Secure Data Logging and Processing with Blockchain and Machine Learning

Leonel Lagos, PhD, PMP® (Principal Investigator)
Himanshu Upadhyay, PhD, PMP® (Co-Principal Investigator)
Team Members: Santosh Joshi, Pranav Gangwani
Applied Research Center
Florida International University (FIU)

Wenbing Zhao, PhD (Co-Principal Investigator)
Cleveland State University (CSU)
Project Description

• Secure Data Logging and Processing with Blockchain and Machine Learning research is focused on the development of platform to securely log and process sensor data in fossil power plant.

• The platform integrates two emerging technologies - blockchain and machine learning, and incorporates several innovative mechanisms to ensure the integrity, reliability, and resiliency of power systems.

• The goal is to protect the power plant from various cyberattacks such as false data injection and denial of service attacks using these technologies.
Various objectives of the research are as follows:

**Objective 1:** Secure authentication and identity verification of sensor nodes, actuators, and other equipment within a network

**Objective 2:** Develop a set of mechanisms that ensure only data sent by legitimate sensors are accepted and stored in the data repository

**Objective 3:** Develop data aggregation methodologies using machine learning / deep learning algorithms to minimize the noise / faulty data

**Objective 4:** Implement the blockchain technology to provide data security using secured IOTA framework & nodes
Secure Data Logging and Processing with Blockchain and Machine Learning

System Architecture

- **Power Plant Components**
  - Furnace
  - Boiler
  - Turbine
  - Stack
  - Condenser
  - Generator
  - Container
  - Pulverizer
  - Coal Supplier
  - Conveyor

- **Fossil Plant Center (FPC)**
  - Sensor Data Acquisition
  - IOTA Node Management
  - ML Model Management
  - Prediction Management
  - Administration
  - System Help

- **IOTA / Blockchain Framework**
  - IOTA Node 1
  - IOTA Private Nodes
  - Tangle

- **Secure Sensor Network (Virtual)**
  - Sensor Gateway
  - Temperature Sensor
  - Pressure Sensor
  - Gas Sensor
  - pH Sensor
  - Air Flow Sensor
  - Particulate Sensor
  - Liquid Flow Sensor
  - Level Sensor
  - Sensor Identification
  - Sensor Authentication

- **Data Aggregator – Artificial Intelligence**
  - Database Server
  - Tensorflow
  - Keras
  - Deep Learning
  - Machine Learning Server
Implementation of a Blockchain-Enabled Secure Sensing Data Processing and Logging System

- This system integrates sensor authentication and identification, data aggregation, storage of raw data in the database server, storage of aggregated data on blockchain, and visualization
Project Tasks

Task 1 - Secure Authentication and Identity Verification of Virtual Sensor Nodes

Task 2 - Data Aggregator / Machine Learning Platform

Task 3 - Secure Logging with Blockchain
Secure Data Logging and Processing with Blockchain and Machine Learning

Research Sponsored by
Department of Energy (DOE) - National Energy Technology Laboratory (NETL)

Developed by
Florida International University (FIU) - Applied Research Center (ARC)
Secure Authentication and Identity Verification of Virtual Sensor Nodes

• **Sensor Data Generation**: Generation of the sensor data based on the range of the sensors on the Fossil Power Plant components (Turbine, Furnace, Boiler, Stack)

• **Sensor Identity Management**: Sensor identification and authentication of every message received from sensors with Elliptic Curve Cryptography

• **Data Storage**: Storage of raw sensor data and aggregated data from data aggregator and IOTA framework

• **Machine Learning**: Build machine learning models and anomaly detection of sensor data
Fossil Power Plant Components and Sensors Considered for Research

The following fossil fuel power plant components and sensors were identified:

- Temperature sensor (Furnace)
- Pressure sensor (Boiler)
- Vibration sensor (Turbine)
- Gas sensor (Stack)
- Gas sensor (Furnace)
- Air Flow sensor (Furnace)
- Particulate sensor (Furnace)
- pH sensor (Boiler)
- Water Level sensor (Boiler)
### Plant Component Manager

