Enhancing Security and Privacy of Permissioned Blockchain using Intel SGX

Batsayan Das  
*TCS Innovation Labs*  
Kolkata, India  
batsayan.das@tcs.com

Srujana Kanchanapalli  
*TCS Innovation Labs*  
Bengaluru, India  
srujana.k@tcs.com

Vigneswaran R  
*TCS Innovation Labs*  
Chennai, India  
vigneswaran.r@tcs.com

Abstract—Permissioned Blockchain systems are envisioned to have immense impact in various sectors such as Finance, Supply Chain, IoT, Insurance and so on. However, the practical aspects and deployment has seen slower progress. One of the critical concerns restraining the practical deployment of blockchains is security and privacy of data and computations within the applications. Our research focuses on making Blockchain more practically deployable by enhancing privacy, security and scalability. In this direction, we leverage Trusted Execution Environment (TEE) to solve several practical challenges in Hyperledger Fabric, an open source permissioned blockchain. We employed Intel's Software Guard Extension (SGX) to improve scalability, privacy and security of the Enterprise Blockchain Network. And implemented a practical process as layer above (on the top) of Hyperledger Fabric and Intel SGX.

Index Terms—Permissioned Blockchain, Enterprise Applications, Hyperledger Fabric, Intel SGX, Intel SGX Protected Code Loader (PCL), TEE, Security, Privacy, Scalability

I. INTRODUCTION

Blockchain is a shared ledger that enables mutually distrusting parties to transact with each other without any central authority. The participants form a peer-to-peer network of nodes enforcing a common ledger of transactions. Fig. 1 shows the sample blockchain network. Blockchain, as its name implies, is a chain of blocks. Each block consists of a set of entries (financial transactions in case of cryptocurrencies) to be included in the blockchain and each new block is chained to the preceding block. All entries are appended to the ledger based on the consensus (agreement) of the involved parties. This ensures that the ledger is always consistent among all the parties.

The properties of a Blockchain are as follows:

- Decentralization: No central authority
- Transparency: Shared ledger
- Integrity: Immutable ledger
- Availability: Peer-to-peer network

A blockchain [13], as used in most cryptocurrencies, does not require any authorization for participants to join or leave the system, and hence is referred to as a permission-less blockchain. However, enterprise applications cannot operate in such models. Enterprise applications operate in a regulated and controlled environments with verified identities for all the parties involved. Hence, the permissioned blockchain models have evolved over the time.

Enterprises are keen on blockchain technology implementation on different sectors (like, supply chain, IoT, Identity management, Insurance, Healthcare and so on). These Permissioned Blockchain systems are envisioned to have immense impact in various sectors. However, the practical aspects and deployment has seen slower progress [16]. One of the critical concerns restraining the practical deployment of blockchains is security and privacy of data and computations within the applications. Our research work focuses on making Blockchain more practically deployable by enhancing privacy, security and scalability. This paper first explores the challenges for the enterprise blockchain networks and the second part explores about Intel SGX a Trusted Execution Environment (TEE) and then our work of implementing the Blockchain with Intel SGX for enterprise application with a secure layer of communication channel.

II. SECURITY, PRIVACY AND SCALABILITY IN BLOCKCHAINS

Making blockchain which is practically deployable involves solving several challenges such as scalability, privacy and security.

A. Security of Computations

The parties (nodes) in blockchain network append entries to the ledger after agreement on validation. The validation
involves business logic that asserts if the entry is valid or invalid. Such business logic is required to guarantee that the validation is sound and secure.

The guarantee involves:

- Security of private computations: The business logic is confidential if necessary. Several enterprise applications require the business logic to remain confidential to protect their business interests.
- Correctness of computations: The correctness of business logic execution has to be guaranteed to ensure robustness of the system. A more trusted execution process embeds higher trust into the system.

B. Privacy of data

Most of the enterprise applications mandate privacy requirements even within the organizations. Besides, the blockchain applications usually involve several organizations transacting with each other and hence require more stringent privacy guarantees. The applications involve computing together on each others’ data while still preserving the privacy. This requires mechanisms to facilitate computations on private data.

C. Scalability

As noted earlier, the entries to the common ledger are appended only after the parties validate the transactions independently and communicate agreement with each other. This induces significant latency into the system, hampering the overall throughput of the applications. Hence, mechanisms to increase the throughput are necessary.

We focus on solving the above mentioned challenges by using trusted execution environment. In particular, we employ Intel’s Software Guard Extension (SGX) [8] to improve scalability, privacy and security in permissioned blockchain setting.

