Implementation of Blockchain to Simulate Data Acquisition in Smart Grid/CPS

Video Link:

1. Abstract:

In this project, blockchain and socket communication have been used to simulate the data acquisition system of any Cyber-physical System, such as a smart grid. Here, four relays have been used as four communication nodes of an IEEE 9-bus system in a smart grid. A data aggregator and a control center have been used to coordinating the whole process. These relays, aggregator and control center have been simulated as six different nodes/servers. To make the data acquisition system decentralized, immutable and non-repudiable, a private blockchain technology has been implemented with a customized consensus mechanism. Moreover, socket communication and encoding technique have been implemented for faster and secured data transmission between nodes. As the data is authenticated and verified by the nodes in several steps while preserving this distributed ledger in a decentralized manner, it would be very difficult for an adversary to manipulate the data considering the effort that would be required to do so.

2. Preparation:

2.1. Example Smart Grid:

In this project, an IEEE 9-bus system have been used as the example of Smart Grid. The topology of IEEE 9-bus system, i.e., the connection of substations through transmission lines, are shown in Figure 1.

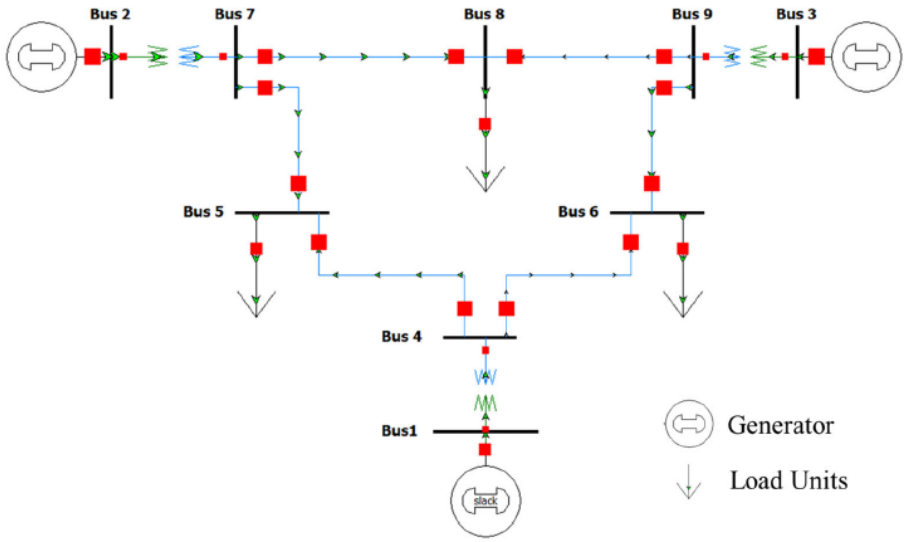
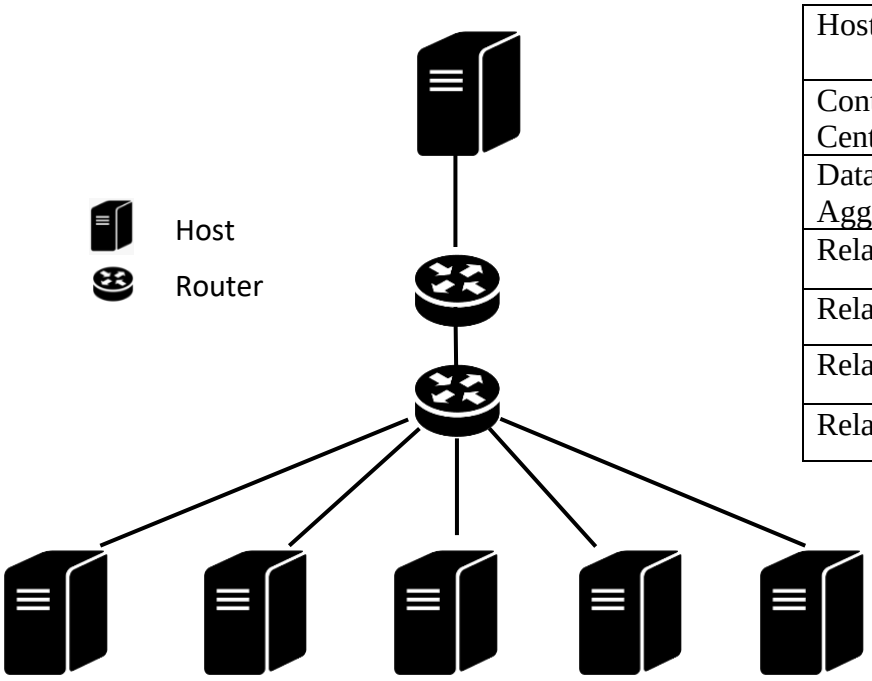


