Encblock: Enclave and Blockchain-enabled Secure Data Sharing and Computing for Power Grids

Derock Xie[1] and Anup Kumar[2]

[1]Kentucky Country Day School, Louisville, KY, US
Email: Derock.Xie@student.kcd.org

[2]University of Louisville, Louisville, KY, US
Email: ak@UofL.edu

***Abstract*—Data sharing and computing are integral aspects of modern power grid networks. It involves the transmission of information about electricity generation, consumption, transmission, and distribution, and should be operated in an efficient and secure way. Traditional approaches conduct mutual authentication and authorization among networks. The interoperability is an issue with the increase of network interconnectivity and complexity of grid data. The paper addresses this issue by introducing a secure data sharing and computing approach that leverages enclaves and blockchain technology, referred to as Encblock. Encblock has a unique design to achieve an enclave-based trusted and confidential SGX computing environment. It is built up with a Dynamic Distribution System (DDS) Pub/Sub (Publisher/Subscriber) middleware and power grid common data model for data compatibility and flexibility for data sharing. A remote attestation protocol is developed to maintain the enclave integrity and authenticity with the external Intel attestation server. Meanwhile, the attestation and data computing results can be securely managed in a blockchain to support various power grid businesses. Substantial experiments are conducted with the data simulations to verify the Encblock performance such as remote attestation latency and the blockchain data transaction capability.**

***Index Terms*—Power grid, Enclave, Data Sharing, Confidential Computing, and Blockchain**

I. Introduction

Data sharing and computing frequently occur in the power grid and industrial control networks. Particularly, the electronic grid is an interconnected network that includes industrial power plants, factories, rural networks, industrial customers, solar/wind farms, power trade networks, and more. This network generates a wealth of data from sensor endpoints such as Phasor Measurement Units (PMUs), protection relays, smart meters, line sensors, transformers, and others. This data is transmitted and utilized across the power grid and its service networks, supporting various operations, trading, management, and services. Security mechanisms are integrated into the grid communication protocols to maintain the grid communicating device authenticity, data integrity, and data confidentiality. For instance, IEC 62351 defines the cybersecurity implementation to protect the data transmitted through IEC 60870-5 series (including IEEE 1815 (DNP3) as well as IEC 61850 series. It allows the power grid network to securely manage the communication key (e.g., certificates), access control, payload encryption and decryption, and secure event logging.

Beyond ensuring data integrity and confidentiality, blockchain has been investigated for its potential to provide additional security features, such as data audibility, privacy-preserving, and redundancy for grid data sharing and computing. Christian Banks et al. [1] take blockchain into consideration to share transaction information among different power grids in a secure, controlled, monitored, and efficient manner. Yuntao Wang et al. [2] developed a blockchain-based data usage tracking scheme to achieve auditable private data sharing and computation. In this scheme, smart contracts for accountability are designed to specify fine-grained data usage policies, and the transaction of the data is recorded in the distributed immutable ledgers. Imran Makhdoom et al. [3] developed a blockchain-based data-sharing scheme with protection of privacy-preserving in a smart city environment. In this scheme, data privacy is maintained by dividing the blockchain network into multiple channels. Each channel has a finite number of authorized parties of specific data types and is controlled by data access control rules described in the smart contracts. In addition to these data security features, blockchain is also considered to promote data sharing in power grid business. The promotion of data sharing is achieved by a blockchain-based security preservation and award framework to encourage the participation of users like power trading customers. Recently, A. Khan et al. [8] advocated the integration of blockchain with deep learning to achieve secure smart power grid scheduling, optimization, and automation such as smart grid-based control management.

Different than the current approaches, this paper proposes an enclave and blockchain-enabled method for secure powder grid data sharing and computing. It considers the scenario that each power grid section, e.g., power plants, solar/wind farms, and power trade networks, is a private network owned and operated by different energy service providers. Traditionally, these private networks are mutually authenticated and authorized with each other such that the energy data can be shared across multiple domains in support of various operations, controls, management, and services. For example, a renewable

energy network connects a set of solar and wind farms to form a rural network which again is connected with the city network with the addition of power plants. Operating with mutual authentication, each domain may need to maintain a security gateway for interaction with external networks. This not only requires interoperability among private network domains but also introduces complexity issues. In order to address these problems, this paper proposes an enclave in the design of a common authentication and authorization agent that can be trusted by all grid network parties. This enclave implemented by Intel SGX [6] can also serve as a trusted computing platform capable of executing grid computing tasks (e.g., data aggregation and fusion) involving multiple grid parties. Confidential computing is achieved without revealing information between any two grid networks and only the evaluated results are presented to external networks. Furthermore, the computing data, tasks, and results can be managed in a blockchain for the purpose of security and grid operations with the use of this information, e.g., the aggregated data produced from private Solar PV inverters is transmitted to an energy trading platform without disclosing sensitive grid information.

Encblock is the first attempt that applies enclave technology for secure grid data sharing and computing. Our contribution to this paper is threefold:

- **Enclave-based Trusted Computing for Grid Networks**: Encblock presents a system model that builds a secure data-sharing and computing environment with the use of enclave technology. It encapsulates DDS middleware, grid data computing modules, and a blockchain agent in a confidential and secret enclave that offers hardware-based memory encryption for the code and data in enclave memory. The use common data model allows compatibility and flexibility for data sharing inside the enclave.
- **Encblock Remote Attestation**: Encblock proposes remote attestation built over the Intel SGX enclave remote attestation procedure to validate the authenticity and integrity of the enclave and ensure the safety of the session. Different than other approaches, the remote attestation is issued by blockchain and the result is reported into the transaction for the purpose of attestation traceability.
- **Blockchain Integration**: Encblock presents a new design to integrate a private or public blockchain. The integration is easy but achieves the goal of supporting grid business without modification of the grid network infrastructure and protocol.

Substantial experiments are conducted with the simulation of power grids, an enclave, and a blockchain in an integrated Encblock system. The results demonstrate Encblock performance such as attestation latency and the blockchain data transaction capability under different configurations.

II. Encblock System Model and Design

Encblock allows a blockchain to be created externally on the power grid networks without directly accessing the grid infrastructure and network. In order for the blockchain to consume the grid data collected or computed from grid networks, security measures should be addressed to ensure data authentication, authorization, integrity, and immutability. For this goal, Encblock is designed to execute in an enclave environment, acting as a multi-party trusted point for secure data exchange and computing between grid networks and the blockchain. On the other hand, data computing inside the enclave with confidentiality and privacy can be maintained among multi-party power grid networks.

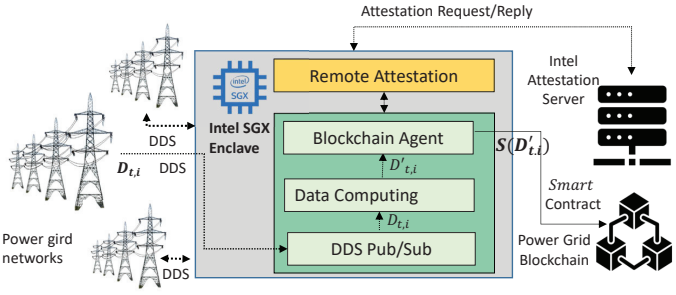

Fig. 1: **Encblock System Model**

A. System Model

Fig. 1 shows the proposed Encblock system architecture where Encblock source codes are securely published and executed under the shield of an authorized SGX enclave. The sensitive data, secret key, and private keys shall be protected in the Enclave from external hardware and software attacks. Remote attestation is a method that establishes a secure channel for a service provider to remotely verify if the enclave program codes are running securely within an enclave. Fig. 1 shows our remote attestation process through Intel Attestation Server [7]. It allows the system to verify if Encblock programming functions are performed honestly or not inside the enclave.

Other than the remote attestation module, Encblock has three software modules as shown in Fig. 1. For providing interoperability, a Dynamic Distribution System (DDS) Pub/Sub (Publisher/Subscriber) middleware is implemented to provide a common interface to interconnect with multiple heterogeneous and private grid domains or networks. Secondly, Computing is a software module for performing trusted data processing, such as fusing the data from power grids. Additionally, Encblock integrates a blockchain agent such as IBM Hyperledger, Ethereum, or others depending on the energy applications, e.g., voltage control, microgrid monitoring, and energy trading. Let $D_{t,i}$ denote the data reading generated in ith grid network at time t. Fig. 1 shows the data flow where $D_{t,i}$ is transmitted to Encblock upon its subscription. $D_{t,i}$ is then computed and the evaluated outcome is $D'_{t,i}$. Secret computing could be conducted in a way that $D'_{t,i}$ is the result of calculation with the use of data from multiple grid while each grid has no knowledge of the data from other grid networks. In the end, $D'_{t,i}$ is published into the energy blockchain by initiating a smart contract, denoted by $S(D'_{t,i})$.

B. DDS Secure Communications and Grid Data Profile

DDS inside the enclave performs data-centric publish/subscribe communications that is well applicable in a large-scale power grid real-time environment. DDS publisher/subscriber code is published into an enclave which creates a session and monitors the code execution during the full session time. Power grid networks could provide channel, topic, and/or type-based publish/subscribe services for Encblock to catch the data $D_{t,i}$ of interests. Let k_E be the encryption key. $D_{t,i}$ is encrypted $\langle D_{t,i} \rangle_{k_E}$ and disseminated to subscribers. Encblock decrypts it with the corresponding decryption key k_D. Symmetric or asymmetric cryptography could be implemented for DDS publishers/subscribers. Symmetric cryptography causes high key distribution complexity since the publisher has to securely distribute the secret key k_D to all subscribers. On the other hand, asymmetric cryptography exhibits higher computational overhead. For both approaches, an authorized center like PKI (Public Key Infrastructure) provides authentication and authorization for the provided data services. For instance, DDS Security by Object Management Group [5] offers a $DDS : Auth : PKI{-}DH$ plugin to implement authentication using a trusted Certificate Authority (CA). Mutual authentication can be conducted between Encblock and grid networks using the RSA (Rivest–Shamir–Adleman) or ECDSA (Elliptic Curve Digital Signature Algorithm) digital signature algorithms. Upon authentication, Diffie-Hellman (DH) or Elliptic Curve Diffie-Hellman (ECDH) Key Agreement Methods could be used to establish shared secrets for Encblock and grid networks. Authentication prevents unauthorized publishers or subscribers from injecting data into the DDS network or receiving data that is not intended for Encblock. Similarly to DDS, MQTT or NATS publisher/subscriber could be integrated in an enclave for secure data $D_{t,i}$ communications.

For interoperability, DNP-3, ANSI C12, DNP, Modbus, Head-End, and other legacy power grid data $D_{t,i}$ could be translated into a common data model before encapsulating them into a publisher message. Common Information Model (CIM) is a standard information model in electrical power transmission and distribution that includes an abstract and formal representation of objects, their attributes, their associations with other objects, and the behavior and operations performed on them inside the electrical network. Specifically, the information model is formally described in a rigid language or diagramming technology, especially using UML (Unified Modeling Language). CIM package dependencies are extended from three standards defined as IEC (International Electrotechnical Commission) 61970, IEC 61968, and IEC 62325. IEC 61970 standard defines data packages of the CIM regarding electrical power transmission. It also defines an XML metadata model for electrical network model exchanges. ICE 61968 series of standards extend the CIM to cover the requirements of electrical distribution such as distribution management systems, outage management systems, planning, metering, work management, geographic information systems, asset management, customer information systems, and enterprise resource planning. IEC 62325 standard defines energy market communications messages for the wholesale energy market. Additionally, IEC 61850 is a standard to achieve interoperability and automation at the substation level. The abstract data models defined in the IEC 61850 cover several communication protocols for intelligent electronic devices at an electrical substation. These protocols are MMS (Manufacturing Message Specification), GOOSE, and SMV (Sampled Measured Values). These protocols can achieve a high-speed response time for protective relaying over substations LANs using high-speed Ethernet. For example, IEC specifies the Recloser data by defining three profiles namely, Event Profile, Reading Profile, and Control Profile to describe the Recloser's events (e.g., open or closed), reads (e.g., voltage, current), and control schedules (e.g., configuration parameters) respectively. Consider a Recloser voltage reading of 110 voltage at12/26/2023 4:35:29 AM GMT-05:00. Data $D_{t,i}$ can then be written in a Recloser data profile, i.e., RecloserReadingProfle.RecloserReading.readingMMXU.PhV.net.cVal.mag.f $= (float)110$; RecloserReadingProfile.messageTimeStamp.seco -nds $= 1700991329$. In addition to the reading and time, the device identification and status information such as id, $name$, and $description$ could be enclosed in the $D_{t,i}$ data profile.

C. Encblock Enclave and Grid Data Computing

Intel SGX enclave is considered for Encblock to accommodate and execute all its software modules (i.e., DDS, grid data computing, remote attestation, and blockchain client) in a trusted execution environment. For example, CREATE is an instruction to start an SGX Enclave Control Structure (SECS) in the Enclave Page Cache (EPC) which is the protected memory for enclaves. EADD instruction then adds pages to the enclave. SGX records the enclave to which the page was added, its virtual address as well as its permission. Subsequently, security restrictions are enforced to ensure the enclave maps the page at the access virtual address. When all enclave pages are added, EINIT instruction creates cryptographic keys that will be used for remote attestation. Such a secure container is desirable for Encblock to initiate and manage its enclave session without directly manipulating these SGX instructions. Additionally, the container provides SGX-extended libraries that enable Encblock inside the enclave to communicate with the Linux kernel that is outside the enclave. This method consists of several shielded syscalls functions. The response of the syscall returns the results back to the enclave. While the SGX-extended library is dynamically linked to the internal codes, Encblock can securely communicate with external systems via three container shields:

- **OS Shield**: Encblock enclave is initiated by a Container running in a underlying trusted OS. It prevents the poison of the container and OS-level attacks such as OS kernel controlling pointers to ensures the enclave's confidentiality and integrity for Encblock trusted computing. Once the enclave is created, Encblock runs in an isolated environment and prevents external attacks from the OS.

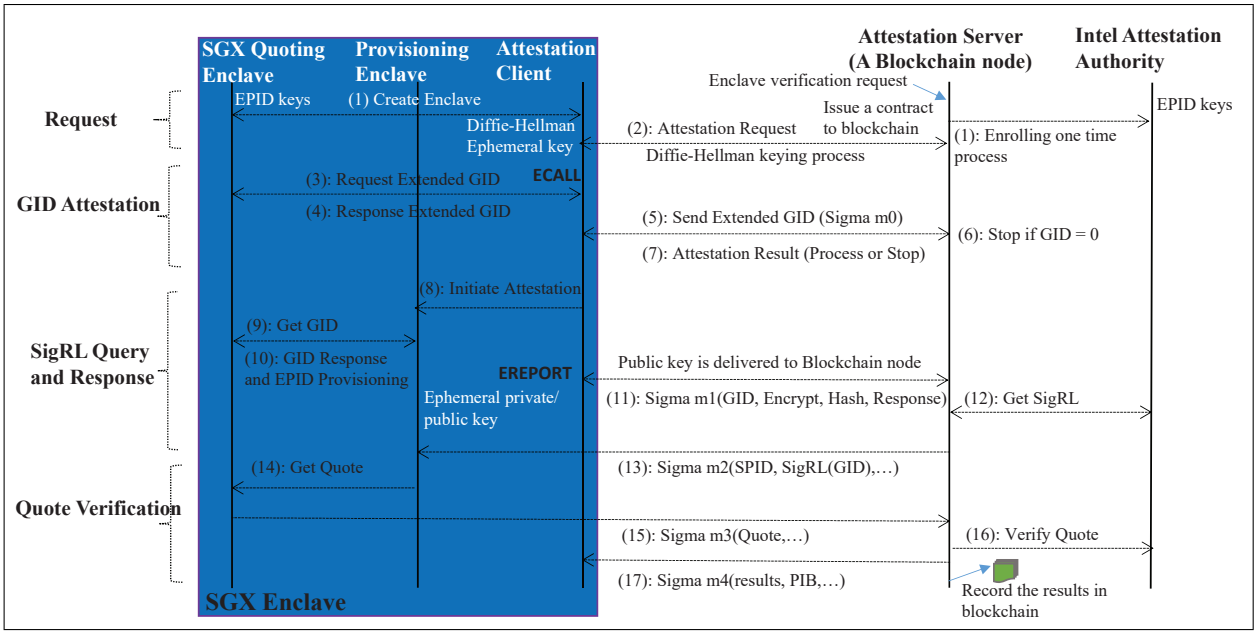

Fig. 2: Encblock Remote Attestation Process

- **File-system Shield**: File-system shield allows Encblock to securely mount its system configuration and source codes into the enclave during the initialization of the container. The configuration data is a collection of DDS configuration, blockchain user account information, and keying information. The compiled source code is the executable binaries (e.g., ELF (Executable and Linkable Format)) of DDS middleware, remote attestation methods, grid data computing algorithms, and a blockchain client. All these executable files are needed to run the designated Encblock functions.
- **Network Shield**: The network shield allows Encblock to securely communicate externally with the grid and blockchain networks. In the Encblock design, the network connection from the enclave to the external networks is established by end-to-end TLS (Transport Layer Security)/DTLS (Datagram Transport Layer Security), and the traffic is encrypted end-to-end. Beyond these capabilities, the enclave network shield should prevent its connection from being redirected to an authorized party.

Grid data computing can be implemented in a variety of ways, depending on grid applications with the use of the decrypted grid sensor data, denoted as $D_{t,i}$. Let's consider a scenario where Encblock receives $D_{t,i}$ as a stream from various grid networks. The raw data, such as the Recloser Identity, is sensitive and should not be disclosed to other parties. Only the computed results, for example, $D'_{t,i}$, which could be statistics, fusion, or learning results of $D_{t,i}$ can be made public. Encblock integrates advanced grid data computing software binaries into a confidential environment without altering any of the participating systems. This allows for secure collaborative confidential computing. The results can then be shared among systems or published externally via blockchain. This preserves data privacy and security while allowing for effective computation and collaboration.

D. Encblock Remote Attestation

Encblock remote attestation is built over the Intel SGX enclave remote attestation procedure to detect if the enclave has been tampered with or not. This validates the authenticity and integrity of the enclave and ensures the safety of the session, the secret (e.g., encryption keys), and the executive binaries. The authenticity of an Intel SGX enclave under the latest Trusted Computing Base is attested. Trusted Computing Base is an entity responsible for protecting the secret provisioned to the enclave (both software and hardware). Also, Encblock remote attestation validates the identity and authority of the enclave and its session data. It also verifies the integrity of the Encblock executable binaries without compromising, which assures the trusted computing of the DDS publisher/subscriber, computing algorithms, blockchain agent, and other enclosed functions. The initiation of the remote attestation processing is started by an authorized party which is a specified blockchain node. For traceability, Encblock considers the attestation to be issued by a blockchain smart contract and the results are saved into the blockchain network through a transaction. The attestation frequency can be scheduled from the blockchain network by means of smart contracts.

Figure 2 illustrates the genetic Encblock remote attestation procedure. At first, the blockchain node acting as the attestation server is securely registered with the Intel attestation server. Then, attestation can be initiated immediately or later any time after the creation of the enclave session. EPID (Intel Enhanced Privacy ID) cryptographic keys will be generated at

the enclave to allow the enclave to sign a message without leaving a trace that can be uniquely backtraced to the signer, a means of the anonymous signing process. The specific attestation can be illustrated with the following steps:

Attestation Initialization: This is the step to initiate a secure channel and start an attestation request. Before issuing a remote attestation, the blockchain node registers itself to the Intel Attestation Authority (i.e., Step 1 in Figure 2). This registration binds the blockchain TLS certificate to a unique Service Provider ID (SPID), and permits access to the service. The blockchain node starts a remote attestation with initiation of a Sigma protocol towards an Enclave (i.e., Step 2 in Figure 2). An ephemeral shared secret key is produced as a result of a Diffie-Hellman keying process. The enclave asks the blockchain node for provision secrets. The blockchain node responds with a remote attestation request message through the secure channel. The request includes SPID and a nonce which is a random number to ensure the freshness of the request.

Group ID (GID) Attestation: This is a process of a local attestation for the blockchain node to verify if the Encblock executable codes inside the enclave are valid or not. The Encblock code is linked to a Quoting Enclave by storing its Enclave Identity. The Enclave Identify of the Encblock code is a cryptographic hash of the enclave log that uniquely represents the executable binaries. It is restricted to access from unauthorized parties. To verify the Enclave Identity, an ECALL (a call to program enclave) is executed inside the enclave towards the Quoting Enclave. ECALL invokes a Request of an Extended GID and returns the results in a Response of the Extended GID, i.e., Steps 3 and 4, where GID is the Group ID of the EPID of the Encblock. GID is then responded back to the Blockchain node (i.e., Step 5). The ECALL result carries out the SGX status (e.g., still true or not as the check) as well as the Diffie Hellmen Key Exchange context. With this information, the blockchain node verifies the Enclave Identify of the executable codes, checks if GID is valid or not, and the results are sent back to enclave, i.e., Steps 6 and 7 respectively. If GID is invalid (i.e., $GID = 0$), the attestation stops, meaning that it is false Encblock code (e.g., tampered). Otherwise, attestation continues.

SigRL Query: This is a SigRL querying process for the enclave to retrieve enclave information (e.g., the originating enclave) from the Intel Attestation Authority such that the corresponding Quote can be obtained for attestation. This process covers Steps 8-13 as shown in Figure 2. At first, the attestation client initiates an attestation request to the Encblock enclave (e.g., Step 8). The request will result in EPID provisioning at the SGX enclave (i.e., Steps 9 and 10). Meanwhile, Encblock generates a pair of ephemeral public and private keys to build a secure channel towards the blockchain node. The private key is for encrypting the querying message. The public key is released to the blockchain node. The querying message is a built-in format of Sigma protocol with message encryption, hash, and response, denoted by $Sigma\ \ m1(GID, Encryption, Hash, Response, \cdots)$. Receiving $Sigma\ m1$, the blockchain node first verifies the authenticity of the message, and then invokes a query of Signature Revocation List, known as SigRL, to the Intel Attestation Authority, i.e., Step 12. SigRL is a list of untrustworthy signatures or verified blacklists, signed by the Intel revocation authority. SigRL is returned back to Enclave (i.e., Step 13) via the blockchain node, Step 13, as shown in the message of $Sigma\ m2\ (EPID, SignRL, SPID\ QUOTE\ type, \cdots)$.

Quote Attestation: This is a process to accomplish the remote attestation by evaluating the enclave quote, including Steps 14-16 as shown in Figure 2. Message $Sigma\ m2$ is forwarded to the quoting enclave. The quoting enclave then calls the hardware enclave instruction GETKEY to get the key to verify the message $Sigma\ m2$. Quoting enclave creates and signs QUOTE using its EPID key such that the QUOTE is only verifiable at the Intel Attestation Authority. QUOTE is then forwarded to the blockchain node using a message $Sigma\ m3\ (QUOTE)$, i.e., Step 15 as shown in Figure 2. The blockchain node uses the EPID public key encapsulated in a certificate to validate the signature of $Sigma\ m3\ (QUOTE)$. It meanwhile checks any parameters included in $Sigma\ m3\ (QUOTE)$, such as the DHKE (Diffie-Hellman Key Exchange) parameters and the application enclave's identity (embedded in the QUOTE). If the quote from the Intel Attestation Server is successfully validated, Encblock extracts the attestation status, MRSIGNER representing the signer of the report, corresponding security version, and a product Identity ID. It then derives the session keys, i.e., SK (secret key), and MK (master key), to generate an attestation response message $Sigma\ m4$ as a response of $Sigma\ m3$. $Sigma\ m4$ carries the trustworthiness of enclave, Platform info Blob (PIB), current time, enclave re-attestation timeout, and other customized information if requested, which is Step 17 as shown in Figure 2. Moreover, the testing results can be recorded in the blockchain via a smart transaction.

E. Blockchain Integration

Encblock offers an integration of the power grid network with either a private or a public blockchain, such as Hyperledger or Ethereum, based on specific business requirements. In this setup, the blockchain client is fully encapsulated within the enclave, enabling secure access to the data $D'_{t,i}$. The integrity of the blockchain client is preserved within the enclave, ensuring a secure process to issue a smart contract to transition data $D'_{t,i}$ into the blockchain. Conversely, blockchain data can be disseminated to a grid network via the client and the DDS publisher, facilitating a mutual data flow between the grid network and the blockchain network. The integration is accomplished without direct interactions between distinct grid and blockchain networks, achieving interoperability without the need for modifying their respective protocols. The blockchain client possesses its own set of security keys for communication with its blockchain network, independent of the DDS security keys. This design ensures a seamless and secure data exchange between the different networks.

III. EXPERIMENTS AND RESULTS

Encblock is simulated in a laboratory environment at InfoBeyond Technology LLC and we performed extensive tests to demonstrate the performance.

A. Settings and Simulation

RTI (Real-time Innovations) simulators are considered in our simulation to generate a large number of different power grid data types to test the Encblock data process. Several example simulators are illustrated below to show how the data in field devices are generated in our experimental setups:

- **Solar Inverter Simulator**: A solar inverter simulator is a program that generates a Solar Reading Profile (i.e., Photovoltaic Power output) and a Solar Status Profile (i.e., grid connection status). The PV power output from the simulator is the calculation of $PV\ Output = (I_F \times s + o) + ((j \times d_R) - j/2)$, where I_F is an input power watt value, s is a scale value, o is an offset value, j is a jitter range value, and d_R is a random number.
- **Load Simulator**: A load simulator is a program to emulate a 24-hour commercial load in a power grid such as in a factory setting. The power consumption simulation value at the hth hour and the mth minute is calculated as $P_{(h:m)} = P_h + |P_{(h+1)} - P_h| \times m/60 + N(\mu, \tau^2)$, where P_h and $P_{(h+1)}$ are two power consumption values of hth and $(h+1)$th hour from a preconfigured 24-hour table, $N(\mu, \tau^2)$ is Gaussian noise which is a random number that follows normal distribution having a mean value of μ and a variance value of τ^2.
- **Recloser Simulator**: A Recloser simulator is a switch connecting a microgrid to the main power grid. It subscribes to Reading, Control, and Event Profiles of every other field device (i.e., Battery, Solar Inverter) in the microgrid. The Recloser simulator is able to island or connect to the main grid and publish Recloser Status and Event Profile data.

Based on the simulators, we build our simulated power grids, such as a microgrid that consists of multiple power grid field devices. These field devices are emulated in a way to generate close-to-operating data, such as device status (e.g., open or closed), power, voltage, current, phase, and frequency. In our simulation, each device runs independently in a thread to emulate its real-time behaviors in the power grid system, using JAVA multithread programming technology. Each simulated device has a configuration file that specifies the device property and device behavior parameters (i.e., data publishing rate).

Real-time DDS middleware is incorporated into the simulation for power grid communications. It integrates the RTI Context DDS which is a DDS solution developed by RTI. Using the RTI library, each grid device is simulated with a DDS publisher and subscriber. As an example, the Recloser at PCC (Point of Common Coupling) detects the grid power unavailability and a Recloser event Data Profile will be generated. The Data Profiles admitting to the IEC 61850 standard are published as an anomaly event through a DDS message. The subscriber at the Encblock enclave can receive the message.

Fig. 3: **Demonstration of** $Sigma\ m2$ **Message**

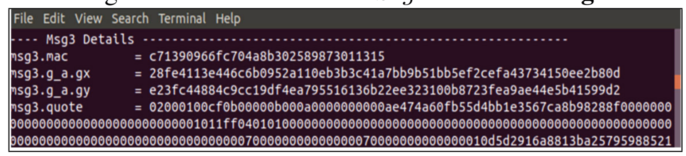

Fig. 4: **Demonstration of** $Sigma\ m3$ **Message**

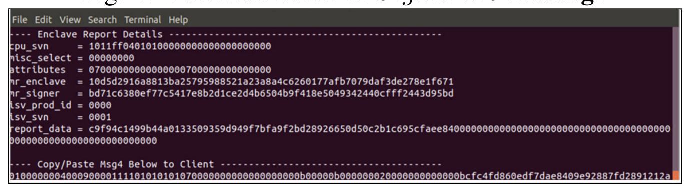

Fig. 5: **Demonstration of** $Sigma\ m4$ **Message**

B. Remote Attestation Performance

Encblock enclave is implemented in a Dell Server of Intel(R) Xeon(R) CPU E3-1220 v5 3.00GHz with 64 GB memory and remote attestation is conducted with the Intel attestation server, as illustrated in Figure 2. Figure 3 shows a $Sigma\ m2$ message that includes the Elliptic Curve point (g_x, g_y), SPID, Quote_type, KDF_id, Sign_ga_gb, MAC (i.e., CMAC), and $SigRL\_size$, where KDF_id is the Key derivation Function ID, CMAC is the Cipher-based Message Authentication Code, and Sign_ga_gb is the shared secret key. Figure 4 shows a $Sigma\ m3$ message where the quote $msg3.quote$ includes a cryptographic hash or measurement of the running enclave which is signed with the Intel EPID key. Figure 5 demonstrates the attestation result of a $Sigma\ m4$ message as a response to a $Sigma\ m3$. Figure 38 shows the detailed enclave report where the enclave is identified by mr_enclave (i.e., enclave identity). If the enclave session is changed, mr_enclave will be changed and cannot be matched on the attestation. The enclave report includes cpu_svn (i.e., CPU security version number), attributes, mr_signer (i.e., signer of the report), isv_prod_id (i.e., product id), isv_svn (i.e., enclave security version number). It also includes the attestation report data such as Platform info Blob (PIB), current time, enclave re-attestation timeout, etc.

Table I shows the experimental results of Encblock session creation and attestation latencies. It first gives the time (i.e., 0.938256 seconds) required to create an enclave session to execute Encblock codes and return the Session Name and Session Hash. The Session Hash is 64 bytes to identify the session uniquely. The attestation delay is 51.665281 seconds

Encblock operations	Latency (seconds)
Create an enclave session to execute the Encblock	0.938256
Attestation request, attestation, and return the result to Enclave	51.665281
Attestation request, attestation, return the result to Enclave, and store the report in blockchain	51.842382

TABLE I: Encblock Operation Performance

which includes all the operations from Step 2 to Step 17 illustrated in Figure 2. This includes the time required to run the attestation software module inside Enclave, the interaction and communication overhead of remote attestation protocol, and the processing delay at the Intel Attestation Server. The result shows the attestation has a significant overhead including high network delay with external Intel Attestation Server and its processing for verifying the enclave. Table I also shows the additional latency required to transit the attestation report to a blockchain created in our laboratory.

C. Grid Data Blockchain Performance

A Hyperledger Fabric private blockchain is built into our simulation that is integrated with Encblock. It simulates the grid data flows as described in Figure 2 where the simulated data (e.g., $PV\ Output$) is collected from a microgrid and saved into the Hyperledger blockchain. A Hyperledger client runs inside an enclave and is able to initiate contracts to the Hyperledger peer network. Blockchain Commitment Latency (t_{Comm}) is defined as the time interval starting from the time that the Encblock receives the DDS subscribed data message to the time that the corresponding transaction is accomplished in the Hyperledger blockchain. In the experiment, Encblock is executed under the servers of different CPU and memory configurations (i) 12 CPU 8GB Memory: Intel Xeon E5-2620 v3 CPU with 12 cores at 2.40 GHz and 8 GB RAM, (ii) 8 CPU 16GB Memory: Intel Core i7-6700 CPU with 8 cores at 3.40 GHz and 16 GB RAM, (iii) 4 CPU 64GB Memory: Intel Xeon E3-1220 v5 CPU with 4 cores at 3.00 GHz and 64 GB RAM, (iv) IBM Cloud: 2 CPU and 4GB RAM. Hyperledger peers are running in the server with the configuration as the Encblock for each scenario. All the servers run Ubuntu 18.04.4 LTS with Linux kernel version 5.3.0.

In the experiments, the simulated power grid creates $30, 40, 50, \cdots, 400$ bursty $PV\ Output$ and other grid data, which will result in bursty Hyperledger transaction requests at the Encblock. These transactions are then submitted to Hyperledger. Figure 6 shows blockchain commitment latency with Encblock. The results show that the commitment latency increases along with the intensity of burst transactions. Encblock with 12 CPU 8GB Memory achieves the least commitment latency and the Encblock with 2 CPU 4GB Memory has the highest latency. The results show that the CUP capability plays a strong impact on transaction processing. This is because in our implementation, the Encblock is executed in multiple threads and the process of a transaction primarily consumes CPU computing resources, e.g., for encryptions. Also, the impact of memory can be indicated from the results. For

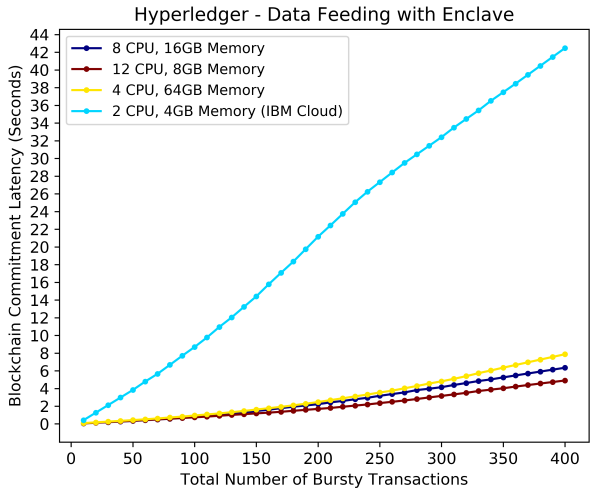

Fig. 6: **Blockchain Commitment Latency with Encblock**

example, the commitment latency for 4 CPU 64GB Memory is significantly higher than 8 CPU 16 GB Memory. This is due to the impact of the increase in memory (16 GB vs. 64 GB) is much less than that of the increase in CPU capability (8 CPUs vs. 4 CPUs). Increasing memory from 16 GB to 64 GB has less impact on the performance, compared to the increase from 4 CPUs to 8 CPUs. Furthermore, the total commitment delay increases when more busty transactions are produced. On the other hand, Encblock performs well for processing bursty transactions in the scenarios of 4, 8, and 12 CPUs. This is because the total commitment latency only slowly increases with the increase of bursty transactions from 30 to 400. The total commitment delay for 400 transactions is accomplished in less than 4 seconds for both 12 CPU 8GB Memory and 8 CPU 16GB Memory.

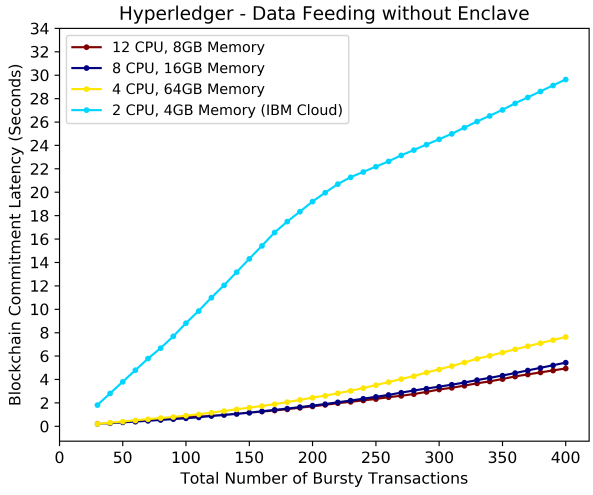

Fig. 7: **Blockchain Commitment Latency without Enclave**

The experiments for Figure 6 are conducted while SGX enclave is disabled. In this case, Encblock is executed with-

out SGX enclave security protection. Comparing the results between Figure 6 and Figure 7, we can see the degradation of the performance caused by the enclave overhead. For example, the average commitment latency is 30 seconds without enclave is degraded to 42 seconds with enclave in the case of 2 CPU 4 GB Memory. On the other hand, the degradation is less with the increase of the CPU capability, e.g., less degradation in 12 CPU 8GB Memory. This is because 12 CPU 8GB Memory provides higher enclave performance compared to 2 CPU 8GB Memory. Meanwhile, Encblock with 12 CPU 8GB Memory still achieves the best performance that has the least commitment latency as shown in Figure 7.