#### Administer Plant Components Below

<table>
<thead>
<tr>
<th>MCU</th>
<th>Components</th>
<th>FPP Component ID</th>
<th>System Component Name</th>
<th>Description</th>
<th>Inserted On</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Boiler</td>
<td>Produces steam</td>
<td>5/20/2020 11:23:13 AM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>Furnace</td>
<td>Burns Coal to heat boiler</td>
<td>5/20/2020 7:03:50 PM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>Turbine</td>
<td>Generates electricity</td>
<td>5/20/2020 7:03:53 PM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>Stack</td>
<td>Expels gases</td>
<td>5/20/2020 7:23:23 PM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>Tubes</td>
<td>Water and air flow</td>
<td>5/20/2020 7:28:38 PM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>Container</td>
<td>Water collection</td>
<td>5/20/2020 7:29:10 PM</td>
</tr>
</tbody>
</table>
## Sensor Management

### Administer Sensor Information Below

<table>
<thead>
<tr>
<th>Sensor ID</th>
<th>Sensor Unique ID</th>
<th>System Component ID</th>
<th>Sensor Name</th>
<th>Description</th>
<th>Sensor Type</th>
<th>Sensor Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0018dc84e</td>
<td>1</td>
<td>Pressure</td>
<td>Boiler sensor</td>
<td>Pressure</td>
<td>Virtual</td>
</tr>
<tr>
<td>2</td>
<td>0018dc82b9</td>
<td>2</td>
<td>Temperature</td>
<td>Furnace sensor</td>
<td>Temperature</td>
<td>Virtual</td>
</tr>
<tr>
<td>3</td>
<td>0018dc983</td>
<td>4</td>
<td>Gas</td>
<td>Stack sensor</td>
<td>Gas</td>
<td>Virtual</td>
</tr>
<tr>
<td>4</td>
<td>0018dc58c</td>
<td>3</td>
<td>Vibration</td>
<td>Turbine sensor</td>
<td>Vibration</td>
<td>Virtual</td>
</tr>
<tr>
<td>7</td>
<td>0018dc983</td>
<td>2</td>
<td>Gas</td>
<td>Furnace sensor</td>
<td>Gas</td>
<td>Virtual</td>
</tr>
</tbody>
</table>
Sensor Dashboard

Dashboard

Furnace Sensors
Latest readings from furnace sensors

- **Temp**: 539.2°F
  - 4/15/2022 11:16:58 AM
- **Air Flow**: 11.24
  - 7/15/2020 12:08:40 PM
- **Gas**: 92.44
  - 4/19/2022 4:33:50 PM
- **Particule**: 16.07
  - 7/15/2020 12:08:41 PM

Boiler Sensors
Latest readings from Boiler sensors

- **Temp**: 478.7°F
  - 4/13/2022 10:26:01 AM
- **Air Flow**: 23.10
  - 04/15/2022 10:01:50 PM
- **Gas**: 15.51
  - 04/16/2022 08:35:11 AM
- **Particule**: 26.36
  - 04/15/2022 02:28:15 PM

Stack Sensor
Latest readings from Stack sensors
## Fossil Power Plant Components and Sensors Value Ranges

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Unit</th>
<th>System Component</th>
<th>Benign Value Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature</td>
<td>°C</td>
<td>Turbine</td>
<td>1.81 – 37.11</td>
</tr>
<tr>
<td>Temperature</td>
<td>°C</td>
<td>Boiler</td>
<td>540 – 570</td>
</tr>
<tr>
<td>Pressure</td>
<td>hPa</td>
<td>Turbine</td>
<td>992.89 – 1033.3</td>
</tr>
<tr>
<td>Pressure</td>
<td>hPa</td>
<td>Boiler</td>
<td>600 – 2465</td>
</tr>
<tr>
<td>Vibration</td>
<td>µm (pk-pk)</td>
<td>Turbine</td>
<td>13.31 – 14.07</td>
</tr>
<tr>
<td>O₂</td>
<td>% in flue</td>
<td>Stack</td>
<td>1.6 – 4.1</td>
</tr>
<tr>
<td>Gas</td>
<td>ppm</td>
<td>Stack</td>
<td>10 – 1000</td>
</tr>
<tr>
<td>Gas</td>
<td>ppm</td>
<td>Boiler</td>
<td>0 – 100</td>
</tr>
<tr>
<td>CO₂ Gas</td>
<td>% in flue</td>
<td>Stack</td>
<td>8 – 10</td>
</tr>
<tr>
<td>NO₂ Gas</td>
<td>% in flue</td>
<td>Stack</td>
<td>10 – 12</td>
</tr>
<tr>
<td>pH</td>
<td>-</td>
<td>Boiler</td>
<td>7 - 9</td>
</tr>
</tbody>
</table>
Sensor Data Generation

