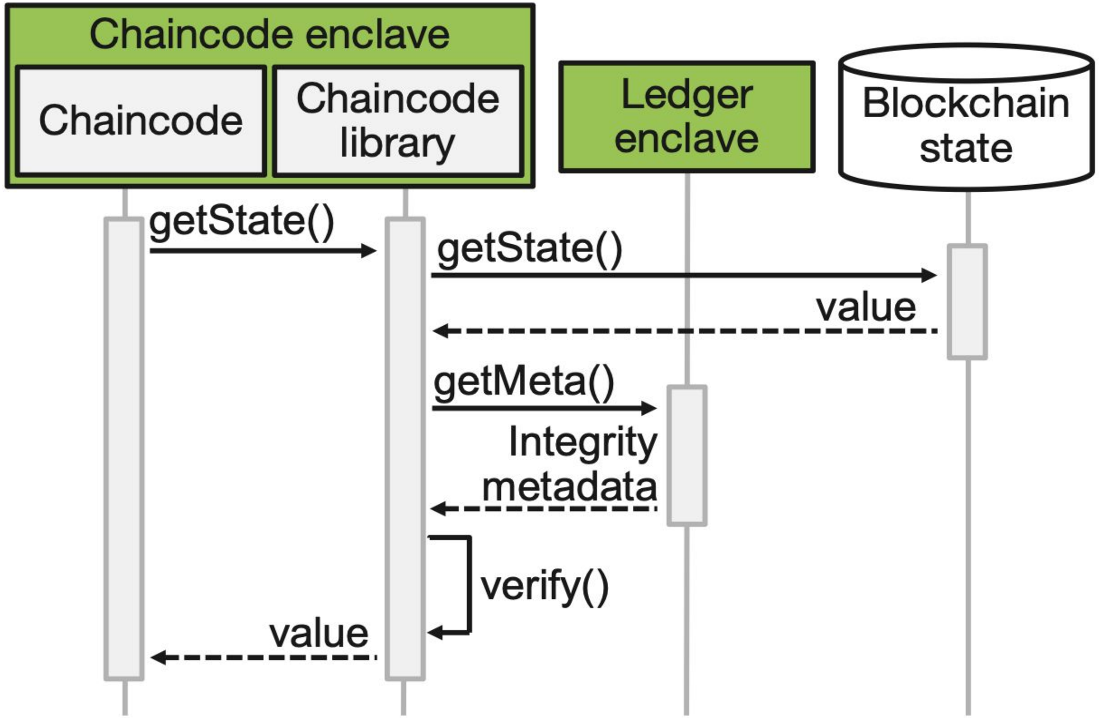
We do this,

Data in worldstate:

```
{
    “Data”: {
        “Encrypted of both alice & bob objects”
    }
}
```

Trusted Ledger Enclave (TLE)[1]

- Having an extra enclave to track the ledger and store the **integrity metadata** of the KVS.
- Advantages:
  - O(1) verification on KVS
- Disadvantages:
  - Original FPC's v1.0 RPC didn't implement due to maintenance difficulties.



[1]M. Brandenburger, C. Cachin, R. Kapitza and A. Sorniotti, "Trusted Computing Meets Blockchain: Rollback Attacks and a Solution for Hyperledger Fabric," *2019 38th Symposium on Reliable Distributed Systems (SRDS)*, Lyon, France, 2019, pp. 324-32409, doi: 10.1109/SRDS47363.2019.00045.
[2]https://www.edgeless.systems/products/ego/

