ORNL/TM-2022/2527

Oak Ridge National Laboratory Pilot Demonstration of an Attestation and Anomaly Detection Framework using Distributed Ledger Technology for Power Grid Infrastructure

Raymond Borges Hink
Gary Hahn
Aaron Werth
Emilio C. Piesciorovsky
Annabelle Lee
William Monday
Yarom Polsky

August 2022

OAK RIDGE National Laboratory

ORNL IS MANAGED BY UT-BATTELLE LLC FOR THE US DEPARTMENT OF ENERGY

ORNL/TM-2022/2527

Electrification and Energy Infrastructures Division

OAK RIDGE NATIONAL LABORATORY PILOT DEMONSTRATION OF AN ATTESTATION AND ANOMALY DETECTION FRAMEWORK USING DISTRIBUTED LEDGER TECHNOLOGY FOR POWER GRID INFRASTRUCTURE

Raymond C. Borges Hink
Gary Hahn
Aaron Werth
Emilio C. Piesciorovsky
Annabelle Lee
William Monday
Yarom Polsky

August 2022

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831-6283
managed by
UT-BATTELLE LLC
for the
US DEPARTMENT OF ENERGY

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

3L	three line
3LG	three line to ground
AC	alternating current
CA	certificate authority
CID	configured IED description
DERs	distributed energy resources
DLT	distributed ledger technology
DNP	Distributed Network Protocol
EmSense	Emulated Sensor
GOOSE	Generic Object-Oriented Substation Events
HLF	Hyperledger Fabric
HMI	human–machine interface
IED	intelligent electronic device
IoT	Internet of Things
LL	line to line
LLG	line to line ground
LTS	long-term support
OT	operational technology
PTP	Precision Time Protocol
RMS	root mean square
RTAC	real-time automation controller
RX	received
SCADA	Supervisory Control and Data Acquisition
SEL	Schweitzer Engineering Laboratories
SLG	single line to ground
SV	Sampled Values
TLS	Transport Layer Security
TM	total measurements
TPM	Trusted Platform Module
TX	transmitted

EXECUTIVE SUMMARY

This report summarizes the design and pilot demonstration of a software framework, we call Cyber Grid Guard referred to hereinafter as Grid Guard, that was created to provide increased data and device trustworthiness to electric grid devices by leveraging distributed ledger technology (DLT), specifically blockchain. Grid Guard contains a combination of core cryptographic methods such as secure hash algorithm (SHA), asymmetric cryptography, private permissioned blockchain, baselining configuration data, consensus algorithm Raft, and the Hyperledger Fabric framework. The system implements a low-energy, fast, and robust enhancement to system trustworthiness within and across electric grid systems such as substations, control centers, and metering infrastructures.

Blockchain is a distributed database structured to provide a practically unalterable (immutable) timeline of stored transactions. By relying on hashing and the Raft consensus algorithm, if an entity tries to illegitimately alter a record at one instance of the database, the other ledger nodes are not altered. They work to cross-reference each other and easily locate any incorrectly added data and remove them. The bulk raw data are stored in an off-chain storage (outside of the blockchain ledger), and a hash of these baseline data is stored in the blockchain ledger via hashing windows of time series and configuration data, after aggregation and filtering. The bulk off-chain data repository is then considered to be trust-anchored using the hashes stored in the blockchain.

To secure the electric grid test bed devices and data, device configuration baselines were compared with those baselines stored in the ledger. Statistical baselines for device configurations, network communication patterns, and high-speed sensor data are calculated and then stored off-chain, and hashes are stored in the ledger. Measurements such as three-phase voltage and current, frequency, breaker status, protection scheme settings, network configuration settings (and other device configuration artifacts), and network traffic features (packet interarrival times) are compared every minute or at other selected time windows.

During Phase 1 of the Grid Guard DLT project, different DLT technologies were studied, and an assessment was performed on DLT technology vulnerabilities, uses, and key characteristics. Multiple DLT consensus protocols were studied, and the Raft algorithm was selected mainly because of its energy-efficient operation in addition to many other benefits. Also, cryptography, public, private, and permissioned or permissionless systems were assessed. Grid Guard implements a permissioned private DLT. Consensus algorithm selection and choice of DLT implementation depended heavily on the use case. For the use case in Phase 1, parameters were selected to measure performance and existing tools for assessment. Benchmarking was performed theoretically and practically. During Phase 2, hashed transactions/blocks were inserted into the ledger every second.

During Phase 2 of the Grid Guard DLT project, a prototype framework was developed and demonstrated for attestation of critical substation devices and data using precision timing systems that use PTP [1] and IRIG-B protocols on a test bed of operational devices that emulated a distribution substation, control center, and power metering infrastructure using real operational technology (OT). The test bed includes OT devices such as protective relays, a human–machine interface, and power meters. To determine when to collect and compare system and network baselines, an initial examination of an anomaly detection capability to identify malicious manipulation of data streams was conducted. The resulting anomaly detection was demonstrated in a set of experiments and leveraged to trigger device artifact attestation checks. Attestation checks occur against device configuration baselines when compared with the immutable blockchain-stored baselines, which provided a cryptographically supported means by which to store baselines.

The electrical substation-grid test bed was created to test the Grid Guard framework. The test bed emulates the operations of a portion of a power grid and SCADA (Supervisory Control and Data Acquisition) systems as closely as possible. The test bed integrates real protocols, mainly IEC 61850 standard protocols, such as the SV (Sampled Values) and the GOOSE (Generic Object-Oriented Substation Events) protocols. The test bed emulates real power conditions using the OpalRT hardware-in-the-loop device, which can create fault situations that cannot be easily tested on real systems.

The electrical substation-grid test bed was created using real measurement, communication, and protection devices that electrical utilities commonly use. In this test bed, use-case scenarios that could be observed in an operational power grid or electrical substation were simulated. The objective of this research with the testbed was to study the impact of faults and cyber events at an electrical substation with inside (protective relays) and outside (power meters) substation devices, as well as internally and between the control center and the substation equipment and between the control center and the metering equipment.

Ongoing activities for the continued development of the Grid Guard DLT attestation framework include the expansion and testing of the DLT platform to understand optimum throughput and performance of the application. This activity involves additional testing of the implementation on the test bed and the implementation on the Commander test bed at the US Department of Energy's Oak Ridge National Laboratory's Grid Research, Integration Deployment Center.

1 INTRODUCTION

Adverse events in recent decades have impacted electric grids. For example, the 2015 attack on the Ukrainian power grid shut down a large power system via malware that sent commands from the control center after the attacker had compromised computers, such as the human–machine interface (HMI) in the control center [2]. Faults, such as the one leading to the 2003 Northeast blackout, have also been harmful [3]. These events are a major cause of concern given the complexities in the national grid. One cyber event or equipment failure can lead to cascading outages or even further damage to the critical infrastructure needed for society to function. A method of facilitating the correct functioning of the components in the electric grid is to verify that the data and devices can be trusted. Verifying that configuration data on devices have not been illegitimately modified from the last known correct settings can protect the overall system. Specifically, the approach of this work is to detect anomalies and discrepancies in the data being shared between devices when compared with the last known correct baselines. This report documents an implementation of a distributed ledger technology (DLT) framework that relies on Hyperledger Fabric, a project and implementation for blockchain, to achieve this verification and even attestation. The data for this work originate from a test bed emulating electric grid systems.

Grid Guard demonstrates a DLT-based remote attestation framework that uses blockchain-based methods for verifying device and data trustworthiness on the electric grid. Under the US Department of Energy's Oak Ridge National Laboratory's DarkNet initiative. Blockchain (using Hyperledger Fabric) was implemented for achieving device attestation and data integrity within and between grid systems, subsystems and apparatus. Experiments with grid devices such as relays, meters, and human–machine interfaces (HMIs) have demonstrated data verification and device attestation on a scaled-down test bed that mimics real-world grid architectures and topologies.

In this study, the approach was to capture power grid data and device configuration settings (artifacts) to better diagnose and respond to cyber events and/or electrical faults, either malicious or no malicious. The data included sensor commands and values sent over IEC 61850 standard protocols, including GOOSE (Generic Object-Oriented Substation Events) and SV (Sampled Values) protocols. An attestation framework that includes DLT was developed to enable the performance of these functions. The framework consists of a set of DLT nodes on a network. In addition, each DLT node was set to a specific geographical location inside or outside the electrical substation.

DLTs store the data from the network and preserve the data immutably and redundantly across the nodes. The data captured include voltage and current as time series data in a raw form as time-sampled alternating current (AC) signals and root mean square (RMS) values. Other data of importance include the configuration data of devices, such as relays and meters on the power grid. The nodes communicate with one another to establish a consensus of the data. The nodes can also manage the situation when some of the nodes are compromised by a cyberattack or malfunction, perhaps owing to equipment failure.

To test the system, a multilayered test bed was developed that emulated, on a small but complete scale, various interconnected systems and subsystems of the power grid. The test bed consists of four main subsystems: a substation, metering infrastructure, a control center, and an underlying hardware-in-the-loop OpalRT substation circuit emulated model (this produces realistic electrical measurements to support creating realistic baseline models).

1.1 DLT OVERVIEW

DLT encompasses various technologies that implement data storage in the form of a shared ledger. Ledgers are append-only data structures, where data can be added but not removed. The contents of the ledger are distributed among designated nodes within a DLT network. Consensus mechanisms enable the shared ledger to remain consistent across the network in the face of threats such as malicious actors or system faults. Peer-to-peer communication protocols, external to the DLT, enable network nodes and participants to update and share ledger data. To provide the necessary functionality to implement a DLT, these components are typically grouped together and made available as DLT platforms. Included here is a summary of the important features of a DLT.

1.2 PERMISSIONLESS AND PERMISSIONED DLTS

There are two general categories of DLTs—permissionless and permissioned. In a permissionless/public DLT, the network is open and available to anyone to participate in the consensus process that blockchains use to validate transactions and data. There are no administrators to control the DLT or define access requirements. In the research for the electric sector, DLT is mostly used for energy transactions—the buying and selling of energy. These DLTs are permissionless and typically focus on IT systems.

The alternative is a permissioned/private DLT that is not publicly accessible. The DLT can only be accessed by users with permissions, and the users may perform only specific actions assigned by an administrator. User identification and authorization is required prior to accessing the DLT.

1.3 CONSENSUS ALGORITHMS

Consensus is the process by which a network of nodes provides a guaranteed ordering of transactions and validates the content of the block of transactions. Once consensus is reached, the decision is final and cannot be modified or reversed, without detection. There are two classes of consensus: lottery-based and voting-based. Lottery-based algorithms include several of the “proof” algorithms, such as proof-of-work and proof-of-stake. Voting-based algorithms include PBFT (practical byzantine fault tolerance) and crash fault tolerance.

1.4 SMART CONTRACTS

A smart contract creates digital assets; reads or writes transactions; orders transaction proposals; and queries transactions in the ledger. Smart contracts do not operate on data external to the ledger. They operate on the data received as arguments to their functions and the data in the ledger. Any data required by a smart contract must be included in the ledger.

1.5 TRANSACTIONS

Users interact with the ledger by sending transactions. Transactions use smart contract functions to create, update, or query assets in the ledger. The first step involves the sender constructing a transaction proposal, which is signed using the private key and sent to a peer. One or more peers with the endorser role will then inspect the proposal and, if valid, allow the transaction process to continue. If the transaction involves a query, then the peer will simply retrieve the data from the ledger and return it. Otherwise, if the transaction invokes a function that updates the ledger, the transaction will then be executed and returned. For the ledger to be updated, it must be prepared for ordering in a block. The ordered transaction is then subject to final validation by the peers and added to the ledger.

1.6 CRYPTOGRAPHY

Cryptography plays an important role in a DLT, including the functionality of the core data structure and the authentication of users and transactions. The main cryptographic primitives that enable these features include cryptographic hashes for data integrity and public key cryptography for authentication.

Cryptographic hash functions map input data of an arbitrary size to a fixed-size output. The output of these functions cannot be used to obtain the original input data. SHA256 is a commonly used standard cryptographic hash algorithm that outputs a 32-byte (256 bits) value.

Blockchains are a common data structure used in distributed ledgers. A blockchain consists of blocks of data that are linked together (i.e., the chain) using cryptographic hashes. These hashes provide immutability for the blockchain in the sense that any modifications of the data within any linked block will result in the calculation of an invalid hash when verifying the blockchain. This will indicate some type of data alteration that may be malicious or result from a failure.

Public key cryptography involves the use of public-private key pairs. The private key must be kept secure and possessed only by its owner, whereas the public key can be shared with and used by anyone. In a DLT, each transaction is signed with a private key. The transaction is verified with the associated public key and the transaction is authenticated. Also included is data integrity. Any alteration of the transaction will result in an invalid signature verification.

1.7 CYBER GRID GUARD

Trustworthiness of devices and data within the electric grid is under intense scrutiny as the attack surface of these networks has substantially increased. Varying degrees of system and network sophistication exist among the layers and levels of the electric grid. In many cases, different entities (including utilities) own and operate different parts of the grid, from generation to distribution. These factors make the nation's smart grid a heterogeneous and complex infrastructure. Furthermore, vast amounts of distributed energy resources (DERs) are integrated, which are "small, modular electricity-generating or storage technologies that are located close to the load they serve" [4]. All these systems are owned and operated by different entities, and these entities rely on each other and external regulatory organizations to optimize energy delivery. A framework of trust is needed across utilities and DER organizations to operate safely and securely in the face of potential electrical faults/failures and cyberattacks.

With the increased vulnerability and risk that exists for adversarial manipulation of information, data, control signals, and so on transported over various communications topologies (e.g., Wi-Fi, wireless networks, the Internet, long-distance fiber networks), data and device trustworthiness are critical. There is ample opportunity for data modification and remote cyberattacks on grid devices. The two-way exchanges of data/information that need to routinely occur among the advanced/automated metering infrastructure, control centers, energy aggregators, end user energy management system, and grid monitoring/control devices/systems to help optimize grid control also present a potential increased security risk (by allowing more communication than previous one-way paths). Electric grid systems are therefore in need of remote attestation methods that can support data and device integrity using robust methods that can accommodate the various generations of existing software and middleware technologies, hardware/devices, and network configurations on the smart grid.

Grid Guard demonstrates a DLT-based remote attestation framework that uses blockchain-based methods for verifying device and data trustworthiness on the electric grid. Under the US Department of Energy's Oak Ridge National Laboratory's DarkNet initiative, these efforts have continued to develop/enhance a secure and energy-efficient solution for trustworthiness. Blockchain (using Hyperledger Fabric) was

implemented for achieving device attestation and data integrity within and between grid systems, subsystems and apparatus. Experiments with grid devices such as relays, meters, and human–machine interfaces (HMIs) have demonstrated data verification and device attestation on a scaled-down test bed that mimics real-world grid architectures and topologies.

The nodes in the Grid Guard framework are considered crash-fault tolerant, meaning that if a majority of the nodes remain uncompromised, the DLT nodes can establish the true state of the data and be used to compare the current system and network data to validate trustworthiness [5]. To compare baselines, various methods were used. For device configuration artifacts, hash values were compared with predetermined baselines from power meters and protective relays.

Grid Guard leverages Oak Ridge National Laboratory's Center for Alternative Synchronization and Timing to provide robust nanosecond-precision timing [6], and software-based processes to create baselines for remote attestation of devices within and between grid systems such as substations, control centers, and the advanced/automated metering infrastructure. The Grid Guard framework has proven to be useful for providing data integrity as well as attestation of device configurations.

When applied to larger data sets (e.g., waveforms or data from high-fidelity sensors), Grid Guard produces hashes that are stored in the blockchain ledgers. Therefore, it is scalable and less computationally intensive than storing records, and it consumes less energy to function. Grid Guard uses the open-source Hyperledger Fabric (HLF) software to operate a blockchain-based distributed ledger that can provide data integrity and attestation of device configurations such as protection schemes and network and device communication settings.

Because the Grid Guard architecture includes devices that are distributed across various locations, remote attestation is required. By monitoring electrical device network traffic sent via IEC 61850 standard protocols such as GOOSE and SV, remote attestation verification is triggered when potentially malicious events/attacks are detected. To provide data integrity, the blockchain employs sliding time windows to compare statistical grid and network data measurements with previously established baselines that are stored using cryptographic hash functions in the distributed ledger.

Grid Guard's attestation framework research and development was led by the need to strengthen the resilience/security of the nation's grid through increasing trustworthiness of devices and data. The ever-evolving smart grid topology, particularly with the deployment of DERs, and communication methods demand a sophisticated mixture of technologies to ensure security and data integrity. The main purpose for Grid Guard is for it to become a framework that helps ensure the trustworthiness of the data, systems, and devices that keep the nation's grid operating safely, reliably, resiliently, and securely. Grid Guard provides attestation using HLF for data measurements and device artifacts and portrays a more comprehensive grid/device health monitoring alternative to existing SCADA (Supervisory Control and Data Acquisition) implementations.

This report is organized as follows: Section 2 includes a literature review, focusing on remote, data and device attestation. Section 3 includes a description of the Grid Guard design. Section 4 describes Grid Guard implementation. Section 5 includes the test bed architecture used for experimentation to test the Grid Guard framework. Section 6 explains data collection in electric grid networks and systems. Section 7 describes the experiments, and Section 8 describes the results of these experiments. Section 9 provides the conclusions and discusses future work.

2 LITERATURE REVIEW

As summarized previously, Grid Guard focuses on two major areas: DLT and data/device attestation. Devices include relays and meters in power systems, specifically at substations, and potentially microgrids and DER devices. Some research has identified the major developments of blockchain when applied to the power grid. However, few publications address permissioned DLTs for the OT environment in the grid. Grid Guard is a permissioned DLT that is deployed in a utility substation and at a utility control center. Sensor data are received from meters at the substation. Grid Guard also implements a data/device attestation methodology using DLT. The DLT remote attestation framework includes anomaly detection of device data and device configuration.

The following is a summary review of literature that focuses on these two topics, in addition to DLT implementations in the electric grid. Table 1 summarize the reviewed literature for general DLT and attestation. The presented analysis is not intended to be comprehensive and instead identifies the most applicable and/or current documents focused on the DLT and attestation areas.

Table 1. General DLT literature review

Author	Areas	Approach
Foti and Vavalis [7]	Energy sector and power grid	Literature review of the use of blockchain technology in the energy sector and the power grid
Andoni et al. [8]	Energy sector	Overview of the principles of blockchain and a literature review of blockchain solutions for the energy industry. Potential use cases in the energy sector: billing, sales and marketing, trading and markets, automation, smart grid applications, and grid management
Sikeridis et al. [9]	Smart grid protection systems	Scalable adaptive protection platform for distributed systems, and a blockchain-based distributed network architecture to enhance data exchange security among the smart grid protection relays
Kong et al. [10]	Power systems	Blockchain multi-chain framework to better manage and protect measurement data in power systems. Measurements from sensors are mined into blocks
Liang et al. [11]	Grid, meters	Distributed blockchain-based protection framework to enhance the self-defensive capability of modern power systems against cyber attacks. To enhance the robustness and security of the power grid, meters as nodes are deployed in a distributed network
Hang and Kim [12]	Grid, sensors	Integrated IoT platform using blockchain technology to guarantee sensing data integrity
Gao et al. [13]	Grid, consumer data	Blockchain-based solution coupled with smart contracts for creating a tamper-proof system for protecting consumer data recorded and transferred onto the smart grid system

As described, much of the DLT research and literature has focused on transaction processing in the electric grid and for Internet of Things (IoT) systems in general, including usage and billing data associated with customers, DER devices, and electric vehicles. These implementations typically deploy permissionless DLTs in the organization's IT environment. Much of this research involved protecting and securing information so that any tampering is detected. Grid Guard is intended to be deployed in an OT environment and address data and device integrity for substations and linked DER devices. DLT is applicable to environments with distributed processing and limited central management. The objective is to ensure the integrity of the data and devices. Table 2 shows the reviewed literature for remote and data/device attestation.

Table 2. Remote and data/device attestation literature review

Author	Areas	Approach
Mathane and Lakshmi [14]	IoT	Different pragmatic approach to define a common and scalable attestation scheme that all IoT devices within an IoT network can deploy
Moro et al. [15]	Smart cities	DLT attestation system comprises a system for authorization and authentication for individual devices and includes an anomalies detection system based on smart contracts
Bare [16]	Remote attestation	Attestation allows a program to authenticate itself, and remote attestation is a means for one system to make reliable statements about the software it is running to another system. The remote party can then make authorization decisions based on that information
Lee-Thorp [17]	Remote attestation	Assessment of the Trusted Computing Group attestation, specifically how practical the attestation specification is and if it meet the needs of designs that propose to take advantage of trusted computing functionality
Arias et al. [18]	Device attestation	A summary of the basics of device attestation. A summary of attestation approaches is classified based on their functionality and reliability
Jenkins and Smith [19]	Remote attestation	A novel attestation architecture distributed attestation network using blockchain to store and share device information is proposed
Sun et al. [20]	Embedded devices attestation	Attestation method that captures both control-flow and data-only attacks on embedded devices. The attestation is based on Operation Execution Integrity
Guttman et al. [21]	Attestation	Five central principles were identified to guide development of attestation systems: (i) attestation must be able to deliver temporally fresh evidence; (ii) comprehensive information about the target should be accessible; (iii) the target, or its owner, should be able to constrain disclosure of information about the target; (iv) attestation claims should have explicit semantics to allow decisions to depend on several claims; and (v) the underlying attestation mechanism must be trustworthy. An architecture is proposed for attestation that is guided by these principles
Valente et al. [22]	Cyber-physical systems	A different form of attestation is introduced that exploits the physical dynamics of the system
Hardjoni and Smith [23]	Blockchain networks attestation architecture	Recent developments were reviewed toward a standard attestation architecture and evidence conveyance protocols. Explores the applicability and benefits of a standard attestation architecture to blockchain networks
Coker et al. [24]	Remote attestation	Five central principles were proposed to guide development of attestation systems. Proposes an architecture attestation guided by these principles
Jain and Vyas [25]	Remote attestation	A modified version of the IBM TPM (Trusted Platform Module) protocol that is secure was proposed. The report describes the design of the protocol and its analysis
Brasser et al. [26]	Attestation for embedded devices	The issue of prover security was considered, including the verifier impersonation, denial-of-service, and replay attacks. Formulates a new roaming adversary model for this scenario and presents trade-offs in countering this threat
Tpm-2 software community [27]	Remote attestation	The paper describes a method that uses a TPM to validate a system integrity by implementing an attestation protocol
Sfyrakis and Gross [28]	Hardware approaches to emote attestation	The paper focuses on remote attestation schemes that use a hardware device and cryptographic primitives to assist with the attestation of nodes in a network infrastructure
Johnson et al. [29]	Remote attestation in embedded systems	The paper reviews the published remote attestation research works from 2003-2020 and includes classification and analysis and areas for future research
Aman et al. [30]	Hybrid remote attestation for IoT	The paper proposes a remote attestation protocol HAtt, which ensures high availability of IoT devices during the software attestation process. The proposed attestation technique uses a randomized approach to attest different parts of an IoT device's memory. Physical unclonable functions are used to protect the secrets of an IoT device from physical attacks
Banks et al. [31]	Remote attestation	The paper describes and evaluates the state-of-the-art for remote attestation, which covers singular attestation of devices as well as newer research in the area of formally verified RA protocols, swarm attestation, and control-ow attestation

DLT facilitates data and device attestation by storing hashes of the data in the ledger and storing the data outside of the ledger in off-chain storage. The hashes, which are immutable, are used to validate the integrity of the data. Because the DLT Grid Guard system is intended to be implemented in a distributed environment, remote attestation is necessary. Remote attestation includes a verifier that validates data from a prover. There are three types of attestation: hardware-based, software-based, and hybrid. Hardware-based remote attestation leverages physical devices/chips and modules to achieve remote attestation. Software-based remote attestation does not rely on any hardware to perform remote attestation. Hybrid remote attestation includes both hardware and software components. Because many of the devices in the electric grid have limited processing and storage capacity, Grid Guard implements software-based remote attestation.