Figure 1. The topology of IEEE 9-bus system.

2.2. Simulation of communication networks

To prepare this implementation, Mininet has been used to simulate a communication network with the topology shown in Figure 2. This network topology mimics but simplifies the communication infrastructure that can be found in today's cyber-physical systems. In this network, a central Control Center communicates with a Data Aggregator, which further communicates with Relay1, Relay2, Relay3, and Relay4.

To simplify, the IP address for each host simulated in Mininet has been fixed. Communication networks used in CPS usually perform two operations: a control operation which can change the configuration or states of physical processes and polling operations which periodically retrieve measurements indicating the state of physical processes. In power grids, each substation contains the following physical measurements: the magnitudes and phasor angles of voltage, real power injection, and reactive power injection.



Host	IP Address	Port
Control Center	10.0.0.1	
Data Aggregator	10.0.0.20	20000
Relay1	10.0.0.11	20001
Relay2	10.0.0.12	20002
Relay3	10.0.0.13	20003
Relay4	10.0.0.14	20004

Figure 2: Network Topology

The measurements collected in each substation are shown in Table 1. To simplify the experiment, the units of all measurements have been removed. We assign each measurement an index number as shown in Table1. Because there are only four relay machines, each relay will be responsible for multiple measurements.

The assignment of the measurements is also specified in Table 1. Specifically, Relay 1 will be responsible for measurements from Bus 1 and Bus 2; Relay 2 will be responsible for measurements from Bus 3 and Bus 4; Relay 3 will be responsible for measurements from Bus 5 and Bus 6; and Relay 4 will be responsible for measurements from Bus 7, Bus 8, and Bus 9.

Substation	Communication Node	Type	Index	Values
Bus1	R1	Vm	11	1.00
		Vp	12	0.00
		P	13	71.95
		Q	14	24.07
Bus2		Vm	21	1.00
		Vp	22	9.67
		P	23	163.00
		Q	24	14.46
Bus3	R2	Vm	31	1.00
		Vp	32	4.77
		P	33	85.00
		Q	34	-3.65
Bus4		Vm	41	0.99
		Vp	42	-2.41
		P	43	0.00
		Q	44	0.00
Bus5	R3	Vm	51	0.98
		Vp	52	-4.02
		P	53	-90.00
		Q	54	-30.00
Bus6		Vm	61	1.01
		Vp	62	1.93
		P	63	0.00
		Q	64	0.00
Bus7	R4	Vm	71	0.99
		Vp	72	0.62
		P	73	-100.00
		Q	74	-35.00
Bus8		Vm	81	1.00
		Vp	82	3.80
		P	83	0.00
		Q	84	0.00
Bus9		Vm	91	0.96
		Vp	92	-4.35
		P	93	0.00
		Q	94	0.00

2.3. Network Protocol

In this implementation, a protocol has been used, based which the Control Center collects measurements from Relays. This protocol is called DNP3m (DNP3 minus), as it represents a simplified version of the DNP3 protocol, which is widely used in US power grid infrastructure. It is an application layer protocol built on top of the TCP protocol.

