Taking IoT Security to the Next Level: Hyperledger Fabric Private Blockchain enabled IoT Middleware

Muhammad Aslam Jarwar*, Sajjad Ali[†], Inayatullah [‡] and Sayed Chhattan Shah[§]
Department of Computing, Sheffield Hallam University, Sheffield, UK.
Department of Information and Communications Engineering,
Hankuk University of Foreign Studies, Seoul 02450, Korea.
*a.jarwar@shu.ac.uk, [†]sajjad@hufs.ac.kr, [‡]C1047487@shu.ac.uk, [§]shah@hufs.ac.kr

***Abstract*—As the growth of the Internet of Things (IoT) persists, it becomes imperative to deliberate on strategies for protecting the security and privacy inside the confines of resource constrained devices and their data, while also preserving optimal performance. This research paper offers an innovative solution at the intersection of IoT middleware and Blockchain technology, specifically the Hyperledger fabric. Through the use of a distributed decentralized ledger, we overcome many of the limitations of current IoT networks. This paper outlines a robust layered IoT model that could be applied to any use case, providing security and privacy at the edge of IoT devices. We conducted an implementation setup to test the model and validate the security measures embedded through Blockchain design. Additionally, we improved IoT devices interoperability through the use of semantic ontologies. Overall, this research contributes to the ongoing effort to create a secure and efficient IoT ecosystem.**

***Index Terms*—Internet of Things, Hyperledger fabric, Blockchain, Security, Privacy.**

I. Introduction

In today's world, we rely heavily on smart devices for everything from managing our homes to monitoring our health. With the rise of digital transformation and automation, we have become increasingly dependent on these devices to help us manage our lives. However, if not properly secured, these devices can be accessed by anyone from anywhere, putting our critical resources at risk of unauthorized manipulation. It is important to ensure that all connected devices are properly secured to prevent potential security breaches and protect our valuable resources. The success of any IoT use case is highly dependent on the way how it is secured from unforeseen threats. This research paper offers a solution that combines IoT applications with Blockchain-based private Hyperledger fabric to provide robust security and privacy for resource constrained IoT devices and their data.

The development of technology is leading us towards decentralization [1], [2]. A decentralized and distributed ledger network is the basis of Blockchain, which chains together blocks of information. Nowadays, IoT devices store their data on cloud databases, which are less secure than private Blockchain due to the ”Immutability, Auditability, Autonomy” that Blockchain provides [3]. The distributed ledger technology uses cryptographic techniques to store information with greater accuracy and security, resulting in a global transformation of information storage and a significant impact on the industry.

When it comes to using Blockchain technology, IoT devices face several challenges. One of the main challenges is their constrained resources. This encompasses limitations in processing power, storage capacity, and battery life, which collectively hinder the feasibility of executing Blockchain software on these devices [4]. It is a challenge to coherently incorporate Hyperledger Fabric software and integrate IoT devices as network connected nodes. Additionally, the expected exponential increase in the number of IoT devices can put a strain on existing Blockchain networks. Another issue in IoT ecosystem is to ensure data security. The devices must establish secure connections to their corresponding backend systems, and the data they transmit must be encrypted before it leaves the device [5]. Finally, the lack of interoperability in IoT devices, caused by diverse protocols and standards, has increased the difficulty to integrate them with Blockchain networks.

In the realm of IoT, one of the technical challenges that is always at the forefront is maintaining the devices that are connected to sensors and managing Blockchain related complexities [6]. Because the hardware devices used in IoT are not particularly advanced when it comes to computing and memory resources, an independent solution is needed that can facilitate communication with sensors and store sensor values. It is important to note that the IoT will not be able to fully integrate into a Blockchain network due to the high hardware requirements. Nevertheless, IoT systems can still benefit from Blockchain technology through the APIs provided by Blockchain network nodes. Therefore, it's imperative that a solution is very much required to enable IoT devices to communicate with Blockchain without the need to install a network node on hardware.