3 CYBER GRID GUARD DESIGN

This section includes an overview of the remote attestation and anomaly detection approach and methodology. The Grid Guard system is a testing ground for the DLT platform to demonstrate attestation and anomaly detection capabilities for electric grid data and electric grid devices. For grid data, the objective is to ensure that the data are within certain bounds and/or are sent at a standard frequency. If the data fall outside these standard bounds (is anomalous), this may trigger an attestation check for the device. The list of devices includes protective relays, meters, real-time automation controllers (RTACs), and HMIs. Network devices include switches, routers, and firewalls. For devices, the focus is on ensuring the integrity of the configuration data (i.e., artifacts) such as protection scheme settings, network settings, and firmware configuration. The robustness of using crystal oscillator grand master clocks for the DLT time-stamping rather than GPS-based timing ensures the system is protected against GPS spoofing attacks, among other weaknesses related to GPS. Timing is provided by the system clock for the node on which it runs (DLT-5). The system clock is kept in sync using the Linux PTP client [32] running on DLT-5.

3.1 ATTESTATION

Two mutually exclusive parties are involved in an attestation scheme: a verifier (the Verifier Module in the Grid Guard framework), and a prover (the device attempting to prove its trustworthiness). Attestation is performed using a challenge-response mechanism upon the verifier's requests. During the execution of an attestation request, the prover does a measurement of a device (through a middleware application). The verifier receives the measurement and then determines whether the measurement represents a valid device state. The Verifier Module uses the hash of the baseline configuration, saved in the ledger to verify the device integrity of the prover. Measurement data such as current, voltage, and interpacket arrival time are also collected from the various Grid Guard devices through IEC 61850 standard protocols such as GOOSE and SV. Data validation is carried out using statistical baselines on these measurements. Windows of statistical baselines are compared to the previous window. Figure 1 includes a high-level view of the Grid Guard attestation framework and Anomaly Detection Module, which is further detailed in the following sections.

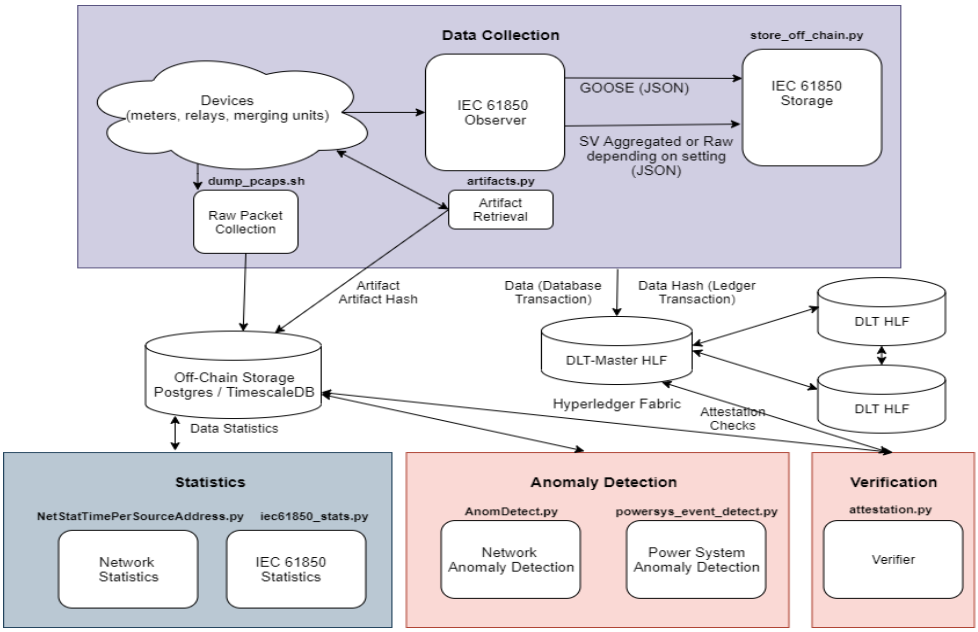


Figure 1. Grid Guard attestation framework and Anomaly Detection Module.

This logical architecture is implemented in the Grid Guard laboratory architecture as illustrated in Figure 2. The Grid Guard Attestation framework collects packets in the network, which come from the relays and smart meters and ultimately derive from sensors. These data include voltage and current data for the three phases associated with the relays. The data are analog when the devices generate the data but are then converted into digital form. The devices package the digital data into packets to be sent over the network. Grid Guard primarily uses IEC 61850 for the main protocol for SCADA network communications. Figure 2 shows the architecture of the communication network and devices used in this study, based on a report by Piesciorovsky et al. [33].

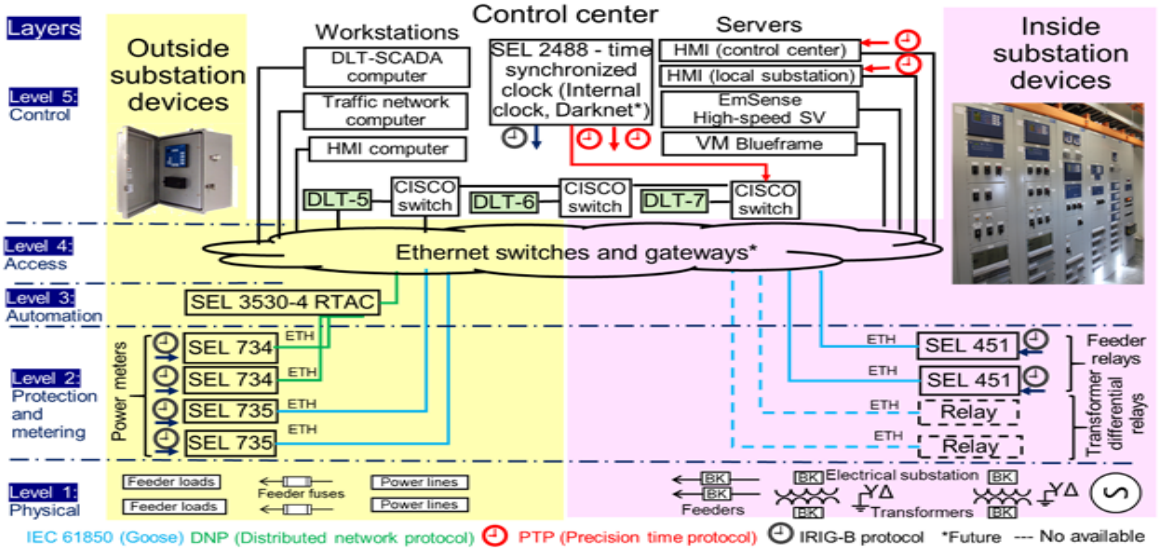


Figure 2. Grid Guard substation test bed implementation.

Several protocols are implemented in the Grid Guard test bed. An overview of these protocols and the Grid Guard test bed devices that generate the data is included here.

3.2 PROTOCOLS

The IEC published the IEC 61850 protocol as an official standard. IEC 61850 is a level 2 protocol in which packets are broadcasted over a network. There are several major types of packets in IEC 61850, including GOOSE and SV.

The GOOSE messages that the Grid Guard relays generate typically contain status information, such as the breaker state for a given relay. Modern relays are considered intelligent electronic devices (IEDs), meaning they are computerized and have networking capability. These relays may also generate other information, including RMS voltage and current. The relays typically send the GOOSE data at lower frequencies than other types of data. Therefore, the time between packets that the relays broadcast is large.

The SV packets are data of raw voltage and current. In contrast to the GOOSE messages, the Grid Guard relays send the SV packets at a very high frequency. These packets carry high-resolution data on the waveforms of voltages and currents associated with the relays.

As described, various devices in the Grid Guard architecture, such as relays and smart meters, produce the data as IEC 61850 packets. Relays used in the Grid Guard test bed are devices that allow a SCADA system to control breakers and gather sensor readings of voltage and current for all three phases. Modern

power systems use AC electricity, which is sinusoidal in nature. The relays receive analog sensor data and sample the sensors at 20 kHz and internally compute RMS values based on the voltage and current. The relays broadcast these values via the network.

3.3 EMSENSE

Because some of the devices in the test bed are limited in the type of IEC 61850 packets they produce, Grid Guard included the development of a device that produces IEC 61850 packets. EmSense (Emulated Sensor) is a device that emulates a high-resolution sensor for a power grid. The device collects raw sensor and voltage data that is derived from Oak Ridge National Laboratory's signature library and packages the data in the form of IEC 61850 SV protocol packets that EmSense broadcasts on the network. In another mode, EmSense generates artificial sinusoidal data that appears as waveforms for voltage and current. EmSense has an internal algorithm for determining the period of a signal based on the data so that the period can be specified as a variable in the IEC 61850 packets. The purpose of EmSense is to allow for experimentation with the Grid Guard architecture where a variety of sensors are represented along with their typical communication traffic. The EmSense device was developed in coordination with the software for receiving and processing the IEC 61850 packets. The receiving software includes a methodology for processing information of high velocity, variety, and volume.

3.4 GRID GUARD DESIGN ASSUMPTIONS

The following are assumptions for the Grid Guard test bed.

General Assumptions

- An asset inventory has been performed for all devices included in the Grid Guard test bed architecture.
    - Data on or sent by a compromised device may or may not be affected by an attacker. Data trustworthiness must therefore be established for all source devices.
        - Measurement and status data being sent from the device cannot be trusted unless the configuration artifact data is successfully verified by the verifier by matching its SHA hash to a known good baseline hash.
- The baseline configuration for devices has not been compromised. Known correct baseline configuration hashes are assumed to be uncompromised. The known correct baseline includes an initial configuration of hardware/software/firmware/settings for all devices.
- Device and network information cannot all be automatically collected for attestation. Some information may have to be collected and entered into the system manually and checked manually. Some data may only be collected by directly connecting to a device or by contacting the vendor.
- Firmware, software, configurations, settings, and tags are periodically checked against the baseline hashes in the Grid Guard DLT.
- The attestation scheme does not include checking updates to device software/firmware prior to implementation in the applicable component.
- The native applications that run on the devices have not been compromised or tampered with and therefore provide a trustworthy baseline. The native applications act as the provers responding with attestation evidence (artifacts of configuration data) when the verifier sends the challenge query.
    - The anomaly detection mechanism detects when a native application has been compromised. The mechanism uses the Grid Guard DLT, which ensures the integrity of the data.

Specific Assumptions for the Test Bed

- The timing system has an independent backup timing source independent from DarkNet and/or the Center for Alternative Synchronization and Timing that can be switched on when connectivity to this system is down. Timing must remain synchronized for all devices.
- Data integrity and message authentication are implemented using cryptographic protocols. A hash-based message authentication code is used for message authentication, and SHA256 is used for data integrity. In addition, HLF includes the TLS (Transport Layer Security) protocol for communications security.
- The anomaly detection framework should detect cyber security attacks, such as man-in-the-middle attacks and message spoofing.

Prerequisites for Test Bed Demonstration

- DLT nodes are located in the substation, metering infrastructure, and control center. As a minimum, three DLT nodes are required to obtain the full benefits of the HLF Raft consensus algorithm. "Raft" is not quite an acronym but its creator attributes the name refers to the algorithm's attributes— i.e., reliable, replicated, redundant, and fault-tolerant. Communication paths are required to link the DLT nodes.
- Asset inventory will be conducted in an automated fashion where possible, with asset discovery tools that leverage vendor asset discovery systems.
    - Integrated methods for asset discovery will be leveraged for IEC 61850.
    - Automated vendor-specific asset discovery tools such as Schweitzer Engineering Laboratories (SEL) Blueframe will be used.
- Assets not identified during the automated asset discovery process must be manually added to the system.
- Asset discovery and enumeration is required prior to implementation of the Grid Guard remote attestation and anomaly detection framework.
- Grid Guard is deployed in a test bed and not in an operational environment. The objective was to demonstrate the implementation of a DLT. Therefore, some cybersecurity devices that are typically deployed in operational environments are not included in the test bed (e.g., firewalls and demilitarized zones). These devices will be considered in the next phase.
- The Blueframe software was used to collect baseline data for the SEL devices. In the next phase, other tools and/or developed software may be used.
- Faults were detected for a subset of the data that was collected.

4 GRID GUARD IMPLEMENTATION

Figure 3 shows the overall data flow of the Grid Guard framework. Network data are read in the test bed from a mirrored port on the switch. All packets of IEC 61850 standard protocols sent by the protective relays, meters, and emulated high-speed sensors are ingested. Statistics are calculated for each type of protocol—GOOSE, SV, and DNP3—and are stored in the off-chain database for baseline comparison purposes. Every 1 min window of data is hashed and stored in the ledger. General network statistics that are protocol-independent are also calculated and stored for baseline comparison. Finally, every device is queried to download its current configuration. Hashes of the network statistics and device configurations are computed and stored in the ledger.

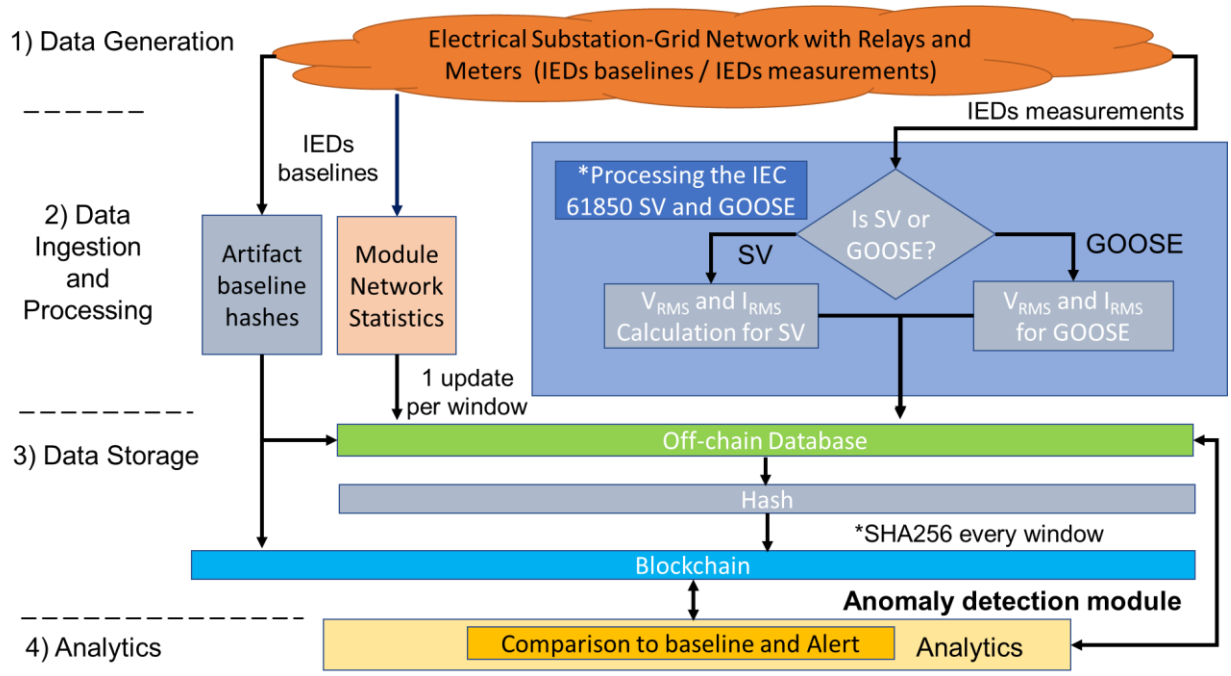


Figure 3. Grid Guard data flow.

4.1 DATA GENERATION, COLLECTION, AND PROCESSING

Data collection occurs at several locations in the framework. The ledger master node (DLT-5 in Figure 2) is connected to a switch mirroring port that captures all network traffic from the substation, metering infrastructure, and control center. Two categories of data are collected: measurement data and configuration (artifact) data. Data in the test bed are collected from the following sources: IEC 61850 packet data from GOOSE and SV messages from the high-speed (high-fidelity sensors); device (e.g., IEDs, smart meters) artifacts, which consists of configuration files; and network data. The collected data are stored in an off-chain database where they can be used for statistical and anomaly analysis after an anomaly is detected. The data are also stored as a hash in a distributed ledger using the HLF DLT platform. This addresses issues with storage of increasing amounts of data in arbitrary sizes by converting them to fixed-size hash values.

Device artifacts are collected using vendor-specific APIs. Currently, the test bed uses SEL's Blueframe API for retrieving configurations and settings from SEL devices. These file archive artifacts are hashed and added to the ledger. Sensor data and statuses are stored off-chain for historical data retention and

long-term analysis and forensics. These data are validated using hashes stored in the ledgers of the DLTs to ensure integrity.

4.1.1 Measurement Data

All IEC 61850 data on the network are captured by first using the software program IEC 61850 Observer. This observer process is implemented using the libiec61850 library [34] and detects any IEC 61850 GOOSE and SV packets on a configured network interface. Packets are (1) received, formatted as JSON (JavaScript Object Notation), and output for other programs to use (GOOSE), or (2) received, aggregated based on the data set values using an RMS calculation, formatted as JSON, and then output (SV). The aggregation phase of the observer for SVs allows the high frequency output of samples (typically 1 kHz or more) by a device such as a real or simulated merging unit to be reduced to a manageable stream of JSON data, which can be consumed by downstream programs and stored. The observer also filters out duplicate packets that result from repeated or heartbeat transmissions. In the case of the SV packets, the observer contains functionality to aggregate the packets.

The following is a summary of the measurement data that are collected in the Grid Guard system:

- Relays:
  - Current magnitude and phase angle
  - Voltage magnitude and phase angle
  - Real power
  - Apparent power
  - Power factor
  - Frequency
  - Time stamp

  Under/over thresholds were calculated for current magnitude, voltage magnitude, and frequency.

- Meters (IEC 61850 GOOSE):
  - Current magnitude and phase angle
  - Voltage magnitude and phase angle
  - Real power
  - Apparent power
  - Power factor
  - Frequency
  - Time stamp

- SV: EmSense
  - Current
  - Voltage

- DNP3 meters:
  - Current magnitude and phase angle
  - Voltage magnitude and phase angle
  - Real power
  - Apparent power
  - Power factor
  - Frequency
  - Time stamp

  Under/over thresholds were calculated for current magnitude.

- All devices on the network:
  - Interarrival packet time
  - Interarrival packet time by source

  Under/over thresholds were calculated for interarrival packet time

Minimum, mean, median, range, and standard deviation statistics were computed over each measurement over 1 min of data.

4.1.2 Configuration Data

A baseline for each selected device, including the relay, smart meter, and network component, was created. This creation occurs on initial system setup or when configuration changes are detected, and/or a Grid Guard system user manually establishes a new baseline. The raw baseline configuration data are stored off-chain, and a hash of the configuration data is stored in the blockchain ledger. The Grid Guard attestation framework triggers the baseline collection process at startup using software to collect the configuration data for each device. The raw data are stored in an off-chain database and the hashed data are stored in the Grid Guard DLT. These configuration data are used for validation in checks when triggered by the anomaly detection. Examples of configuration data include the following:

- Protection schemes (group settings)
- Device configurations
- Network settings (port, frequency, data sent/received)
- Tags for IEC 61850 and similar items for other protocols (e.g., registers for Modbus and identifiers in DNP3)
- Firmware, program settings, and status (reclose enabled/ground enabled, breaker status, long-term settings—GOOSE messages)

The listed artifacts are examples, and additional artifacts may be available. Configuration data availability is determined by the vendor and the vendor's proprietary software either directly or via vendor-specific software and tools for asset discovery and connectivity. In addition, software developed tools may be used. In the Grid Guard architecture, HLF uses the Raft protocol. Figure 4 shows the high-level overview of Grid Guard attestation and anomaly detection framework. The steps of the Grid Guard sequence of events are as follows:

- **S*tep 1.*** The Data Ingestion Module calculates statistics for the network packets, GOOSE, SV, and configuration baselines and then stores them in the off-chain database.
- ***Step 2.*** The Anomaly Detection Module checks statistical windows of data from the network and GOOSE/SV payloads against prior baseline threshold values.
- ***Step 3.*** The Anomaly Detection Module triggers the Verifier Module when an anomalous event is detected.
- ***Step 4.a.*** The Verifier Module initiates remote attestation communication to the middleware application to validate the current configuration settings of the IEDs.
- ***Step 4.b.*** The Verifier Module requests a device measurement, which consists of configuration settings from the various IEDs across the multiple systems (substation, control enter, and metering infrastructure).
- ***Step 5.*** After the Verifier Module retrieves the most recent configuration data from the devices via the middleware application, that data are hashed and compared to previously stored baseline device hashes in the on-chain DLT. If the hashes do not match, then the device integrity has been compromised.

- ***Step 6.*** The off-chain data are continuously trust-anchored to the ledger using SHA256. Off-chain data are also verified every time an anomalous event is detected using prior baseline data. The off-chain storage is reverified at random times, as well.

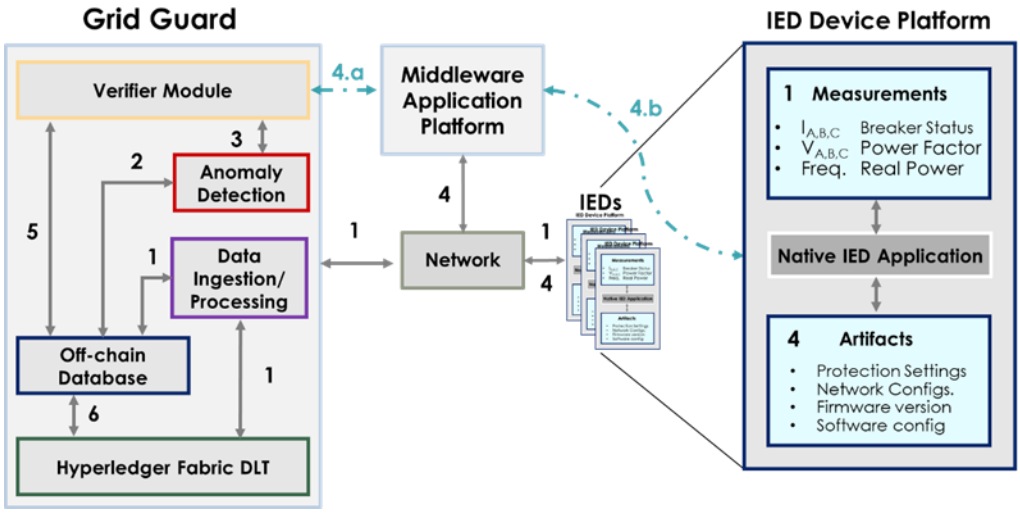


Figure 4. High-level overview of Grid Guard attestation and anomaly detection framework.

4.1.3 Data Storage and DLT Processing

The software module responsible for data storage receives the JSON-formatted IEC 61850 packet data. These data are inserted into the off-chain data database while simultaneously being hashed and stored in the blockchain ledger. The off-chain data store is currently a PostgreSQL instance with the TimescaleDB extension. Postgres is a feature-rich and widely used SQL database that provides support for JSON as a data type and provides efficient handling of time series data using TimescaleDB. This allows for flexibility when implementing and assessing schemas during development. Tables in the database are used for storing IEC 61850 GOOSE and SV packet data, network statistics, artifact data, anomalies and other events, hash window time stamps, and the keys used for accessing ledger data.

A mechanism was developed for a Verifier Module to monitor the generated blocks of data for attestation, including measurement data profiles used to make an initial determination of potential device compromise. This attestation process uses the hashed data from the ledger to provide remote attestation. A data model and metadata for the various devices were developed. Data are to be stored off-chain and on-chain. The highlighted portion of Figure 5 identifies the data collected by the Grid Guard system. There was a disadvantage in the implementation of the attestation process as of the completion of Phase 2 of DarkNet: there was a substantial delay between when the Blue Frame API checks the devices for new artifacts, which then makes the artifacts available for another software such as the Verifier Module to retrieve the artifacts. In the experiments of this work, Blueframe was initially configured to do this check every 5 min but later was configured to every 1 min. If an artifact is changed in the middle of this cycle and an attestation check is made, then the Verifier Module would not see the update until later when the API was updated and another check was made. However, since the completion of Phase 2 of DarkNet, the Verifier Module was enhanced to bypass having to use Blueframe and is therefore far more efficient. As a result, the Verifier Module can see the newest update.

The off-chain storage is the main storage for the raw measurement and configuration data. The blockchain ledger stores hashes of the off-chain data. This process of only storing hashes significantly reduces the amount of storage and speed required to support a large number of devices on a network. High-speed

sensors were simulated by replaying traffic on the network. The sensor data were baselined and hashes were stored in the DLT. If necessary, the sensor data were filtered or aggregated. Waveform data were aggregated into RMS current/voltage data.