The application layer structure is shown in Figure 3:

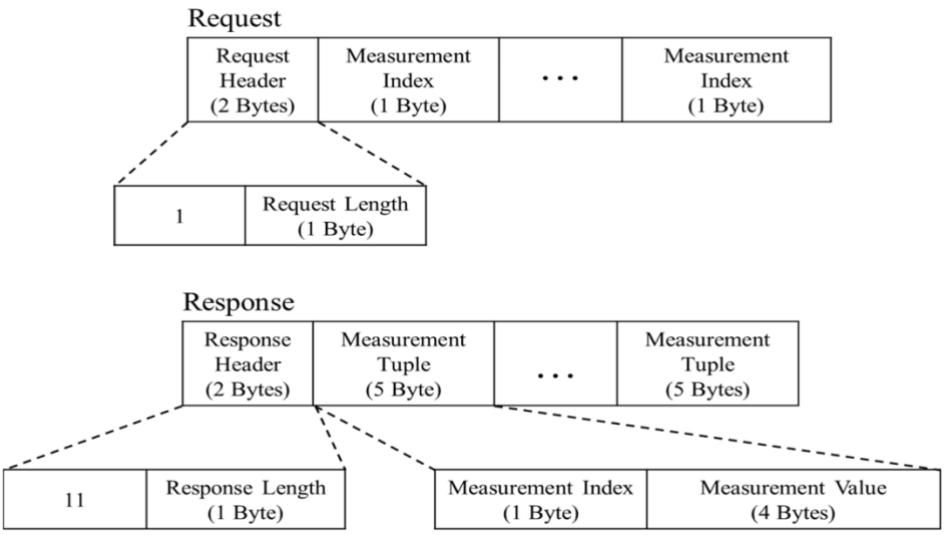


Figure 3: Application layer of DNP3m protocol

- Request: The "Request Header" contains two bytes. The first byte indicates that this is the request of the DNP3m protocol; the value of this byte has been set as "1". The second byte of the "Request Header" contains the length of this whole request message. Following the "Request Header," the request contains indices of the measurements that the Control Center want to collect. For example, if the request is issued from the Control Center to collect measurements of indices 11, 12, 13, 14, 21, 22, 23, and 24, the request should include all these indices following the "Request Header" with each index occupying one byte. Here, the Control Center collects all measurements listed in Table 1.

- Response. The "Response Header" contains two bytes. The first byte indicates that this is the response of the DNP3m protocol; the value of this byte should be set as "11". The second byte of the "Response Header" contains the length of this whole response message. Following the "Response Header," the response contains multiple "Measurement Tuples". Each Measurement Tuple contains a Measurement Index occupying one byte and a Measurement Value occupying 4 bytes. In this tuple, the measurement index corresponding to the index included in the request and the measurement value is the corresponding value shown in Table 1.

2.4. Blockchain Mechanism

In the blockchain implementation part, a private blockchain has been implemented. Here, a customized consensus mechanism has been developed for selecting mining node for different time scale. Also, for hashing the block, python 'hashlib' module has been used. With this module, Sha256 algorithm has been used for hashing function. For exchanging data as strings, 'UTF-8' encoding technique has been used along with DNP3m protocol.

Authentication and verification of blocks has also been carried out by the nodes and the data aggregator in several steps.

3. Project Environment

This project has been implemented in a virtual machine environment built in a virtual box with below configuration.

Virtual Machine OS: Ubuntu 20.04 (64-bit)
Base Memory: 8 GB
Storage: 20 GB
Mininet Version: 2.2.2
Python Version: Python3

4. Main Design

The whole process can be divided into four major components depicted in Figure 4.

- **Control Center**
  Control Center (CC) is responsible for generating control and data acquisition command. It has its own storage and blockchain copy

- **Data Aggregator**
  Data Aggregator (DA), after receiving data acquisition command from CC, collects data from field devices, selects mining node, coordinate the verification process and finally updates control center about the new block

