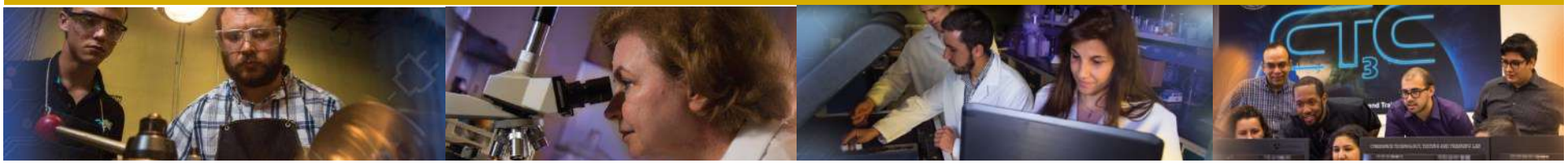

# Secure Data Logging and Processing with Blockchain and Machine Learning

**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)**

FLORIDA INTERNATIONAL UNIVERSITY



NATIONAL ENERGY TECHNOLOGY LABORATORY

# Project Description

FIU Applied Research Center

- 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

CLEVELAND STATE UNIVERSITY 1964

NATIONAL ENERGY TECHNOLOGY LABORATORY

# Project Objectives

FIU Applied Research Center

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

CLEVELAND STATE UNIVERSITY 1964

NATIONAL ENERGY TECHNOLOGY LABORATORY

# Project Tasks

FIU Applied Research Center

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

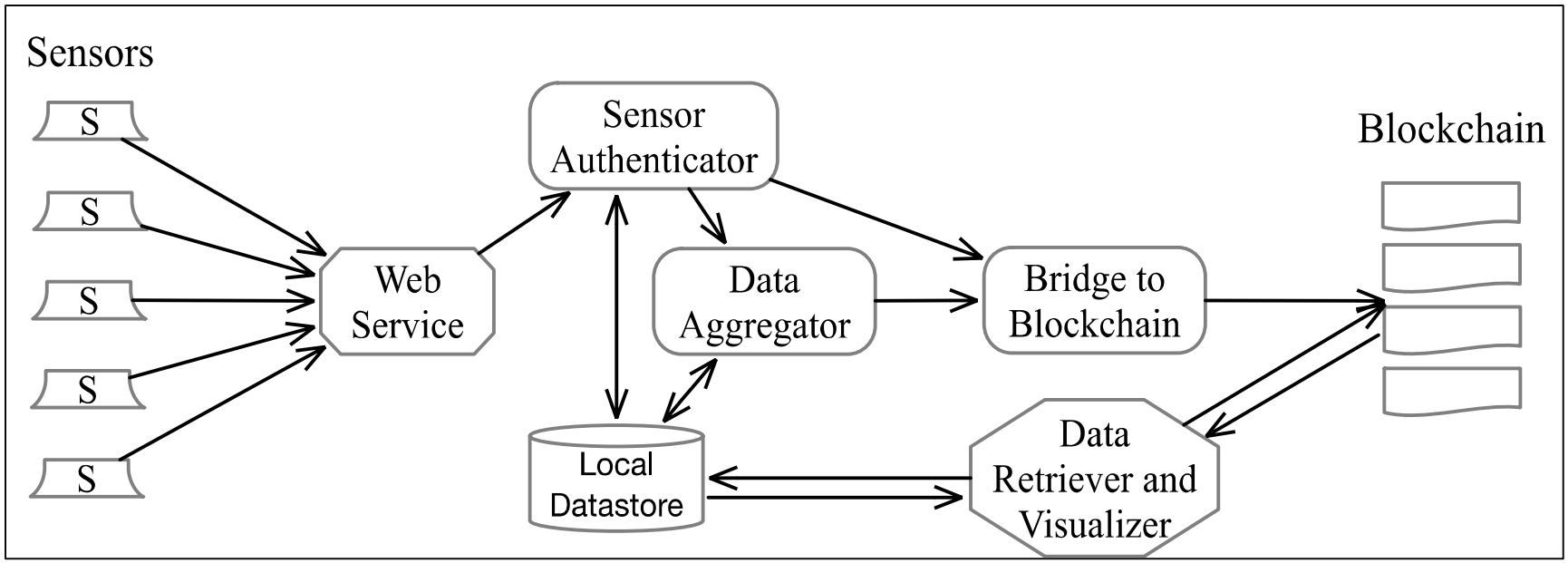
**Task 2 - Data Aggregator Machine Learning Platform**

**Task 3 - Secure Logging with Blockchain**

CLEVELAND STATE UNIVERSITY 1964

# Implementation of a Blockchain-Enabled Secure Sensing Data Processing and Logging System

- This system integrates sensor authentication and identification, data aggregation, local storage of raw data, storage of aggregated data on blockchain, and visualization



NATIONAL ENERGY TECHNOLOGY LABORATORY

# System Design Goals

FIU Applied Research Center

- The system only accepts sensing data submitted by eligible sensors
- The system only stores a small fraction of the amount of sensing data collected from sensors on the public blockchain for safe-keeping, both to not overwhelm the blockchain and to minimize the associated financial cost
- The system offers an efficient way to retrieve the data both logged on the public blockchain, and the raw data stored at the local data store