In this paper, we present a solution for securing IoT systems using a Blockchain-based Hyper-ledger fabric architecture for any use case. Our design aims to enhance the access control of IoT device's and secure transmission of their real-time data through private Blockchain enabled mechanism. We adopted a deductive approach for this research, conducting scientific investigations from multiple sources. The hypothesis we generated is crucial in determining the efficacy of the proposed solution. We validated hypotheses based on several key variables, including Capacitive Soil Moisture Sensor's real time data transmission, hardware communication through serial port and secure Rest API network communication. We developed and evaluated our proposed solution using various technologies, such as NodeJs, ElectronJs, Python, Arduino, React Native, and Protégé.

Overall, this paper contributes to the following advances.

- A security and privacy-focused architectural design for low-constrained IoT devices.
- Implementation design of Blockchain Hyperledger Fabric for IoT data.
- An interoperable scheme of data representation and cross communication for IoT devices through Virtual Objects (VOs) and semantic ontologies.
- Introduced access and control mechanism and secure data transmission between device and network through implementing JWT (Json Web Token).

II. Related Work

Security, privacy and efficient proliferation of sensor data from IoT devices to the decision ends is highly critical to the success of IoT systems. Blockchain technology supports a strong, scalable solution to these IoT limitations. However, the practical implementations still suffer several challenges. Numerous challenges are associated with constrained IoT devices such as limited computational power, low memory footprint, and inadequate power resources. Several research studies have contributed to solve numerous IoT challenges with Blockchain technologies [7]–[9]. Similar to our approach, other researchers [10] have developed a Blockchain platform based on Hyperledger Fabric incorporated with IoT edge network. Their implementation involved Raspberry Pi devices along with virtual machines on VMware. However, their system suffered interoperability issues in heterogeneous IoT environment; as well this system was not tested for large-scale IoT scenarios. To mitigate the vulnerabilities and trust issues in IoT enabled smart city transportation systems, authors in [11] proposed a Blockchain-oriented solution, however, we believe that the proposed solution lacks interoperable IoT design and absences scalability. Moreover, [12] proposed a private Blockchain network on Hyperledger Fabric for healthcare industry usecase. However, this research was evaluated to assess a trust-based healthcare scenario, which lacks design scalability and proper security and privacy tests.

III. Blockchain Hyperledger fabric unified IoT middleware

A. System Architecture

As shown in Fig. 1, there is a well-defined architecture for a Blockchain Hyperledger Fabric unified IoT system. The layered architecture is organized with several functional components at each level, ensuring seamless integration and efficient operation. The sensor layer is responsible for collecting data from IoT devices such as Arduino and Raspberry Pi. The virtualization layer enables the semantic processing of sensor data to support interoperability. The middleware layer provides sensor data proliferation schemes using technologies like MQTT and APIs. The blockchain layer supports a distributed ledger technology protocol that securely stores all digitalized sensor data and manages smart contracts. Finally, the services layer provides user access to the system with a mobile application dashboard, enabling secure access and control over data and sensor-generated feeds from the distributed ledger.

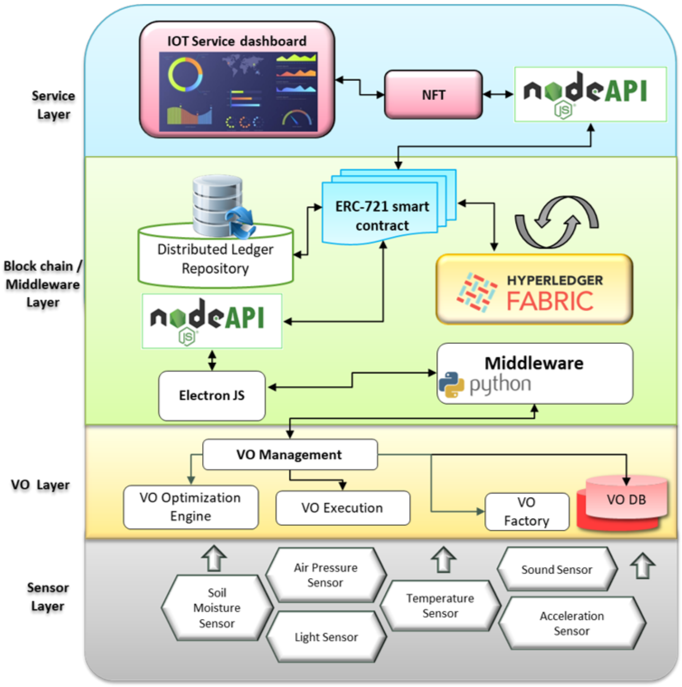


Fig. 1: Layered Architecture of Private Blockchain Hyperledger Fabric for IoT

B. Sensor Layer

In the proposed architecture, the Arduino Sketch is employed at the sensor layer to retrieve sensor values from the Arduino sensor kit. Data is being collected from multiple sensors encompassing various parameters such as soil moisture level, air pressure, temperature, acceleration, light intensity, and sound. The sensor layer is designed to be versatile, allowing for the integration of many types of sensors within the context of services or applications.

C. VO Layer

The proposed architecture includes a layer that focuses on modules designed to represent streaming data with Virtual Objects (VOs). Essentially, VOs are digital representations of real-world objects, such as devices, sensors, and information sources [13]. These VOs represent semantic sensory data

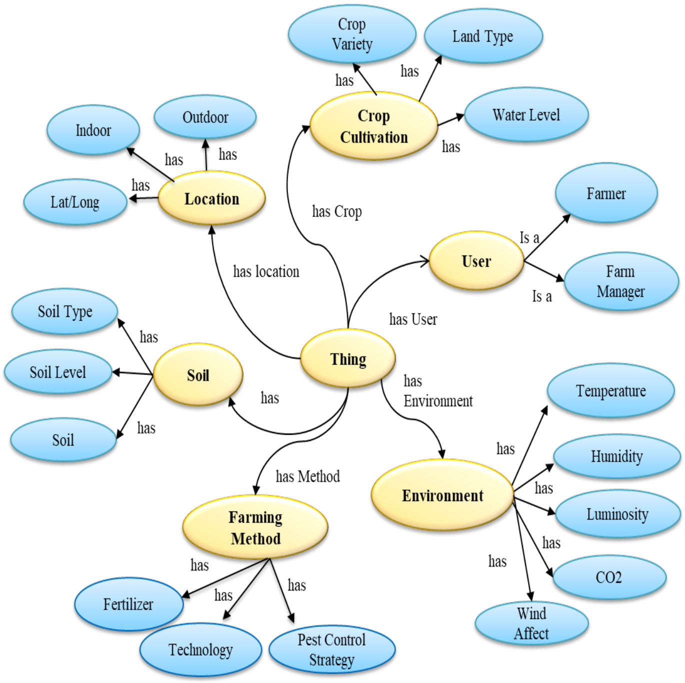
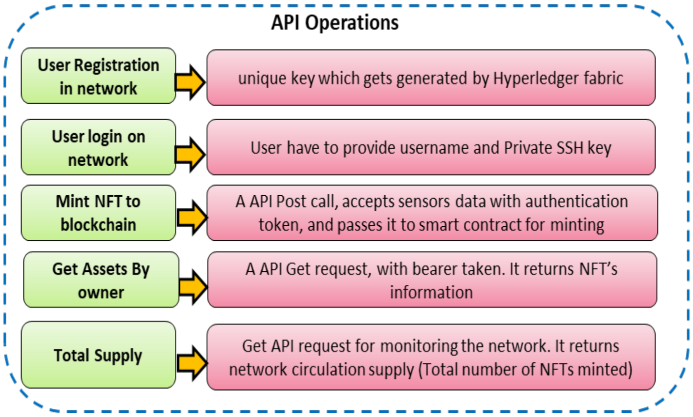


Fig. 2: VO Ontology Template


API Operations

User Registration in network	unique key which gets generated by Hyperledger fabric
User login on network	User have to provide username and Private SSH key
Mint NFT to blockchain	A API Post call, accepts sensors data with authentication token, and passes it to smart contract for minting
Get Assets By owner	A API Get request, with bearer taken. It returns NFT's information
Total Supply	Get API request for monitoring the network. It returns network circulation supply (Total number of NFTs minted)

Fig. 3: API for communication with Blockchain network

that is annotated with associated information from the VO Ontology. This layer is vital because it addresses the issue of heterogeneity and interoperability in IoT and enables the smooth composition of services over a diverse range of devices and objects [14], [15]. In order to facilitate the management of VOs, a number of functional components have been developed inside this layer. For example, the execution of the VO service involves the use of vendor-oriented API directives and data formats, which are then interpreted by the system to facilitate communication between the connected sensor and VO. Each VO serves as a representation of the data originating from individual objects or sensors. The responsibility of the VO management service is to handle the core activities of the VO, which include creating new VOs by utilising the VO factory and applying semantic annotation to sensor data through the VO semantic annotation service.

D. Blockchain Hyperledger fabric Layer

The initial component of this layer offers a Python-based protocol to accomplish real-time data processing and extracting sensor feed while ensuring safe connectivity.The subsequent component encompasses many functionalities that facilitate the establishment of blockchain hyperledger fabric for setting up a blockchain network.

The purpose of python implementation protocol is to handle streaming data, which is used to send the sensor feed to several required endpoints. Here, we are wrapping sensor information in JSON format and passing it to MQTT. Additionally, the Main purpose of python implemented data management function is to broadcast messages to all the connected devices, i.e., Raspberry-based ElectronJs application and Android application.

On the other side, we have developed a secure Web API Endpoint, and we have used Raspberry Pi frontend screens with ElectronJs environment. At the first stage of our application, we have initialized the MQTT protocol broker for capturing sensor values which our middleware is broadcasting and further sending that information to our ElectronJs application where the end point for API's is initialized. We have developed multiple secure API calls to support the required operations (see Fig. 3).

The system's implementation of the blockchain Hyperledger fabric involves a server-side network application. To achieve this, we have set up a small-scale network with 8 GB Memory, 2 Core Processor, 160 GB SSD Disk with 5 TB Transfer rate, and deployed a Hyperledger fabric network on AWS (Amazon Web Services) with the endpoint written in NodeJs. Additionally, we have implemented the Hyperledger fabric version 2.0 SDK for decentralized storage and set up 01 channel and 01 organization as per our scenario for robust transactions. To facilitate communication with Hyperledger fabric, we have deployed secure NodeJs APIs on the server end, which performs several operations (see Fig. 3).

Our Blockchain Hyperledger fabric model has its own official ERC-721 smart contract for minting. We have implemented the smart contract in our network and adapted it as per our data need. The first function is the minting token function. This function is executed by secure NodeJs API on the server side. Before minting a sensor value to an NFT into the network it checks several parameters: 1) Authorization: Check the authorization, If the client is authorized to mint NFT then its sensor value will get stored on a distributed ledger. 2) Token_ID: As per blockchain-based architecture, every NFT that gets generated on the network has some unique identity (Token_ID). We handled that process in NodeJs by providing auto increment id to every new sensor value to achieve uniqueness. 3) Token URI: This is the sensor value in our scenario. It will be stored on a distributed ledger against the owner. The second function is the client-minted NFT Function. The function of smart contract communicates with Hyperledger fabric and checks the authentication of owner, if it is valid then it fetches the list of NFTs which are listed against owner. The third function is the total supply function. This function of ERC-721 smart contract calculates the total number of NFTs generated and handled by this contract in the network and returns the total supply of all NFTs on network.