Our Approach

Merkle Tree Approach (MTA) Architecture

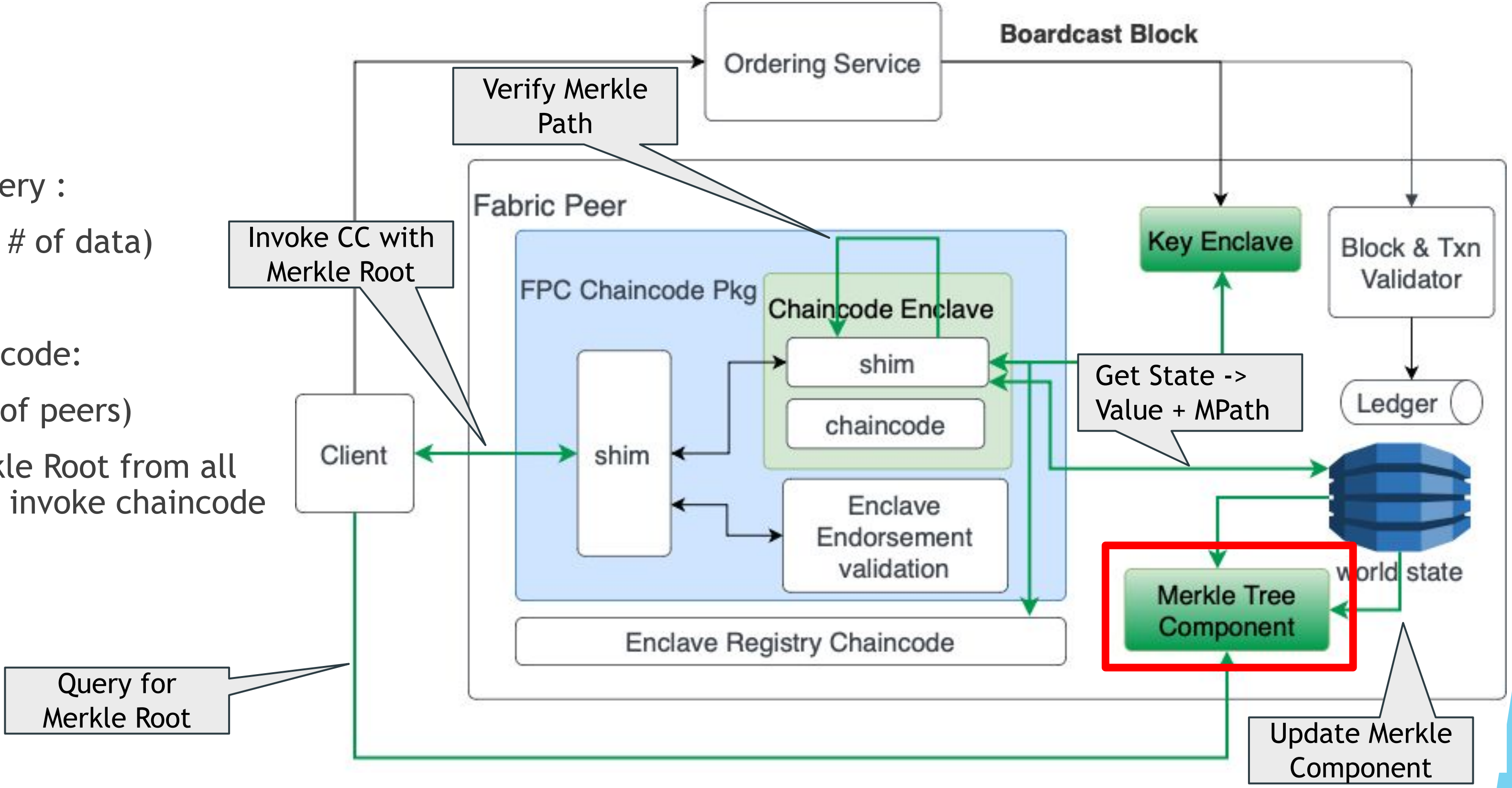
Update & query :

O(log n) (n = # of data)

Invoke Chaincode:

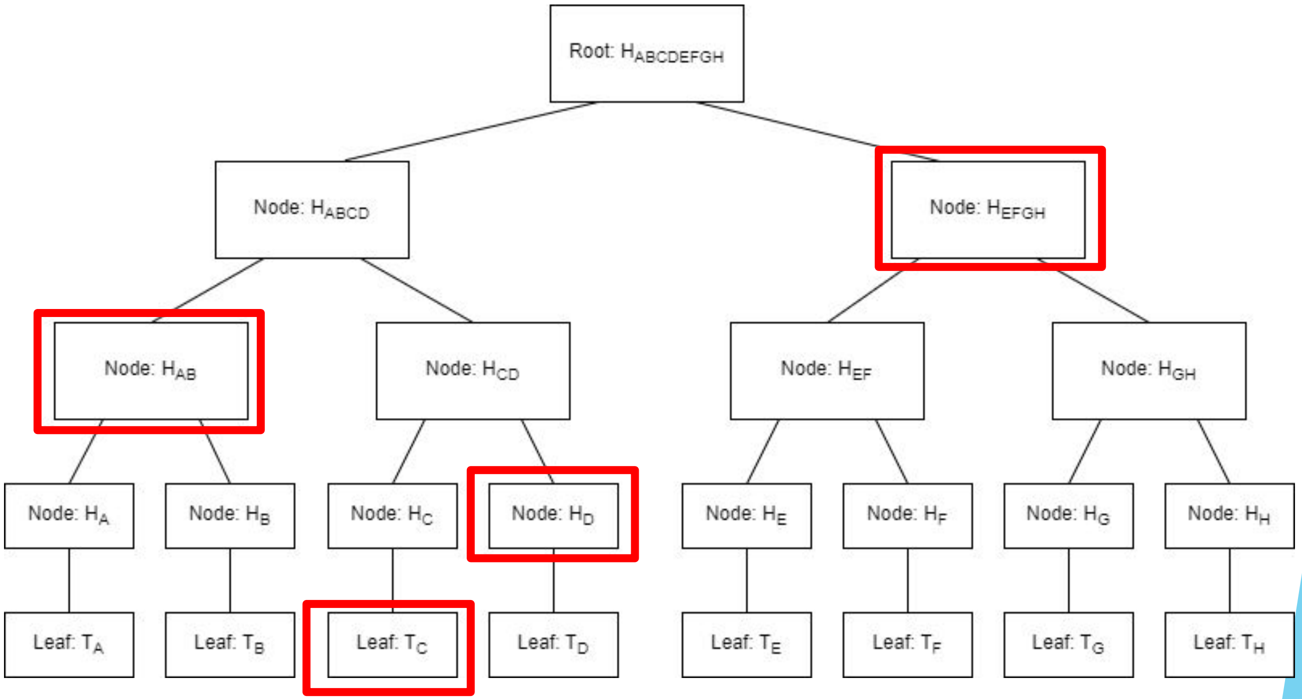
O(m) (m = # of peers)

Require Merkle Root from all peers before invoke chaincode



How we maintain the merkle tree

- Each Leaf contain Each Key Value Pair from wordstate.
- During Get State, Return Leaf + related Merkle Path



MTA Execution Flow

Addition Step:

(0. Query Merkle Root)

(1a. Agree Merkle Root)

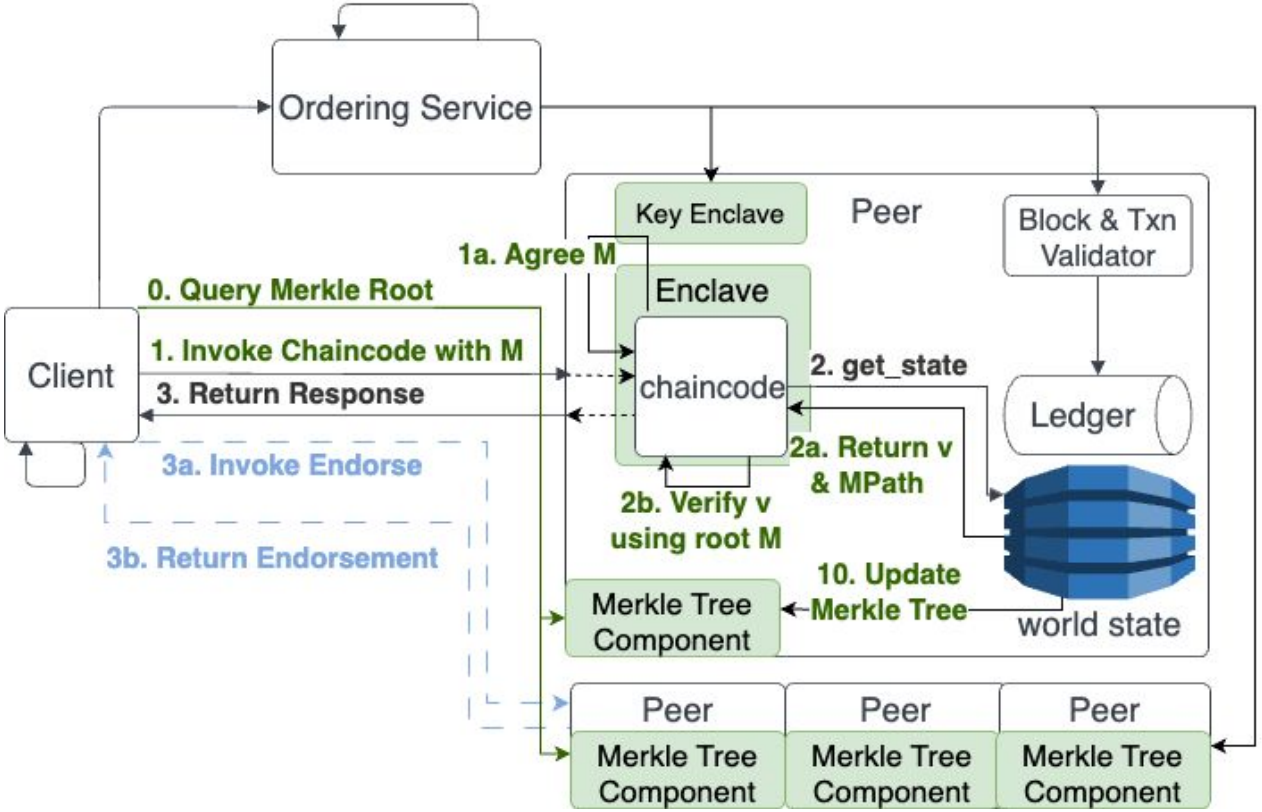
(2b. Verify value)

(10. Update Merkle Tree)

Modify Step:

(1. Invoke CC with M)

(2a. Return Value & MPath)



Evaluation

- Trusting Computing Base (TCB) Size
- Performance Impact
- Security Analysis

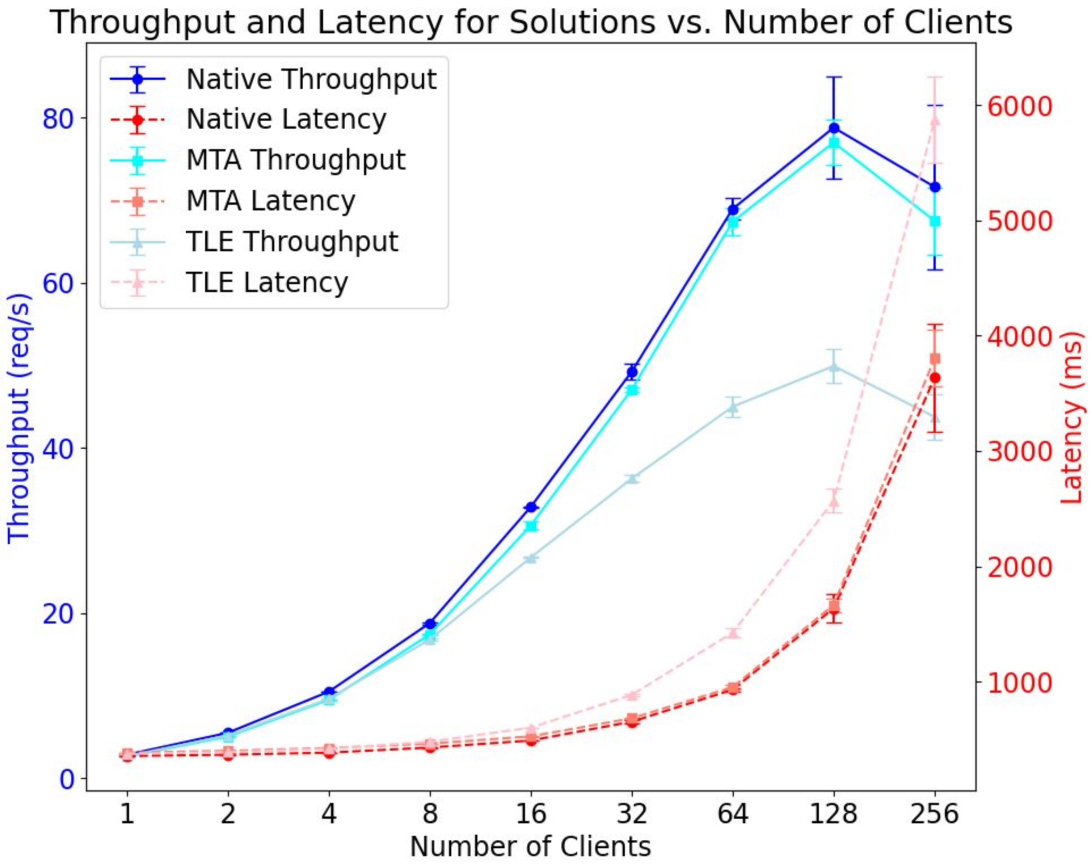
Trusting Computing Base (TCB)

	Own Enclave component (inside TCB)	Enclave STUB (Inside TCB)	Client SDK (Outside TCB)	Fabric Peer (Outside TCB)
SKVS	0	141	0	0
TLE	6700+	162	0	0
Merkle	0	534	105	2000+

Shows how many lines of code (LoCs) **added** for each solution

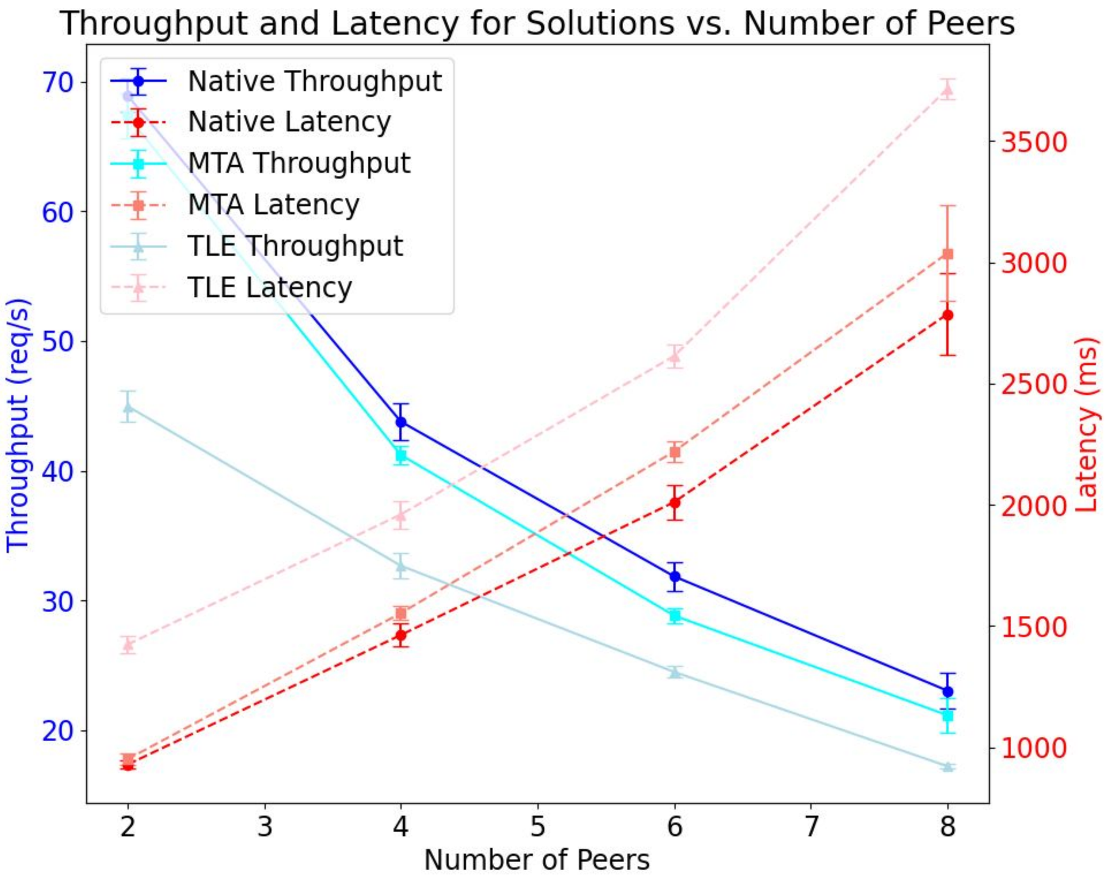
Test on # of clients.
(Expectation: Client++, Throughput++)

- All Solution reach saturation at 128 client
- MTA perform just as good as Native FPC
- TLE perform decrease as faster as the # of client increase



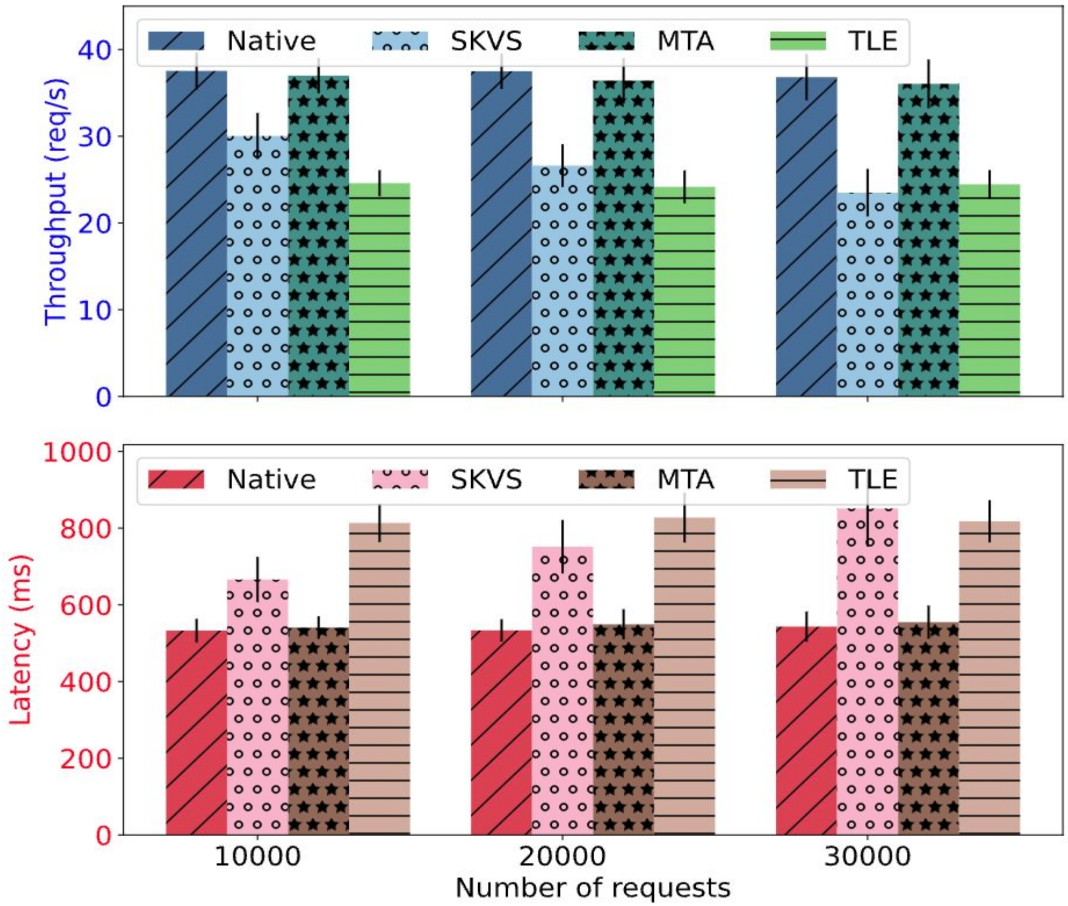
Test on # of peers.
(Expectation: Peers++, Latency++, MTA will become worst)

- Native: linear latency increase
- MTA: superlinear latency increase. (because we query each peer 1-by-1)
- TLE: Still perform the worst



Test on Ext Secret Keeper

- MTA: perform as good as Native
- TLE: has the worst performance
- SKVS: performance drop as txn number increase
  - Read & Write the whole data for each action.
  - More Data -> More Time to decrypt & encrypt



Full Article[1]

Revisiting Rollbacks on Smart Contracts in TEE-protected Private Blockchains

Chen Chang Lew*
ETH Zürich
lewchenchang@gmail.com

Christof Ferreira Torres
ETH Zürich
christof.torres@inf.ethz.ch

Shweta Shinde
ETH Zürich
shweta.shinde@inf.ethz.ch

Marcus Brandenburger
IBM Research
bur@zurich.ibm.com

***Abstract*—Blockchain technology offers decentralized security but fails to ensure data confidentiality due to its inherent data replication across all network nodes. To address these confidentiality challenges, integrating blockchains with Trusted Execution Environments (TEEs), such as Intel SGX, offers a viable solution. This approach, by encrypting all data outside the SGX enclave and making them unrecognizable to untrusted network nodes, ensures secure processing of data and computations within TEEs. Fabric Private Chaincode (FPC), an enhancement of Hyperledger Fabric, demonstrates this integration by securing smart contracts in enclaves, thereby enhancing confidentiality. However, FPC's reliance on states stored on the blockchain introduces vulnerabilities, especially to rollback attacks. This work provides a detailed analysis of rollback attacks in FPC, evaluates existing protection mechanisms, and proposes a solution: a Merkle Tree approach implemented in an FPC application named Secret Keeper. Through experimental validation, this solution shows significant security enhancements against rollback attacks within FPC contexts.**

This paper delves into potential solutions within the FPC framework. The original FPC documentation suggests a strawman approach of consolidating all values under a singular state, a method that proves secure but inefficient and unscalable. It also discusses a Trusted Ledger Enclave solution, which was excluded from the FPC RFC [5] due to high maintenance costs and suboptimal performance. We propose a Merkle tree-based solution that retains up to 95% of FPC's original throughput without rollback protection, demonstrating a minor compromise in efficiency for significantly improved security.

Our contributions are multifaceted, extending from theoretical exploration to practical application:

- **Feasibility Analysis:** We assess the practicality of existing rollback protection mechanisms from literature in the context of the FPC, considering their efficiency and effectiveness in Section 2.6.
- **Solution Prototyping and Implementation:** We implement the Single Key-Value Storage and Trusted Ledger Enclave solutions as described in

[1] https://github.com/chenchanglew/systex2024/blob/2bd229b4d4c4e6be9648c411aa8e4d744eff3524/systex24-final56.pdf

Lessons Learned

Solutions	Advantages	Disadvantages	Comment
SKVS	- Small TCB (141 LoC) - Easy to implemented - no Fabric Peer's Modification	- Bad performance in concurrent transactions - Always load all data in the enclave	- Good for application dependent on High Read Operation.
TLE	- Transparent for the clients. - In theory good performance (O(1) for retrieve & update)	- Mediocre Performance (peak 65% of Native FPC) - Need to edit Fabric Peer's code - Relatively large TCB (6000+, plus fabric peer code)	- Good for integrate with old system. (Just require to update the chaincode)
MTA	- Small TCB (500+ LoC) - Great Performance (95% of Native FPC)	- Need to edit Fabric Peer's code - Need to edit Client side SDK - Cannot integrate with old system.	- Good for new application that require rollback protection.

Thanks!

Artifact links (feel free to play around and break it XD)

Reference Slides

- Intel SGX: https://www.intel.com/content/www/us/en/developer/tools/software-guard-extensions/overview.html
- AMD SEV: https://www.amd.com/en/developer/sev.html
- ARM Trustzone: https://www.arm.com/technologies/trustzone-for-cortex-m#:~:text=Arm%20TrustZone%20technology%20is%20used,Learn%20More
- Hyperledger Fabric: https://www.hyperledger.org/use/fabric
- Fabric Private Chaincode: https://github.com/hyperledger/fabric-private-chaincode
- SGX Monotonic counters: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjb8I3dr8D_AhVe_7sIHTj3BhgQFnoECAcQAQ&url=https%3A%2F%2Fcdrdv2-public.intel.com%2F671564%2Fintel-sgx-platform-services.pdf&usg=AOvVaw3Cbr31cwfEXIgDGYZoXsgr
- ROTE: Sinisa Matetic et al. "ROTE: Rollback Protection for Trusted Execution". In: Pro- ceedings of the 26th USENIX Conference on Security Symposium. SEC'17. Vancouver, BC, Canada: USENIX Association, 2017, pp. 1289–1306. isbn: 9781931971409. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/matetic
- Trusted Ledger Enclave (TLE): M. Brandenburger, C. Cachin, R. Kapitza and A. Sorniotti, "Trusted Computing Meets Blockchain: Rollback Attacks and a Solution for Hyperledger Fabric," *2019 38th Symposium on Reliable Distributed Systems (SRDS)*, Lyon, France, 2019, pp. 324-32409, doi: 10.1109/SRDS47363.2019.00045. https://ieeexplore.ieee.org/document/9049585
- Formal Verification: [1]Saharsh Agrawal and Karen Tu. "Enabling Verifiable Execution of Distributed Secure Enclave Platforms". In Berkeley EECS-2021-153, https://www2.eecs.berkeley.edu/Pubs/TechRpts/2021/EECS-2021-153.html
- Merkle Tree: https://en.wikipedia.org/wiki/Merkle_tree