CLEVELAND STATE UNIVERSITY 1964

NATIONAL ENERGY TECHNOLOGY LABORATORY

# Major Components in the System

FIU Applied Research Center

- **Web service:** The interaction between the sensors and the sensing data processing and logging system is enabled by the CherryPy, a Python-based Web framework
  - ✓ This framework makes it very easy to develop object-oriented Python programs on the server side
  - ✓ All messages between the sensors and the system are encoded as JSON documents running on top of the HTTP protocol (through the POST or GET methods)
- **Sensor authenticator:** This component is in charge of handling the initial sensor enrollment phase and authenticating every message received from any sensor
- **Data aggregator:** This component is in charge of aggregating the received raw sensing data, compute the aggregated data with the corresponding Merkle root, store both the raw data and the Merkle root on the local datastore, pass the aggregated data and the Merkle root to the bridge to blockchain for safe-keeping

CLEVELAND STATE UNIVERSITY 1964

NATIONAL ENERGY TECHNOLOGY LABORATORY

# Major Components in the System

FIU Applied Research Center

- **Local datastore:** This component stores the raw data in conjunction with the Merkle root. In our implementation, we use MongoDB, but any database system could be used such as MySQL with minor modification to the code
- **Bridge to blockchain**: This component handles the communication between our system and the public blockchain (in our current implementation is IOTA)
- **Sensors:** Various sensors would be configured to collect sensing data and report them to the server components
  - For the convenience of development, we used a python program to simulate various sensors

CLEVELAND STATE UNIVERSITY 1964

# Major Components in the System

- **Data retriever and visualizer:** This is a stand-alone component to fetch the raw data stored at the local datastore or the aggregated data stored on the public blockchain. All such operations are read-only. This component would provide a graphic interface for data query and visual display via various charts. This component has several objectives:
  - ✓ to facilitate decision making and planning based on the aggregated data;
  - ✓ to facilitate reporting and auditing as needed
  - ✓ to perform forensics analysis in case of issues with the system being monitored

❖ **This component is under development**

NATIONAL ENERGY TECHNOLOGY LABORATORY

# Sensor Identification and Authentication

FIU Applied Research Center

- Sensor identity management is essential in mitigating the spoofing attacks, Sybil attacks, and injecting attacks because all these attacks exploit weakness in sensor authentication
- In recent literature on sensor identity management, a very popular theme is to use a physical unclonable function (PUF) to authenticate sensors
- All proposed PUF-based solutions in the literature require a trusted node to store the sensor ID and the set of CRPs. The trusted node is either co-located with the authenticator or is only accessible by the authenticator
- Unfortunately, we do not think this is an acceptable approach for practical systems because once the so-called “trusted” node is compromised, for example, by leaking the stored set of CRPs, anyone who has possession of the set of CRPs could impersonate the sensor

CLEVELAND STATE UNIVERSITY 1964

NATIONAL ENERGY TECHNOLOGY LABORATORY

# Sensor Identification and Authentication

FIU Applied Research Center

- Inspired by some of the design principles of public blockchain, we decide to perform sensor identification and authentication based on the public-key cryptography, and rely on the immutability offered by blockchain to store the enrollment information
- The biggest difference between the proposed approach and the PUF-based approach is that the sensor never shares its private information to the authenticator and the blockchain
- Indeed, the public blockchain operates exactly like this and the private key is held only by the user. The authentication of the user in blockchain is done entirely by verifying the digital signature. Essentially, the digital signature is the zero-knowledge proof of the possession of the private key. We take the same approach for sensor identity management

CLEVELAND STATE UNIVERSITY 1964

NATIONAL ENERGY TECHNOLOGY LABORATORY

# Sensor Identification and Authentication

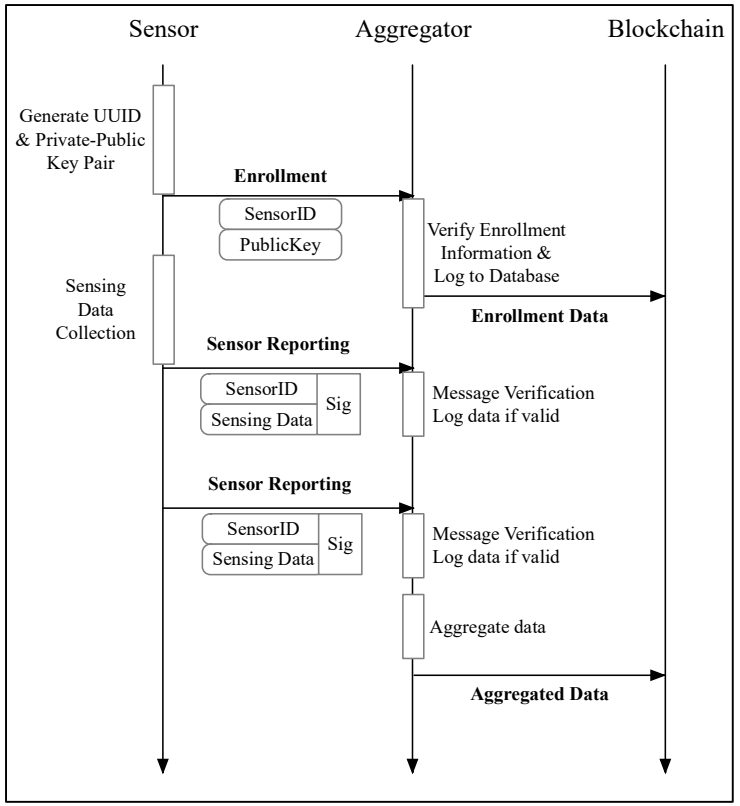
FIU Applied Research Center

- When we adopt the blockchain approach for sensor authentication, however, we must ensure that each sensor has one and only one identity. We accomplish by the following two mechanisms:
  - ✓ When configuring a sensor, a unique identifier and one pair of private-public keys are created. The unique identifier will be used as the sensor ID. The sensor ID and the public key will be included in all messages sent to the authenticator that verifies the messages. Furthermore, all messages will be digitally signed using the private key.
  - ✓ Before the sensor can report a sensing reading to the aggregator, it must first enroll with the aggregator. The enrollment message contains the sensor ID and the public key. The aggregator will store the sensor ID and the corresponding public key locally as well as uploading to the blockchain for safe keeping
  - ✓ Because the enrollment phase is the most vulnerable step in the system, it must be done in a controlled environment. For example, the sensor device should be placed physically together with the computer running our system in close vicinity in the same room. Allowing any device to enroll without physical security will be detrimental to the security of the authentication scheme used here