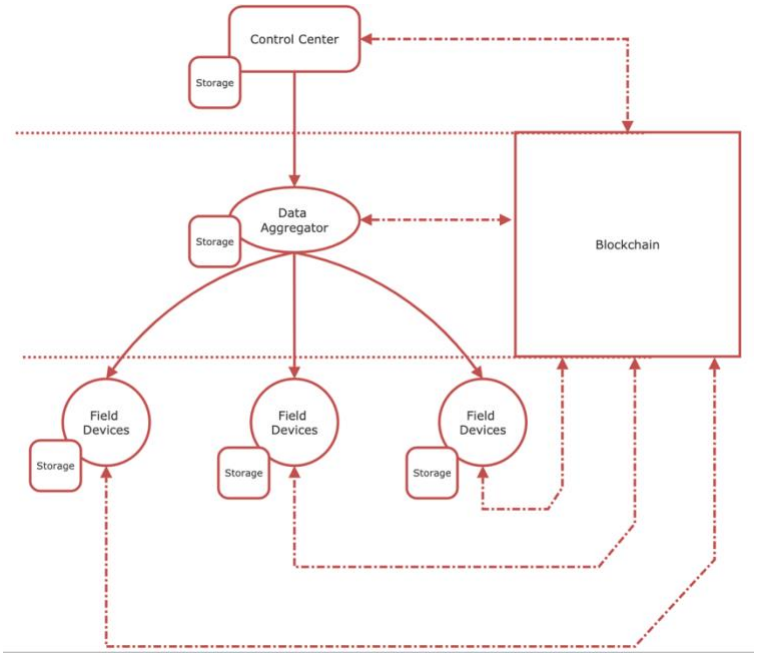


Figure 4: Main Design

- **Field Devices**
  Field devices can be the actuators, sensors, PMU, smart meters, relays etc. These are responsible for sending data to DA. One of the field devices are being selected as Mining Node by DA in each cycle while others takes part in block verification process.

- **Blockchain**
  Blockchain implementation starts with the selection of Mining node and ends temporarily with the addition of new blocks in a repetitive manner. In this part, data preprocessing, data aggregation, existing blockchain verification, hashing, new block creation, block verification, block addition occurs as intermediate process. Also, for the simplicity, one block has been limited to one data measurement from all the field devices and separate blockchain copies has been generated as a .zip file for every 10 blocks.

5. Operational procedure

The whole operational procedure can be described by the flow chart given in Figure 5.

Step-1: Process starts

Step-2: CC initiates data acquisition process by sending specific indices to DA. Here, for simplicity, measurement data for all the indices from all four relays have been requested. Also, CC packs the indices in a request header following the DNPm protocol and sends it to DA over a newly created TCP socket. The IP address of CC is 10.0.0.1 which acts as a client to DA server (10.0.0.20)

Step-3: Encrypted request reaches to DA server

Step-4: DA unpack the request and again packs four separate requests for field devices which include measurement indices according to the assigned relays. It also uses DNP3m protocol to create the request packs. DA also creates TCP socket connection with four relays with four different ports. The IP address of DA is 10.0.0.20 and the port is 20000. It acts as server to CC and client to field devices.

Step-5: DA sends the request to relays and wait for the response

Step-6: Each relay unpacks the DA request and create response pack with their measured data according to the indices following the DNP3m protocol. Then they send the responses to DA. The IP addresses of relays are 10.0.0.11, 10.0.12, 10.0.0.13, 10.0.0.14 and the ports are 20001, 20002, 20003, 20004. The relays act as clients to DA

Step-7: Response packs reaches to DA server

Step-8: In this step, DA sends its own copy of blockchain to all nodes (relays) to check whether they have the same copy of the blockchain or not. If they have the same exact copy of blockchain as DA, they send replies ‘already updated’; otherwise ‘Blockchain manipulated’

Step-9: If any node does not have updated blockchain, it immediately stops the program. On the other hand, if all the nodes have the updated blockchain, DA starts calculating the average of the measurement data.

Step-10: Mining Node Selection- If the average of the measured data of a node is the closest to the actual average (average of all 36 measurement) DA selects that node as the mining node for that cycle. DA also sends the entire measurement data from all nodes (or relays) to the mining node using 'UTF-8' encryption.

Step-11: If a node is being selected as mining node, it first aggregates the data into four categories (Voltage Magnitude, Vm; Voltage Phasor, Vp; Real Power, P; Reactive Power, Q). So, even somehow the mining node is compromised, the attacker would only get the sum value of data for all relays.

Step-12: In this step, Mining node mines the data, hashes the data and creates new block. While doing so, it also verifies the previous block. On the other hand, other three nodes wait for the DA to command them to verify this newly created block

Step-13: Mining Node sends the newly created block to DA for verification purpose.

Step-14: DA sends this new block with previous block to all nodes except the mining node for verification. In the verification part, nodes matche their copies of blockchain with the copy from DA and also matches the hash values between newly created block (previous hash value) and previous block (current hash value)

Step-15: All the nodes except the Mining node send their verification responses ('chain verified') to DA.

Step-16: DA gets all the responses and starts its' own verification process.

Step-17: If that passes too, DA sends command to all nodes except mining node (as mining node have already added the block in its' chain) to add that new block to their chain. On the other hand, if any of the received verification responses is negative, the new block is discarded, and program comes to an end

Step-18: All nodes update their chain with new block, Then, DA sends the updated copy of the blockchain to CC and one cycle completes. But, CC initiates data acquisition process again after 5 sec and cycle repeats.

When 10 blocks have been added to any chain, for simplicity, that chain is zipped into a separate file following incremental naming convention.

6. Evaluation:

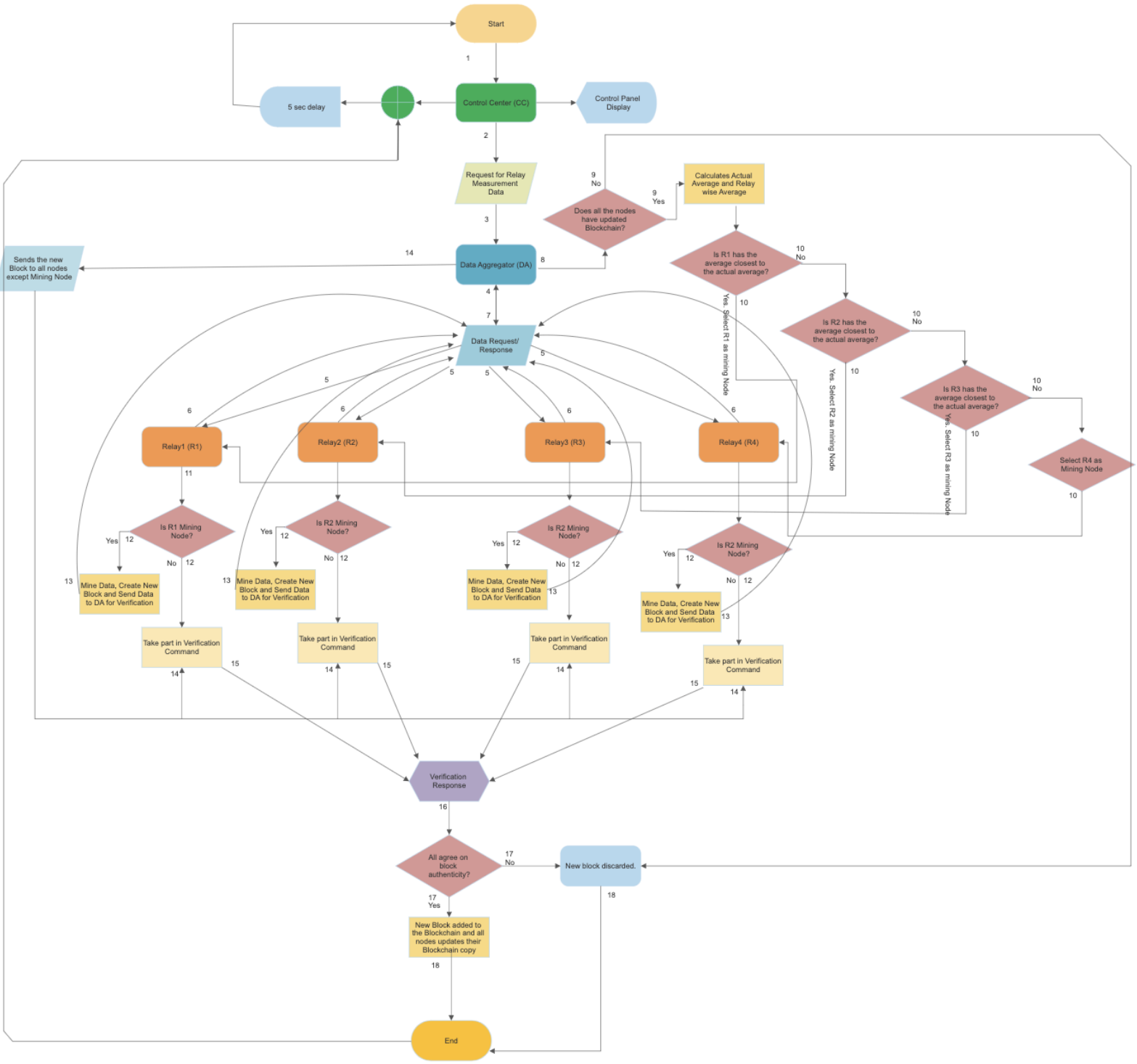


Figure 4: Flow chart of working principle

- **Purpose of the blockchain in the smart grid:**

Present CPSs like smart grids are turning into decentralized structure from a centralized structure. Centralized forms of CPS have several disadvantages including the chances of the cyber-attacks and non-resiliency. To be a complete decentralized grid, present smart grids need to adopt the blockchain based data acquisition system over current centralized SCADA system. Blockchain technology can enable decentralization to a smart grid and at the same time it can come to a great help to protect and secure the data.

- **Decentralization:**

As all the components of this implementation has the updated blockchain at the end of each cycles, the data can be treated as decentralized data.

- **Immutability:**

If a malicious actor tries to manipulate a data, the hash will be changed. He would need to mine all the node after the compromised node to hide the data manipulation

- **Non-repudiation:**

Each and every measured record is stored in the blockchain which ensures the non-repudiation characteristic of the blockchain