- Research was conducted to analyze the various sensors mounted on the FPP system components.
- Investigated normal sensor value ranges for different sensors.
- Synthetic data generated based on the range (Normal / Benign and Malicious) by selecting between lower range – offset and higher range + offset.
Generate Sensor Data

Enter Information Below To Generate Sensor Data

Select Component: Furnace
Select Sensor: Gas - 00F88983
Sensor Behavior: Normal
Batch Size: 10
Offset: 0

Generate Sensor Data

<table>
<thead>
<tr>
<th>Sensor ID</th>
<th>Sensor Value</th>
<th>Sensor Time</th>
<th>Inserted On</th>
</tr>
</thead>
<tbody>
<tr>
<td>00F88983</td>
<td>15.02</td>
<td>04/19/2022 16:34:49</td>
<td>4/19/2022 4:33:49 PM</td>
</tr>
<tr>
<td>00F88983</td>
<td>21.37</td>
<td>04/19/2022 16:35:49</td>
<td>4/19/2022 4:33:49 PM</td>
</tr>
<tr>
<td>00F88983</td>
<td>13.63</td>
<td>04/19/2022 16:36:49</td>
<td>4/19/2022 4:33:49 PM</td>
</tr>
<tr>
<td>00F88983</td>
<td>62.58</td>
<td>04/19/2022 16:37:49</td>
<td>4/19/2022 4:33:49 PM</td>
</tr>
<tr>
<td>00F88983</td>
<td>51.5</td>
<td>04/19/2022 16:38:49</td>
<td>4/19/2022 4:33:50 PM</td>
</tr>
<tr>
<td>00F88983</td>
<td>77.48</td>
<td>04/19/2022 16:39:49</td>
<td>4/19/2022 4:33:50 PM</td>
</tr>
<tr>
<td>00F88983</td>
<td>29.29</td>
<td>04/19/2022 16:40:49</td>
<td>4/19/2022 4:33:50 PM</td>
</tr>
<tr>
<td>00F88983</td>
<td>64.69</td>
<td>04/19/2022 16:41:49</td>
<td>4/19/2022 4:33:50 PM</td>
</tr>
<tr>
<td>00F88983</td>
<td>74.59</td>
<td>04/19/2022 16:42:49</td>
<td>4/19/2022 4:33:50 PM</td>
</tr>
</tbody>
</table>
Sensor Data Visualization

Select Component And Sensor To View

Select Component: Furnace
Select Sensor: Gas
Start Date: mm/dd/yyyy
End Date: mm/dd/yyyy

Sensor Value | Inserted On
-------------|------------
92.44        | 4/19/2022 4:33:50 PM
74.59        | 4/19/2022 4:33:50 PM
64.69        | 4/19/2022 4:33:50 PM
29.29        | 4/19/2022 4:33:50 PM
77.48        | 4/19/2022 4:33:50 PM
51.5         | 4/19/2022 4:33:50 PM
62.58        | 4/19/2022 4:33:49 PM
13.63        | 4/19/2022 4:33:49 PM
21.37        | 4/19/2022 4:33:49 PM
15.02        | 4/19/2022 4:33:49 PM
Sensor Identification and Authentication

• Sensor identity management is essential in mitigating the different attacks which exploit weakness in sensor authentication
  ➢ Injection attacks: The attacker would inject faulty or fake sensing data in the system
  ➢ Sybil attacks: A compromised sensor would impersonate as many legitimate sensors as possible
  ➢ Spoofing attacks: An attacker would pretend to be a legitimate sensor

• Sensor identification and authentication is implemented using Elliptic Curve Cryptography
Sensor Identification and Authentication

Sensor Identity

Select Component And Sensor To View

Select Component:  
Select Sensor:  
Start Date: mm/dd/yyyy  
End Date: mm/dd/yyyy  