CLEVELAND STATE UNIVERSITY 1964

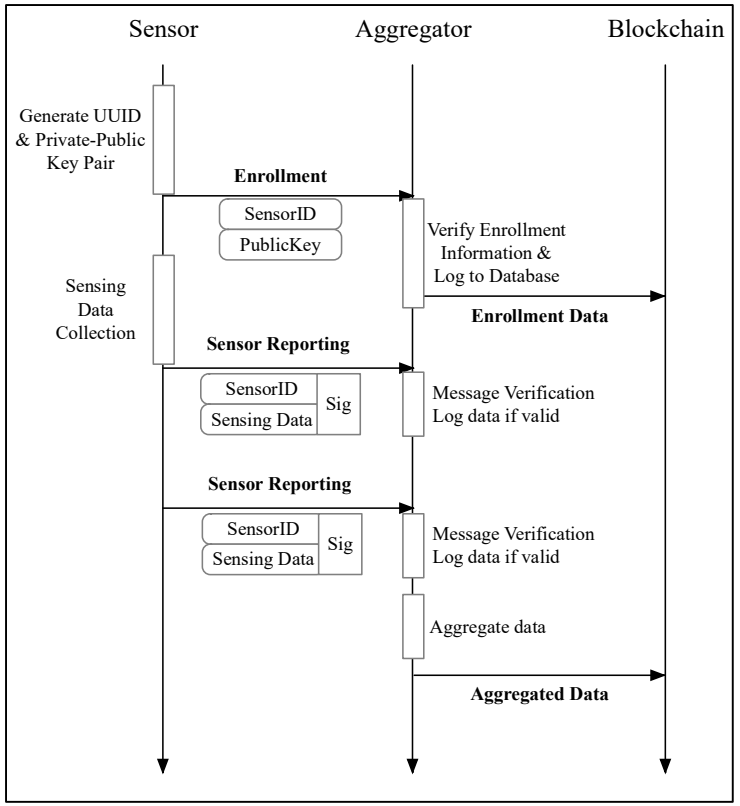
# Sensor Identification and Authentication

- As soon as a sensor device is turned on and configured with the unique sensor ID and a pair of public-private keys, it attempts to enroll itself with the authenticator
- The sensor ID and the private key are written in stable storage at the sensor device so that if the sensor device is rebooted the same sensor and the private key will be used
- The device would perform enrollment only the first time it is configured
  - ✓ Like many public blockchains, we also choose to use the Elliptic Curve Cryptography (ECC)
  - ✓ In ECC, the private key is generated using a passphrase as the seed. The public key is derived from the private key
  - ✓ ECC public key and the signature are both much shorter than those produced by RSA



# Sensor Identification and Authentication

- The sensor authenticator will accept an enrollment request if it is for a new device
- It will add an entry for the device in its enrollment table. The entry consists of a sensor ID and the corresponding public key
- The authenticator can create a transaction and store the enrollment entry immediately to the blockchain, or it may choose to aggregate a set of enrollment entries and then record them on the blockchain

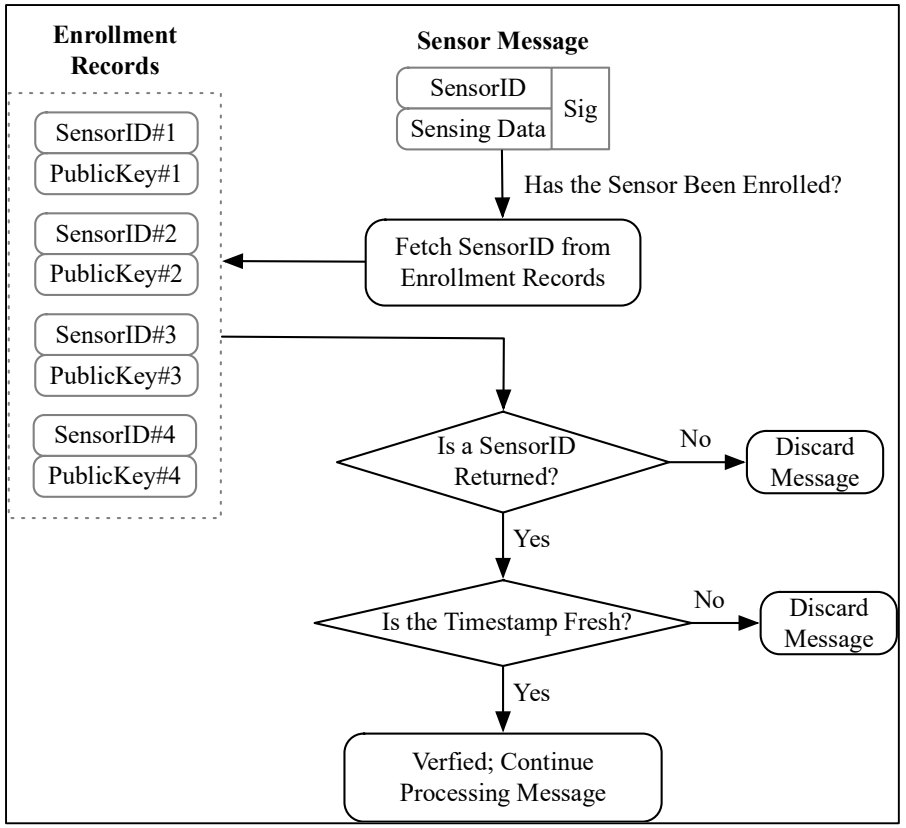


NATIONAL ENERGY TECHNOLOGY LABORATORY

# Sensor Identification and Authentication

FIU Applied Research Center

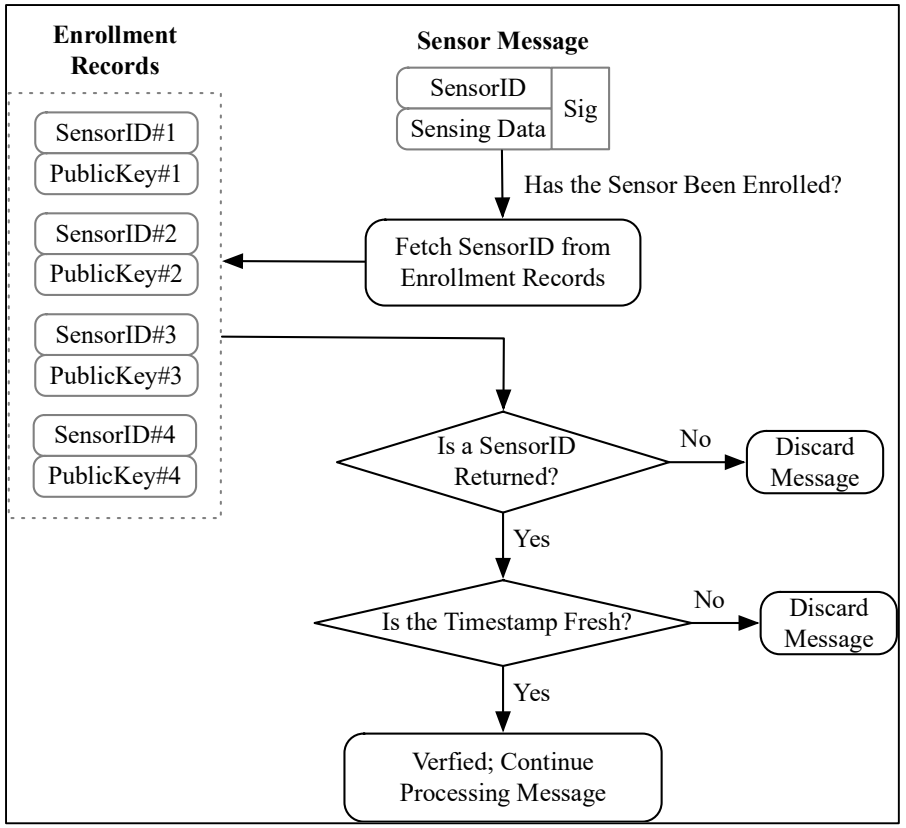
- The authenticator verifies every message received from a sensor
- All messages sent by a sensor must be digitally signed and it must include a sensor ID and the current timestamp
- The sensor authenticator would first retrieve the sensor ID from the message, then it will perform a lookup in its enrollment table
- If an entry is found, the corresponding public key is retrieved




CLEVELAND STATE UNIVERSITY 1964

# Sensor Identification and Authentication

- The public key is then used to verify the digital signature
- The message will be discarded if the signature validation fails and so does if an entry is not found
- To detect replay attacks, the authenticator would further check the timestamp included in the message
- If the timestamp is the same or older than the last seen timestamp from the sensor, the message is also discarded
- The message is accepted only when it has passed all these verification steps