- **Scalability:**

More and more nodes can be added as field devices as well as multiple data aggregators can also be incorporated in order to harness the data with the slight modification and configuration of the code.

- **Security:**

Security has been implemented by introducing DNP3m protocol and encoding mechanism. Also, no relays except the mining node can know others' data. Moreover, indices have been used to cover up the identity of relays and only DA knows the indices-relays mapping. On a different note, as individual measurement data is aggregated into four broad categories, if , somehow, an attacker compromises the mining node, it won't get the relay wise individual measurement data.

- **Consensus and Incentive:**

A customized census mechanism has been developed to select the mining node in a random fashion. At first, each relay shares their own data to DA and then DA calculates the actual average and relay wise average. The relay which has the average closest to the actual average is selected as the mining node. As, no relays knows other data, it is difficult for an attacker to increase the possibility of being selected as the mining node. If a mining node successfully mines the data, it is rewarded with 10 points. However, in this private blockchain, the need of an incentive mechanism should be re-evaluated

7. Limitation and Future Works:

- For simplicity, exception handling part has not been incorporated in current code
- Also, the code execution stops immediately if it detects any manipulation.
- Better consensus mechanism needs to be deployed. Current consensus mechanism neither handles the case of a node to be selected as mining node in a repetitive manner nor incorporate the scenario of multiple nodes for same cycle (it would happen if, somehow, two nodes have the same distance from the actual average)
- Pseudonyms and different types of encryption mechanisms can be used to hide the identity of nodes and secure the data
- Smart contract can be deployed in the future version of this project
- Honeypot and other IDSs can be incorporated to tighten the security a bit more.

8. Conclusion:

The project demonstrates and simulate an example data acquisition scenario with blockchain technology that can be extended to any CPS and IoT devices. Mininet has been used to simulate the network topology on a virtual machine. Blockchain concept has been adopted to distribute the data and decentralize the whole data acquisition procedure. At the same time, certain security measures have been taken into account to protect data and verify the authenticity of data.