III. INTEL SGX

A TEE enforces trust in an untrusted environment through software and hardware. Intel’s SGX (a set of extensions to Intel’s architecture) [4] [1] provides an isolated and secure execution environment called enclave for a user program without trusting any privilege software (such as hyper-visor or operating system). A TEE guarantees privacy and confidentiality (of both data/code and computation) with hardware security. The computations are performed in a secure container (enclave), which is deployed at the untrusted remote machine with the help of remote attestation. The enclave creation service runs at privilege level 0 (Ring 0) which is the most privileged level for any operating system. The deployment of instructions with Ring 0 privilege ensures higher security guarantees and more trust.

A. Application Design

Application design with Intel SGX requires that the application be divided into two components:

- Trusted component: Trusted application part of Intel SGX is the enclave. The instructions residing in the trusted enclave are isolated from the other application environment.

An application can have more than one trusted component or enclave.

- Untrusted component: Untrusted component comprises of rest of the application and any of its modules. It is important to note from the standpoint of an enclave, operating system and Virtual Machine Manager (VMM) are considered untrusted components.

For better security it is desirable to structure minimal trusted-untrusted components interaction. While enclaves can leave the protected memory region and call functions in the untrusted component (through the use of a special instruction), limiting these dependencies will strengthen the enclave against potential attacks.

B. Basic Control Flow of SGX

![Basic flow of control with SGX](image)

Fig. 2. Basic flow of control with SGX

Basic control flow of SGX [2] as shown in the Fig. 2. The important components of the flow are described as follows:

- Untrusted: It refers to code or construct that runs in the application environment outside the enclave.
- Trusted: It refers to code or construct that runs in the TEE inside the enclave.
- ECALL: A call from the application into an interface function within the enclave.
- OCALL: A call made from within the enclave to the application.
- Untrusted Run-Time System (uRTS): Code that executes outside the enclave environment and performs functions such as loading and manipulating an enclave (For example, destroying an enclave). Making calls (ECALLs) to an enclave and receiving calls (OCALLs) from an enclave.
- Trusted Run-Time System (tRTS): Code that executes within the enclave environment and performs functions such as receiving calls (ECALLs) from the application and making calls outside (OCALLs) the enclave. It helps in managing the enclave.
C. Features

There are three main functionalities (Fig. 3) that enclaves achieve listed as follows:

- Isolation - Instructions and data inside the enclave protected memory cannot be read or modified by any process external to the enclave. Hence, data and computation inside the execution are isolated from the untrusted application thus preserving privacy of data and computation.

- Sealing and Unsealing - The enclave platform uses secret key to encrypt data to the platform (that is, sealing), or to decrypt data already on the platform (that is, unsealing). Data passed to the host environment is encrypted and authenticated with a hardware-resident key. The data can only be accessed within the enclave when necessary. This limits the exposure of private application-critical data within the trusted zone.

- Attestation - A special signing key and instructions are used to provide an unforgeable report attesting to instructions, static data, and (hardware-specific) meta-data of an enclave, as well as outputs of computations performed inside the enclave. The data can only be accessed within the enclave when necessary. This limits the exposure of private application-critical data within the trusted zone.

These features of Intel SGX can be leveraged to solve the challenges mentioned in the previous section.

IV. HYPERLEDGER FABRIC WITH INTEL SGX

A. Application design

This section describes architecture and application flow of the client application interacting with enclave environment.

Similar approach in Fabric Secure Chaincode solution [14] to leverage a TEE in Hyperledger Fabric relies on modifications to the core platform to execute the whole smartcontracts within a TEE. Also the Coco framework [15] proposes ”running the Ethereum Virtual Machine itself within the TEE”. However, this approach is not ideal since it is important to keep the trusted code base small and contained which is the motivation for our solution to execute only certain parts of the smartcontracts inside TEE.

1) Architecture: Our design realizes leveraging Intel SGX in the Hyperledger Fabric (HLF) [12] environment through design of a security layer that enables communication between the blockchain application and the Intel SGX.

The enclave application architecture (Fig. 4) consists of three parts described as follows:

1. Fabric Node is a client application (HLF BC network node executes both chaincode and fabric code) which is executed on a machine (or node).
2. The application’s Untrusted part.
3. The application’s Trusted part.

2) Application Flow: - Fabric chaincode (application code) sends the request to the untrusted part.
- Untrusted part checks and send the required ECALL request to the trusted part.
- Trusted part executes the request and sends the reply to the untrusted part.
- Untrusted part sends the response back to the fabric chaincode.

Note that all this communication is under secure channel.

B. Security layer implementation

This design model has an SGX application (which has untrusted and trusted parts), and a Blockchain Node (which we use as client, representing blockchain network). This section explains the secure layer design implementation.

This section explains the implementation details in two modules; first module explains the key generation and client registration process, and the second module explains the data sharing between client (Blockchain Node) and SGX (Enclave or Trusted) using session key and Rivest-Shamir-Adleman (RSA) key.
This secure layer is implemented and tested with Intel SGX Protected Code Loader (PCL) [10]. The Intel SGX PCL [11] is intended to protect Intellectual Property (IP) within the code for Intel SGX enclave applications running on the Linux OS. The enclave shared object is encrypted at build time. It is decrypted at enclave load time. So, adversaries cannot reverse engineer the binary enclave shared object.

1) Module 1: Key Generation and User Registration: In this module the following activities will be performed:
- Key Generation: RSA key pair is generated at Client and Enclave side.
- User/Client Registration: In this process a Client ID and clients Pub.key is shared to the enclave. Similarly, enclaves Pub.key is shared with client for securing the subsequent communications.

Fig. 5 shows the flow of Key Generation and Client Registration process.

2) Module 2: Data sharing between Client and Enclave (SGX): In this module, client shares the payload or data with the enclave (SGX) using our secure communication design. Data is shared between enclave and client in a secure way using session key. This session key is randomly generated symmetric key. A new session key is generated for every transaction (for sharing inputs or payloads) to communicate in a secure way. After enclave receives the payload, it executes the business logic and send the result back to client in a secure way.

Fig. 6 shows the process flow of data sharing between Client and Enclave.

Blockchain Node with Golang 1.11 crypto library for generation of key, encryption, and decryption.

3) Future proposal - Design proposal for Future Implementation: Data sharing with Signature:

Signature will help in avoiding man in the middle attack in case of Client Registration as well as Data or payload Sharing process of our secure layer communication. This section is very similar to the Module 2. Additionally signature will be passed to enclave for verification. The Fig. 7 shows the flow and steps involved in the process of data sharing with signature.

C. Realizing our Blockchain objectives

We now describe how the features of Intel SGX realizes our objectives of improving security, privacy and scalability in blockchains.

1. Security Of Computations: Security of computations in blockchain applications can be attributed to security of private computation and correctness of computation.

The independent parties in the blockchain network may require to preserve the confidentiality of computations. Hence, it is not possible to share such computations with other parties for transaction validation. The TEE executes the business logic and attests the result guaranteeing the correctness of the computation. Thus, the TEE ensures security of private computations and embeds trust in the correctness of business logic execution.

SGX Application side uses intel-sgx-ssl library [9] and
2. Privacy of data: Privacy of data is of utmost importance in applications involving several parties. Blockchain applications require several parties computing together on each others data while still preserving the privacy.

The isolation of TEE facilitates preserving the privacy of data while allowing computations on data. The involved parties can provide encrypted private data to the TEE. The enclave (TEE) decrypts the data within the trusted zone and computes necessary operation on the data. The TEE then sends attested results that can be verified and accepted by the parties without revealing private information to each other.

3. Scalability: The entries to the common ledger are appended after the parties validate the transactions independently and communicate agreement with each other. This induces significant latency into the system hampering the overall throughput of the applications.

TEE can be leveraged to improve the scalability in network. Instead of individual parties validating the transactions, a TEE (enclave) can be employed to validate the transaction and endorse the validity. This way, all the parties need not validate transactions independently, rather just verify the endorsement of the enclave before appending the transaction into the blockchain. This decreases the communication rounds necessary for parties to reach agreement on the validity and improves the transaction throughput in the network.

Please note that this solution is not specific to Hyperledger Fabric. This solution is in general applicable to any blockchain system.

V. OTHER RELATED WORK

Fabric Secure Chaincode solution [14] leverages Intel SGX with Hyperledger Fabric for the same purposes. However, in this the core platform is modified to execute the entire smartcontract within TEE. Similarly, the Coco framework [15] proposes “running the Ethereum Virtual Machine itself within the TEE”.

However, it is important to keep the trusted code base small and contained which is the motivation for our solution to execute only certain parts of the smartcontracts inside TEE.

VI. CONCLUSION

We leverage Intel SGX, to improve the privacy, security and scalability of blockchain applications. Though there are other alternate solutions such as Zero-Knowledge proofs and verifiable computing, a trusted execution environment is the most practical solution in terms of viability and efficiency. Our security layer implementation can be utilized in any blockchain application by appropriately scheming the application to utilize our feature implementation to improve security, privacy and scalability in blockchain applications.

ACKNOWLEDGMENT

We thank Mr. Sitaram Chamarty, Dr. Rajan M A, and Mr. Nitesh Emmadi from Tata Consultancy Services for their insightful feedback. We also thank anonymous reviewers for their valuable comments in improving the paper.

REFERENCES