The flow chart in Fig. 4 summarizes the processes executed systematically through our proposed architecture. First, the communication channel is established with the secure IoT device endpoints where the MQTT protocol broker is initialized for capturing data. The sensor data is validated and passed on to the middleware through a serial communication link. Second, the middleware broadcasts and forwards the information to ElectronJs application where a secure API endpoint is initialized. Secure APIs communicate with the Blockchain network. Third, the Blockchain Hyper-ledger fabric enables transactions on decentralized storage through an implementation of a server-side application. The secure NodeJs API's deployed on the server side facilitate communication with Hyperledger fabric. Finally, the minting of sensor data to NFTs takes place. This minting is based on our ERC-721 smart contract execution. As an initial procedure, prior to minting, an authorization check is performed and a unique ID is generated for each NFT, then sensor data is stored on a distributed ledger. Later, a function of smart contract fetches a list of client-minted NFTs against each owner and renders sensor data to the requesting user dashboard.

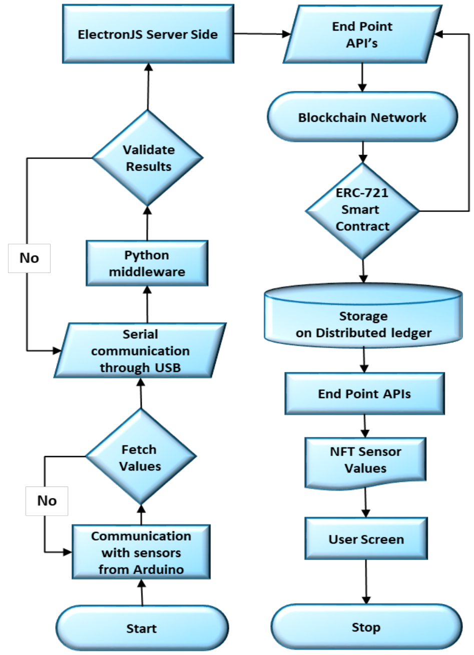


Fig. 4: Systematic flow chart of sequence of operations in the proposed system

E. Service Layer

At this layer, we have developed the mobile application dashboard in React Native platform. This dashboard serves as a conduit through which users can access and view intricate details of sensor-generated data emanating from the distributed ledger. To realize this pivotal functionality, we have implemented a secure QR code scanning mechanism. This procedure yields a user-specific security token, which subsequently becomes instrumental in effecting subsequent requests to secure APIs. Said API, in turn, furnishes sensor-generated data in the form of NFTs. The secure API connectivity and display of NFT values implementation validate the user token values. Additionally, we have meticulously developed a secure API call mechanism whereby we incorporate the security bearer token as a safeguarding parameter within the API call URL. Following this configuration, we await to receive a response from the secure API. Subsequently, we proceed with the analysis and manipulation of the NFT sensor data, channeling it for presentation on the service dashboard.

IV. Results and Discussion

A. 4.1 Hypothesis Formulation

Through the use of experimental and case study research strategies, we were able to design and develop hypotheses. We tested these hypotheses through various scenarios, ultimately leading to successful results. Our hypotheses were based on several key variables, including Capacitive Soil Moisture Sensor percentage, MQTT-based data exchange, and secure API based communication. In formulating our hypotheses, we determined that both capacitance and conductance types of soil moisture sensors were available. However, due to their ability to mitigate the impact of ionic activities commonly found in cultivated soil, capacitance-type sensors proved to be advantageous. We selected a capacitance-type sensor with a voltage of 1.0, which had an analog signal value range of 0 to 1023. To accurately determine soil moisture levels, an implementation was required to precisely calculate sensor values. To address this issue, we utilized a relative range percentage formulation as shown in equation 1.

$$Percentage = ((CRV{-}MN) * 100)/(MX - MN) \quad (1)$$

In Equation. 1 CRV represents current raw value, MN represents minimum value recorded during specific interval of time and MX represents maximum value recorded during specific interval of time.

B. Formulation of datasets to test hypothesis

To perform capacitive soil moisture sensor relative percentage calculation, we have derived a set of instructions in Arduino for testing the soil moisture sensor's values. To gather the necessary data, we conducted two rounds of tests – one with the sensor in a dry place and the other with it submerged in a water jar (as shown in Fig. 5). We have generated the dataset (see Table . I) of sensor-based values for calculation of

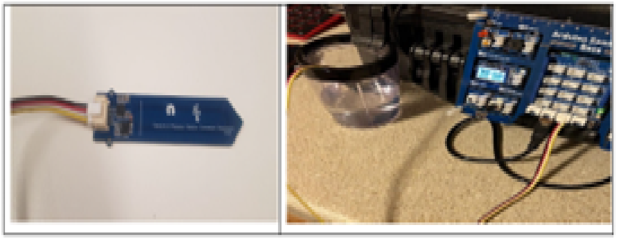


Fig. 5: Hardware setup

TABLE I: Excerpt view of soil sensor dataset generated during the experiment

Set 1 (Dry)	Set 2 (Wet)
501	719
499	720
489	720
495	717
490	723
495	721

soil moisture in percentage. Moreover, we have implemented the formulation to find the desired result.

To enable hardware communication through Serial port and MQTT, we have tested the serial communication with baud rate 9600, but unfortunately, we failed to retrieve data from serial port “ttyACM0” in ElectronJs application which is deployed on Raspberry Pi. Alternatively, we changed our approach as per the hypothesis generated, first, we read sensors from serial communication then we initialized a communication protocol developed in Python. In addition, we implemented the communication stream with ElectronJS via MQTT. Once the MQTT broker is initialized and receives value from USB serial communication, it broadcasts the message to ElectronJS and mobile app. To support security in communication between devices and networks. We used secure API endpoints for providing enhanced user authentication mechanisms while communicating with Blockchain [16]. Furthermore, it offers digital signature and encryption methodologies to ensure the secure transmission and authentication of data. As a direct outcome of this investigation, our prototype has successfully devised a robust authentication mechanism.

C. Proof of concept and analysis

We have developed a multi-tier application interface for agricultural industry users enabling them to effectively oversee and regulate their agricultural operations through the continuous monitoring of critical parameters such as soil moisture, temperature, air pressure, and various other sensor-derived data. The provision of access and governance over this data is facilitated through the utilization of IoT application dashboard, fortified with NFTs to ensure robust security and operational efficiency. The implementation model of our developed proof of concept (see Fig. 6) shows multiple hardware types that are engaged together to develop an ecosystem for IoT to communicate with private blockchain distributed ledger. Here, Raspberry Pi unit is attached to Arduino Uno R3 with a USB serial cable, and on top of it we have an embedded Arduino sensor kit.

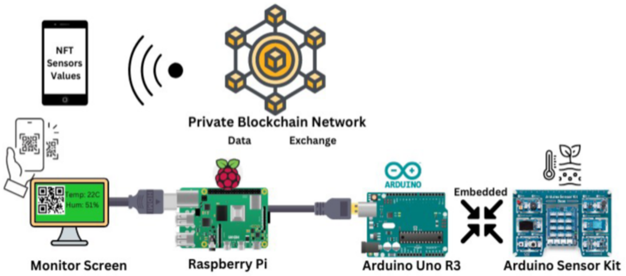
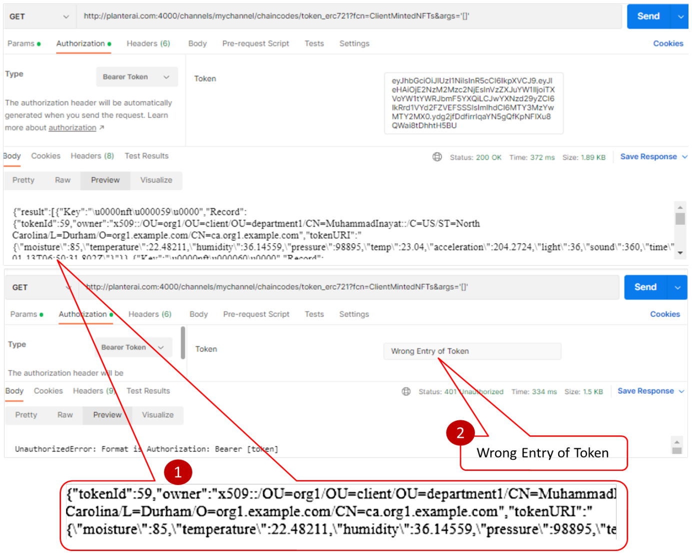