Sensor UnqiueID  Sensor Value  Sensor Behavior  Sensor Hash  Sensor Verification  Inserted On
00F7d84c  2074.87  0  System.Byte[]  1  4/18/2022 4:38:35 PM
00F7d84c  1402  0  System.Byte[]  1  4/18/2022 4:38:35 PM
00F7d84c  2046.09  0  System.Byte[]  1  4/18/2022 4:38:35 PM
00F7d84c  641.67  0  System.Byte[]  1  4/18/2022 4:38:35 PM
00F7d84c  1961.17  0  System.Byte[]  1  4/18/2022 4:38:35 PM
00F7d84c  1058.08  0  System.Byte[]  1  4/18/2022 4:38:35 PM
00F7d84c  949.64  0  System.Byte[]  1  4/18/2022 4:38:35 PM
00F7d84c  2418.51  0  System.Byte[]  1  4/18/2022 4:38:35 PM
00F7d84c  1201.26  0  System.Byte[]  1  4/18/2022 4:38:35 PM
00F7d84c  713.22  0  System.Byte[]  1  4/18/2022 4:38:35 PM
Sensor Identification and Authentication

- A private key is generated for every system component
- A signature is generated from the data string which is encoded using UTF-8
- The data string is a concatenation of the properties of the sensor that include the sensor ID, batch number, sensor output value, and time.
- A public key is used to verify the signature and record is updated in the database.

```python
def append_with_Signature(data_string, priv_key, public_key):
    signer_priv_key = SigningKey.from_pem(priv_key)
    Encode String
    data_bytes = data_string.encode("utf-8")
    # Sign Data
    signature = signer_priv_key.sign(data_bytes)
    return signature

def validateSignatureInt(signature, public_key, data_string):
    verify_key = VerifyingKey.from_pem(public_key)
    data_bytes = data_string.encode("utf-8")
    try:
        valid = verify_key.verify(signature, data_bytes)
        return 1
    except BadSignatureError:
        print("BAD SIGNATURE")
        return 0
    #finally:
    #return valid_signature
```
Data Aggregation & Machine Learning Platform

• The following Machine learning and Deep Learning algorithms for anomaly detection are implemented:
  ➢ One-class Support Vector Machines.
  ➢ Isolation Forest
  ➢ Elliptic Envelope
  ➢ Local Outlier Factor
  ➢ Agglomerative Clustering
  ➢ AutoEncoders

• Scikit-Learn, Keras, and TensorFlow frameworks are used for implementation
Machine Learning Prediction

Prediction Building

Build Prediction From Built Models

Select Component: Boiler  Select Sensor: Pressure  Prediction Name:  Prediction Description:

Start Date: mm/dd/yyyy  End Date: mm/dd/yyyy  Select Model: Temp Sensor Model  Select Algorithm:  

Go  Clear

© 2022 - Secure Data Logging and Processing with Blockchain and Machine Learning

FIU-ARC
## Machine Learning Model Management

### Manage Models

<table>
<thead>
<tr>
<th>Model ID</th>
<th>Model Name</th>
<th>Description</th>
<th>UserName</th>
<th>Model Type</th>
<th>Feature Name</th>
<th>Label Name</th>
<th>Is Active</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Temp Sensor Model</td>
<td>Temperature Sensor Model</td>
<td>Admin</td>
<td>User</td>
<td>SensorValue</td>
<td>TemperatureSensor</td>
<td>☑</td>
<td>Completed</td>
</tr>
<tr>
<td>2</td>
<td>Gas</td>
<td>Gas Sensor Model</td>
<td>Admin</td>
<td>User</td>
<td>SensorValue</td>
<td>GasSensor</td>
<td>☑</td>
<td>Created</td>
</tr>
<tr>
<td>3</td>
<td>Gas Sensor Model</td>
<td>Furnace Sensor Model</td>
<td>Admin</td>
<td>User</td>
<td>SensorValue</td>
<td>VirtualGasSensor</td>
<td>☑</td>
<td>Created</td>
</tr>
</tbody>
</table>

© 2022 - Secure Data Logging and Processing with Blockchain and Machine Learning

FIU-ARC
## Anomaly Detection Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-class SVM</td>
<td>0.9875</td>
</tr>
<tr>
<td>Elliptic Envelope</td>
<td>0.93375</td>
</tr>
<tr>
<td>Isolation Forest</td>
<td>0.9875</td>
</tr>
<tr>
<td>Local Outlier Factor</td>
<td>0.9875</td>
</tr>
<tr>
<td>Agglomerative Clustering</td>
<td>0.4825</td>
</tr>
<tr>
<td>AutoEncoder</td>
<td>0.9775</td>
</tr>
</tbody>
</table>
Secure Logging with Blockchain

The objectives of the IOTA implementation include:

- Security
- High Scalability and Throughput
- Data Integrity
- Privacy
- Traceability and Auditability
- Secure Storage
IOTA Private Framework at FIU