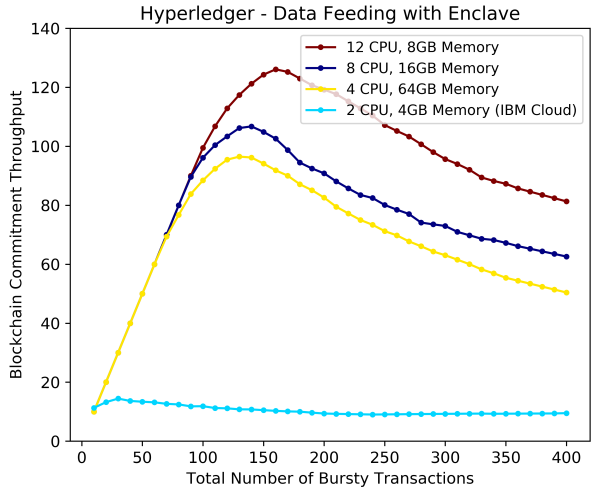

Fig. 8: **Commitment Throughput with Enclave**

Blockchain Commitment Throughput (α_{comm}) is defined as the total number of accomplished blockchain transactions per second by an Encblock through transactions. The maximal commitment throughput represents the achievable Encblock transaction capability. A larger throughput indicates a higher grid data transaction capability. Figure 8 shows that different configurations achieve different maximal processing capabilities. Given the scenario of 12 CPU 8GB Memory, the commitment throughput increases with the incoming of more transactions. Encblock is capable of processing the transaction before the saturation of the request reaches a peak. The peak throughput stands for the maximal Encblock processing capability, e.g., about 128 transactions per second for the case of 12 CPU 8GB Memory. It indicates the maximal utilization of the resource at the enclave. After this, the processing capability decreases due to the extra requests that hold the computing resources (e.g., multithreads and blockchain transactions). This allows us to control the power grid data rate in a way that prevents saturation and meanwhile maximizes the processing capabilities.

Figure 9 presents the experimental results of the throughput without enclave, e.g., SGX enclave is disabled. Comparing the result with and without the enclave, we can see that the SGX enclave does impact the blockchain transaction performance, due to the enclave overhead caused by additional data encryption, management, and remote attestation. The impact is augmented by the increase in transaction request intensity.

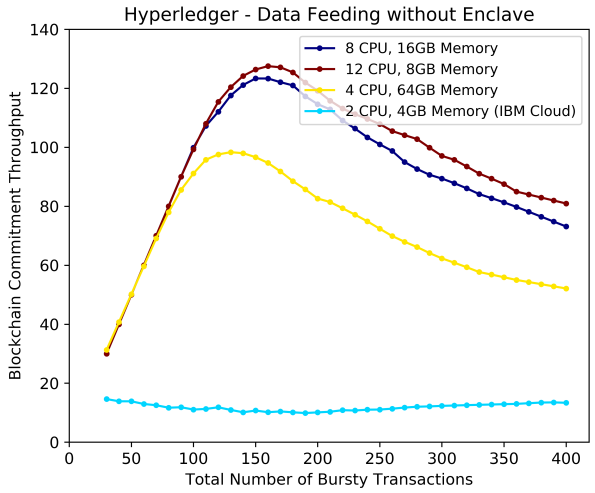

Fig. 9: **Commitment Throughput without Enclave**

IV. Conclusion

The interoperability of grid networks with external networks is not well addressed in the research and industrial efforts. Encblock is the attempt to fill up this gap by means of enclave-based trusted computing. It allows different network domains to be connected in a trusted environment for data sharing and computing without modification of the network infrastructure. For such a trusted environment, a remote attestation protocol is developed to ensure the confidentiality and integrity of the enclave. Further, the proposed Encblock integrates with a blockchain to support power grid business with the use of the grid data. Our experimental results show the performance of the Encblock design.

References

[1] C. Banks et al., "Blockchain for Power Grids," 2019 SoutheastCon, Huntsville, AL, USA, 2019, pp. 1-5, 2019.

[2] Y. Wang et al., "SPDS: A Secure and Auditable Private Data Sharing Scheme for Smart Grid Based on Blockchain," in IEEE Transactions on Industrial Informatics, vol. 17, no. 11, pp. 7688-7699, Nov. 2021.

[3] Imran Makhdoom, Ian Zhou, Mehran Abolhasan, Justin Lipman, Wei Ni, "PrivySharing: A Blockchain-based Framework for Privacy-preserving and Secure Data Sharing in Smart Cities," Computers & Security, Volume 88, 2020, 101653, ISSN 0167-4048.

[4] O. Samuel, N. Javaid, M. Awais, Z. Ahmed, M. Imran and M. Guizani, "A Blockchain Model for Fair Data Sharing in Deregulated Smart Grids," in Proceedding of GLOBECOM, 2019.

[5] Object Management Group, "DDS Secuirty," Version 1.1, 2018, https://www.omg.org/spec/DDS-SECURITY/1.1

[6] Intel, "Intel Software Guard Extensions," Online: https://www.intel.com/content/www/us/en/architecture-and-technology/software-guard-extensions.html.

[7] Intel, "Strengthen Enclave Trust with Attestation," Online: https://www.intel.com/content/www/us/en/developer/tools/software-guard-extensions/attestation-services.html.

[8] A. Khan, A. Laghari, M. Rashid, H. Li, A. Javed, T. Gadekallu, "Artificial Intelligence and Blockchain Technology for Secure Smart Grid and Power Distribution Automation: A State-of-the-Art Review," Sustainable Energy Technologies and Assessments, v. 57, 2023.