Fig. 6: Implementation model


GET http://planterai.com:4000/channels/mychannel/chaincodes/token_erc721?fcn=ClientMintedNFTs&args='[]'

Type: Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about authorization ↗

Token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE2NzM2Mzc2NjEsInVzZXJuYW1lIjoiTXVoYW1tYWRJbmF5YXQiLCJwYXNzd29yZCI6IkRrd1VYd2FZVEFSSSIsImlhdCI6MTY3MzYwMTY2MX0.ydg2jfDdfirrlqaYN5gQfKpNFlXu8QWai8tDhhtH5BU

Status: 200 OK Time: 372 ms Size: 1.89 KB

```
{"result":[{"Key":"\u0000nft\u000059\u0000","Record":
{"tokenId":59,"owner":"x509::/OU=org1/OU=client/OU=department1/CN=MuhammadInayat::/C=US/ST=North
Carolina/L=Durham/O=org1.example.com/CN=ca.org1.example.com","tokenURI":"
{\"moisture\":85,\"temperature\":22.48211,\"humidity\":36.14559,\"pressure\":98895,\"temp\":23.04,\"acceleration\":204.2724,\"light\":36,\"sound\":360,\"time\"
01-13T06:50:31.802Z\"}"}},{"Key":"\u0000nft\u000060\u0000","Record":
```

GET http://planterai.com:4000/channels/mychannel/chaincodes/token_erc721?fcn=ClientMintedNFTs&args='[]'

Type: Bearer Token

The authorization header will be

Token: Wrong Entry of Token

Status: 401 Unauthorized Time: 334 ms Size: 1.5 KB

```
UnauthorizedError: Format is Authorization: Bearer [token]
```

1

```
{"tokenId":59,"owner":"x509::/OU=org1/OU=client/OU=department1/CN=Muhammadl
Carolina/L=Durham/O=org1.example.com/CN=ca.org1.example.com","tokenURI":"
{\"moisture\":85,\"temperature\":22.48211,\"humidity\":36.14559,\"pressure\":98895,\"te
```

2 Wrong Entry of Token

Fig. 7: Data storing and retrieving from Blockchain using correct and incorrect authentication

Streaming data from IoT devices were represented with VOs and used to evaluate the secure private blockchain Hyperledger fabric network in real time. The sensor’s streaming data are sent to the ElectronJs application to get scaffold in a distributed ledger. The user authentication was implemented with secure tokens. For testing we made two secure calls to the network using the Postman platform, One was with the correct authentication token and the other one was with wrong authentication, therefore, we were able to obtain the intended outcomes. (see Fig. 7).

V. Conclusion

This paper presents a novel middleware solution that addresses the security and privacy concerns that arise in IoT environments. Our approach entails the utilization of virtual object ontology and private Blockchain Hyperledger fabric. This combination enables the development of a decentralized middleware that exhibits automation, reliability, and interoperability. With our layered architectural design and prototype implementation, we have successfully integrated IoT networks with Blockchain, ensuring secure data transmission between

IoT devices and networks. As an added measure of security, we have also developed a protocol in Python that resolves communication issues between Arduino and Raspberry Pi devices and implemented a secure token for user authentication. Our model has been rigorously tested and validated through access to NFT's via the IoT application service dashboard. We believe that our solution provides a robust and reliable approach to securing IoT environments.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science and ICT Korea (No. 2021R1F1A1045933) and in part by HUFS research funds 2023.

References

[1] Alfonso Panarello, Nachiket Tapas, Giovanni Merlino, Francesco Longo, and Antonio Puliafito. Blockchain and iot integration: A systematic survey. *Sensors*, 18(8):2575, 2018.

[2] Md Ashraf Uddin, Andrew Stranieri, Iqbal Gondal, and Venki Balasubramanian. A survey on the adoption of blockchain in iot: Challenges and solutions. *Blockchain: Research and Applications*, 2(2):100006, 2021.

[3] Allan Cook, Michael Robinson, Mohamed Amine Ferrag, Leandros A Maglaras, Ying He, Kevin Jones, and Helge Janicke. Internet of cloud: Security and privacy issues. *Cloud Computing for Optimization: Foundations, Applications, and Challenges*, pages 271–301, 2018.

[4] Salma Salimi, Paola Torrico Morón, Jorge Pena Queralta, and Tomi Westerlund. Secure heterogeneous multi-robot collaboration and docking with hyperledger fabric blockchain. *arXiv preprint arXiv:2206.15242*, 2022.

[5] Harshit Gupta, Amir Vahid Dastjerdi, Soumya K. Ghosh, and Rajkumar Buyya. ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments. *Software: Practice and Experience*, 47(9):1275–1296, 2017.

[6] M Shyamala Devi, R Suguna, Aparna Shashikant Joshi, and Rupali Amit Bagate. Design of iot blockchain based smart agriculture for enlightening safety and security. In *Emerging Technologies in Computer Engineering: Microservices in Big Data Analytics: Second International Conference, ICETCE 2019, Jaipur, India, February 1–2, 2019, Revised Selected Papers 2*, pages 7–19. Springer, 2019.

[7] Mohammad Saidur Rahman, M.A.P. Chamikara, Ibrahim Khalil, and Abdelaziz Bouras. Blockchain-of-blockchains: An interoperable blockchain platform for ensuring iot data integrity in smart city. *Journal of Industrial Information Integration*, 30:100408, 2022.

[8] Meryem Ammi, Shatha Alarabi, and Elhadj Benkhelifa. Customized blockchain-based architecture for secure smart home for lightweight iot. *Information Processing & Management*, 58(3):102482, 2021.

[9] Aitizaz Ali, Mohammed Amin Almaiah, Fahima Hajjej, Muhammad Fermi Pasha, Ong Huey Fang, Rahim Khan, Jason Teo, and Muhammad Zakarya. An industrial iot-based blockchain-enabled secure searchable encryption approach for healthcare systems using neural network. *Sensors*, 22(2), 2022.

[10] Houshyar Honar Pajooh, Mohammad Rashid, Fakhrul Alam, and Serge Demidenko. Hyperledger fabric blockchain for securing the edge internet of things. *Sensors*, 21(2), 2021.

[11] Khizar Abbas, Lo'Ai A Tawalbeh, Ahsan Rafiq, Ammar Muthanna, Ibrahim A Elgendy, and Ahmed A Abd El-Latif. Convergence of blockchain and iot for secure transportation systems in smart cities. *Security and Communication Networks*, 2021:1–13, 2021.

[12] Ghassan Al-Sumaidaee, Rami Alkhudary, Zeljko Zilic, and Andraws Swidan. Performance analysis of a private blockchain network built on hyperledger fabric for healthcare. *Information Processing & Management*, 60(2):103160, 2023.

[13] Michele Nitti, Virginia Pilloni, Giuseppe Colistra, and Luigi Atzori. The virtual object as a major element of the internet of things: a survey. *IEEE Communications Surveys & Tutorials*, 18(2):1228–1240, 2015.

[14] Muhammad Aslam Jarwar, Sajjad Ali, and Ilyoung Chong. Microservices model to enhance the availability of data for buildings energy efficiency management services. *Energies*, 12(3):360, 2019.

[15] Muhammad Golam Kibria, Sajjad Ali, Muhammad Aslam Jarwar, and Ilyoung Chong. A framework to support data interoperability in web objects based iot environments. In *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 29–31. IEEE, 2017.

[16] Madhusudan Singh, Abhiraj Singh, and Shiho Kim. Blockchain: A game changer for securing iot data. In *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, pages 51–55, 2018.