NATIONAL ENERGY TECHNOLOGY LABORATORY

# Sensing Data Aggregation and Logging

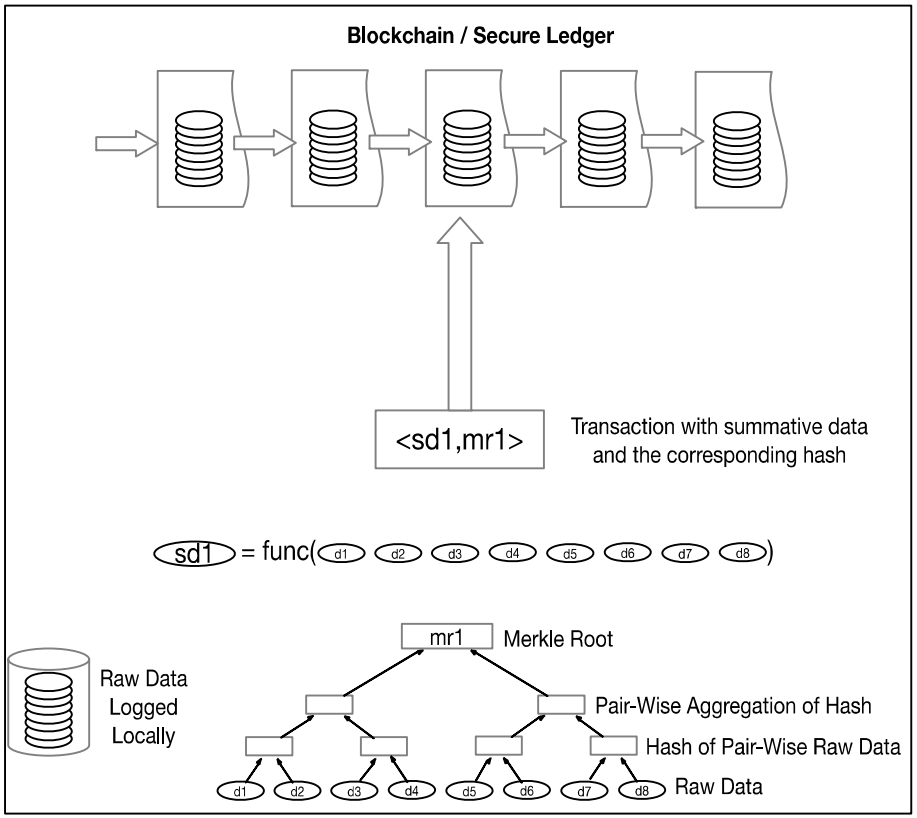
FIU Applied Research Center

- Sensing data are aggregated periodically, and only the aggregated data are placed on the blockchain for safe-keeping for several reasons we have stated previously
- How to aggregate data varies based on the application needs
  - ✓ In our implementation, we choose to aggregate data per sensor ID per sensor type, and compute the mean, maximum, minimum, and standard deviation values of the batch of the data
  - ✓ We could also apply filtering to minimize jitters, or even using machine learning to remove faulty data
- To ensure the security of the raw data, we designed a strong linkage mechanism to extend the immutability property offered by the public blockchain to the raw data stored at the local datastore
- The basic idea is to compute the root of a Merkle tree (or Merkle root for short) using the raw data items as the leaf nodes for the tree, and putting the root together with the aggregated data on the blockchain

CLEVELAND STATE UNIVERSITY 1964

# Sensing Data Aggregation and Logging

- The Merkle root is also stored together with each batch of raw data, both as a checksum for integrity check on the raw data, and as a way to match the raw data and the aggregation
- This Merkle root establishes the strong link between the aggregated data placed on the blockchain, and the corresponding raw data recorded at the local datastore



NATIONAL ENERGY TECHNOLOGY LABORATORY

# Sensing Data Retrieval

FIU Applied Research Center

- Because the public blockchain is shared among many users, it is essential to have a way to query the blockchain for our own aggregated data
- Every blockchain offers a way to users to query. Typically, each transaction placed on the blockchain is assigned a transaction ID and it is returned to the user, which enables the user to query that particular transaction at a later time
- Some public blockchains offer a more desirable way to query the data on the blockchain. For example, IOTA allows to query one or more transactions based on a user-assigned tag. In our system, we use such a feature to search the aggregated data on the granularity of day

CLEVELAND STATE UNIVERSITY 1964

NATIONAL ENERGY TECHNOLOGY LABORATORY

# Sensing Data Retrieval

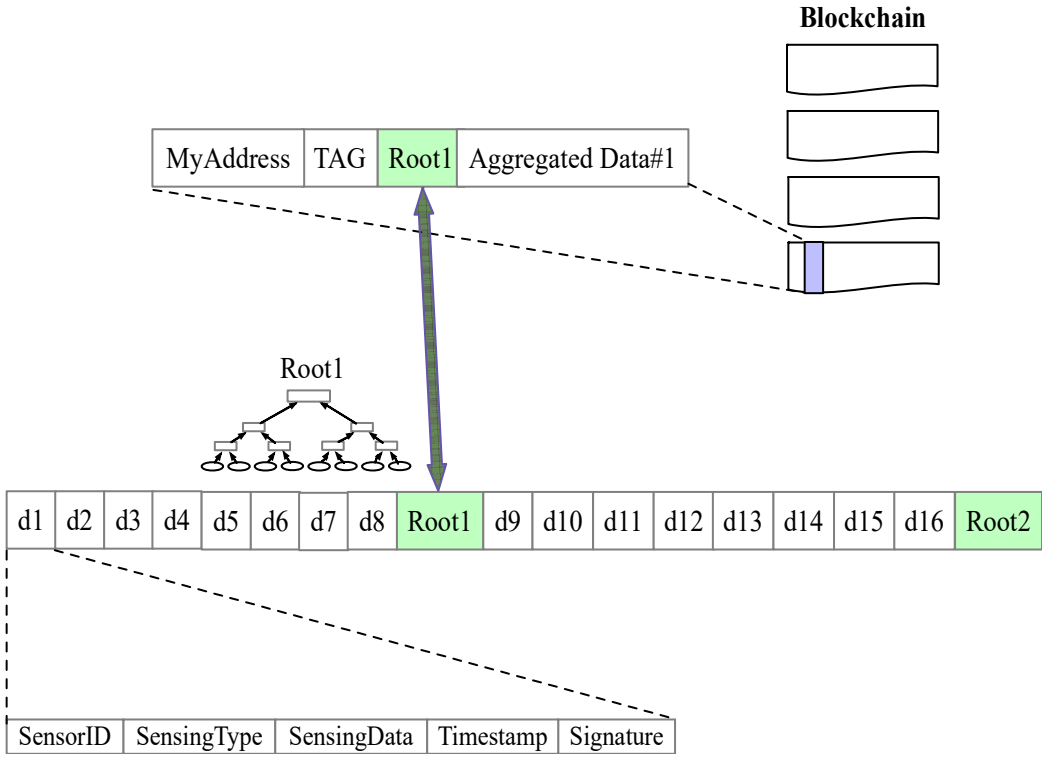
FIU Applied Research Center

- Because IOTA allows only alphabetical letters and number 9 in the tag, we have to develop a mechanism to encode the year, month, and day as alphabetical letters and as concise as possible
- The idea is to use Roman numerals to convert the year and day into letters, and use the short-hand 3-letter notation for the month. In between the year and month, month and day, number 9 is used as the separator
- For example, 2021 March 4 would be denoted as MMXXI9MAR9IV. The mechanism can be trivially extended to allow search with finer granularity, such as hour, sensor ID, and sensor type

CLEVELAND STATE UNIVERSITY 1964

# Sensing Data Retrieval

- The local storage of the raw data are designed to facilitate the search of the corresponding batch of raw data given the aggregated data
- Because the Merkle root is always stored together with the batch of raw data, and with the corresponding aggregated data, **the root is used to search for the batch of raw data**
- Once the root is found in the log, the preceding raw data items can then be retrieved

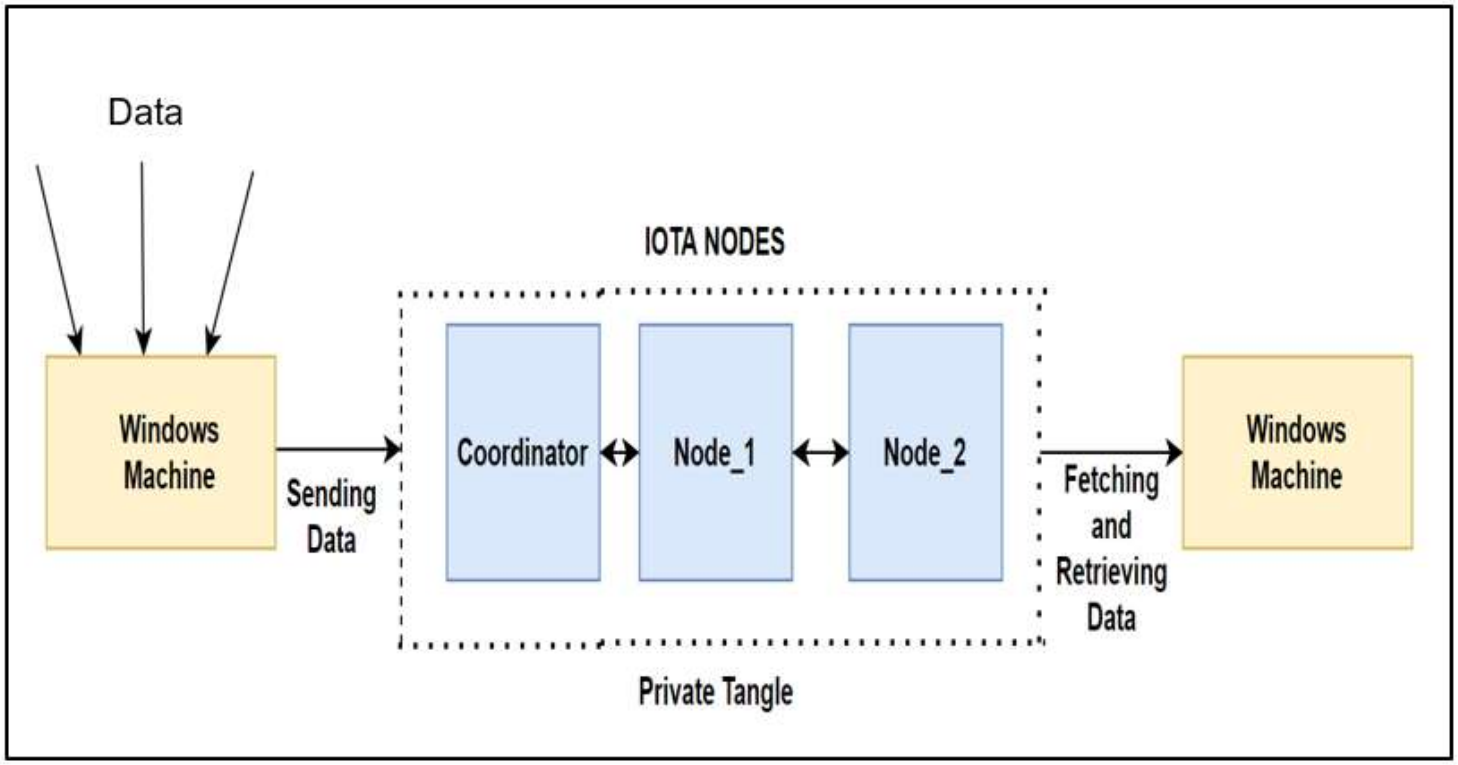


NATIONAL ENERGY TECHNOLOGY LABORATORY

# IOTA Private Framework at FIU

FIU Applied Research Center

A three (Coordinator, Neighbor node- Node_1, Neighbor node- Node_2) node private tangle has been implemented as shown below:



NATIONAL ENERGY TECHNOLOGY LABORATORY

# Coordinator Node

FIU Applied Research Center

- The Coordinator is a client that sends signed messages called milestones that nodes trust and use to confirm messages
- In the current IOTA networks, nodes rely on the Coordinator to reach a consensus, therefore each one is hard-coded with the address of a Coordinator
- Nodes use this address to validate the Coordinator's signatures in milestones
- Messages in the Tangle are considered for confirmation only when they are directly or indirectly referenced by a milestone (signed Txn) that has been validated by nodes
- To allow them to recognize milestones, all nodes in the same IOTA network are configured with the **Merkle root address** of a Coordinator that they trust to confirm messages

CLEVELAND STATE UNIVERSITY 1964

NATIONAL ENERGY TECHNOLOGY LABORATORY

# Neighbor Nodes (Node_1 & Node_2)

FIU Applied Research Center

- Neighbors are nodes that are mutually connected and that 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 starts to ask its neighbors for them to become synchronized with the rest of the network

CLEVELAND STATE UNIVERSITY 1964

NATIONAL ENERGY TECHNOLOGY LABORATORY

# Sending Data to Tangle

FIU Applied Research Center

- A Python Script (Send.py) was developed to perform following functions
  - ➢ Establishes a connection to our private tangle using a REST API
  - ➢ Generates a random address for sending transactions
  - ➢ Loads the sensor data into JSON format
  - ➢ Encrypts the sensor data using a symmetrical key
  - ➢ Prepares the transaction bundle and bundle hash and sends it to our private tangle


```
"1": {
    "S.R No.": "1",
    "ts": "1594512094.3859744",
    "device": "b8:27:eb:bf:9d:51",
    "co": "0.004955938648391245",
    "humidity": "51.0",
    "light": "False",
    "lpg": "0.00765082227055719",
    "motion": "False",
    "smoke": "0.02041127012241292",
    "temp": "22.7"
},
"2": {
    "S.R No.": "2",
    "ts": "1594512094.7355676",
    "device": "00:0f:00:70:91:0a",
    "co": "0.00284008860710157",
    "humidity": "76.0",
    "light": "False",
    "lpg": "0.005114383400977071",
    "motion": "False",
    "smoke": "0.013274836704851536",
    "temp": "19.700000762939453"
},
```


CLEVELAND STATE UNIVERSITY 1964

NATIONAL ENERGY TECHNOLOGY LABORATORY

# Fetching Data from Tangle

FIU Applied Research Center

- A Python Script (Fetch.py) was developed to fetch the transactions and the data from the tangle
  - ➢ It fetches the whole bundle using a bundle hash
  - ➢ Concatenates all the transactions in the bundle
  - ➢ Decrypts the data using the symmetrical encryption key

- This implementation ensures that the security, integrity and privacy of the data is obtained as the data is stored in a private tamper-proof ledger

CLEVELAND STATE UNIVERSITY 1964

NATIONAL ENERGY TECHNOLOGY LABORATORY

# Sensor Identification

FIU Applied Research Center

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)

CLEVELAND STATE UNIVERSITY 1964

NATIONAL ENERGY TECHNOLOGY LABORATORY

# Sensor Data Acquisition

FIU Applied Research Center

- The team deployed two python scripts
  1. LiveInsert.py
  2. Virtual.py
- Virtual.py script generates artificial sensor data from virtual sensors based on their range of parameters and aggregates it into the Raspberry Pi
- LiveInsert.py script stores the live sensor data from Raspberry Pi into the database

```python
import paho.mqtt.client as mqtt
import pymssql        # Necessary for database
import time
import random
import numpy as np
from multiprocessing import freeze_support, Pool, Process

#creates a list of topics to subrscibe to
mqtt_topics = [("pressure",2),("temperature",2),("gas",2),("Vibration",2)]

# Connection string for database (Driver is linux specific)
dataBaseConnection = pymssql.connect("10.102.206.162\MSSQLServer2017")

# Reads all values from the table
def read(dataBaseConnection):
    print("Reading")
    cursor = dataBaseConnection.cursor()
    cursor.execute("SensorData_GetAll")
    for row in cursor:
        print(f'{row}')
```

CLEVELAND STATE UNIVERSITY 1964

NATIONAL ENERGY TECHNOLOGY LABORATORY

# Machine Learning Implementation

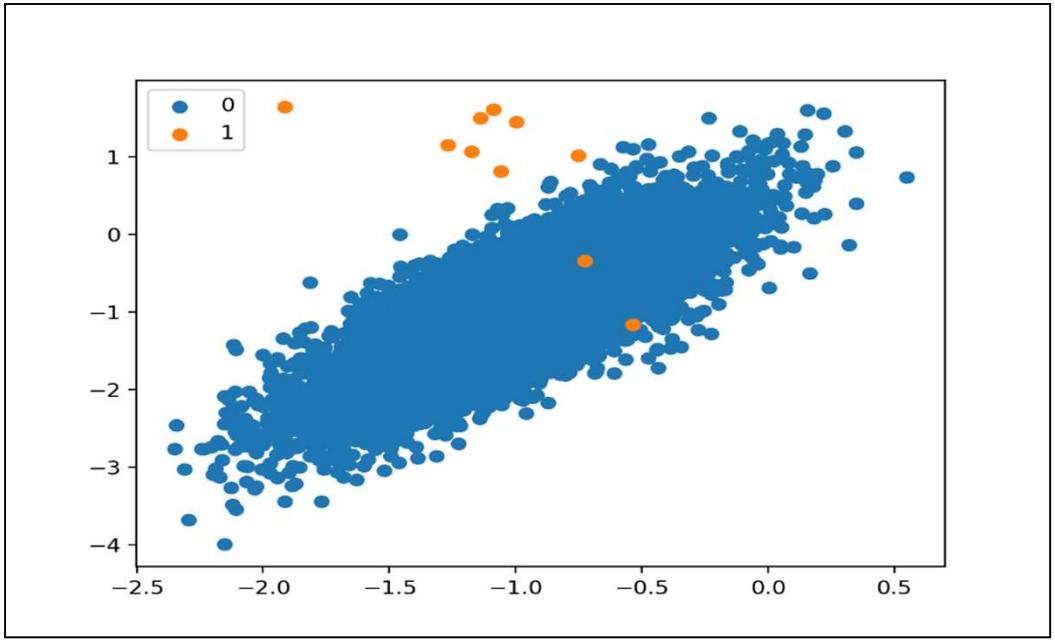
FIU Applied Research Center

- Continued research on multiple machine learning algorithms to perform sensor data analytics
- The algorithms reviewed include:
  - One-Class Support Vector Machine
  - Recurrent-Neural Network - Long Short Term Memory (RNN-LSTM)
  - Isolation Forest
- Implemented python scripts to build models and predictions using the sensor data
- Sensor data anomalies predicted using machine learning model will be stored in IOTA Framework

CLEVELAND STATE UNIVERSITY 1964

# One-Class Support Vector Machine

- One-class SVM algorithm is effective for imbalanced classification datasets where there are none or very few examples of the minority class, or datasets where there is no coherent structure to separate the classes that could be learned by a supervised algorithm

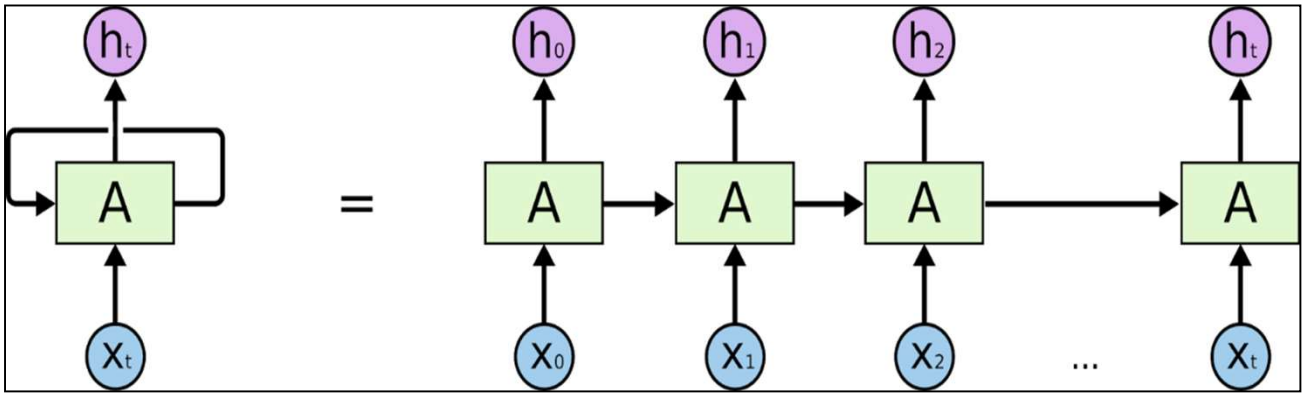


NATIONAL ENERGY TECHNOLOGY LABORATORY

# RNN-LSTM

FIU Applied Research Center

- LSTM networks are a type of recurrent neural networks (RNN) capable of learning order dependence in sequence prediction problems
- A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. LSTM networks are well-suited to classifying, processing and making predictions based on time-series sensor data, since there can be lags of unknown duration between important events in a time series



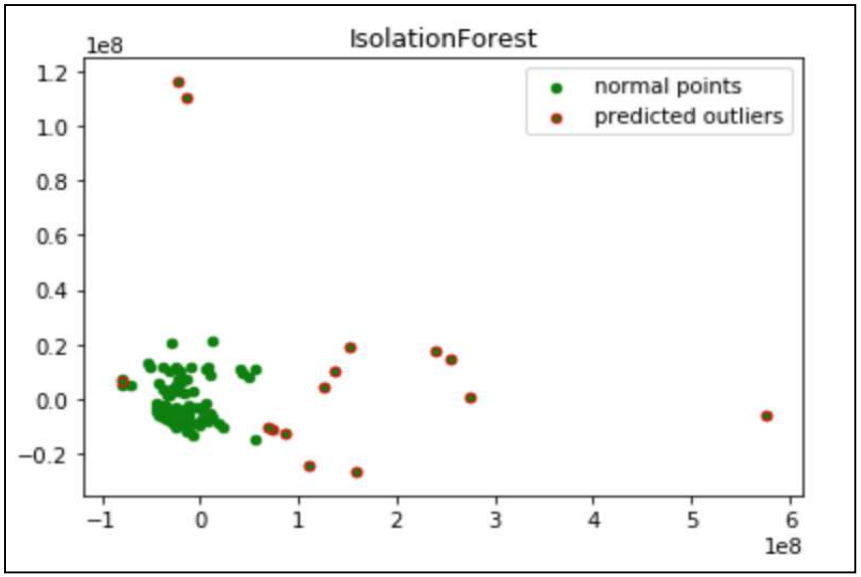

CLEVELAND STATE UNIVERSITY 1964

NATIONAL ENERGY TECHNOLOGY LABORATORY

# Isolation Forest

FIU Applied Research Center

- Isolation Forest is a tree-based ensemble method, built based on decision trees
- In these trees, partitions are created by randomly selecting a feature and then selecting a random split value between the minimum and maximum value of the selected feature
- Isolation Forest explicitly identifies anomalies instead of profiling normal data points




CLEVELAND STATE UNIVERSITY 1964

NATIONAL ENERGY TECHNOLOGY LABORATORY

# Concluding Remarks

FIU Applied Research Center

- Secure authentication of sensor nodes
  - Explore using SRAM-based physically unclonable functions (PUFs) to generate unique "digital fingerprints" as device IDs. The cryptographic hash of each device ID is uploaded to a blockchain instance
  - While accepting the sensor data from a device locally, it is authenticated by comparing the hash included in the message and the one that is present in that blockchain
- Integration of machine learning module with data aggregation
- Secure two-level logging with IOTA distributed ledger
- Visualization of sensor data with sophisticated analytics functionalities

CLEVELAND STATE UNIVERSITY 1964

NATIONAL ENERGY TECHNOLOGY LABORATORY

# Q & A

FIU Applied Research Center

# Thank You

CLEVELAND STATE UNIVERSITY 1964

# Acknowledgment & Disclaimer

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