A three node (Coordinator, Neighbor node- Node_1, Neighbor node- Node_2) private tangle is implemented at FIU
Coordinator and Neighbor Nodes

Coordinator Node:
• The Coordinator is a client that sends signed messages called milestones that nodes trust and use to confirm messages
• Nodes rely on the Coordinator to reach a consensus, therefore each one is hard-coded with the address of the Coordinator
• Nodes use this address to validate the Coordinator's signatures in milestones

Neighbor Nodes:
• Neighbor nodes are mutually connected and communicate directly with each other on the same IOTA network
• To synchronize their ledgers with the rest of the network, all nodes send and receive transactions among their neighbors
• After receiving a new transaction, nodes check that they have the transaction's history in its ledger
• If a node is missing any transactions, it communicates with neighbor node to synchronize with the rest of the network
Aggregation and Checksum Generation

Aggregation:
• Sensor data aggregation is performed in batches and insert into the tangle.
• Calculated the mean, median, mode, Std-Dev, Variance, Min and Max of the generated sensor data batch.
• This gives some meaningful insights about the generated data such as the range, average sensor value and how the sample of data is distributed.

Checksum:
• Checksum is calculated to verify the integrity of the data during transfer.
• Checksum is an aggregated hash value of the sensor data using SHA-256.
• The output hash value will always be the same if the input is same.
• Small change in the input value will result in a completely different hash value.
IOTA Storage and Retrieval of the Sensor Data

Sensor Data Storage into IOTA

- The IOTA transaction is encoded as JSON structure and stored in the IOTA tangle.
- The transaction is broadcasted as a message in the private tangle.
- Once the transaction is sent to the IOTA, a Unique Message ID is generated which refers to the current transaction.

Sensor Data Retrieval from IOTA:

- This module queries the private tangle with the Message ID and the IOTA nodes sends the response after fetching the data from the tangle.
- The retrieved data is then decoded into human readable format and stored with transaction timestamp.

```json
netl_structure = {
    "sensor_id" : sensor_id,
    "system_batch_id" : batch_id,
    "checksums" : checksum,
    "mean" : sensor_mean,
    "mode" : sensor_mode,
    "median" : sensor_median,
    "min" : sensor_min,
    "max" : sensor_max,
    "standard_deviation" : std_dev,
    "variance" : sensor_var
}
```

IOTA Data Structure
IOTA Data Storage in Centralized Database

- Once the data was fetched and retrieved from the private tangle, the data is stored in the centralized database

<table>
<thead>
<tr>
<th>Columns</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SensorID</td>
<td>6ad99030</td>
</tr>
<tr>
<td>SensorBatchID</td>
<td>6ad99030_1</td>
</tr>
<tr>
<td>Checksum</td>
<td>6b186fa5019b4e92bf5713de60efc1f8fdf8604d44496e833c3e4b48b59fbb05</td>
</tr>
<tr>
<td>SensorMean</td>
<td>554.995</td>
</tr>
<tr>
<td>SensorMode</td>
<td>544.57</td>
</tr>
<tr>
<td>SensorMedian</td>
<td>555.04</td>
</tr>
<tr>
<td>SensorStandardDeviation</td>
<td>8.679</td>
</tr>
<tr>
<td>SensorVariance</td>
<td>75.32</td>
</tr>
<tr>
<td>SensorMin</td>
<td>540.00</td>
</tr>
<tr>
<td>SensorMax</td>
<td>569.997</td>
</tr>
<tr>
<td>TransactionTimestamp</td>
<td>2022-03-29 21:46:38.000</td>
</tr>
<tr>
<td>MessageID</td>
<td>8e23514bc49702c6d9986fb75ac8ae29637d75e05934e4d680bad16fe33ba5df</td>
</tr>
</tbody>
</table>
System Demonstration
“Secure Data Logging and Processing with Blockchain and Machine Learning”
Conclusion

• Prototype for Secure Logging and Processing with Blockchain and Machine Learning for FPP is developed.
• Authentication and Identification of Sensor Data.
• Machine learning platform for Anomaly Detection and Data Aggregation.
• Secure two-level logging with IOTA distributed ledger.
• Blockchain technology to provide sensor data security using IOTA framework.
• Visualization of sensor data with sophisticated analytics capabilities.
Q & A

Thank You
Acknowledgment: "This material is based upon work supported by the Department of Energy Award Number DE- FE0031745."

Disclaimer: "This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof."