Student: Rohan Deshamudre

Responsible Professor: Kaitai Liang

Merging smart contracts with trusted

1. Introduction:

Hyperledger fabric: private and permissioned distributed ledger technology platform that supports creation of smart contracts.

Smart contracts: programs or transaction protocols that automatically execute when pre-set conditions are met. In default state cannot maintain confidentiality and are susceptible to attacks.

Intel SGX: provides a secure computing base that guarantees data confidentiality, integrity and

2. Research Question:

" How can an e-auctions smart contract be executed within Intel SGX trusted execution environment to enhance its security?"

3. Method:

- **Literature Study**
  - Hyerpledger fabric smart contracts and applications
  - Smart contract analysis & vulnerabilities
  - Compare different trusted hardwares and research e-auction systems
- **Create smart contract**
  - Use test-network and write sample application
  - Write e-auction contract and application in Go and JS
  - Explore encryption and chaincode execution in Intel SGX
- **Enhance security**
  - Write chaincode for secure transactions and remote attestation within Intel SGX
  - Implement attestation and bid encryption
  - Test the contract and evaluate security enhancements

References: [1] Marcus brandenburger, Christian Cachin. Blockchain and Trusted Computing: Problems, Pitfalls, and a Solution for Hyperledger Fabric, 2018

[2] Hyperledger, Hyperledger fabric docs, A Blockchain Platform for the Enterprise, 2020

[3] Chunxiao Mu Dan Wang, Jindong Zhao. Research on blockchain-based e-bidding system, 2021.

[4] Hyperledger. Hyperledger fabric introduction, 2020.

4. System Design

Figure 1: Architecture of the prototype

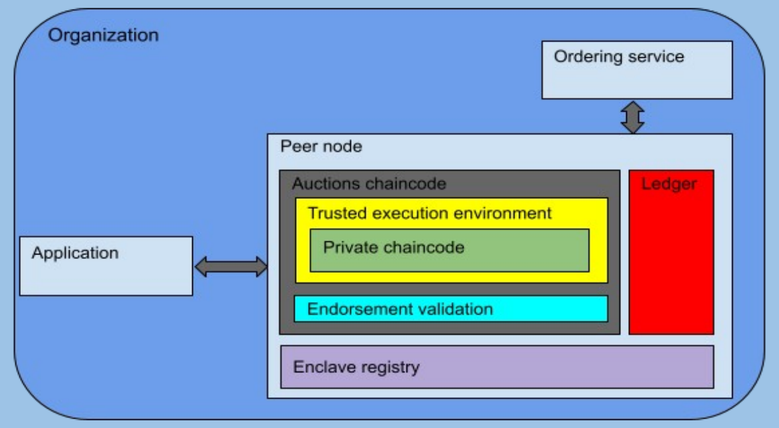


Figure 2: Structure of an organisation in the auction

5. Implementation

```
type BlindAuction struct {
    EventType       string                `json:"eventType"`
    SellingItem     string                `json:"sellingItem"`
    Seller          string                `json:"seller"`
    Price           int                   `json:"price"`
    Organisations   []string              `json:"organisations"`
    HiddenBids      map[string]HiddenBid  `json:"hiddenBids"`
    RevealedBids    map[string]RevealedBids `json:"revealedBid"`
    WinningBidder   string                `json:"winningBidder"`
    AuctionState    string                `json:"auctionState"`
}
```

Components of a Blind Auction

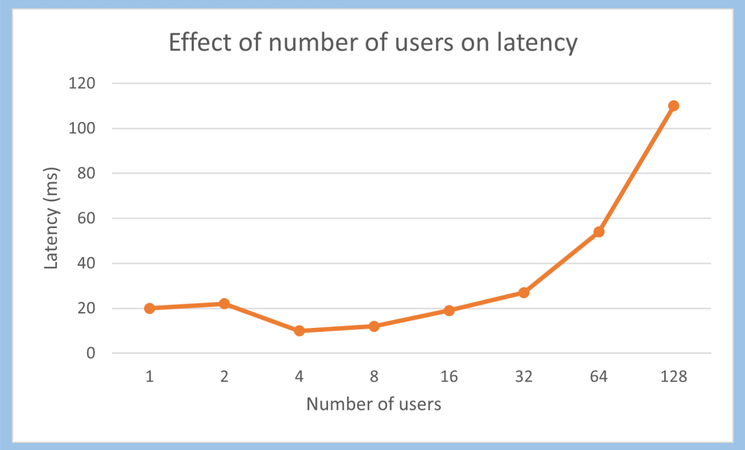
```
type HiddenBid struct {
    Organisation    string      `json:"organisation"`
    BidHash         string      `json:"bidHash"`
}

type RevealedBid struct {
    EventType       string      `json:"eventType"`
    Price           int         `json:"price"`
    Organisation    string      `json:"organisation"`
    Bidder          string      `json:"bidder"`
}
```

Structure of a bid in hidden and revealed form

6.Prototype

A client uses the application to interact with the e-auctions smart contract. The bidding chaincode runs in an enclave whereas the rest runs in the untrusted part of peer. SGX is used for attestation of chaincode and encryption of data.

7. Measurements



Graph representing the change in latency as the number of users of the application increases.

8. Conclusion

Execution of chaincode consisting of private data inside the trusted execution environment of Intel SGX enclaves ensures data confidentiality and protects it from malicious attacks. However there were still some limitations. Due to enclaves being vulnerable to risks such as rollback attacks, untrusted peers could reset enclave to change order of transactions leading to breach in privacy. With the proposed solution, the TCB is minimized, rollback attacks and other vulnerabilities are handled and the data is secured.