IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 December 2022, accepted 28 February 2023, date of publication 3 March 2023, date of current version 8 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3252030

RESEARCH ARTICLE

A Blockchain-Facilitated Secure Sensing Data Processing and Logging System

WENBING ZHAO[1], (Senior Member, IEEE), IZDEHAR M. ALDYAFLAH[1], PRANAV GANGWANI[2], (Graduate Student Member, IEEE), SANTOSH JOSHI[2], HIMANSHU UPADHYAY[2], AND LEONEL LAGOS[2]

[1]Department of Electrical Engineering and Computer Science, Cleveland State University, Cleveland, OH 44115, USA
[2]Applied Research Center, Florida International University, Miami, FL 33174, USA

Corresponding author: Wenbing Zhao (wenbing@ieee.org)

This work was supported by the United States Department of Energy under Award DE-FE0031745.

ABSTRACT In this paper, we present the design, implementation, and evaluation of a secure sensing data processing and logging system. The system is inspired and enabled by the blockchain technology. In this system, a public blockchain is used as immutable datastore to log the most critical data needed to secure the system. Furthermore, several innovative blockchain-inspired mechanisms have been incorporated into the system to provide additional security for the system's operations. The first priority in securing sensing data processing and logging is admission control, *i.e.,* only legitimate sensing data are accepted for processing and logging. This is achieved via a sensor identification and authentication mechanism. The second priority is to ensure that the logged data remain intact overtime. This is achieved by storing a small amount of data condensed from the raw sensing data on a public blockchain. A Merkel-tree based mechanism is devised to link the raw sensing data stored off-chain to the condensed data placed on public blockchain. This mechanism passes the data immutability property of a public blockchain to the raw sensing data stored off-chain. Third, the raw sensing data stored off-chain are secured with a self-protection mechanism where the raw sensing data are grouped into chained blocks with a moderate amount of proof-of-work. This scheme prevents an adversary from making arbitrary changes to the logged data within a short period of time. Fourth, mechanisms are developed to facilitate the search of the condensed data placed on the public blockchain and the verification of the raw sensing data using the condensed data placed on the public blockchain. The system is implemented in Python except the graphical user interface, which is developed using C#. The functionality and feasibility of the system have been evaluated locally and with two public blockchain systems, one is the IOTA Shimmer test network, and the other is Ethereum.

INDEX TERMS Blockchain, cyber-physical systems, data immutability, data processing and logging, distributed ledger, Merkle tree, sensor identification and authentication, security, Ethereum, IOTA.

I. INTRODUCTION

We have seen increasing use of wireless sensors and Internet of Things with wireless transmission capability in many physical systems, such as power plants, to collect the runtime status of key operations in such systems [1]. The sensing data would help make faster and more accurate decision regarding key operations [2]. This also sharply increases the criticality of these data because a wrong decision made based on compromised data could lead to disastrous consequences. Indeed, this potential issue has been long recognized and well studied [3], [4], [5], [6], [7], [8], [9], [10].

Although the attack vectors for wireless sensors are well-understood [3], [4], [5], [6], [7], [8], [9], there still lack effective solutions for securing wireless sensor based systems. For example, spoofing attacks, Sybil attacks, and injecting attacks are all exploiting the weakness in sensor identification and authentication. The current state of art

The associate editor coordinating the review of this manuscript and approving it for publication was Zijian Zhang.

is to use the physical unclonable function (PUF) as the foundation to identify each physical sensor [11], [12], [13], [14], [15], [16], [17]. However, the proposed mechanism requires the use of a trusted entity to keep the set of challenges and responses (CRPs) generated by the PUF to authenticate the sensor. Using a trusted entity in a system would make this component the most vulnerable point in the system. The blockchain technology offers an innovative user authentication mechanism based on the non-interactive zero-knowledge principle [18], which we have adapted for the purpose of sensor identification and authentication.

Yet another concern is regarding the integrity of the logged sensing data. Logged sensing data are still important asset because they are essential for forensic analysis in case of unexpected incidence in the physical system that has been monitored by these sensors, and they are also essential for satisfying government regulations regarding critical infrastructures. One naive solution could be to place all sensing data on large public blockchain to ensure data immutability. This is not practical for several reasons: (1) the rate of sensing data generated would exceed the throughput of these blockchain systems; (2) the sensing data could be considered confidential and therefore unsuitable to store on a public blockchain; and (3) storing a large amount of data on public blockchain would incur unacceptable level of financial cost. We propose a mechanism that protects the integrity of the large amount of sensing data stored either locally or on commercial cloud storage services using a small amount of data placed on the public blockchain. Furthermore, we adopted the proof-of-work mechanism to create a barrier to the modification of the logged raw sensing data. Raw sensing data are grouped into blocks that are chained a together, in a way rather similar to the blockchain design in Bitcoin [19]. Each block of raw sensing data is then condensed into one transaction to be placed on the public blockchain. The structure of the condensed transaction facilitates the finding of the corresponding block of raw sensing data, and the verification of the integrity of the raw sensing data.

The system is implemented in Python except the graphical user interface, which is developed using C#. To demonstrate the generic design of our system, the functionality and performance of the system have been evaluated locally and with two public blockchain systems, one is the IOTA Shimmer test network [20], and the other is Ethereum [21].

In summary, this paper make the following research contributions:

- First, a blockchain-inspired mechanism for sensor identification and authentication is proposed. This mechanism does not require the use of any trusted entity to store confidential data.
- Second, the system is further secured by grouping the raw sensing data into chained blocks with a moderate amount of proof-of-work. This scheme prevents anyone from making arbitrary changes to the logged data within a short period of time. Again, this scheme is inspired by the blockchain design and constitutes a self-protection mechanism.
- Third, only a small amount of condensed sensing data (*i.e.,* one transaction per block of raw sensing data) are transmitted to the blockchain for safe-keeping.
- Fourth, a clustered indexing scheme is developed to facilitate efficient searching of the condensed data placed on the public blockchain. Furthermore, the design of the block structure and the condensed data structure makes it efficient to locate and verify the raw sensing data as needed.
- Fifth, a working prototype is developed for the system and the operation of the system can be demonstrated with a graphical user interface.

II. RELATED WORK

It is challenging to secure wireless sensor-based systems. That is why we have adopted a multi-faceted approach where we incorporated blockchain-facilitated mechanisms in sensor identification management and secure off-chain sensing data storage. In this section, we go over related work from several perspectives and contrast with out approach.

A. CYBERATTACKS ON SENSOR SYSTEMS

Security in wireless sensor networks has been throughly studied and typical attack vectors are well-understood [3], [4], [5], [6], [7], [8], [9], [10]. Major types of attacks include firmware attacks, identity-based attacks, denial of service attacks, routing-based attacks, and application-level attacks. In this paper, we focus on the mitigation of identity-based attacks.

Identity-based attacks include spoofing, Sybil, false data injection, and node replication attacks. The root vulnerability for these attacks is the lack of robust sensor identification and authentication. The spoofing attack is an attack in which the adversary would impersonate a legitimate sensor. The fact that a sensor could be impersonated means that the credential of the legitimate sensor has been exposed or has been stolen. The Sybil attack is an attack in which one adversary could create many sensor identities. Only a system that does not have proper sensor authentication could allow the Sybil attack. The false data injection attack is a more insidious attack in which the adversary would manipulate the sensing values with the aim of causing a wrong decision being made. The false data injection attack is possible only if a system does not have proper sensor identification or the sensor itself can be compromised. This is also the case for node replication attacks.

In recent years, the physical unclonable function (PUF) has been touted as the most effective way to authenticate sensors [11], [12], [13], [14], [15], [16], [17]. The foundation for PUF is that the integrated circuit in each sensor has some unique, but deterministic, characteristics. On the one hand, the characteristics can be used to differentiate different sensors. On the other hand, the deterministic characteristics can be used to reliably authenticate each sensor. More

specifically, The PUF of a sensor can be used to generate a unique serial number deterministically. This unique number is typically used as the sensor device identifier (ID). Furthermore, the PUF of a sensor is capable of generate unpredictable but deterministic responses to a small set of inputs. Using the unique sensor device ID alone for sensor authentication is not sound because once the ID is known, anyone could impersonate the sensor that bears this ID, which would lead to a spoofing attack. If an adversary could collect a large set of such IDs, the adversary could launch a Sybil attack.

A much more robust sensor authentication mechanism is to use the set of challenges and responses (CRPs) to the PUF. When a sensor is being authenticated, the system would send each of the inputs in the CRPs and demand the sensor to provide the corresponding response. If the sensor is a legitimate one, the sensor would be able to generate the correct responses using the built-in PUF. If the sensor could respond properly for all the inputs, then the sensor is authenticated. Although this mechanism appears to be robust against attacks such as spoofing attacks and Sybil attacks, it requires the system to store the CRPs in a trusted node for authentication. We argue that this is not a sound approach for designing secure practical systems because the ''trusted'' node will inevitably become the single point of failure and the storage of confidential data in a central place will lead to the leak or theft of such data either due to operator error and/or cyberattacks. We note that this is not a baseless claim. On the contrary, the evidence is abundant as seen by numerous news reports on massive data breaches and high-impact ransomware incidents. For example, in 2017, the Equifax data breach resulted in the theft of confidential identity information of 147.9 million Americans. In May 2021, the largest cyberattack on the United States oil infrastructure took place where all billing systems of Colonial Pipeline were locked by a ransomware attack and the company had to pay the 75 Bitcoin random as demanded by the attacker. A lessor known fact is that 100 GB data were stolen from Colonia Pipeline prior to the ransomware attack.

In this paper, we adapt a user authentication mechanism used in Bitcoin for sensor identification and authentication. This mechanism has several advantages over the traditional approach outlined above. More details will be provided in Section IV-A.

B. THE BLOCKCHAIN TECHNOLOGY

The blockchain technology [19] has been reviewed thoroughly [4], [5], [6], [8], [22], [23], [24], [25], [26], [27], [28], In [2], we enumerated key operations that could be benefited from the blockchain technology, including sensing data acquisition, data storage, data communication and coordination, access control, command and control, identity management, software management, and data facilitated assessment. In [29], we identified a list of fundamental properties of public blockchain that can be useful to various applications, including data immutability, data provenance, privacy, transparency, censorship resistance, fault tolerance, security, and atomic contract execution.

C. INTEGRATION OF SENSORS AND BLOCKCHAIN

The integration of sensors and IoT devices with blockchain has attracted a great amount of research. The subject has been well reviewed [30], [31], [32], [33], [34].

A large number of blockchain applications have been proposed, including using blockchain to help secure wireless sensor based systems [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48]. Indeed, the blockchain properties can be used to mitigate various threats to the sensing data. The most fundamental blockchain property is data immutability. With data immutability of the blockchain, critical data can be stored on the blockchain to facilitate the integrity and availability of the sensing data. More specifically, blockchain has been used to facilitate sensor (IoT) authentication [49], [50], [51], [52], [53], access control [52], [54], anonymity/privacy [48], [51], [55], [56], [57], accountability [58], and trust [59].

In this paper, we use the data immutability benefit of the public blockchain. The sensor identification and authentication mechanism is also inspired and facilitated by the blockchain technology. *Unlike existing solutions, sensors are not interacting with the blockchain directly in our approach*. In the system we propose, a sensor processing and logging entity mediates the sensors and the blockchain. There are several benefits with our approach: (1) the demand on blockchain throughput is drastically reduced by storing the summary of each block of raw sensing data on the public blockchain; (2) the system is not locked in to any specific blockchain, which is beneficial for the long-term survivability of the system; (3) the sensors are not involved in blockchain operation, particularly not involved in the decentralized consensus process. In a way, this is a form of off-chain approach. The payment channel from Lightning Network [60] is perhaps the most well-known off-chain solution that could drastically reduce the throughput need of a public blockchain because only a single anchor transaction is needed to establish a payment channel and another settlement transaction is needed to close off the channel. Previously, we proposed to use Merkle tree [61] to create a fingerprint for a batch of raw sensing data and place the fingerprint on the public blockchain as a way to pass the data immutability property of the public blockchain to the locally stored batch of raw sensing data [62]. This mechanism is adopted in the current work. The comparison of this study and the related work is summarized in Table 1.

Like all engineered systems, making tradeoff is inevitable. Our system places higher demand on the processing power of the sensing nodes. To ensure sensor identification and authentication, the sensor must be able to digitally sign each message using a public key algorithm (*i.e.,* elliptic curve digital signature algorithm). Although an implementation of the elliptic curve cryptosystem has been proposed on 8-bit microcontroller [63], there is currently no application

TABLE 1. Comparison between this paper and related work.

Approach	Security Properties	References	Drawback
Sensor-blockchain direct interaction	Authentication	[49]–[53]	Higher demand on blockchain throughput
	Access control	[52], [54]	
	Anonymity/privacy	[48], [51], [55]–[57]	
	Accountability	[58]	
	Trust	[59]	
Sensor-blockchain indirect interaction	Authentication & data security	This paper	Higher demand on sensor processing power; sensor sampling rate is limited to 10Hz or below

programming interface support for microcontroller-based sensing devices. As such, our approach precludes the use of these sensing devices. ARM CPU powered Raspberry Pi zero/1/2/3/4 product lines are good candidates for use in our system [64]. This can be regarded as a limitation/drawback of our approach. Also due to this limitation, our system cannot support sensing sampling rate higher than 10Hz if the Raspberry Pi devices are used.

An earlier design of the system was reported in a 6-page conference paper [65]. Compared with this earlier work, significant improvements have been made in the current system design and implementation. First, the earlier system was tied to the IOTA public distributed ledger. The current system can work with other public blockchains with a custom clustered indexing component, which facilitates the search for the condensed data placed on the public blockchain. The index is stored both locally and on the InterPLanetary File System (IPFS) (https://docs.ipfs.tech/concepts/what-is-ipfs/), which offers decentralized storage of files. Second, the earlier system lacked a concrete mechanism to ensure that the sensor enrollment phase is secure. The current system incorporates a specific mechanism that only legitimate sensors that are physically located together with system may enroll in the system by forming an ad-hoc network with static IP addresses. Third, the earlier system lacked a self-protection mechanism to resist the modification of the raw sensing data logged locally (other than the explicit integrity check with the data placed on the public blockchain). The current system incorporates a self-protection mechanism that makes the modification to the raw sensing data a time-consuming processing (the older the data are, the more time consuming to change the data). Finally, it is inefficient to locate the raw sensing data given an aggregated sample placed on the public blockchain in the earlier system. The current system makes it much more efficient to do so with the new block structure and the structure of the condensed data.

III. OVERVIEW OF THE SYSTEM

In this section, we present the threat model for our system, provides a concise security analysis, and introduce the main components of the system.

A. THREAT MODEL

We describe the threat model in terms of what have proposed in [9] for wireless sensor networks. This is a unified threat model based on the nature of the attacks and security violations where attacks are categorized based on two dimensions: (1) the layer in which the attack takes place or is targeting to (*i.e.,* application layer, communication layer, and perception layer), and (2) the security property that will be violated if the attack is successful (*i.e.,* integrity, confidentiality, and availability). The main types of attacks are illustrated in Fig. 1, including the application-level attacks, routing attacks, identity-based attacks, denial of service attacks, and firmware attacks.

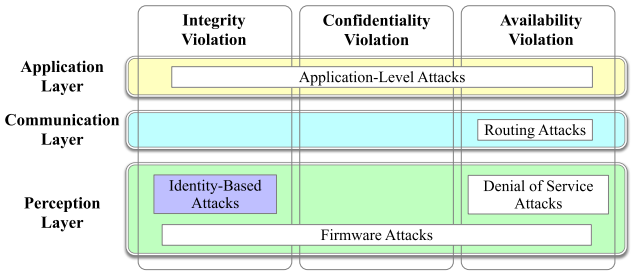


FIGURE 1. The threat model with main types of attacks.

There are a large number of application-level attacks, such as malware, virus, worm, trojan horse, adware, man-in-the-middle attacks, session hijacking, SQL injection attacks, traffic analysis, eavesdropping, side channel attacks, and various denial of service attacks [9]. Example routing attacks include the desynchronization of TSCH frames and 6LWPAN exploit attacks [9]. The attacks on the perception (*i.e.,* sensing) layer can be divided into firmware attacks, identity-based attacks, and denial of service attacks. There are various forms of denial of service attacks at the perception layer, such as the sensor overwhelming attack, exhaustion attack, sleep deprivation attack, collision attack, jamming attack, interference attack, resource depletion attack, node destruction attack, node malfunctioning attack, and node outage attack [9]. Firmware attacks include RFID attack, reprogram attack, booting attack, malicious code injection attack, and node capture attack. Identity-based attacks include the spoofing attack, node replication attack, Sybil attack, and false data injection attack.

The attacks on the application layer are actually general attacks that could happen to any networked systems. The routing attacks are only applicable for systems that use ad-hoc wireless sensor networks. With the evolvement of Internet of Things and improved networking infrastructure, the need for forming ad-hoc wireless sensor network is becoming diminished. Hence, we do not consider the attacks in the

application and communication layers. Firmware attacks and denial of service attacks are beyond the scope of this paper. In summary, the focus of this study is to mitigate identity-based attacks.

B. SECURITY ANALYSIS

The goal of the system is to ensure secure sensing data processing and logging. More specifically, we assume that the system might be subject to:

- Identity-based cyberattacks, such as spoofing, Sybil, false data injection, and node replication attacks.
- Hard drive failures, which may corrupt the logged data.
- Cyberattacks on the raw sensing data.

To mitigate these attacks and issues, a number of blockchain-facilitated mechanisms are incorporated in the system, which will be elaborated in sections IV-A, IV-B, and IV-C. The system has the following characteristics:

- Only sensing data transmitted by registered sensors are accepted for processing and logging.
- Only a small amount of sensing data are placed on the public blockchain for safe-keeping. Sensor enrollment data are also placed on the public blockchain for safe-keeping.
- The sensing data placed on the public blockchain can be used to verify the raw sensing data efficiently.
- The raw sensing data are stored in chained blocks, and each block of data incorporates a moderate amount of proof-of-work for self protection against malicious modification. Although this mechanism does not guarantee data immutability, it does limit the amount of data modified within a period of time.

As we have argued previously in Section II, the popular PUF-based approach to sensor identification and authentication is problematic because placing critical data in a trusted node would make the node the target of cyberattacks. Therefore, we decide to take a completely different approach that does not require the use of any trusted node. Fundamentally, the authentication is a process that verifies an entity is indeed who that entity claims to be. As such, it is inevitable for the entity to be authenticated to prove that it indeed possesses some secret that is known to this particular entity and not any other entity. The PUF-approach essentially requires the entity to share its secret with the authenticator. As evidenced by the user identification mechanism adopted by Bitcoin and many other public blockchain systems [18], this is completely unnecessary and in fact should be avoided.

In many blockchain systems such as Bitcoin, user identification is based on the public-key cryptography, where each user is required to create a pair of security keys, one of which is made public (*i.e.,* public key), and the other must remain secret at all times (*i.e.,* private key). The private key is used to generate a digital signature that can be verified using the public key, which is exactly how a public blockchain verifies a user. The digital signature can only be generated by a user that possess the private key, and the private key is also unique to the user. Hence, this satisfies the requirement of user authentication. In this user authentication scheme, the user does not share any secret with the authenticator at all, which is much more robust.