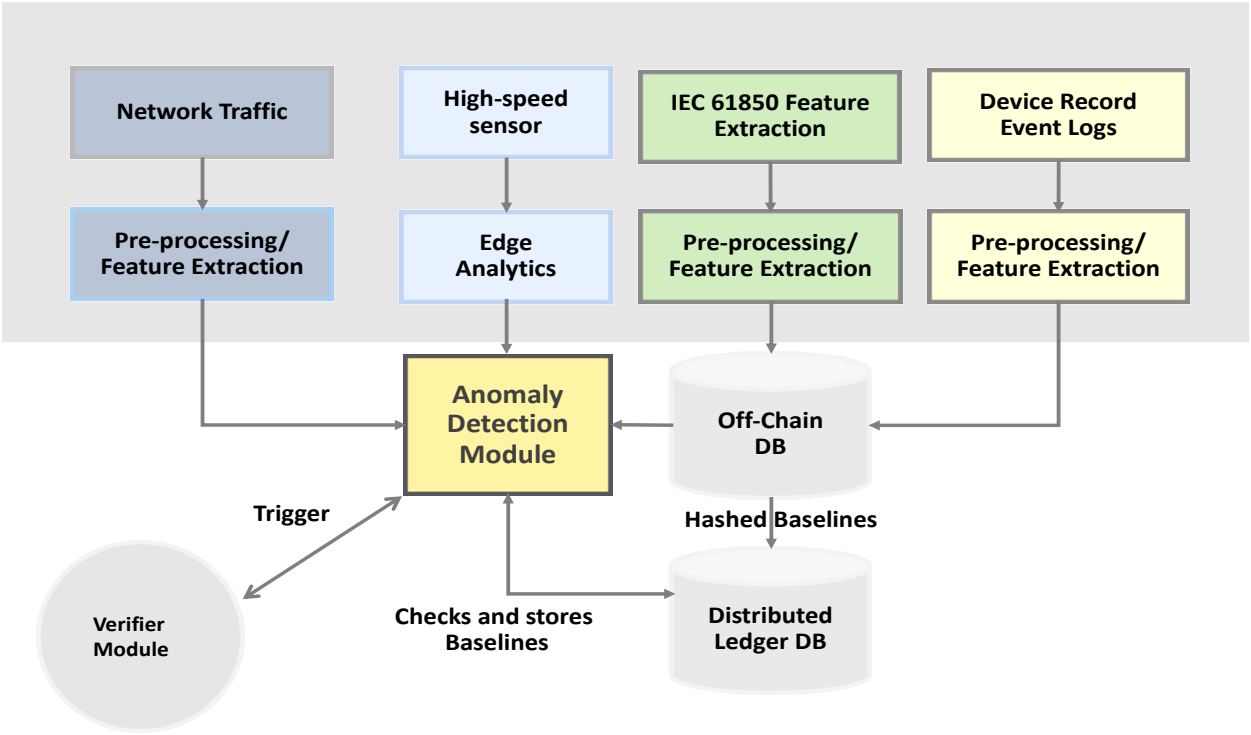


Figure 5. Grid Guard data collection and storage.

The storage of measurement data—which is constantly being transmitted at various frequencies and grows with the number of network devices—presents some potential performance and storage issues. To address these issues, the GOOSE and SV packet data produced by the test bed are first aggregated or filtered, when possible, and then hashed using static window periods. This allows arbitrary amounts of data within a specific window of time to be mapped to a fixed-size value. The hashing is done by combining the window data and using it as input to the SHA256 cryptographic hash function to get a 32-byte hash value.

The Storage Module separates the packets it receives based on the IEC 61850 protocol. The current configuration for each type of IEC 61850 packet (GOOSE and SV) as a different source for the purpose of creating ledger keys. It then initializes a ledger data object by creating a key if necessary (and inserting the key in an off-chain database table for convenience) and initiates a hash window using the time stamp of the first packet it receives. Packet data are appended to the window until the end of the window period has been reached. At this point, the hash is created and sent to the blockchain ledger, and the window data are inserted into the off-chain database.

The window hash is created by joining all the JSON-formatted packet data in the window into a single, compact (no whitespace) UTF-8 byte-string, which is provided as input to a SHA256 hash function. The resulting hash value is converted into a hex string and used along with the time stamps of the first and last packets in the window as the arguments to an update transaction that is sent to the blockchain ledger. The Storage Module inserts all raw packet data within the window into the off-chain database in the appropriate table, along with the start/end time stamps of the hash window used in the update transaction.

There are several considerations related to determining the hash window size. One is the possibility of data being compromised in the period between data collection and hashing. This period starts when the data are received and ends with the transaction containing the window hash is successfully added to the blockchain ledger. Another consideration is ledger parameters such as the block creation time and smart contract implementation. Finally, computational performance, storage, and network latency constraints should also be considered. For these reasons, a 1 min hash window was selected as an acceptable window period that would allow enough data to be collected to reduce ledger storage and transaction processing concerns while also being short enough to reduce risk of not detecting a compromise.

4.1.4 Anomaly Detection Module

The Anomaly Detection Module in the Grid Guard attestation framework performs anomaly detection, busing the data stored in the database of the off-chain network and the hashes in the blockchain ledger. Figure 6 illustrates the anomaly detection framework.

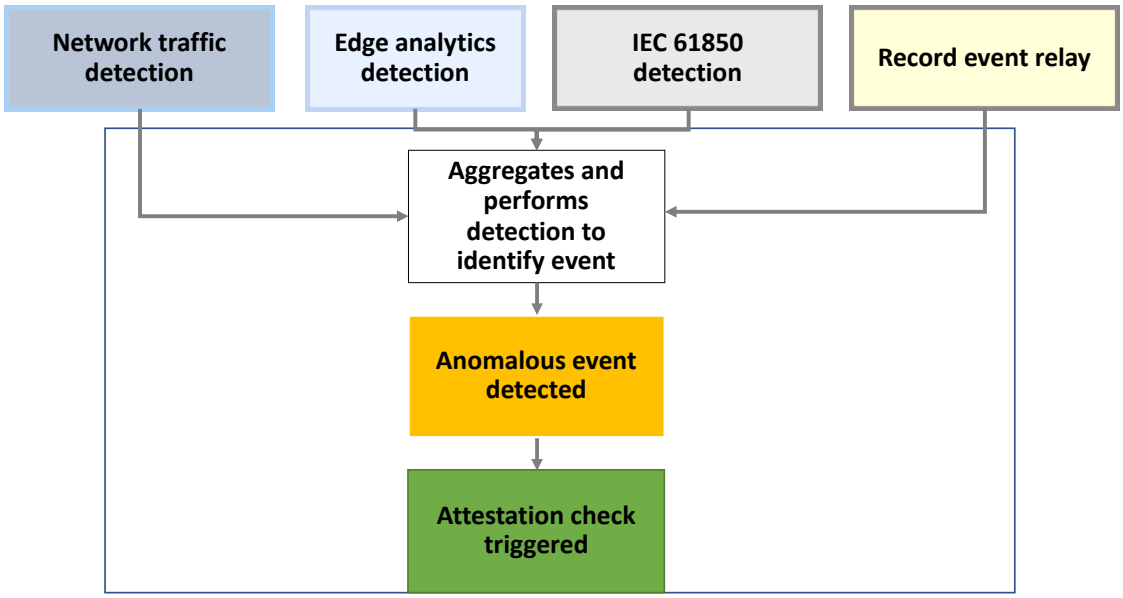


Figure 6. Anomaly detection framework.

DLT-5 is the master node and is used to configure and update the other two DLT nodes. The nodes (specifically the peer and orderer components running on the three DLT nodes) communicate using the Hyperledger gossip protocol. Data are sent as transactions to the DLT-5 node, which upon hashing, ordering, and validation is sent to the other two DLT nodes. Currently, the DLT-5 node is queried when performing attestation checks.

The anomaly detection component of Grid Guard comprises two software modules: the Network Traffic Statistics Module and the Power System Anomalies Module that uses the IEC 61850 data.

Network Traffic Statistics Module: The first module continuously queries network statistics. The network statistics module handles collecting network traffic via packet captures, calculating statistics, and then inserting them into an off-chain network statistics database table. When one of the statistics has exceeded a threshold, a network anomaly event is inserted into the database and a device artifact attestation check is initiated.

Power System Anomalies Module: To retrieve and store the artifacts from the protective relays, power meters, network devices, and IEDs, the vendor-specific API (Blueframe) and Grid Guard–developed software are used. The anomaly detection software only stores a hash of the statistical baseline patterns in the blockchain ledger for comparison. These statistics are useful to establish a profile of behavior for the sensor data and network when running experiments under normal conditions. When the Grid Guard framework has collected new data into the database, these new data may be compared with the statistics to determine if the profile of the new data is similar to or significantly different from the established profile. A second software component collects and stores a window of data of a predetermined length (1 min in pilot demonstration) of data including multiple data streams to establish a statistical baseline for network communication and sensor patterns.

When data or configuration/settings/parameters do not match the baseline, an alert is triggered for that device, indicating the new configuration hash and last known good configuration hash. The source of anomalous data is identified in terms of its IP address and/or MAC address. A system operator then manually needs to verify if the change was authorized, but in future demonstrations, this may be partially automated. Much of this determination on whether an anomaly has occurred is based on threshold checking of the data. When an attestation check event is triggered, the attestation scheme repeats the data verification step to compare the newly acquired data window with the stored baselines from the DLT. Anomalous data does not automatically imply that a cybersecurity compromise has occurred; it could be a result of a device failure or misconfiguration.

During verification, the data may be discarded unless an anomaly is detected, and then the data are stored for post-mortem analysis. In the Grid Guard pilot demonstration, all data were saved regardless of whether an anomaly was detected.

4.2 GRID GUARD HLF IMPLEMENTATION

HLF is an open-source permissioned DLT platform designed for general-purpose usage and was developed by IBM. It is blockchain-based with a modular architecture. The main roles and functions of HLF are split into network components, which can be executed on one node in simple configurations or split across many nodes in larger networks. As a permissioned platform, HLF implements a system for authentication of users and authorization using policies. HLF also supports the use of smart contracts in addition to other features, such as private data for subsets of users.

4.2.1 HLF Components

Running an HLF network involves three main components. Peers are network nodes that maintain a copy of the ledger, which consists of a world state database and the ledger blockchain. Data in the ledger are represented as key-value pairs. The world state database is a key-value database that keeps the current key-value pairs in the ledger to facilitate fast queries of recent data from the peer, and the blockchain is stored as a file [35]. The world state database is configurable, with HLF supporting LevelDB as the default. CouchDB is another supported option, which—as a document database—allows for more advanced querying of ledger values stored as JSON. Peers can also be configured as validator nodes, playing a role in executing smart contract functions to validate transactions.

Orderers serve an important role in keeping the ledger data in a consistent state. Blocks of transactions are ordered and then sent to peers for final approval. Orderers use the Raft consensus protocol to agree on the transaction ordering and are also involved in performing authorization [36].

Certificate authorities (CAs) are the third main component. While optional, CAs are an important part of the public-key infrastructure that is integral to the functioning of the HLF platform in a production

environment. Cryptographic material such as keys and certificates can be generated by various tools and deployed to nodes and users without using a CA, but this becomes burdensome to manage in larger networks. HLF is modular and provides support for other CA platforms in addition to its own CA component.

4.2.2 HLF Permissioned Network

One of the main features of HLF is that it is a permissioned DLT platform. This means that users must authenticate to the platform before being able to use the ledger. Permissioned platforms are typically implemented in use cases in which a small number of organizations or groups control the DLT network and limit access to authorized users. HLF uses a concept of Membership Service Providers to represent the identities of users in the network. These identities may be supported by certificates from the CA(s) (the Grid Guard framework currently does not implement a CA). Policies are another important HLF concept that are used to define what users are authorized to do.

4.2.3 HLF Chaincode

HLF supports the use of smart contracts to implement logic for handling transactions. Smart contracts are often referred to as chaincode in the context of HLF, which is the term used for the packaging and deployment of smart contracts in the network. HLF provides a software development kit for the development of smart contracts using a variety of popular programming languages, namely JavaScript, Java, and Go.

4.2.4 HLF Grid Guard Network Configuration

The test bed HLF network is based on three servers that function as nodes within the HLF DLT network (i.e., DLT nodes). Three nodes were chosen as a starting point for the physical architecture to enable establishing a minimal setup that could handle at least one node failure and still function. Each of the DLT nodes has two Ethernet interfaces. One interface is configured for the 10.0.0.x network, which is used for HLF communication. One of the nodes had its second interface configured for the 192.168.100.x network, which is used by the test bed relay and meter devices. This node is responsible for using that interface to receive data from the power devices for ingestion into the blockchain ledger. Ubuntu Linux 20.04.2 LTS is used as the operating system on each DLT node. Figure 7 shows the HLF DLT network.

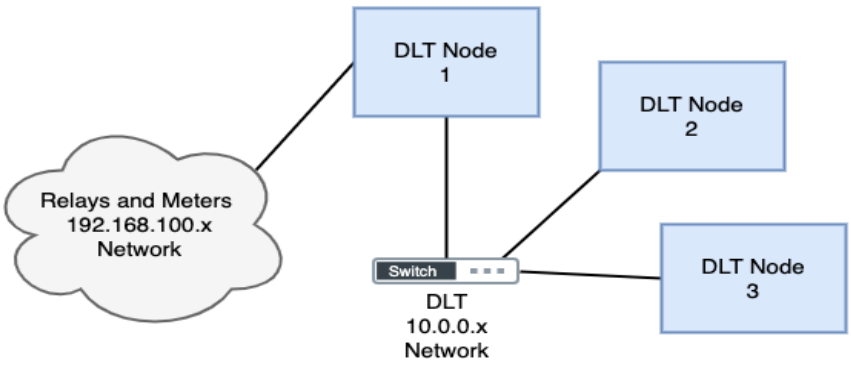


Figure 7. HLF DLT network.

Figure 8 shows the HLF Fabric network using Docker to execute HLF version 2.2 components on each node. These components include the peer, orderer, and CA. Each node is configured to run an endorsing peer—responsible for validating and executing transactions—and an orderer. The ordering service uses Raft consensus. The use of three orderer nodes enables the network to tolerate at least one failure while

maintaining a quorum of Raft nodes [36]. In addition, one node runs the CA component, which is currently unused. The cryptographic material backing the ledger Membership Service Profile identities are generated manually and distributed to other nodes using SSH. Future work would involve using the CA component for certificate management.

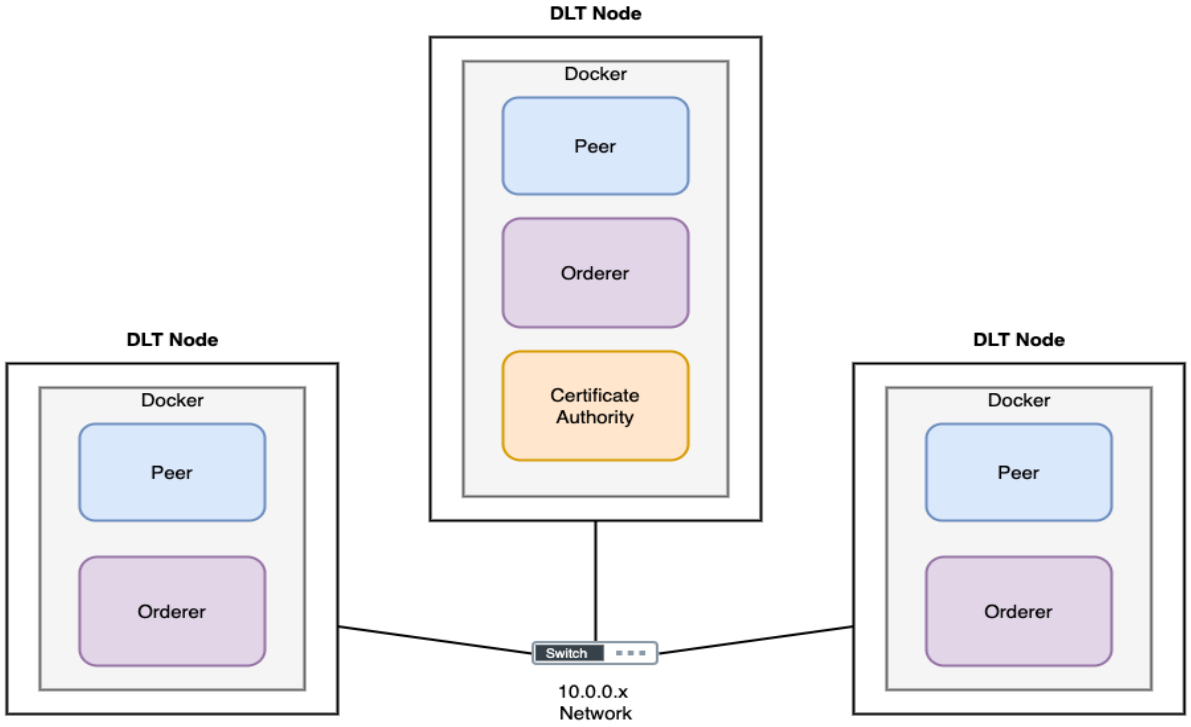


Figure 8. HLF network using Docker.

Docker Swarm is used to deploy, run, and manage the components as Docker containers across the three nodes. Docker Swarm defines two types of nodes for managing and running containers. One of the DLT nodes is designated as the manager node, and the other two serve as worker nodes, with all three running containers [36]. DLT-5 is the master node and DLT-6 and DLT-7 are the slave nodes. The configuration for all the HLF components was created using a Docker-Compose file in YAML [37] format and used as the Docker Swarm service.

Scripts were created for automating various HLF network operations. The main script is responsible for starting or stopping the network. When starting, other scripts are called to handle initialization operations. These include deploying chaincode.

The HLF peer and orderer components are configured using the *core.yaml* file for the peer [38] and the *orderer.yaml* file for the orderer [39]. The settings in these files are overridden in the Docker-Compose file using corresponding environment variables defined by the HLF Docker images. These settings include log levels, listener addresses and ports, and TLS configuration [40]. The Docker-Compose configuration also designates data directories external to the containers for the peer and orderer components on each node. This allows for easier access to the ledger data on the host file system. These directories are typically cleared in between runs for experimentation.

The world state database uses the default LevelDB. This could be configured as CouchDB in the future to take advantage of enhanced JSON document querying as usage requirements become clearer. Although Docker Swarm was chosen as the initial orchestration platform due to its simplicity, future work would

involve examining using Kubernetes as an alternative for the production environment. This platform is increasingly becoming the deployment tool of choice in the HLF community.

4.2.5 HLF Grid Guard Smart Contracts

In HLF, smart contracts define the functions used to send transactions to the ledger. These functions implement the logic involving how data are created, updated, or queried from the ledger and enforces constraints. Smart contracts can be grouped together and deployed as chaincode. Currently, the chaincode for this test bed implements a smart contract for each type of data. The MeasurementHashAudit smart contract handles measurement-related data, such as IEC 61850 GOOSE and SV packet data, and the ArtifactHashAudit smart contract handles device artifacts. Each ledger entry consists of a key to uniquely identify and lookup the associated value, and the value itself. The key can be a single field, or it can be a composite key consisting of multiple fields. The value is always a data object in JSON format for the chaincode being used in this test bed.

The MeasureHashAudit smart contract provides functions for storing and querying windows of hashed measurement data. At least three approaches have been identified for implementing the measurement smart contract. These approaches have various advantages and disadvantages associated and are related to implementation complexity, usage, and impact on the underlying world state database and storage. Although evaluation of these approaches is ongoing as the system is further developed and tested, an initial implementation is described here.

Each entry consists of a composite key with an ID field representing the measurement data source, and a time stamp representing the beginning of the measurement hashes it contains. The time stamp string contains the date and UTC time in ISO 8601 format, which provides chronological ordering of the strings when sorting.

The value is a data object that includes a string field containing a URI or description of the off-chain data source, the number of window hashes contained in the object, and an array of hashes containing all the hash windows for the period beginning at the key's time stamp. Several other fields describe the period represented by the key, including the period length and units. Each element of the hash window array contains a hash and the start and end time stamps for the measurement data. For example, a key-value entry in the ledger representing the 1 min hash windows of all IEC 61850 GOOSE packet data in the test bed for a 24 h period beginning on January 24, 2022, would consist of a composite key with the ID `iec61850_goose` and the time stamp string `2022-01-24T00:00:00Z`. The corresponding data object would then be as follows:

```
{
  "created": "2022-01-22T00:00:00Z",
  "modified": "2022-01-22T00:00:00Z",
  "source": "postgresql://localhost/example_db",
  "period_length": 1,
  "period_units": "minutes",
  "hashes_max": 1440,
  "hashes": [
    {
      "hash":
"afb3a31b79dd1782f5305cbce13c8be82b281b49ca95cde5473363517f7b6b10",
      "start_ts": "2022-01-22T00:00:00Z",
      "end_ts": "2022-01-22T00:00:59Z"
    },
```

```
    {
      "hash":
"4590c52db328dc89ef26039d31047f9ac7f89586371a486081bddb7585fc23f1",
      "start_ts": "2022-01-22T00:01:01Z",
      "end_ts": "2022-01-22T00:01:59Z"
    },
    ...
  ]
}
```

The ArtifactHashAudit smart contract provides functions for storing and querying device artifact data. Each entry consists of a composite key with three fields: the ID of the artifact source, the ID of the artifact belonging to the source for which the hash was generated, and the ISO 8601 time stamp string.

The value is a data object containing a field that points to the off-chain data source for the artifact and another field for the hash value. For example, a device artifact representing an archive of device settings and configuration files provided by the Blueframe API would have a key consisting of its source (device) ID `20411f6b-5d31-4a89-8427-1ee9c2c9afb1`, the artifact ID `81b3e1784769a4ea0bf4e612dfe881e6`, and the time stamp `2022-01-22T15:31:47.158354Z` and the corresponding data object as follows:

```
{
  "source": "postgresql://localhost/example_db",
  "hash":
"50ae4bd89152abec9ccfdccace49348a66e2610f6cae758789bee80228eab047"
}
```

4.3 POWER SYSTEM ONE-LINE DIAGRAM

The power system was based on an electrical substation with two power transformers of 10 MVA, and primary and secondary voltages of 34.5 and 12.47 kV. The electrical grid was a 12.47 kV power system that could be connected as a radial configuration. The electrical grid has power meters and fuses on feeders. The inside substation devices were two SEL-451 protective relays, and the outside substation devices were two SEL-735 and two SEL-734 power meters. Figure 9 shows the one-line diagram of the electrical substation-grid configuration. The electrical substation was based on a sectionalized bus configuration [41]. This arrangement is based on two single bus schemes, each tied together with bus sectionalizing breakers. The sectionalizing breakers may be operated normally open or closed, depending on system requirements. This electrical substation configuration allows the removal from the service a bus fault or breaker failure, to keep service with another breaker and/or bus if it is necessary. The sectionalized bus configuration allows a flexible operation, higher reliability than a single bus scheme, isolation of bus sections for maintenance, and the loss of only part of the substation for a breaker failure or a bus fault [41]. The sectionalized bus configuration is shown in Figure 9A. The electrical grid was connected to the substation feeders through two breakers. The power meters or outside substation devices measured the phase current and phase to neutral voltages of fuse feeders. The electrical grid is shown in Figure 9B. The electrical substation and grid protection schemes were based on using overcurrent relays and fuses, respectively.

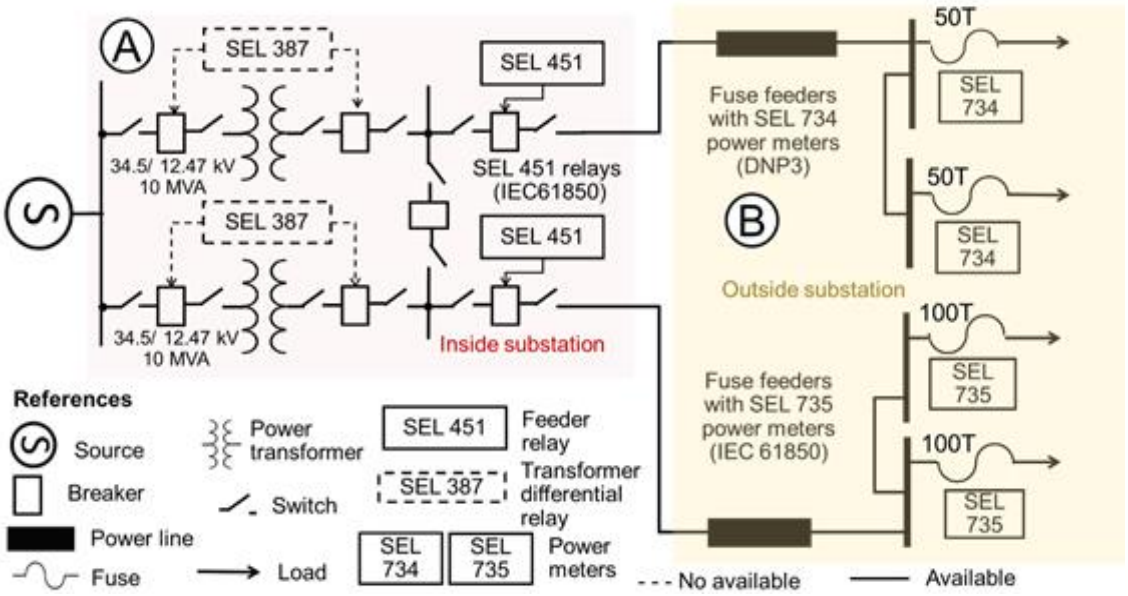


Figure 9. One-line diagram of electrical substation-grid test bed power system [33].

The electrical protection system of the substation and power grid was provided by two substation feeders that have two breakers. Each substation feeder has a SEL-451 relay as a protective device. The breakers were connected to power lines and two power loads as shown in Figure 9. The protection devices of the power loads were fuses, and the currents and voltages of these fuses were measured by the SEL-735 and SEL- 734 power meters. Based on the electrical substation-grid test bed (Figure 9), the radial power system configuration with outside substation devices and maximum load currents is shown in Table 3.

Table 3. Radial power system configuration with outside substation devices and maximum load currents

Power lines	Load feeders	Power meters	Maximum load currents[a] (A)	Type of Medium voltage fuse
Power line 1	1	SEL-734	35	50 T
	2	SEL-734	35	50 T
Power line 2	3	SEL-735	70	100 T
	4	SEL-735	70	100 T

[a]Maximum load currents could be modified during the simulations by setting different power loads

4.4 ELECTRICAL SUBSTATION-GRID TEST BED WORKSTATIONS, EQUIPMENT, AND SOFTWARE

The electrical substation-grid test bed was set at the Advanced Power System Protection lab space at Oak Ridge National Laboratory's Grid Research, Integration Deployment Center. The test bed was a real framework to study the scalability of DLT data architecture and vulnerability assessment in an electrical substation with inside (protective relays) and outside (power meters) devices. The test bed had the real-time simulator rack (A), the inside/outside substation devices rack (B), and the DLT communication rack (C). The main purpose of this test bed was to generate different power system scenarios, such as normal operation and electrical fault events. The electrical substation-grid test bed was created to perform the electrical fault and cyber event tests for inside (protective relays) and outside (power meters) devices with IEC 61850 and/or DNP protocols. Figure 10 shows the electrical substation-grid test bed.

Figure 10. Electrical substation-grid test bed with DLT and inside/outside devices for cyber event detection [33].

In Figure 10, the electrical substation-grid test bed with DLT and inside/outside devices for cyber event detection was formed by the following systems:

1. The electrical substation-grid test bed system that was represented the utility source, power transformers, breakers, power lines, bus, and loads
2. The electrical protection and measurement system that was given by the SEL-451 protective relays (inside substation devices) and SEL-734/SEL-735 power meters (outside substation devices)
3. The time synchronization system given by the timing synchronized sources and time clock displays
4. The communication system with ethernet switches, RTAC, and firewalls
5. The Grid Guard framework with CISCO ethernet switches and DLT devices

The electrical substation-grid test bed has four main workstations (computers) that are connected to the electrical-substation network of the test bed. The host computer (1) that was used to run the tests generated the phase currents and phase to neutral voltages measured by the protective relays and power meters, and breaker pole states measured by the protective relays. The HMI computer (2) was used to set the protective relays and power meters. The HMI was also used to change relay settings during the event tests to study the impact of the event time and collected data at the electrical substation-grid test bed. The DLT-SCADA computer (3) was used to collect the measurements from substation inside (protective relays) and outside (power meters) devices and detected cyber events. The Blueframe computer (4) was focused on retrieving and storing the artifacts from IEDs such as power meters and protective relays. For the Grid Guard project, it is necessary, especially during the simulations, to be able to collect and record relevant data. These data can be used for later analysis. Since the devices (e.g., protective relays, power meters) that produce the data are synchronized with the master clock, the time stamps associated with the data are correlated. Several computers are used in this collection of data. The electrical substation-grid test bed includes six computers located at desks and racks. The four computer workstations set on the desk are shown in Figure 11, and they are the host computer (F), HMI computer (G), traffic network

computer (H) and DLT-SCADA computer (I). In addition, the control center HMI, local HMI, Blueframe, and EmSense high-speed SV servers/computers are in the test bed.

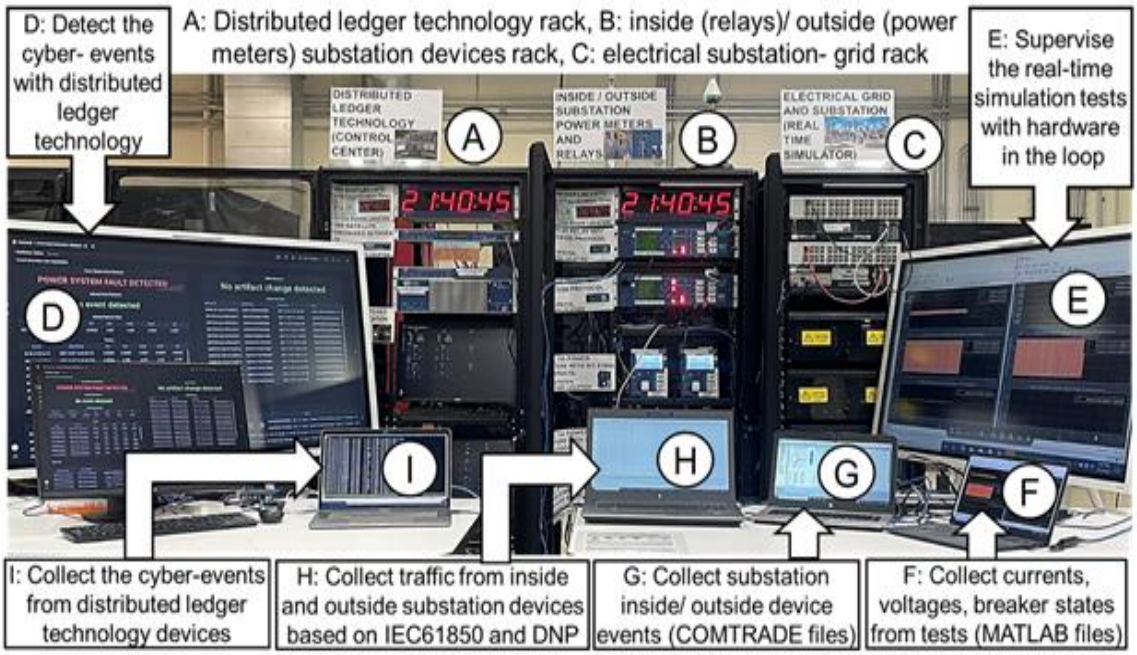


Figure 11. Electrical substation-grid test bed and workstations [33].

This electrical substation-grid test bed was implemented by using a real-time simulator with hardware-in-the-loop. The MATLAB/Simulink software was used to create the electrical substation-grid test bed model. The RT-LAB software was used to create the RT-LAB project configuration and integrate the electrical substation-grid test bed model with the real-time simulator. Also, the RT-LAB software was used to run the power system simulation tests. The AcSELerator Quickset software was used to set the SEL-451 protective relays [42] and SEL-734/SEL-735 power meters [43, 44]. These devices were connected to the HMI computer to measure currents and voltages from SEL protective relays and power meters. The IEC 61850 protocol transmits the GOOSE messages that were configured with the GOOSE data set of protective relays and power meters before being installed. The SEL-451 protective relays and SEL-735 power meters were set with CID files to create the GOOSE messages. The AcSELerator Architect software was used to create and download the IEC 61850 CID files for the SEL-451 protective relays and SEL-735 power meters. The SEL-734 power meters had DNP instead of the IEC 61850 (GOOSE) protocol. The SEL-734 power meters were connected to a remote terminal unit or RTAC. The RTAC polled data from the power meters with DNP and transmitted the measurements from the SEL-734 power meters. The AcSELerator RTAC software was used to create the configuration for the SEL-3530-4 RTAC [45] and SEL-734 power meters. The AcSELerator Diagram Builder software was downloaded to create the future SCADA architecture for the electrical substation-grid test bed. The Wireshark software was used to collect and verify the GOOSE messages from the SEL-451 protective relays and SEL-735 power meters, and DNP messages from the SEL-734 power meters. The Synchrowave software was used to plot and analyze the COMTRADE events from SEL-451 relays, SEL-734 and SEL-735 power meters after running the simulations. Table 3 lists the software applications to build the electrical substation-grid test bed.

Table 3. Software application to build the electrical substation-grid test bed

Software	Applications
MATLAB/Simulink	• To create the electrical substation-grid model
RT-LAB	• To create the RT-LAB project configuration • To run the simulations
AcSELerator Quickset	• To set the SEL-451 protective relays with IEC 61850 • To set the SEL-735 power meters with IEC 61850 • To set the SEL-734 power meters with DNP • To communicate with the SEL protective relays and power meters • To use HMI of SEL protective relays and power meters
AcSELerator Architect	• To create the IEC 61850 CID files for the SEL-451 protective relays • To create the IEC 61850 CID files for the SEL-735 power meters • To download the IEC 61850 CID files into the SEL-451 protective relays • To download the IEC 61850 CID files into the SEL-735 power meters
AcSELerator RTAC	• To create the architecture for the SEL-3530-4 RTAC and SEL-734 power meters • To communicate and download the configuration to SEL-3530-4 RTAC • To create the configuration for the SEL-3555 RTAC with SCADA[a]
AcSELerator Diagram Builder	• To create the SCADA project for the electrical substation-grid[a]
Wireshark	• To collect and verify the GOOSE messages from relays and power meters • To collect and verify the DNP messages from SEL-734 power meters
Blueframe	• To retrieve and store the artifacts from protective relays and power meters
Synchrowave	• To plot and analyze the COMTRADE events from protective relays and power meters

[a]Future tasks

4.5 THE RT-LAB PROJECT FOR THE ELECTRICAL SUBSTATION-GRID TEST BED

The RT-LAB project implementation for the electrical substation-grid test bed included wiring protective relays and power meters with a real time simulator. The MATLAB/Simulink and RT-LAB software were used to create the RT-LAB project configuration. This RT-LAB project configuration was implemented in the host computer that deployed the RT-LAB project configuration in the target computer (real time simulator) and run the simulations with the hardware-in-the-loop. This RT-LAB project configuration was created using two subsystems, one master block (SM_Master) with the simulated electrical

substation-grid test bed circuit, and another block to perform the SC_Console. The SC_Console block checked the simulation tests. The phase currents and phase to neutral voltages from the SEL-451 protective relays and SEL-734/SEL-735 power meters, and the breaker pole states of the electrical substation feeders, were collected during the simulation from the SC_Console block. Figure 12A and 12B show the SM_Master (electrical substation/grid circuit) and SC_Console (scope supervision) of the RT-LAB project configuration.

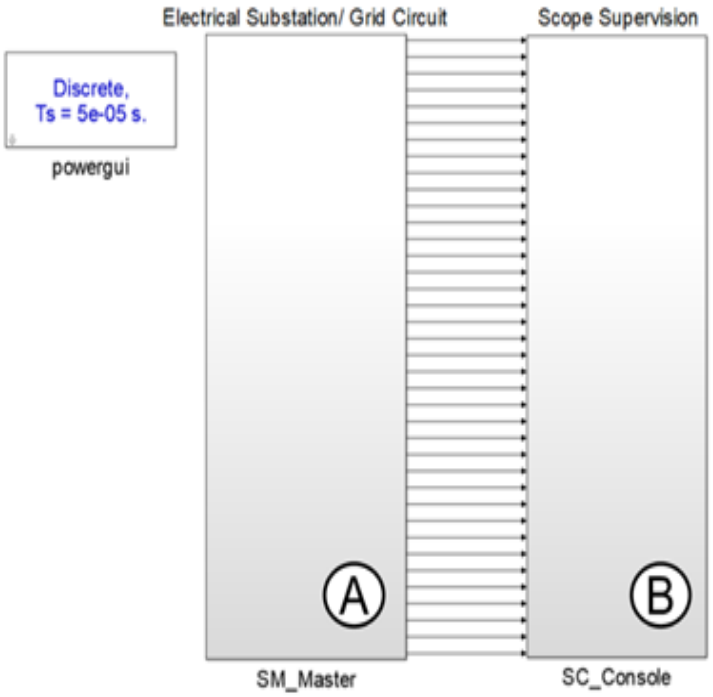


Figure 12. (A) SM_Master and (B) SC_Console subsystems.

Inside the SM_Master block (Figure 12A), the electrical substation-grid test bed circuit (Figure 13) was set. The electrical substation was implemented using a sectionalized bus configuration that has two 34.5/12.47 kV power transformers of 10 MVA connected in parallel. The electrical substation had two breaker feeders of 12.47 kV that were controlled by two SEL-451 protective relays-in-the-loop; therefore, the A, B, and C phase currents and phase to neutral voltages were collected from the breaker feeder locations. Each breaker feeder was connected to a radial power grid, with two 12.47 kV power lines connected to power loads. The protection devices of the power loads were medium voltage fuses. One power line had two power loads with 50 T fuses [36], and the other power line had two power loads with 100 T fuses [46]. The A, B, and C phase currents and phase to neutral voltages for the 50 and 100 T fuses were measured with the SEL-734 and SEL-735 power meters, respectively, based on the one-line diagram of electrical substation-grid test bed in Figure 9. The electrical substation-grid test bed system is shown in Figure 13 and includes the utility source, electrical substation, power lines, and power load feeders. In Figure 13A, the SEL-451 protective relays measured the A, B, and C phase currents and phase to neutral voltages from the breaker feeder locations, as well as the breaker trip/close signals. In Figure 13B, the SEL-734 and SEL-735 power meters measured the A, B, and C phase currents and phase to neutral voltages from the fuse feeders. The fault block (Figure 13D) set the single line to ground (SLG), line to line ground (LLG), line to line (LL), three line to ground (3LG), and three line (3L) faults at any location of the electrical substation-grid. The fault block (Figure 13D) was triggered by an external signal generated by a fault signal circuit (Figure 13C) that set the time to start the fault state in the fault block.

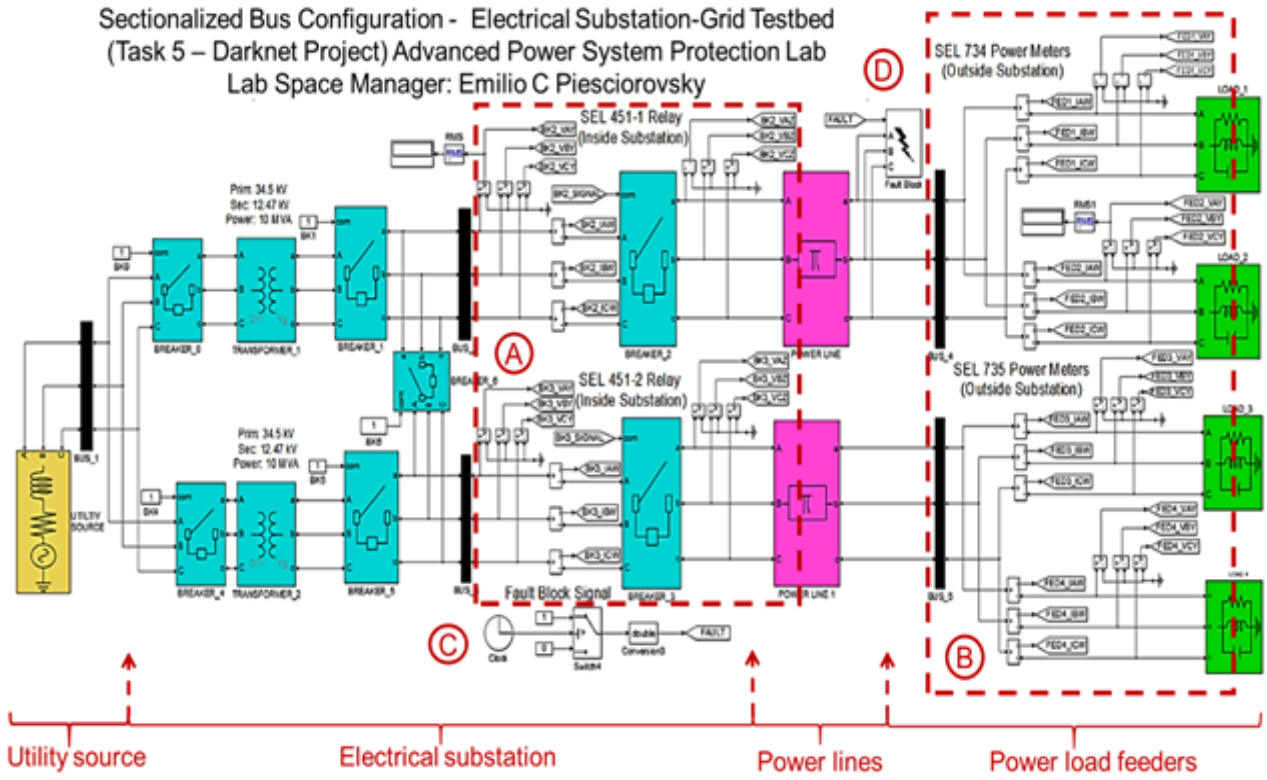


Figure 13. Electrical substation-grid test bed system [33].

Inside the SM_Master block (Figure 12A), the data acquisition circuit (Figure 14) was set with the OpWrite File block that recorded the data from the SEL-451 protective relays and SEL-734 and SEL-735 power meters during the simulation. For the electrical substation breaker feeders provided by the SEL-451 protective relays-in-the-loop, the A, B, and C phase primary currents, phase to neutral voltages, and breaker trip signals were collected (Figure 14A,B). For the power load feeders provided by the SEL-734 and SEL-735 power meters-in-the-loop, the A, B, and C phase primary currents and phase to neutral voltages were collected (Figure 14C,D).

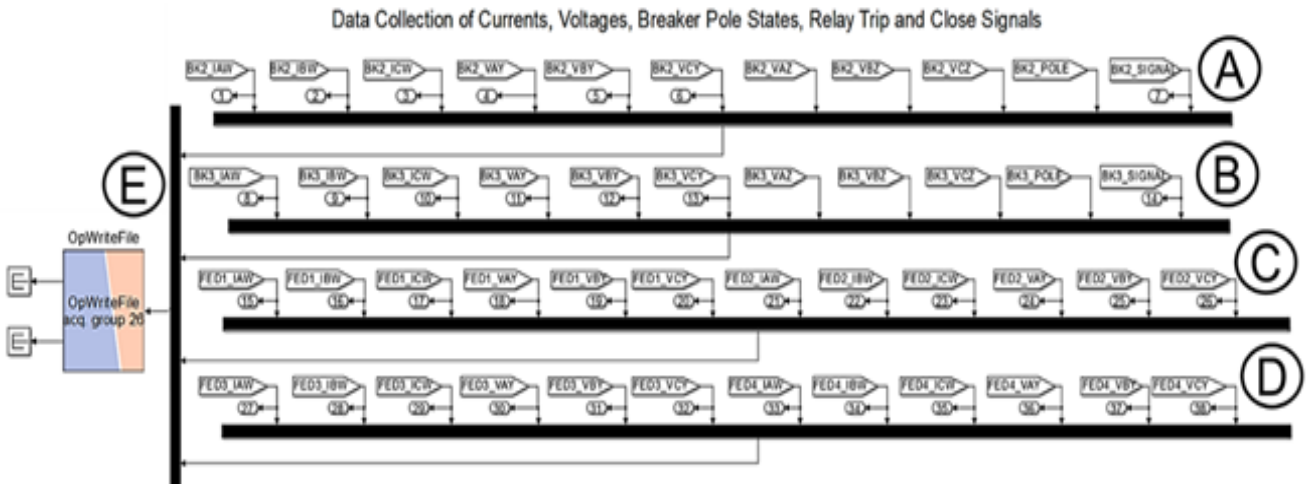


Figure 14. Data acquisition circuit to collect signals from (A, B) the SEL-451 protective relays and (C, D) SEL-734 and SEL-735 power meters with (E) the OpWrite File block.

Inside the SC_console subsystem (Figure 12B), the OpComm block (Figure 15A) and scopes (Figure 15B–E) were set to supervise the simulations. The Opcomm block collected the signals simulated from the S_Master subsystem (Figure 12A). Then, the scopes were open during the simulations to supervise the experiments. The scopes for the electrical substation breaker feeders were provided by the SEL-451 protective relays-in-the-loop that measured the A, B, and C phase primary currents, phase to neutral voltages, and breaker pole state signals (Figure 15B,C). The scopes for the power load feeders provided by the SEL-734 and SEL-735 power meters-in-the-loop measured the A, B, and C phase primary currents and phase to neutral voltages (Figure 15D,E).

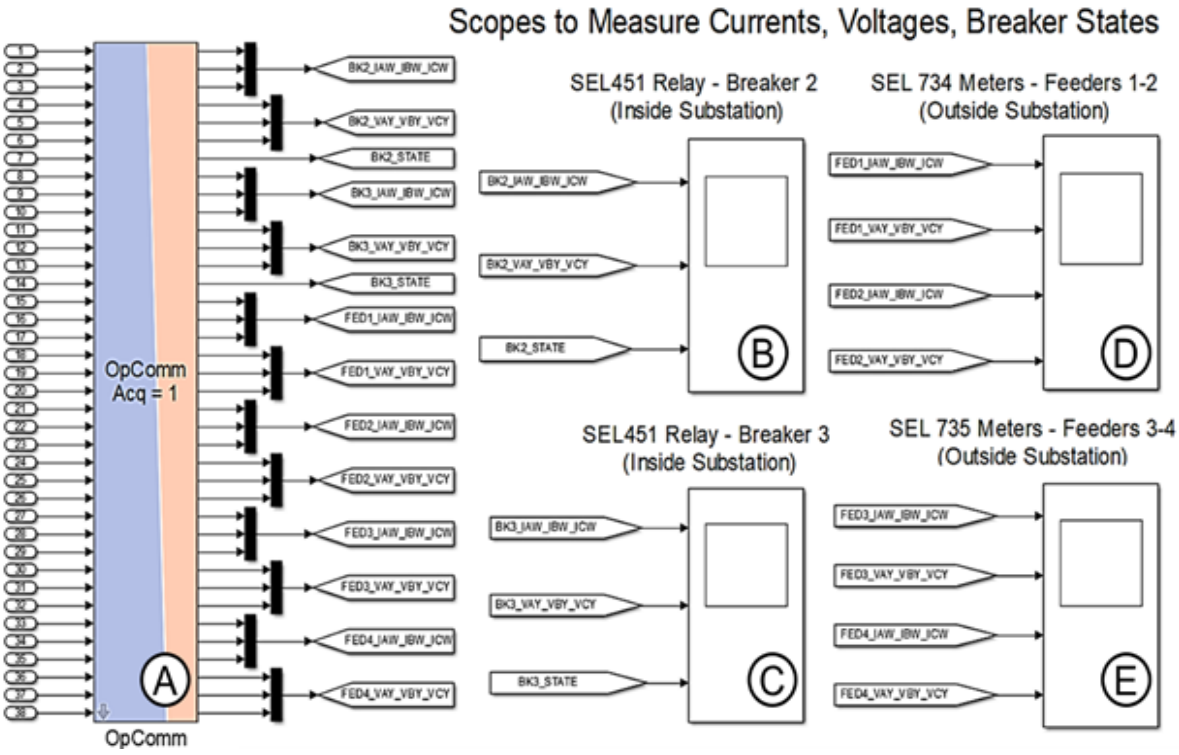


Figure 15. (A) The OpComm block with scopes for (B, C) the SEL-451 protective relays-in-the-loop and (D, E) SEL-734 and SEL-735 power meters-in-the-loop.

Figure 16 shows the signals measured during the simulation of a SLG electrical fault located at the end of the power line (Figure 13D) for one SEL 451 protective relay and two SEL 735 power meters. Scope 1 measured the signals from the SEL-451 protective relay that was connected to the faulted power line. Scope 4 measured the signals from the SEL-735 power meters that were wired at power load feeders of the non-faulted power line. Then, the signals for the SEL-451 relays and SEL-735 power meters could be supervised during the simulation. In this case, the A, B, and C phase primary currents (Figure 16A), phase to neutral voltages (Figure 16B), and breaker pole state signals (Figure 16C) for the SE-451 protective relay were measured at a faulted power line. In addition, the A, B, and C phase primary currents (Figure 16D–F) and phase to neutral voltages (Figure 16E–G) for the SEL-735 power meters were measured at a non-faulted power line.

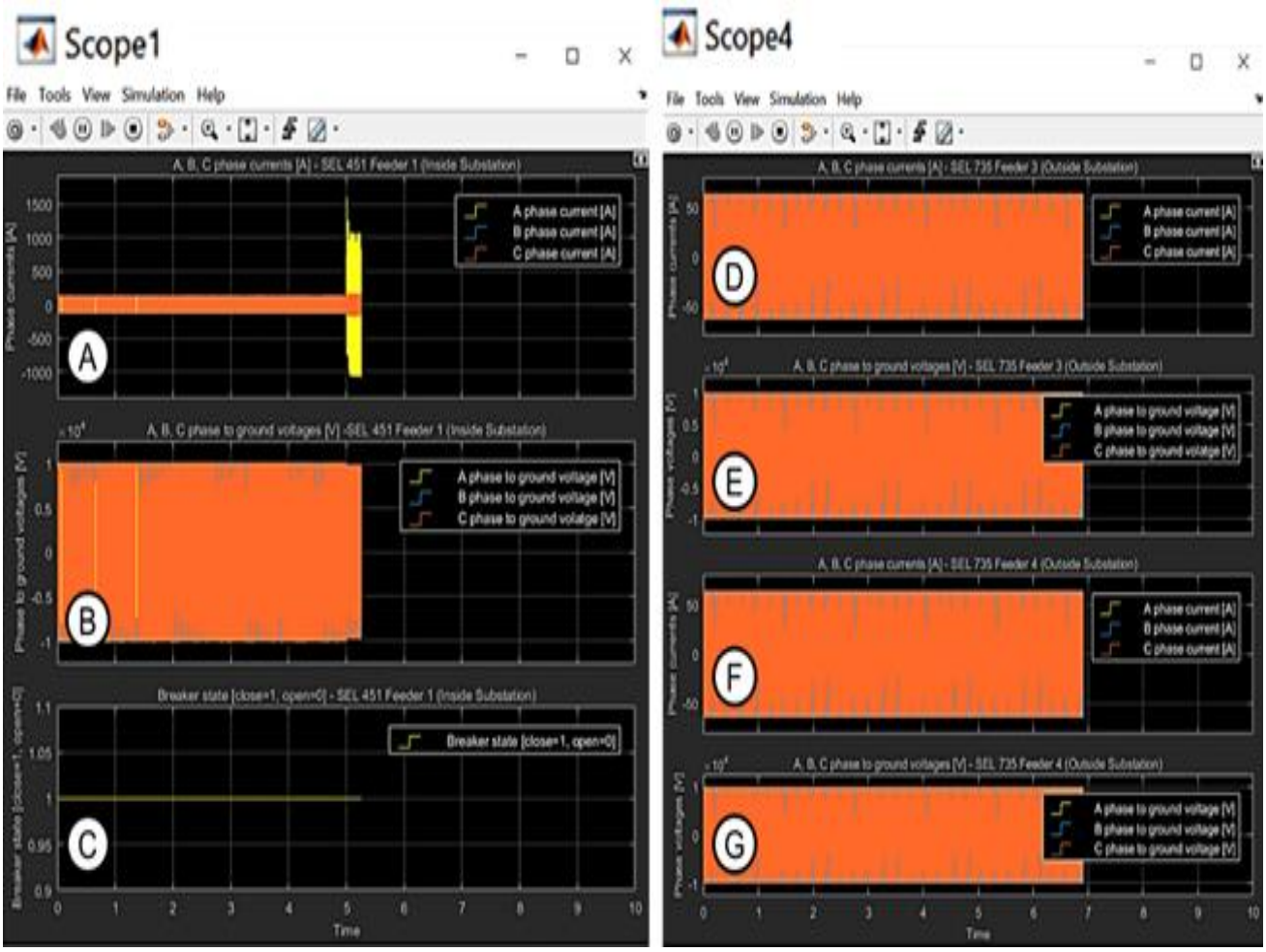


Figure 16. Scope for (A–C) the SEL-451 protective relay and (D–G) the SEL-735 power meters.

4.6 MEASURED FEATURE CATEGORIES AND TOTAL MEASUREMENTS WITH DLT

In the electrical substation-grid test bed, the Grid Guard framework measured the GOOSE messages of two SEL-451 relays and two SEL-735 meters. The measurements from these devices were given by analog signals, digital signals, and time stamps. The measurements that represent the analog signals were given by numerical values that could be estimated based on statistic values (minimum, maximum, mean, range, and standard deviation) such as A, B, and C phase voltages and currents, frequency, and real and reactive power. However, the digital signals were given by Boolean values and represent the breaker states (close or open), and time stamp signals were not estimated based on statistic values. In addition, Table 4 lists the measurements that were collected.

Table 4. Characteristics of the Grid Guard framework at the electrical substation-grid test bed

Network-retrieved data				Data statistics	Device-retrieved data
Protective relay GOOSE data set points with statistics, R_{GDPS} and no statistics, R_{GDNPS}	**Meter GOOSE data set points with statistics, M_{GDPS} and no statistics*, M_{GDNPS}**	**EmSense SV data set points, E_{SVDPS}**	**All devices on network, I_{PTS}**	**Statistic values, S_V**	**IED category settings**
QTY:16, 2*	QTY:16, 1*	QTY: 6	QTY: 2	QTY: 5	QTY:14
-$I_{A,B,C}$ mag. (3) -$I_{A,B,C}$ angle (3) -$V_{A,B,C}$ magnitude(3) -$V_{A,B,C}$ angle (3) -W power (1) -Var power (1) -Power factor (1) -Frequency (1) -Breaker state*(1) -Time stamp*(1)	-$I_{A,B,C}$ mag. (3) -$I_{A,B,C}$ angle (3) -$V_{A,B,C}$ magnitude(3) -$V_{A,B,C}$ angle (3) -W power (1) -Var power (1) -Power factor (1) -Frequency (1) -Time stamp*(1)	- Phase A, B, C RMS current (3) - Phase A, B, C RMS voltage (3)	-Interarrival packet time (1) -Interarrival packet time by source (1)	-Minimum -Maximum -Mean -Range -Standard deviation	-Automation -Port -Group -Breaker -Monitor -Alias -Global -Report -Protection -Front panel -Output -Bay control -Notes -DNP

QTY: quantity.

The measured feature categories (*MFC*) are calculated by Eq. (1),

$$MFC = S_V\,(R_{GDPS} + M_{GDPS} + E_{SVDP} + I_{PTS}) + [R_{GDPNS} + M_{GDPNS}], \quad (1)$$

where *MFC* are the measured feature categories, R_{GDPS} is the relay (GOOSE) data set points with statistics, M_{GDPS} is the meter (GOOSE) data set points with statistics, E_{SVDP} is the EmSense (SV) data set points, I_{PT} is the number of interarrival packet times with statistics, R_{GDPNS} is the relay GOOSE data set points with no statistics, and M_{GDPNS} is the meter GOOSE data set points with no statistics.

The total number of measurements in the Grid Guard framework depends on the number of power meters and protective relays connected to the Grid Guard framework. The total measurements in the Grid Guard framework were calculated by multiplying Eq. (1) by the number of meters and relays using the GOOSE messages at the electrical substation-grid test bed. The total measurements at the Grid Guard framework are given by Eq. (2),

$$TM = S_V\,(N_R\,[R_{GDPS} + E_{SVDP}] + N_M[M_{GDPS} + E_{SVDP}] + I_{PTS}) + [(N_R \times R_{GDPNS}) + (N_M \times M_{GDPNS})], \quad (2)$$

where *TM* is the total measurements, N_R is the number of relays, and N_M is the number of meters.

From Table 4 and Eqs. (1) and (2),

MFC = 5 × (16 + 16 + 6 + 2) + [2+1] = 5 × (40) + [3] = 203 (measured feature categories)
MFC = measured feature categories
R_{GDPS} = relay GOOSE data set points with statistics (16)
R_{GDPNS} = relay GOOSE data set points with no statistics (2)

M_{GDPS} = meter GOOSE data set points with statistics (16)
M_{GDPNS} = meter GOOSE data set points with no statistics (1)
E_{SVDP} = EmSense SV data set points (6)
I_{PTS} = number of interarrival packet times (2)
S_V = 5 statistics values (minimum, maximum, mean, range, and standard deviation)

The total measurements T_M at the Grid Guard framework will depend on the measured feature categories and number of meters N_M and relays N_R. Then, the total measurement at the Grid Guard framework is calculated by Eq. (2).

$$TM = S_V\ (N_R\ [R_{GDPS} + E_{SVDP}] + N_M[M_{GDPS} + E_{SVDP}] + I_{PTS}) + [(N_R \times R_{GDPNS}) + (N_M \times M_{GDPNS})]$$

$TM = 5 \times (\{2 \times 16\} + \{2 \times 16\} + \{2 \times 6\} + \{2 \times 6\} + 2) + [(2 \times 2) + (2 \times 1)]$
$TM = 5 \times (32 + 32 + 12 + 12 + 2) + [4 + 2]$
$TM = 5 \times (90) + [6] = 456$ (total measurements)

4.7 PROCEDURE TO SET THRESHOLD VALUES TO DETECT ELECTRICAL FAULTS

The anomaly detection algorithm was created to detect the electrical faults at the power system implemented in the electrical substation-grid test bed based on finding the maximum and minimum RMS current magnitudes to detect the electrical faults, and verifying possible maximum load RMS current.

The maximum RMS current magnitude was calculated by finding the minimum electrical fault phase RMS current magnitude. Then, the SLG, LLG, LL, and 3LG electrical faults were set at the test bed to measure all electrical fault phase RMS currents and find the minimum electrical fault phase RMS current magnitude. The minimum RMS current magnitude was calculated by implementing a power flow simulation at the electrical substation-grid test bed to detect the maximum load RMS current. Then, the threshold value to detect the electrical faults was selected with a value between the "1.5 × *Irms max load*" that represents the possible maximum RMS phase current at normal operation, and the *"Irms min faultt"* that represents the minimum electrical fault RMS phase current. Figure 17 shows the flowchart to calculate the RMS phase current magnitude threshold to set the DLT algorithm for detecting the electrical fault events at the electrical substation-grid test bed for the feeder relays.

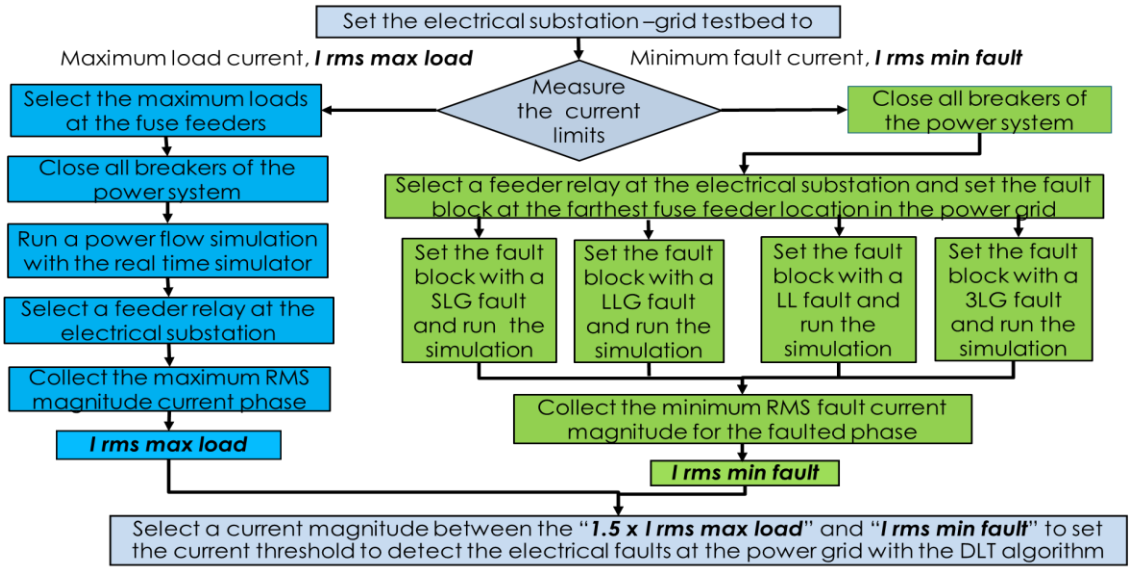


Figure 17. Flowchart to calculate the RMS current magnitude threshold to set the DLT algorithm for detecting the electrical fault events at the substation feeder relays.

Based on the electrical substation-grid test bed [33], the SEL-451 relays located at the electrical substation feeders were considered with a maximum load current of 100 A, and the minimum electrical fault current was 751 A (SLG electrical fault). Then, from Figure 17, the selected RMS current magnitude threshold was 200 A to set the DLT algorithm for detecting the electrical fault events at the substation feeder relays.

5 DATA IN ELECTRIC GRID NETWORKS AND SYSTEMS

As devices transmit and broadcast packets through the network, certain other computers receive the packets and store their data in a central database. The HMI displays the stored data for a user to view. Typically, a conventional SCADA master performs the function of collection, storage, and displaying. However, in this work, an additional technology is used to ensure greater security of the data.

5.1 DATA COLLECTION USING CONVENTIONAL SCADA

In a modern power system, there is a centralized data collection and aggregation system—the SCADA system. This is a conventional system that is commonly used by most utilities. Data are collected in this system and aggregated up incrementally, usually to a central control and monitoring center. Typically, devices—generically remote terminal units and master terminal units—aggregate and forward the data. Within some utilities, these devices are RTACs, and they also manage protocol translation in addition to data aggregation. Finally, the data are visualized and stored in a central database—a historian. The data may then be viewed with a HMI both locally at the substation and in master HMIs at the central utility control center.

5.2 VUNERABILITIES OF CONVENTIONAL SCADA AND OPPORTUNITIES FOR NEW TECHNOLOGY

In the conventional SCADA approach to data collection, certain vulnerabilities exist. For example, historians, which store aggregated system event logs, are generally centralized databases. Therefore, historians are a single point of failure in traditional SCADA. Historians usually offer no data tampering prevention intrinsically and must be protected with additional security devices (e.g., firewalls) and therefore are prone to exploitation or failure from a cyberattack or other harmful events. If the main computer associated with the historian is compromised for example, then the data can be potentially lost or modified, and there is no innate way to detect how the data was tampered with or to reconstruct the data. Several remediations exist to address these vulnerabilities, and blockchain has proven vastly superior for being tamper-resistant and difficult to exploit.

6 EXPERIMENTS AND TESTING

To determine how well the Grid Guard framework could collect and validate data and how it responds to various events, multiple experiments were performed within the test bed under various circumstances including normal conditions, cyber events and electrical fault events. These main categories of events were selected because of their relevance to power systems. To run the experiments on the test bed, the Opal RT hardware-in-the-loop must run a specific simulation of the power system appropriate to each experiment. Another reason to perform these experiments was to collect data for later analysis for improving the system performance. Also, experiments to determine the overall performance of the ledger technology were conducted. It is especially important to determine how well the technology can perform under a high volume of information on the network.

6.1 EXPERIMENTS AND CATEGORIES

In general, the purpose of these experiments is to achieve the attestation of the test bed emulating power grid simulations, which can be grouped into categories including (1) normal load events, (2) cyber events, (3) electrical fault events, and (4) co-occurring cyber and electrical fault events. The cyber events were defined as an attempt by an engineer to set a bad setting in a protective relay by mistake or an attempt by a malicious entity to set an undesirable setting. This may be intentional or unintentional and negatively impacts the electrical infrastructure network or system. Both intentional and unintentional cyber events could have the same results despite their different nature. The experiments demonstrate that the DLT devices can capture the relevant data of the power system from the protective relays inside the electrical substation and the power meters outside the electrical substation. The attestation and data verification could be evaluated satisfactorily by using the Grid Guard framework.

Experiments 1.a and 1.b: The first category (normal load events) consisted of two experiments: Experiment 1.a was performed with a normal MATLAB/Simulink model of the electrical substation-grid and metering infrastructure with no electrical faults simulated. Experiment 1.b was essentially the same but incorporated the EmSense device, which broadcast IEC 61850 SV packets in addition to the GOOSE packets sent by the SEL test bed devices. The experiment was created to provide more variety in the network traffic, especially since the high-fidelity traffic is required for the project.

Experiments 2.a and 2.b: The second category (cyber events) consisted of two experiments involving normal load simulations at the electrical substation-grid test bed that were subjected to various cyber events and phenomena on the power grid and communication network. Experiment 2.a involved the command injection to change the current transformer ratio setting (non-desired situation) of a protective relay located inside the electrical substation. Experiment 2.b involved the command injection to open a feeder breaker (non-desired situation) with a protective relay inside the electrical substation.

Experiments 3.a, 3.b, 3.c, and 3.d: The third category (electrical fault events) involved various types of electrical faults at the electrical substation-grid test bed. These electrical faults were performed at the load feeders where the power meters were located. Then, the protective relays located inside the electrical substation implemented backup protection devices by clearing these electrical faults. All the electrical faults were introduced at 50 s into simulations of 100 s. These experiments were performed for a SLG (3.a), LL (3.b), LLG (3.c) and 3LG (3.d) electrical faults.

Experiment 4.a: The fourth category (cyber and electrical fault events) involved the possibility that a cyber event can occur in tandem with an electrical fault. This experiment addressed the situation and response of the protective relay to a single line to ground electrical fault and an added cyber event. Experiment 4.a consisted of a cyber event of a command injection to change the current transformer ratio setting on the protective relay with an added naturally occurring SLG electrical fault.

All experiments were run at the electrical substation-grid test bed (real-time simulator with hardware-in-the-loop). The experiments were run with the RT-LAB software that integrates the MATLAB/Simulink libraries. The experiments have a time step of 50 µs to provide a real-time simulation for the power grid, and each simulation was set at 100 s for consistency and to compare the data. Table 5 summarizes the experiments performed with the Grid Guard framework.

Table 5. Major categories and experiments performed

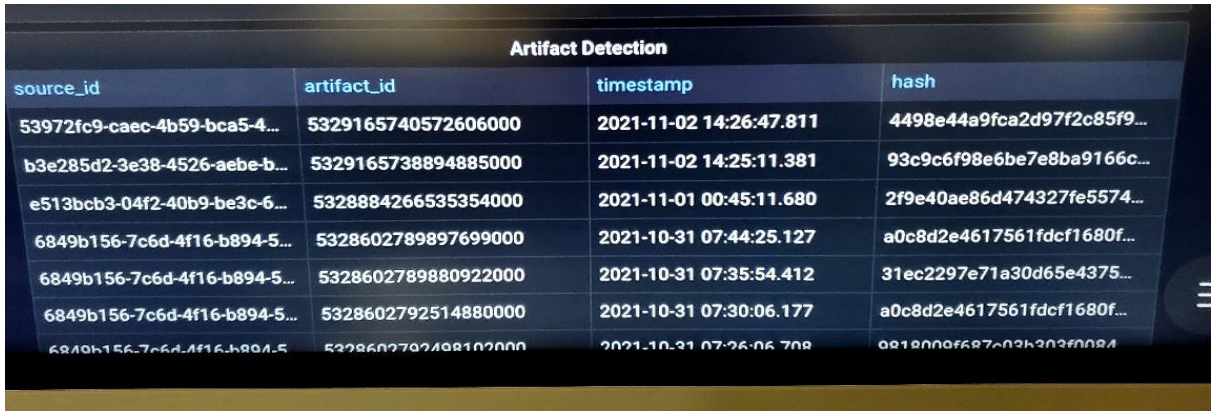
Category	Exp. ID	Description	Simulation	Added activities
1. Normal load events	1.a	Normal conditions with no electrical faults or cyber events	Model with load power flow	N/A
	1.b	Normal conditions with EmSense and no electrical faults or cyber events	Model with load power flow	EmSense running and sending IEC 61850 SV packets
2. Cyber events	2.a	Normal load condition with no electrical faults or cyber events and change of setting variables	Model with load power flow	Command injection (change of the current transformer ratio setting on the protective relay)
	2.b	Normal load condition with no electrical faults or cyber events and opening a breaker	Model with load power flow	Command injection (open breaker)
3. Electrical fault events	3.a	The SLG electrical fault occurs at 50 s into the experiment	Model with the SLG electrical fault (at 50 s)	N/A
	3.b	The LL electrical fault occurs at 50 s into the experiment	Model with the LL electrical fault (at 50 s)	N/A
	3.c	The LLG electrical fault occurs at 50 s into the experiment	Model with the LLG electrical fault (at 50 s)	N/A
	3.d	The 3LG electrical fault occurs at 50 s into the experiment	Model with the 3LG electrical fault (at 50 s)	N/A
4. Cyber and electrical fault events	4.a	Both an electrical fault and a cyber event occur at 100 s into the simulation	Model with the SLG electrical fault (at 50 s)	Command injection (change of the current transformer ratio setting on the protective relay with a SLG electrical fault)

The Anomaly Detection Module triggers the Verifier Module to query the Blueframe computer to retrieve and store the artifacts from IEDs such as protective relays and smart meters to the ledger via device standard protocols and HTTPS. API requests from the Verifier Module stored the configuration artifacts and network traffic in the ledger for anomaly detection and post-mortem forensic analysis.

By using the Anomaly Detection and Verifier Modules, electrical faults and cyber events could be detected and differentiated. A SLG electrical fault was executed in the OpalRT simulation that used a denial-of-service cyber event. Additional fine tuning of parameters will be needed to learn baselines for detecting multiclass events to differentiate the type of anomaly and to determine if the devices are still trustworthy after the events. Figure 18 shows the DLT screen used to detect the power system fault events and artifact changes at the electrical substation-grid test bed [33].

Figure 18. DLT screen to detect power system fault events and artifact changes at the electrical substation-grid test bed [33].

In the DLT screen, the hashes for the configuration files are displayed for the devices at the electrical substation-grid test bed. Data were collected from protective relays and power meters that were verified against stored hashed baselines in the blockchain for trustworthiness. If a difference hash is detected, then an alert is triggered through the graphical interface. The specific setting changed can be identified in post-mortem analysis by reviewing the off-chain storage for the last known correct baseline configuration file. Figure 19 shows the hashes for the configuration files on the devices at the electrical substation-grid test bed.



Artifact Detection

source_id	artifact_id	timestamp	hash
53972fc9-caec-4b59-bca5-4...	5329165740572606000	2021-11-02 14:26:47.811	4498e44a9fca2d97f2c85f9...
b3e285d2-3e38-4526-aebe-b...	5329165738894885000	2021-11-02 14:25:11.381	93c9c6f98e6be7e8ba9166c...
e513bcb3-04f2-40b9-be3c-6...	5328884266535354000	2021-11-01 00:45:11.680	2f9e40ae86d474327fe5574...
6849b156-7c6d-4f16-b894-5...	5328602789897699000	2021-10-31 07:44:25.127	a0c8d2e4617561fdcf1680f...
6849b156-7c6d-4f16-b894-5...	5328602789880922000	2021-10-31 07:35:54.412	31ec2297e71a30d65e4375...
6849b156-7c6d-4f16-b894-5...	5328602792514880000	2021-10-31 07:30:06.177	a0c8d2e4617561fdcf1680f...

Figure 19. Hashes for configuration files on the devices at the electrical substation-grid test bed with DLT.

In the Grid Guard test bed, the power system events based on electrical faults were detected by using the hashes and storing the statistical baselines for RMS values of the phase A, B and C currents of the protective relays over a specified time window. The experiment was conducted with the DLT devices that measured the A, B, and C phase RMS current magnitudes to attempt to detect electrical faults by comparing the pickup RMS current magnitude versus the A, B, and C phase RMS current magnitudes for the feeder protective relay located at the electrical substation. In the DLT algorithm, to detect the power system events for the electrical faults, the DLT pickup RMS current magnitude was set to not trip the fault event detection at the maximum load current and trip the fault event detection at minimum electrical fault current, represented by Eqs. (3) and (4).

$$I_{DLT-fault\ event\ pickup} > I_{rms\ max\ load}, \tag{3}$$

$$I_{DLT-fault\ event\ pickup} \geq I_{rms\ min\ fault}\ , \tag{4}$$

where $I_{DLT\text{-}fault\ event\ pickup}$ is the value greater than the maximum load current $I_{\ rms\ max\ load}$ and minimum electrical fault current $I_{rms\ min\ fault}$ for the feeder protective relay located at the electrical substation. The electrical fault event detection was not tripped for the maximum load RMS current magnitude and was tripped for the phase RMS current magnitude greater or equal to the minimum electrical fault current magnitude.

For the test, a simple manually configured RMS threshold value was determined to be an accurate measure to use. An electrical fault was then triggered in the OpalRT simulation shown Figure 20. Figure 20 shows the screen with voltages, currents, and breaker states of feeder protective relays at the electrical substation.

Figure 20. Screen with voltages, currents, and breaker states of feeder protective relays.

6.2 DATA COLLECTION

For all experiments, data were collected for analysis. The sources of these data include the PCAP file generated from Wireshark, MATLAB file of simulation, record event from the relays, and database entries form the DLT for comparison.

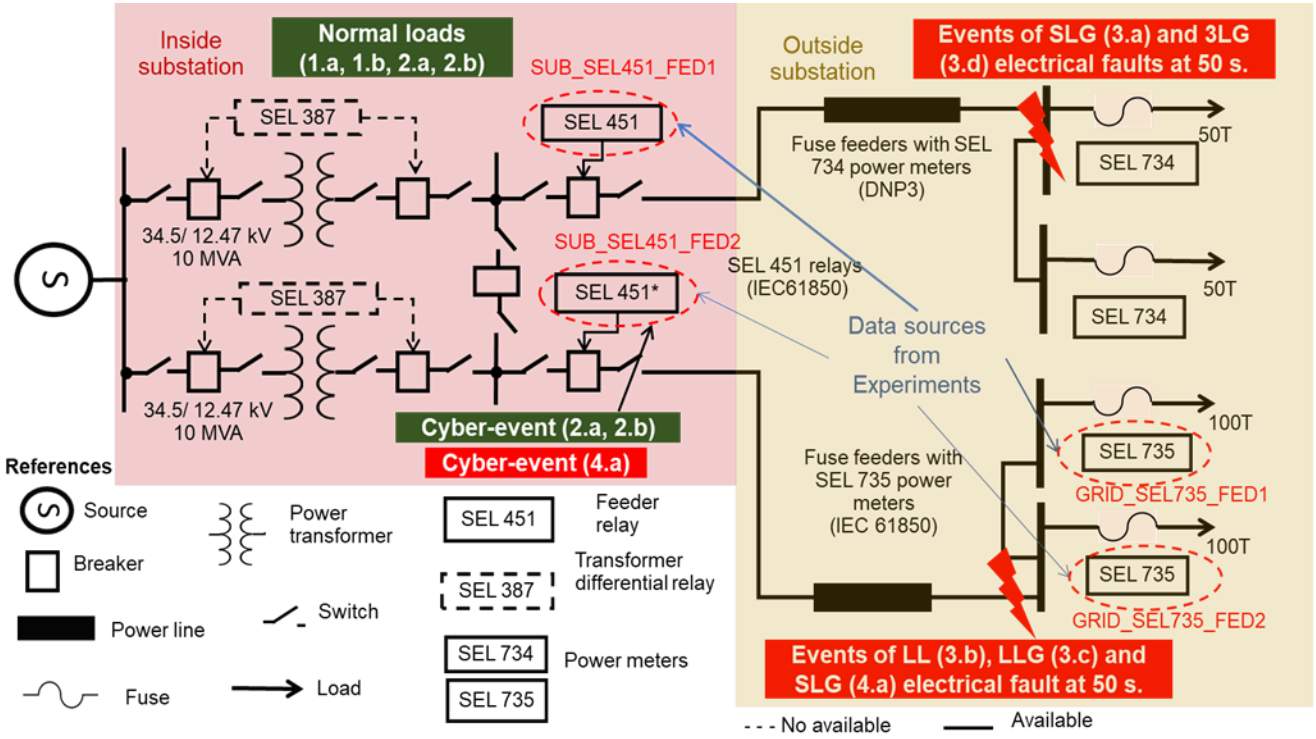
The purpose of the PCAP data was to have a record of the relevant traffic for later analysis or for understanding protocols associated with the relays. The data could be used to create better tools and scripts for changing the state of the breakers and to understand the GOOSE/CV packets and other protocols, such as DNP3. The PCAP data were also useful to aid in software development of the DLTs since replaying the PCAP data emulates some of the experiments that were performed and allow for simple and efficient tests of the software. The MATLAB files served as a record of the raw data in the simulation. The data were useful as ground truth data to be compared with other data in the experiments, such as the data from the relays or the data that the DLTs received. Grid Guard received data from the network and stored this data in off-chain databases for future reference and analysis with anomaly detection.

7 RESULTS OF EXPERIMENTS

The results of the experiments are divided into two main sets: results of experiments under various conditions that include normal, cyber event, and electrical fault scenarios; and results on performance testing of the Grid Guard framework.

7.1 EXPERIMENTAL RESULTS UNDER VARIOUS CONDITIONS

For these experiments, the raw data and the data stored in the database of the master DLT node were collected. Also, each experiment was analyzed to understand the behavior of the voltage and current and why the behavior occurs. The results of these experiments are associated with four devices in the power system: two SEL-451 protective relays and two SEL-735 power meters. Figure 21 shows the electrical substation-grid diagram of test bed and event descriptions for the experiments. The cyber events were applied to the SUB_SEL451_FED2 relay, and the electrical faults (SLG, LL, LLG and 3LG) were applied at 50 s in the 50 and 100 T fuse feeder.



Experiments	Event descriptions
1.a	Normal load
1.b	Normal load with EmSense
2.a	Normal load with cyber-event (change current transformer ratio setting of SEL 451* relay)
2.b	Normal load with cyber-event (open breaker from SEL 451* relay)
3.a	SLG electrical fault at 50T fuse feeder
3.b	LL electrical fault at 100T fuse feeder
3.c	LLG electrical fault at 100T fuse feeder
3.d	3LG electrical fault at 50T fuse feeder
4.a	SLG electrical fault at 100T fuse feeder and cyber-event (change current transformer ratio of SEL 451* relay)

Note: The cyber-events represent an operator that modified the breaker status and/or the relay settings by mistake.

Figure 21. Electrical substation-grid diagram and event descriptions for experiments.

7.1.1 Experiments with Normal Load Events

Experiment 1.a is represented by the normal load case based on the diagram of Figure 21. Figures 22 and 23 shows the data captured versus time during the experiment. Figure 22 shows the currents for the relays (Figure 22-A and B) and meters (Figures 22-C and D), and Figure 23 shows the voltages for the relays (Figure 23-A and B) and meters (Figures 23-C and D). The measured currents and voltages were constant for the duration of the experiment as expected since there were no electrical faults or cyber-events. The DLT master node observed this behavior based on the packets that it received. In Figure 22-C and D, the currents didn't show the same magnitudes for the balanced loads, because of using amplifiers instead of the low-voltage level for the interface of SEL 735 meters [33], however measurement errors up to 5% for general monitoring could be accepted by electrical utilities.

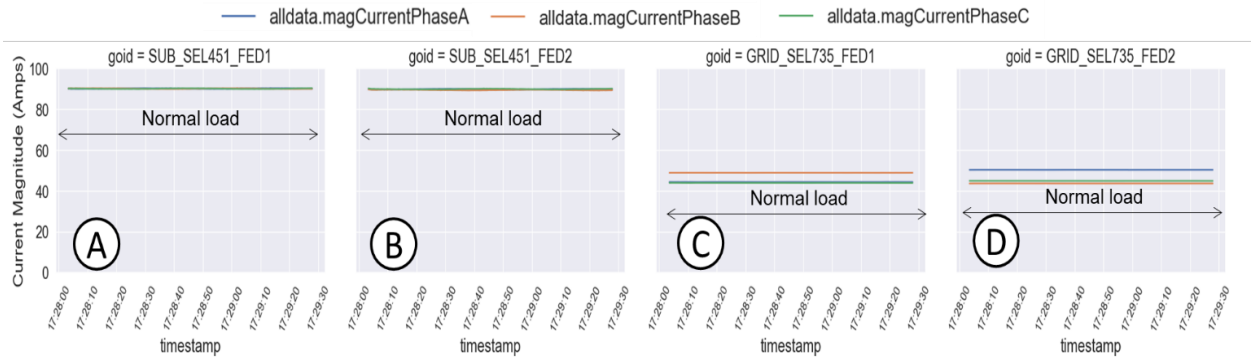


Figure 22. DLT current data from (A, B) the protective relays and (C, D) power meters for Experiment 1.a.

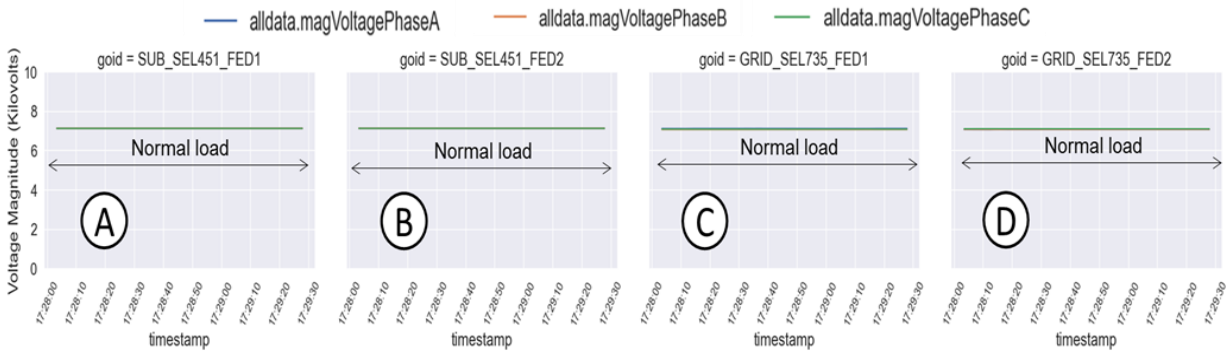


Figure 23. DLT voltage data from (A, B) the protective relays and (C, D) power meters for Experiment 1.a.

Experiment 1.b is represented by the normal load case with the EmSense device, based on the diagram of Figure 21. For the normal case, in which the EmSense device broadcast SV packets over the network, the DLT master node was still able to receive the broadcasted GOOSE packets from the four relevant devices without interference. The packets indicated the normal behavior for the relays and meters, meaning that EmSense did not impact the packets coming from the other devices. Figures 24 and 25 shows the data captured versus time during the experiment. Figure 24 shows the phase currents for the relays (Figure 24-A and B) and meters (Figures 24-C and D), and Figure 25 shows the phase voltages for the relays (Figure 25-A and B) and meters (Figures 25-C and D). The phase currents and voltages were constant for the duration of the experiment as expected since there were no electrical faults or cyber-events. In Figure 24-C and D, the phase currents didn't show the same magnitudes for the balanced loads, because of using amplifiers instead of the low-voltage level interface for the SEL 735 meters [33], errors up to 5% for general monitoring could be accepted by electrical utilities.

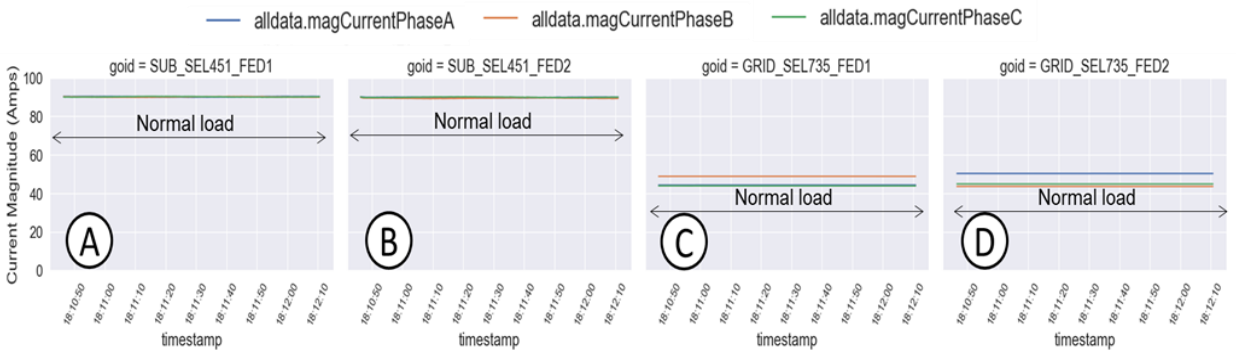


Figure 24. DLT current data from (A, B) the protective relays and (C, D) power meters for Experiment 1.b.

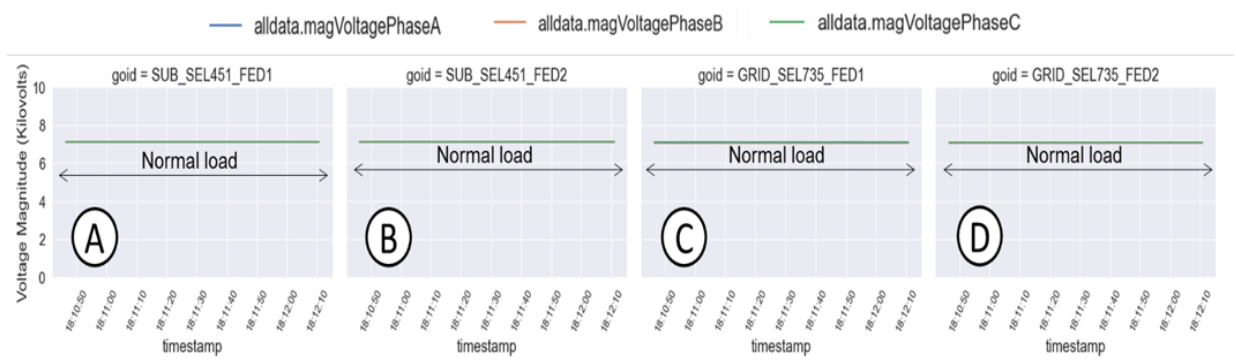


Figure 25. DLT voltage data from (A, B) the protective relays and (C, D) power meters for Experiment 1.b.

7.1.2 Experiments with Cyber Events

Experiment 2.a is represented by the normal load with cyber-event (change of current transformer ratio setting of SEL 451 relay*) based on the diagram of Figure 21. Figures 26 and 27 shows the data captured versus time during the experiment. Figure 26 shows the phase currents for the relays (Figure 26-A and B) and meters (Figures 26-C and D), and Figure 27 shows the phase voltages for the relays (Figure 27-A and B) and meters (Figures 27-C and D). For the command injection of a cyber-event (Figure 26-B), the overall behavior was not affected from the DLT Master's perspective except for the phase currents measured by the "SUB_SEL451_FED2" relay that dropped drastically when the current transformer ratio setting was changed from 80 to 1. In Figure 26-C and D, the phase currents did not show the same magnitudes for the balanced loads, because of using amplifiers instead of the low-voltage interface level for the SEL 735 meters [33], errors up to 5% for general monitoring could be accepted by electrical utilities.

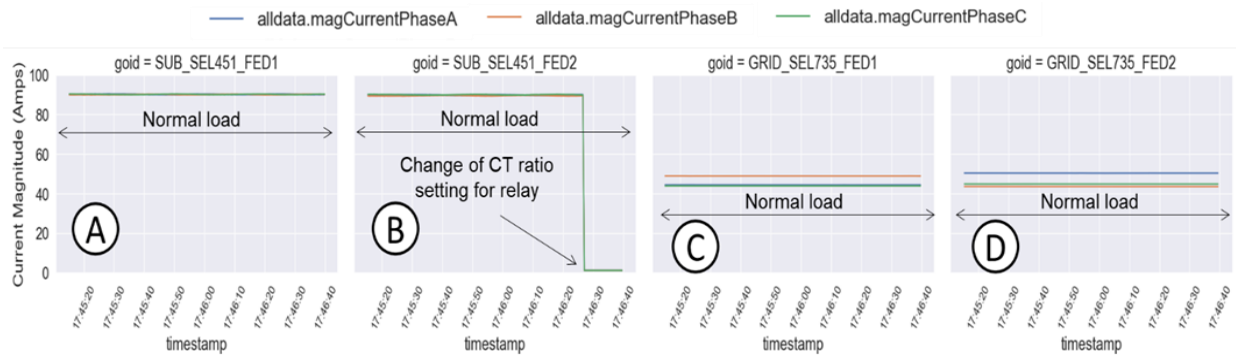


Figure 26. DLT current data from (A, B) the protective relays and (C, D) power meters for Experiment 2.a.

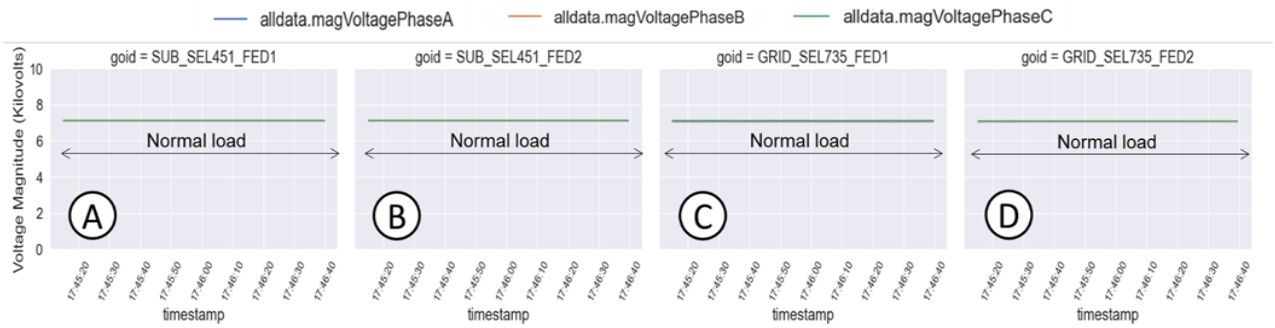


Figure 27. DLT voltage data from (A, B) the protective relays and (C, D) power meters for Experiment 2.a.

Experiment 2.b is represented by the normal load with cyber-event (open breaker from SEL 451 relay*) based on the diagram of Figure 21. Figures 28 and 29 shows the data captured versus time during the experiment. Figure 28 shows the phase currents for the relays (Figure 28-A and B) and meters (Figures 28-C and D), and Figure 29 shows the phase voltages for the relays (Figure 29-A and B) and meters (Figures 29-C and D). For the cyber-event case when the breaker was opened, the phase currents dropped to zero (Figure 28-B), and the nominal voltages (Figure 29-B) were measured for the "SUB_SEL451_FED2" relay, which was approximately at 50 s from starting the simulation. The DLT Master Node observed the behavior for the measured phase currents and voltages of the "SUB_SEL451_FED2" relay. In addition, when the breaker was opened for the "SUB_SEL451_FED2" relay, the phase currents (Figure 28-C and D) and voltages (Figure 29-C and D) of the SEL 735 meters decreased up to zero. It was because the SEL 735 meters were in the same circuit path of the "SUB_SEL451_FED2" relay. In Figure 28-C and D, the phase currents did not show the same magnitudes for the balanced loads, because of using amplifiers instead of the low-voltage level interface for the SEL 735 meters [33], errors up to 5% for general monitoring could be accepted by electrical utilities.

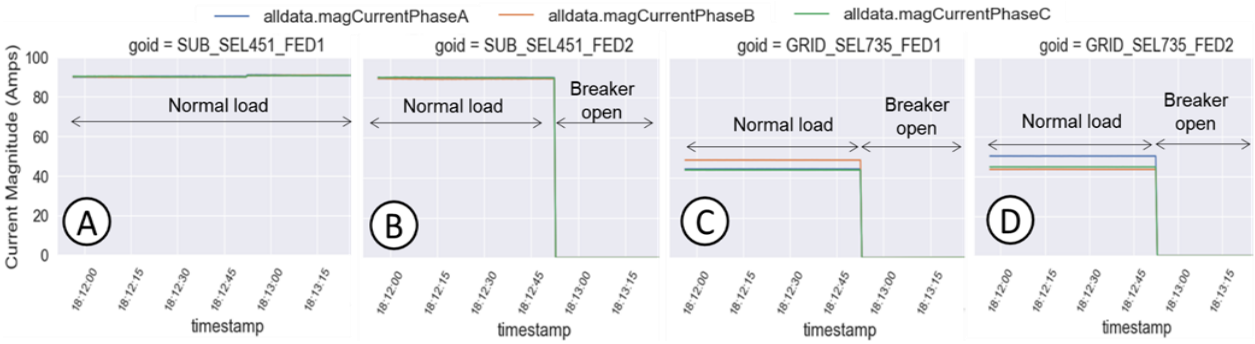


Figure 28. DLT current data from (A, B) the protective relays and (C, D) power meters for Experiment 2.b.

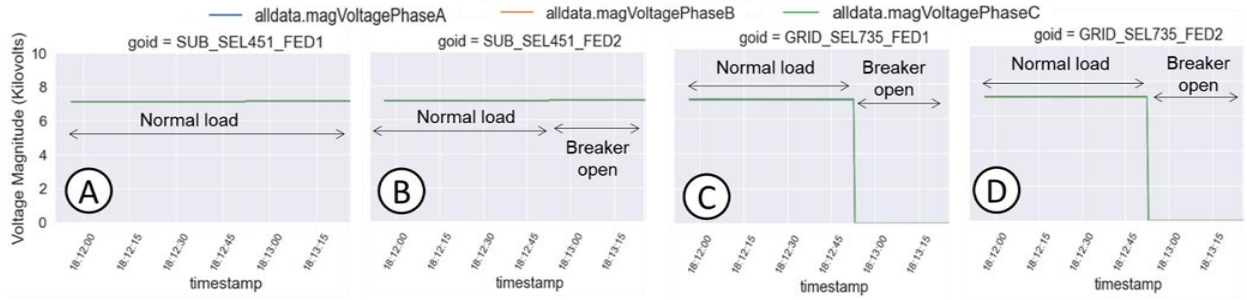


Figure 29. DLT voltage data from (A, B) the protective relays and (C, D) power meters for Experiment 2.b.

7.1.3 Experiments with Electrical Fault Events

Experiment 3.a is represented by the SLG electrical fault at the 50 T fuse feeder based on the diagram of Figure 21. During this SLG electrical fault affecting phase A, the DLT master node observed a significant increase in the current of phase A for the "SUB_SEL451_FED1" relay (Figure 30-A). This situation is because the phase A was grounded at 50 T fuse feeder bus. Once the phase A current increased at the fault state (Figure 30-A), the "SUB_SEL451_FED1" relay detected it, and the relay opened the breaker. Then, after the SLG electrical fault was cleared at the post-fault state, the phase currents dropped to zero (Figure 30-A), and the nominal phase voltages were measured (Figure 31-A). The electrical circuit path that was not affected directly by the SLG electrical measured a short-time disturbance for the phase currents (Figure 30-B, C and D) and voltages (Figure 31-B, C and D).

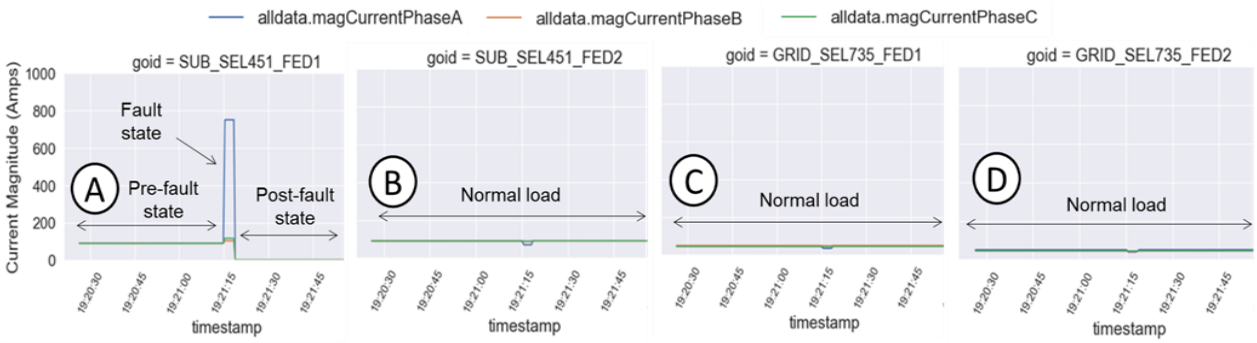


Figure 30. DLT current data from (A, B) the protective relays and (C, D) power meters for Experiment 3.a.

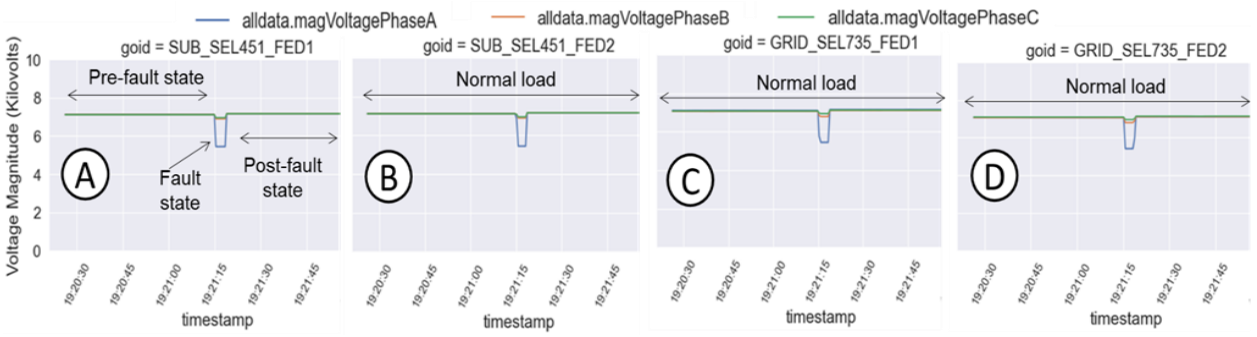


Figure 31. DLT voltage data from (A, B) the protective relays and (C, D) power meters for Experiment 3.a.

Experiment 3.b is represented by the LL electrical fault at the 100 T fuse feeder based on the diagram of Figure 21. During this LL electrical fault affecting phase A and B, the DLT master node observed a significant increase in the current of phase A and B for the "SUB_SEL451_FED2" relay (Figure 32-B). This situation is because the phase A and B were faulted (without grounding) at the 100 T fuse feeder bus. Once the phase A and B current increased at the fault state (Figure 32-B), the "SUB_SEL451_FED2"relay detected it, and the relay opened the breaker. Then, after the LL electrical fault was cleared by the breaker at the post fault state, the measured phase currents from the "SUB_SEL451_FED2" relay dropped to zero (Figure 32-B). However, the nominal phase voltages were measured at the post-fault state (Figure 33-B). When the breaker was tripped by the "SUB_SEL451_FED2" relay, the SEL 735 meters had shown how the phase currents (Figure 32-C and D) and voltages (Figure 33-C and D) dropped to zero. The electrical circuit path that was not affected directly for the LL electrical fault measured a short-time disturbance for the phase currents (Figure 32-A) and voltages (Figure 33-A).

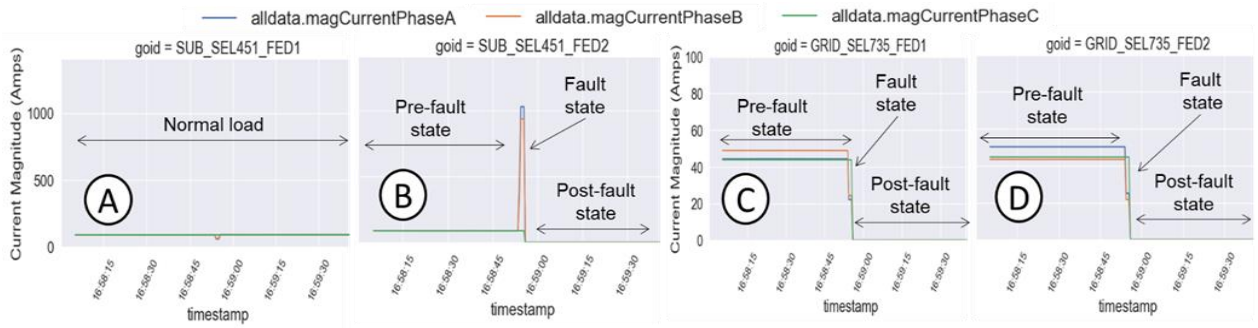


Figure 32. DLT current data from (A, B) the protective relays and (C, D) power meters for Experiment 3.b.

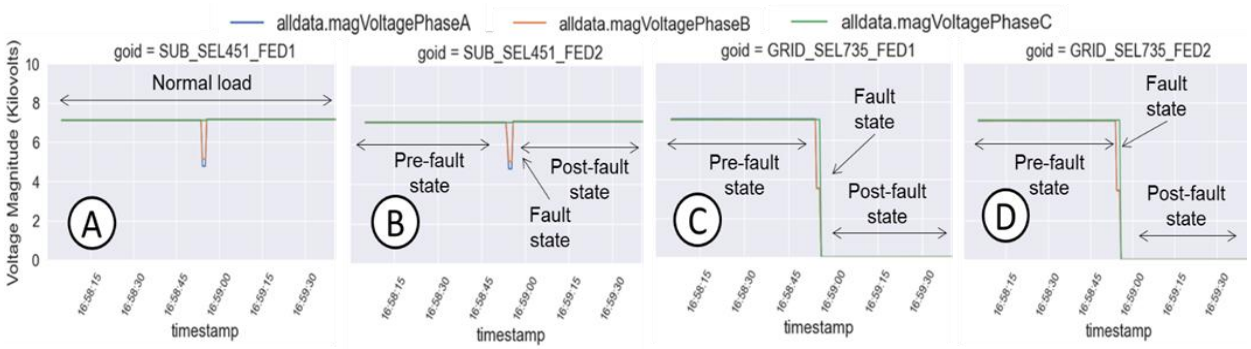


Figure 33. DLT voltage data from (A, B) the protective relays and (C, D) power meters for Experiment 3.b.

Experiment 3.c is represented by the LLG electrical fault at the 100T fuse feeder based on the diagram of Figure 21. During this LLG electrical fault affecting phase A and B, the DLT master node observed a significant increase in the current of phase A and B for the "SUB_SEL451_FED2" relay (Figure 34-B). This situation is because the phase A and B were grounding at the 100T fuse feeder bus. Once the phase A and B current increased at the fault state (Figure 34-B), the "SUB_SEL451_FED2" relay detected it, and the relay opened the breaker. When the breaker was tripped at the post fault state, the phase currents from the "SUB_SEL451_FED2" relay dropped to zero (Figure 34-B). However, the nominal phase voltages from the "SUB_SEL451_FED2" relay were measured (Figure 34-B). The LLG electrical faults produced an overvoltage in the non-faulted power line (phase C) at the fault location, and it was measured by the SEL 735 power meters (Figure 35 C and D). The electrical circuit path that was not affected directly by the LLG electrical fault measured a short-time disturbance for the phase currents (Figure 34-A) and voltages (Figure 35-A).

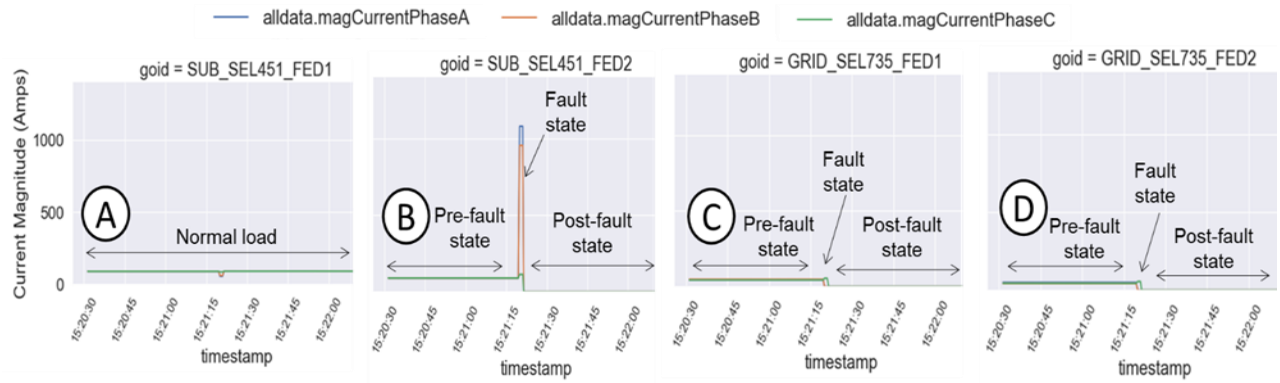


Figure 34. DLT current data from (A, B) the protective relays and (C, D) power meters for Experiment 3.c.

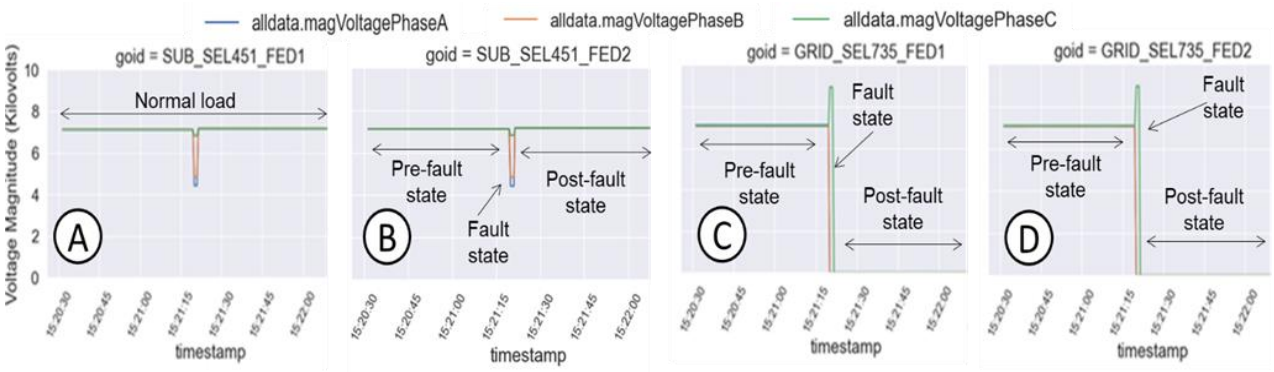


Figure 35. DLT voltage data from (A, B) the protective relays and (C, D) power meters for Experiment 3.c.

Experiment 3.d is represented by the 3LG electrical fault at the 50T fuse feeder based on the diagram of Figure 21. During this 3LG electrical fault affecting the phase A, B and C, the DLT master node observed a significant increase in all phase currents for the "SUB_SEL451_FED1" relay (Figure 36-A). This situation is because the phase A, B and C were grounded at the 50T fuse feeder bus. Once all phase currents increased at the fault state (Figure 36-A), the "SUB_SEL451_FED1" relay detected it, and the relay opened the breaker. Then, the measured phase currents from the "SUB_SEL451_FED1" relay dropped to zero at the post-fault state (Figure 36-A). However, the nominal phase voltages were measured from the "SUB_SEL451_FED1" relay at the post-fault state (Figure 37-A). The electrical circuit path that was not affected directly by the 3LG electrical fault measured a short-time disturbance for the phase currents (Figure 36-B, C and D) and voltages (Figure 37-B, C and D).

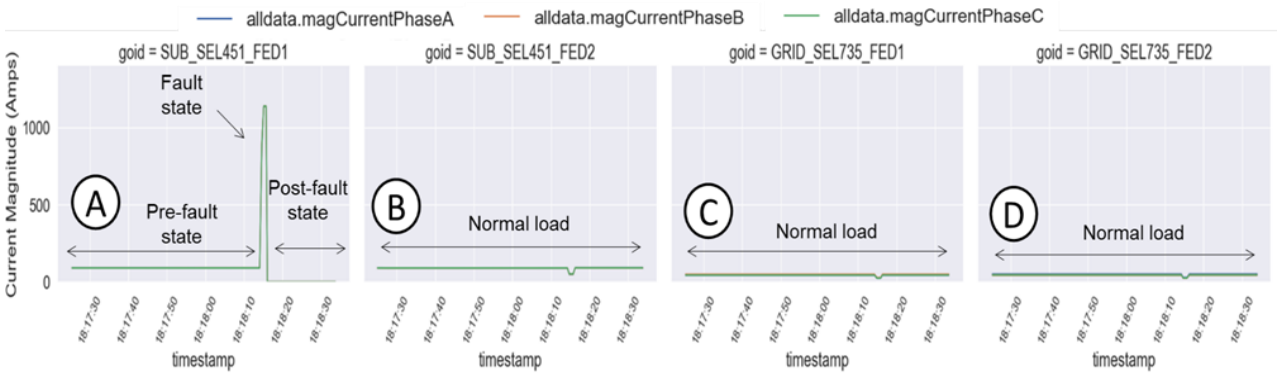


Figure 36. DLT current data from (A, B) the protective relays and (C, D) power meters for Experiment 3.d.

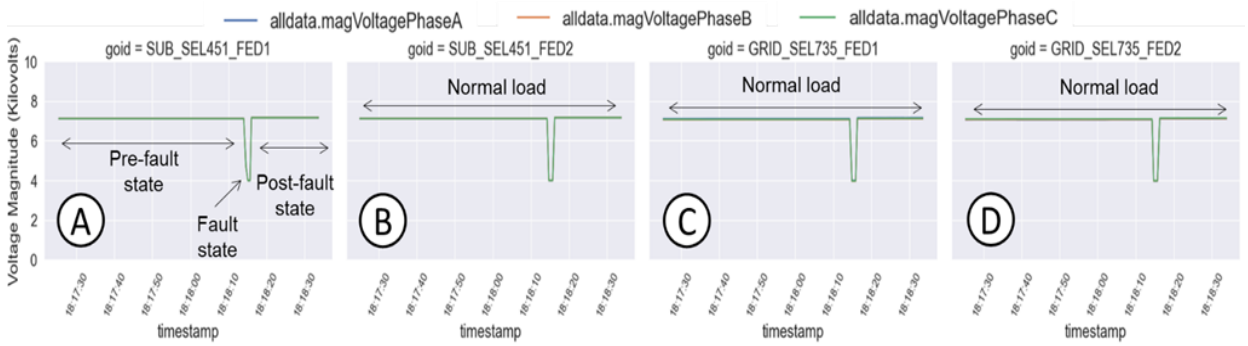


Figure 37. DLT voltage data from (A, B) the protective relays and (C, D) power meters for Experiment 3.d.

7.1.4 Experiment with Combined Cyber and Electrical Fault Events

Experiment 4.a is represented by the SLG electrical fault at the 100T fuse feeder and cyber-event (change the current transformer ratio setting of SEL 451* relay), based on the diagram of Figure 21. Before the application of the SLG electrical fault, the current transformer ratio of the "SUB_SEL451_FED2" relay was changed from 80 to 1, and the measured phase currents decreased drastically (Figure 38-B). Then, the SLG electrical fault affecting the phase A was performed at roughly 50 s. into the simulation. During this experiment, the DLT master node observed a non-significant increase in the current of phase A for the "SUB_SEL451_FED2" relay (Figure 38-B). This situation is because the current transformer ratio of the "SUB_SEL451_FED2" relay was modified. However the relay tripped because the inverse time overcurrent setting did not depend on the current transformer ratio setting. Once the phase A current increased at the fault state, the "SUB_SEL451_FED2" relay detected it, and the relay opened the breaker. Then, the measured phase currents from the "SUB_SEL451_FED2" dropped to zero at the post-fault state (Figure 38-B). In addition, the nominal phase voltages from the "SUB_SEL451_FED2" relay were measured at the post-fault state (Figure 39-B). The electrical circuit path that was not affected directly by the SLG electrical fault measured a short-time disturbance for the phase currents (Figure 38-A) and voltages (Figure 39-A).

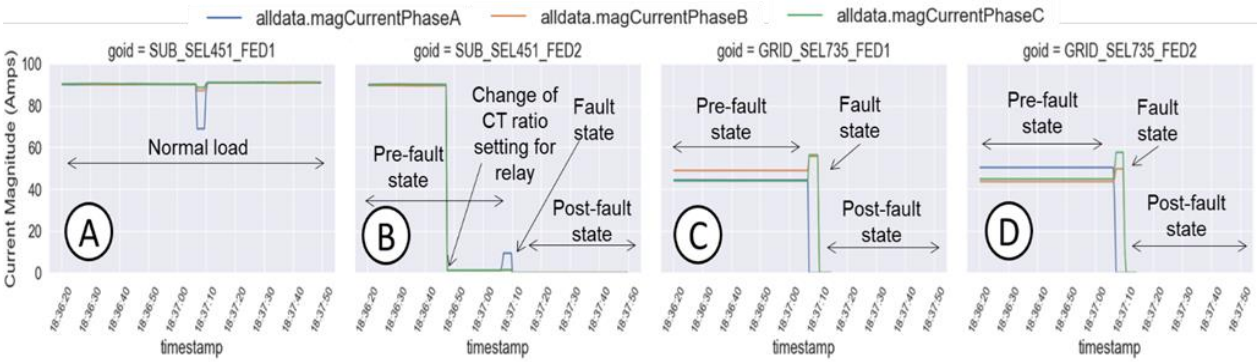


Figure 38. DLT current data from (A, B) the protective relays and (C, D) power meters for Experiment 4.a.

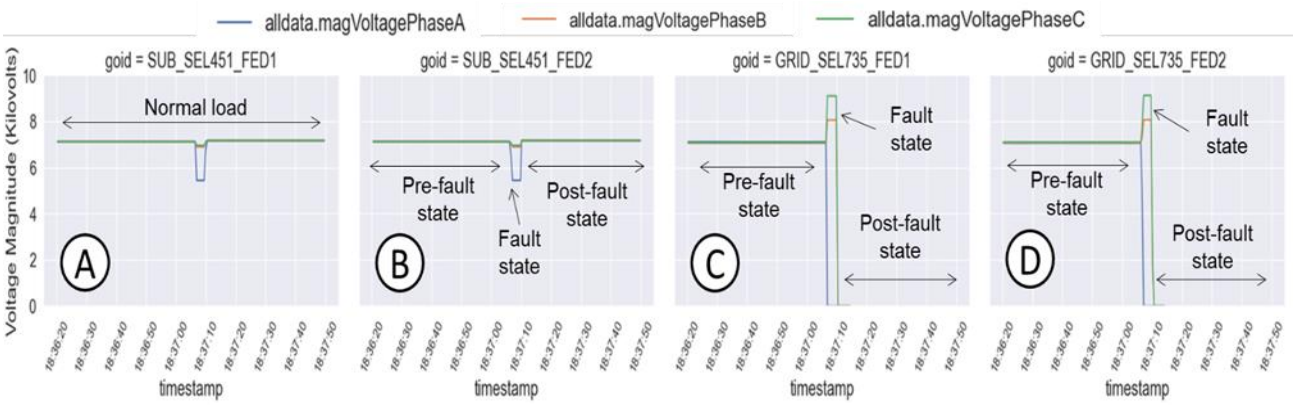


Figure 39. DLT voltage data from (A, B) the protective relays and (C, D) power meters for Experiment 4.a.

7.2 EXPERIMENTAL RESULTS ON PERFORMANCE

7.2.1 Hyperledger Caliper

Hyperledger Caliper is a benchmarking platform developed under the Hyperledger umbrella of projects. It is designed to support four types of DLT platforms, including Ethereum and HLF. Caliper collection benchmark configuration information and network details. It then operates on the DLT, called the system under test, to produce results [47].

The configuration provided by the user includes the workload, benchmark configuration, and network configuration. Workload modules consist of the transaction logic needed to send or query ledger data. These modules are implemented in NodeJS and adheres to the Caliper workload API. For HLF, workload modules define the interaction with chaincode functions. The benchmark configuration contains the settings for how Caliper will use the workload module, such as the number of transactions to execute and send rate. The network configuration contains the details needed for Caliper to connect to the DLT network, including addresses, ports, and cryptographic artifacts such as certificates and keys.

In addition to the transaction benchmark results that Caliper collects, such as transaction throughput and latency, monitors can be added to the benchmark configuration for collecting results from other external sources. Currently, Caliper supports monitoring processes, Docker containers, and Prometheus metrics [48]. The statistics or metrics that these monitors are configured to retrieve will be added to the results report generated upon completion of the benchmarking.

7.2.2 HLF Transaction Benchmarking Process

Hyperledger Caliper was used to collect benchmark performance results concerning the smart contracts implemented to send and query measurements and artifact hashes. The evaluated metrics include the transaction throughput (TPS); the minimum, maximum, and average transaction latency; and the average CPU usage (percentage) and average memory usage (percentage) of each node running HLF peer and orderer components.

The test bed contains three server machines with AMD Ryzen 9 3950X 16-core CPUs and 32 GB of RAM to function as DLT nodes. Each node hosts a HLF peer and orderer component. Hyperledger Caliper was set up on the DLT5 node, using the most recent 0.4.3 version of the Hyperledger Caliper Docker image to address issues with running the current stable version. The program is launched from a Docker-Compose file.

Benchmark configuration files were created for each type of smart contract to test. The measurement benchmark file defined 6 rounds of sending transactions using fixed send rates of 50, 100, 150, 200, 500, and 1,000 TPS for 3,000 transactions per round. A seventh round involved querying the ledger for 30 s. The artifact benchmark defined a similar set of rounds; however, it removed the 500 and 1,000 TPS rounds and reduced the number of transactions per round to 1,000 for simplicity and to reflect the lower expected number of artifact transactions in the actual system. The benchmark configuration files also contain the Prometheus queries to use for collecting metrics at the end of each round.

Workload modules consist of NodeJS files that implement the Caliper workload API with logic to send transactions and query data using MeasurementHashAudit and ArtifactHashAudit functions. Each type of transaction for each smart contract was broken out into its own file. The modules that log transactions generate random data for the key and hash, whereas the query logic simply uses an existing key.

The network configuration contains the details for connecting to the HLF network. This includes the paths to the private key and signed certificate for the user identity that Caliper uses to access the network and the connection profile. The connection profile is generated by a helper script and allows Caliper to access the HLF network using a gateway [49].

The overall benchmarking process involved modifying the MaxMessageCount parameter of the BatchSize section inside the HLF channel configuration before each Caliper run. This configuration was contained in the *configtx.yaml* file. The maximum message count of each batch defines the maximum number of transactions, or messages, to include in the data section of each block, analogous to the maximum ledger block size [50]. This setting, combined with the other parameters in the BatchSize section, can be modified to tune performance of ledger block creation based on the transaction characteristics of a particular network channel.

7.2.3 Benchmarking Results from Caliper Framework on DLT

The results for each batch size and smart contract were saved to an HTML report file output by Caliper. Python was used to process the HTML tables containing the results in each file and generate plot images.

Figure 40 shows the measured throughput versus the send rate for each batch size up to a send rate of 2,000 TPS. At a send rate of 500 TPS or more, although there is an increase in throughput to a maximum of ~490 TPS, the results fail to match the send rates. A batch size of 25 results in the highest throughput for send rates of 1,000 TPS and above for the ArtifactHashAudit contract, but the overall results are similar for both contracts.

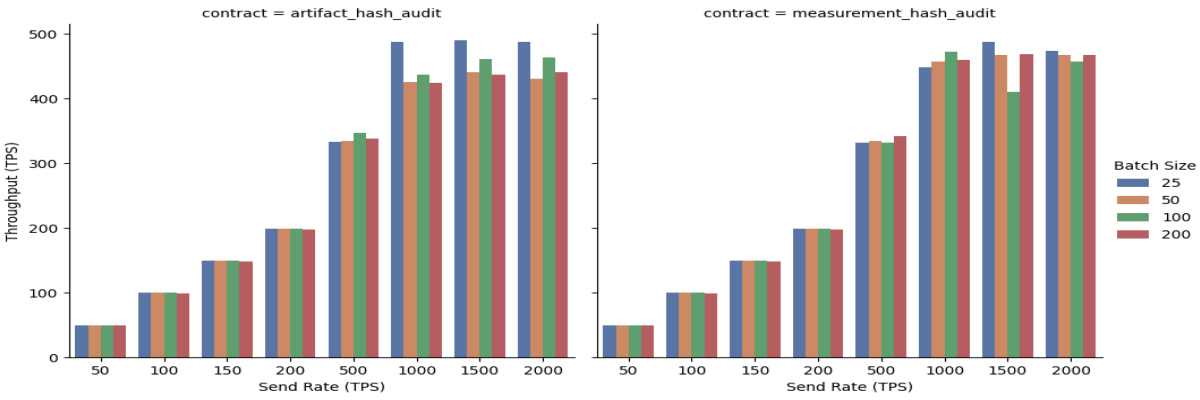


Figure 40. Throughput results of the batch sizes.

Figure 41 shows the measured average latency in seconds observed for each send rate and batch size. The latency decreases for each send rate up to 500 TPS, where it becomes mostly steady except for a batch size of 25, which increases significantly. The latency shows a significant initial drop in latency for batch sizes of 100 and 200, most likely due to batch timeouts taking effect when receiving transactions at slower rates.

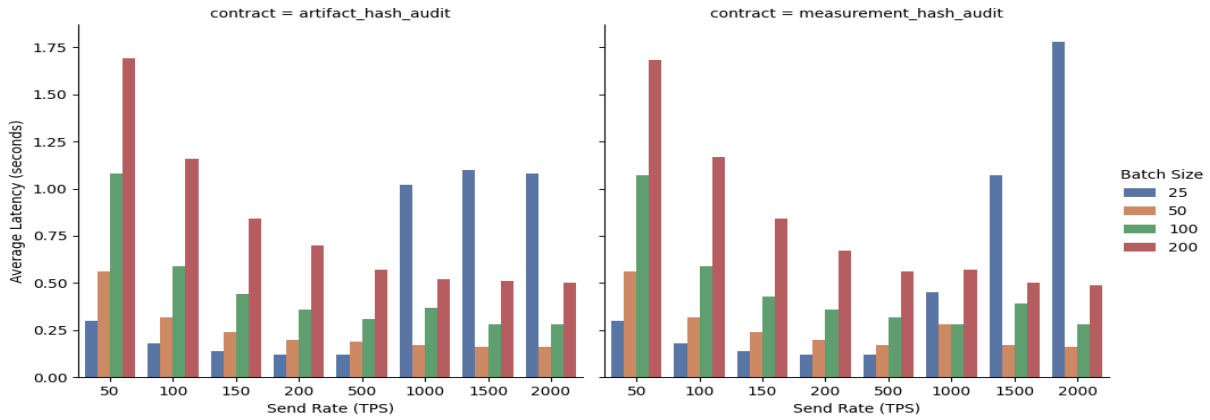


Figure 41. Average latency results of the batch sizes.

Figure 42 shows the average CPU usage of the DLT nodes for each send rate and batch size. While average CPU usage for the ArtifactHashAudit contract results in a maximum of about 4.5%, the results show variability between batch sizes for each send rate. For the MeasurementHashAudit contract, average CPU usage increases mildly for each send rate up to about 1,000 TPS, where there is some minor variability. Overall, minimal impact on CPU usage was observed.

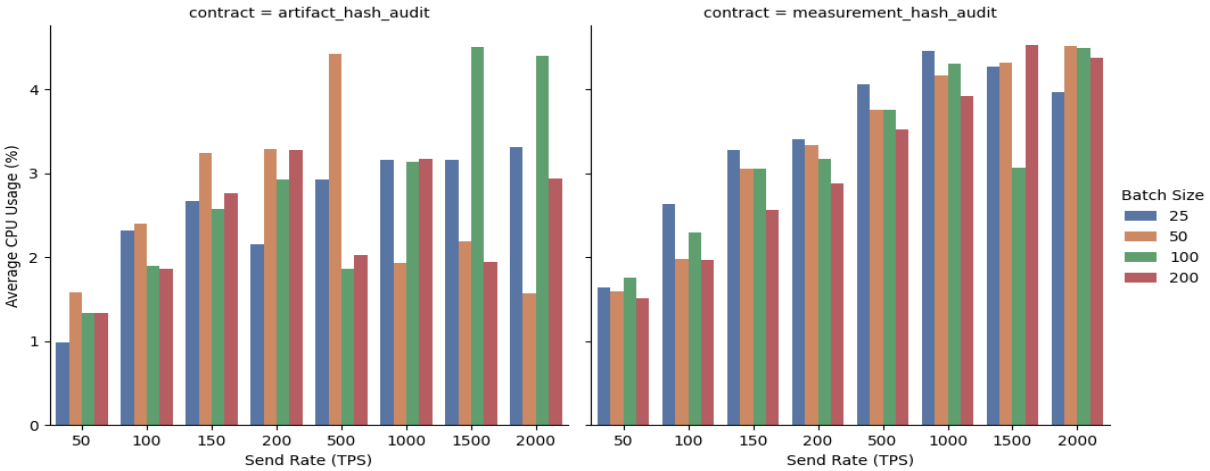


Figure 42. Average CPU usage results of the batch sizes.

Figure 43 shows the average memory usage of the DLT nodes for each send rate and batch size. While the overall amount of memory used during the benchmarks is slightly higher relative to CPU usage, the memory used for each send rate remained mostly constant. Overall, the observed impact of the benchmark tests on memory is low.

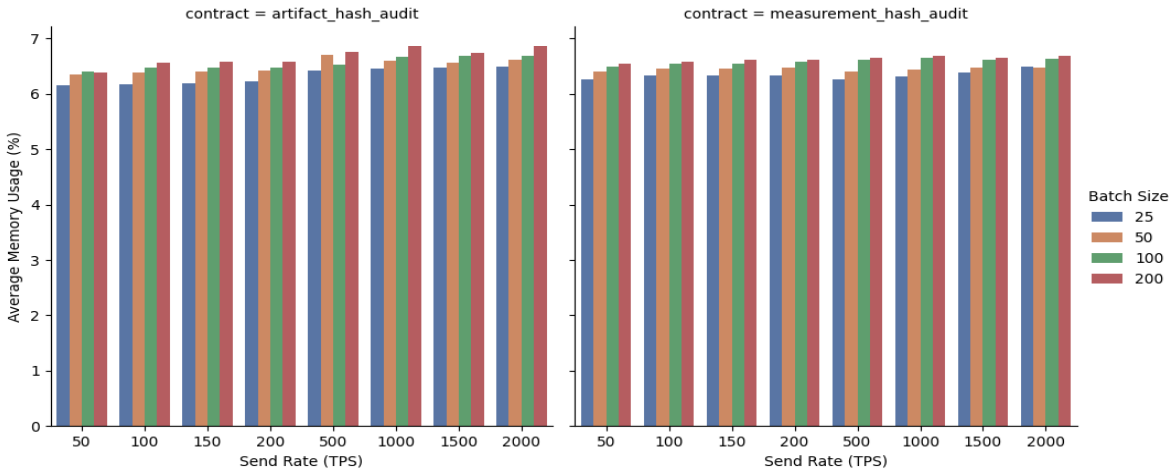


Figure 43. Average memory usage results of the batch sizes.

7.2.4 Performance Results from High Packet SV Traffic

Experiments were conducted to test the ability of the Grid Guard framework to handle high-velocity, high-volume packet traffic. These experiments involved using the EmSense devices connected to the main network of the Grid Guard test bed. Specifically, multiple EmSense devices were used starting with one connected to the network. Once activated, the EmSense began transmitting SV packets. Following that, a second EmSense began transmitting in addition to the first. In the same pattern, a third EmSense was activated 1 min after the second EmSense and so on until six EmSense devices were transmitting SV packets. Each EmSense device transmitted packets at a constant rate. The following figures show a Grafana dashboard to illustrate the results on Benchmarking for the DLT. These results demonstrate that the DLT can process large quantities of data quickly. Specifically, Figure 44 shows the network traffic both transmitted (TX) and received (RX). Figure 44 and Figure 45 depict the traffic level in bits per second over time. Figure 45 shows the benchmarking results for SV message storage in long-term memory and in-queue.

Figure 44. Benchmarking results with EmSense—(top left) networking data traffic transmitted by the Grid Guard framework, (top right) network data traffic received, (bottom left) off-chain data storage, and (bottom right) on-chain data storage.

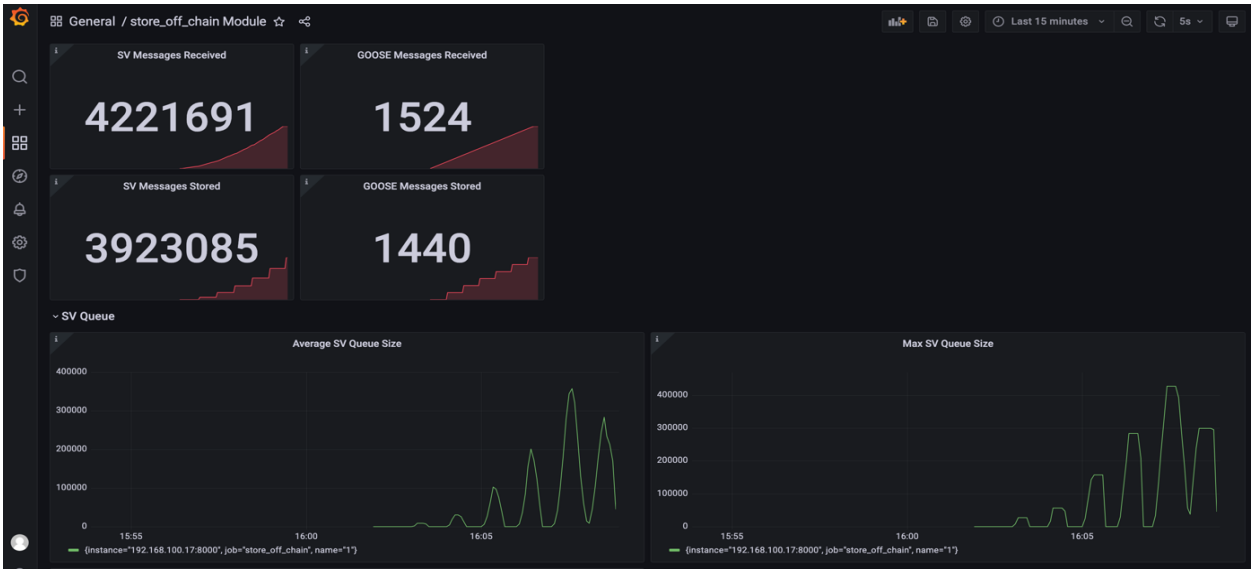


Figure 45. Benchmarking results with EmSense—(top) SV messages stored and received, (bottom left) average SV queue size, and (bottom right) max SV queue size.

As expected, the received graph shows a sequence of plateau at increasing levels similar to a staircase. With the activation of each EmSense device, the graph goes to a new level. Also of interest are the off-chain data and in particular how much of the data are used to store the data that are ingested over time. The off-chain data itself grows very quickly compared with on-chain data since the off-chain data consist of the actual data of the packets, where the on-chain data are hashes of windows.

7.3 OVERALL ANALYSIS AND DISCUSSION

The results demonstrate the effectiveness of the attestation framework to attest to system changes and anomaly detection in helping to flag specific events based on statistical threshold values. The attestation framework can support the detection of system changes by itself, but when combined with an anomaly detection framework, it has a lower system resource requirement and may be more likely to catch system changes. Additional experiments are being conducted to study the advantages and disadvantages of using anomaly detection to trigger the attestation checks. Moreover, these initial demonstrations and experiments prove that the framework can handle stress and high data bandwidth, such as multiple high-fidelity sensors in the 10 kHz and above range. The research shows that the data are captured correctly and attested to by the Grid Guard framework using the blockchain DLT and may be also used for additional post-mortem analysis in addition to or alongside historian data for a confidence analysis with more than historian data alone given the data tampering resistance of the DLT.

8 CONCLUSIONS AND FUTURE WORK

Overall, Phase 2 of the DarkNet project on DLT was successful. This report describes the development of a preliminary system that can ingest data from the network and secure the data with the blockchain. The DLTs can manage even very high-speed data when processing this data from high-fidelity sensors, such as EmSense.

Future work will include developing the concepts and ideas of this project further to deploy the technology at real substations or other environments, such as DERs or a microgrid. In so doing, the technology developed would be agnostic to the environment where it is deployed and will enable handling multiple SCADA protocols and types of edge devices that include relays of various brands. Future work will also include creating a better set of potential cyber event scenarios in which the cryptographic keys are compromised. This work will improve the understanding of how the DLTs would be able to respond to compromised nodes and such scenarios.

9 REFERENCES

[1] *IEEE 1588*, NIST. Available online (accessed August 2, 2022): https://www.nist.gov/el/intelligent-systems-division-73500/ieee-1588.

[2] "ICS Alert (IR-ALERT-H-16-056-01): Cyber-Attack Against Ukrainian Critical Infrastructure, Cybersecurity & Infrastructure Security Agency." Available online (accessed August 3, 2022): https://www.cisa.gov/uscert/ics/alerts/IR-ALERT-H-16-056-01.

[3] J. R. Minkel, "The 2003 Northeast Blackout--Five Years Later," *Scientific American* 13: 1–3, 2008.

[4] N. R. Friedman, *Distributed energy resources interconnection systems: Technology review and research needs*, National Renewable Energy Laboratory, 2002.

[5] "Raft consensus in swarm mode," Docker. Available online (accessed August 2, 2022): https://docs.docker.com/engine/swarm/Raft/.

[6] "Alternative Precision Timing Services for the Nation's Power Grid," US Department of Energy Office of Electricity. Available online (accessed August 2, 2022): https://www.energy.gov/oe/articles/alternative-precision-timing-services-nations-power-grid.

[7] M. Foti and M. Vavalis, "What blockchain can do for power grids?" *Blockchain: Research and Applications* 2(1): 100008, 2021. https://doi.org/10.1016/j.bcra.2021.100008.

[8] M. Andoni, V. Robu, D. Flynn, S. Abram, D. Geach, D. Jenkins, P. McCallum, and A. Peacock, "Blockchain technology in the energy sector: A systematic review of challenges and opportunities," *Renewable and Sustainable Energy Reviews* 100: 143–174, 2019. https://doi.org/10.1016/j.rser.2018.10.014.

[9] D. Sikeridis, A. Bidram, M. Devetsikiotis, and M. J. Reno, "A Blockchain-based Mechanism for Secure Data Exchange in Smart Grid Protection Systems," *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, January 10–13, 2020, Las Vegas, Nevada, 1–6. https://ieeexplore.ieee.org/document/9045368.

[10] X. Kong, J. Zhang, H. Wang, and J. Shu, "Framework of Decentralized Multi-chain Data Management for Power Systems," *CSEE Journal of Power and Energy Systems* 6(2): 458–468, 2020. https://ieeexplore.ieee.org/iel7/7054730/9084208/08779800.pdf.

[11] G. Liang, S. R. Weller, F. Luo, J. Zhao, and Z. Y. Dong, "Distributed Blockchain-based Data Protection Framework for Modern Power Systems Against Cyber Attacks," *IEEE Transactions on Smart Grid* 10(3): 3162–3173, 2019. https://ieeexplore.ieee.org/document/8326530.

[12] L. Hang and D. Kim, "Design and Implementation of an Integrated IoT Blockchain Platform for Sensing Data Integrity," *Sensors* 19(10): 1–26, 2019. https://doi.org/10.3390/s19102228.

[13] J. Gao, K. O. Asamoah, E. B. Sifah, A. Smahi, Q. Xia, H. Xia, X. Zhang, and G. Dong, "GridMonitoring: Secured Sovereign Blockchain Based Monitoring on Smart Grid," *IEEE Access* 6: 9917–9925, 2018. https://ieeexplore.ieee.org/document/8303679.

[14] V. Mathane and P. V. Lakshmi, "Multi-Layer Attestation for Internet of Things using Blockchain," *International Journal of Engineering and Advanced Technology (IJEAT)* 9(3): 995–1000, 2020. https://doi.org/10.35940/ijeat.C4719.029320.

[15] E. Pioli Moro and A. K. Duke, "Distributed Ledger Technologies and the Internet of Things: A Device Attestation System for Smart Cities," *The JBBA* 1–7, 2020. https://doi.org/10.31585/jbba-3-1-(7)2020.

[16] J. C. Bare, "Attestation and Trusted Computing," *CSEP 590: Practical Aspects of Modern Cryptography* 1–10, 2009.

[17] A. Lee-Thorp, *Attestation in Trusted Computing: Challenges and Potential Solutions*, Royal Holloway University of London, 1–79, 2010. https://www.ma.rhul.ac.uk/static/techrep/2010/RHUL-MA-2010-09.pdf.

[18] O. Arias, F. Rahman, M. Tehranipoor, and Y. Jim, "Device Attestation: Past, Present, and Future," *Design, Automation and Test in Europe Conference and Exhibition*, Dresden, Germany, March 19–23, 2018, 473–478. https://ieeexplore.ieee.org/document/8342055.

[19] I. R Jenkins and S. W. Smith, "Distributed IoT Attestation Via Blockchain." *20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing*, May 11-14, 2020, Melbourne, Australia, 798-801. https://ieeexplore.ieee.org/document/9139696.

[20] Z. Sun, B. Feng , L. Lu, S. Jha, "OAT: Attesting Operation Integrity of Embedded Devices", *2020 IEEE Symposium on Security and Privacy*, May 18-21, 2020, San Francisco, CA, 1443-1449. https://ieeexplore.ieee.org/document/9152803.

[21] J. Guttman, A. Herzog, J. Millen, L. Monk, J. Ramsdell, J. Sheehy, B. Sniffen, G. Coker, P. Loscocco, "Attestation: Evidence and Trust," MITRE Technical Report, MITRE 080072, Center for Integrated Intelligence Systems Bedford, Massachusetts, March 2008, 1-39. https://www.mitre.org/sites/default/files/pdf/07_0186.pdf.

[22] J. Valente, C. Barreto, A. A. Cardenas, "Cyber-Physical Systems Attestation," *2014 IEEE Conference on Distributed Computing in Sensor Systems*, May 26-28, 2014, Marina Del Rey, CA, 354-357. https://ieeexplore.ieee.org/document/6846189.

[23] T. Hardjono and N. Smith, "Towards an attestation architecture for blockchain networks," *World Wide Web*, 24: 1587–1615, 2021 https://doi.org/10.1007/s11280-021-00869-4.

[24] G. Coker, J. Guttman, P. Loscocco, A. Herzog, J. Millen, B. O'Hanlon, J, Ramsdell, A. Segall, J. Sheehy and B. Sniffen, "Principles of Remote Attestation," *International Journal of Information Security*, 10, 63-81, 2011. https://doi.org/10.1007/s10207-011-0124-7.

[25] L. Jain and J. Vyas, "Security Analysis of Remote Attestation," CS259 Report, Stanford University. https://seclab.stanford.edu/pcl/cs259/projects/cs259_final_lavina_jayesh/CS259_report_lavina_jayesh.pdf

[26] F. Brasser, K. B. Rasmussen, A. R. Sadeghi, G. Tsudik, "Remote Attestation for Low-End Embedded Devices: the Prover's Perspective," *53nd Design Automation Conference*, June 5-9, 2016, Austin, TX, 1-6. https://ieeexplore.ieee.org/document/7544334.

[27] tpm2-software community, Remote Attestation, Dec. 18, 2019, Available online (accessed 2022-7-21): https://tpm2-software.github.io/tpm2-tss/getting-started/2019/12/18/Remote-Attestation.html.

[28] I. Sfyrakis and T. Gross, "A Survey on Hardware Approaches for Remote Attestation in Network Infrastructures," Newcastle University, July 2020, 1-20. https://arxiv.org/abs/2005.12453.

[29] W. Johnson, S. Ghafoor, S. Prowell, "A Taxonomy and Review of Remote Attestation Schemes in Embedded Systems," *IEEE Access*, 9: 142390-142410, 2021. https://ieeexplore.ieee.org/abstract/document/9565863/.

[30] A. M. Naveed, B. M. Haroon, S. Dash, J. Wen Wong, J. Xu, H. Wei Lim, B. Sikdar, "HAtt: *H*ybrid Remote *Att*estation for the Internot of Things with High Availability," *IEEE Internet of Things Journal*, 7 (8): 7220-7233, 2020. https://ieeexplore.ieee.org/document/9047883.

[31] A. Banks, M. Kisielk, P. Korsholm, "Remote Attestation: A Literature Review," IT University of Copenhagen , May 2021, 1-34. https://arxiv.org/abs/2105.02466.

[32] The Linux PTP Project. Available online (accessed 2022-8-2): http://linuxptp.sourceforge.net/.

[33] E. C. Piesciorovsky, R. Borges Hink, A. Werth, G. Hahn., A. Lee, J. Richards, Y. Polsky, "Technical Report: Assessment of the Electrical Substation-Grid Test Bed with Inside/Outside Devices and Distributed Ledger," Oak Ridge National Laboratory, Electrification and Energy Infrastructures Division, Report: ORNL/TM-2022/1840, 1-102, April 2022. https://doi.org/10.2172/1864423.

[34] Libiec61850, Github. Available online (accessed 2022-8-2): https://github.com/mz-automation/libiec61850.

[35] Ledger, Hyperledger Fabric. Available online (accessed 2022-7-21): https://hyperledger-fabric.readthedocs.io/en/release-2.2/ledger/ledger.html.

[36] The Ordering Service, Hyperledger Fabric. Available online (accessed 2022-7-21): https://hyperledger-fabric.readthedocs.io/en/release-2.2/orderer/ordering_service.html.

[37] YAML Ain't Markup Language (YAML™) version 1.2Available online (accessed 2022-8-2): https://yaml.org/spec/1.2.2/.

[38] Hyperledger Fabric, Sample Config – Release 2.2, Github. Available online (accessed 2022-7-21): https://github.com/hyperledger/fabric/blob/release-2.2/sampleconfig/core.yaml.

[39] Available online (accessed 2022-7-21): https://github.com/hyperledger/fabric/blob/release-2.2/sampleconfig/orderer.yaml.

[40] Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations: NIST SP 800-52 Rev. 2. Available online (accessed 2022-8-3): https://csrc.nist.gov/News/2019/nist-publishes-sp-800-52-revision-2.

[41] C. Edvard, "Six common bus configurations in substations up to 354 kV," Power Substation/Transmission and Distribution, March 18, 2019. Electrical Engineering Portal. Available online (accessed 2022-7-21): https://electrical-engineering-portal.com/bus-configurations-substations-345-kv.

[42] SEL-451-5 Protection, Automation, and Bay Control System and SEL-400 Series Relays Instruction Manual, Schweitzer Engineering Laboratories Inc. Available online (accessed 2022-7-21): https://selinc.com/products/451/docs/.

[43] SEL-734 Advanced Metering System Instruction Manual, Schweitzer Engineering Laboratories Inc. Available online (accessed 2022-7-21): https://selinc.com/products/734/docs/.

[44] SEL-735 Power Quality and Revenue Meter Instruction Manual, Schweitzer Engineering Laboratories Inc. Available online (accessed 2022-7-21): https://selinc.com/products/735/docs/.

[45] SEL-3530 Real-Time Automation Controller (RTAC) Instruction Manual, Schweitzer Engineering Laboratories Inc. Available online (accessed 2022-7-21): https://selinc.com/products/3530/docs/.

[46] Total Clearing Time-Current Characteristic Curves, Positrol® Fuse Links–S&C "T" Speed (TCC 170-6-2). Available online (accessed 2022-7-21): http://www.sandc.com/en/products--services/products/sm--sml-power-fuses/.

[47] Architecture, Hyperledger Caliper. Available online (accessed 2022-7-21): https://hyperledger.github.io/caliper/vNext/architecture/.

[48] Resource and Transaction Monitors, Hyperledger Caliper. Available online (accessed 2022-7-21): https://hyperledger.github.io/caliper/vNext/caliper-monitors/.

[49] Connection Profile, Hyperledger Fabric. Available online (accessed 2022-7-21): https://hyperledger-fabric.readthedocs.io/en/release-2.2/developapps/connectionprofile.html.

[50] Hyperledger Fabric, Sample Config, Github. Available online (accessed 2022-7-21): https://github.com/hyperledger/fabric/blob/main/sampleconfig/configtx.yaml.