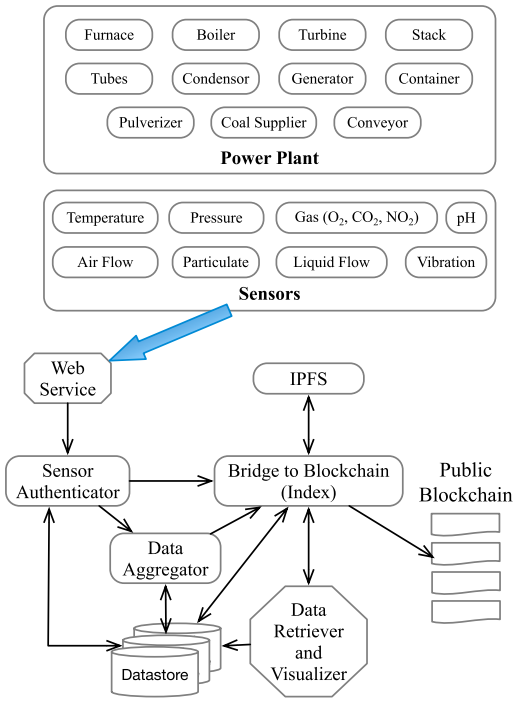


FIGURE 2. The main components in the secure sensing data processing and logging system.

Because any user could create a pair of keys, the user authentication mechanism in blockchain allows a form of anonymity to its users. Exactly because of this, the mechanism cannot be used as it is to mitigate identity-based cyberattacks in sensor-based systems. In blockchain, the user authentication is done to ensure that the user is the owner of the coin as indicated by its address. In the sensor-based system, we must restrict the set of public keys such that only the public keys for the actual sensors in the system can be recognized. A public key outside this set of restricted set of public keys must be excluded from sensor authentication consideration. To satisfy this requirement on a restricted set of public key, we need a sensor enrollment phase.

Because any sensor admitted during the enrollment phase would be regarded as a legitimate sensor provided that the sensor can provide a valid digital signature, the enrollment phase constitutes the most vulnerable stage for sensor authentication. If an adversary could successfully enroll a sensor under its control, the security of the entire system would be broken. In Section IV-A, we will present a mechanism to ensure the security of the sensor enrollment phase.

C. MAIN COMPONENTS OF THE SYSTEM

As shown in Fig. 2, the main components of the system include the physical system, sensors, Web service,

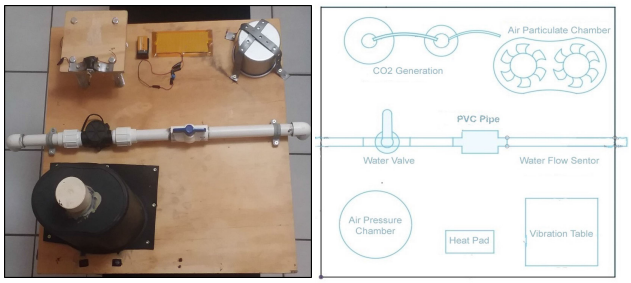


FIGURE 3. A portion of the mock testbed for sensing.

sensor authenticator, data aggregator, datastore, bridge to blockchain, and data retriever and visualizer.

1) PHYSICAL SYSTEM

A physical system will be monitored by a set of sensors. In our study, we consider a fossil fuel power plant as the physical system. This could be replaced by other physical systems, with potentially a different set of sensors.

2) SENSORS

A set of wireless sensors are used to monitor the physical system. For a fossil fuel power plant, various sensors are used to monitor the status of key components in the power plant. For example, the turbine is monitored using the temperature sensor, pressure sensor, and vibration sensor, and the boiler is monitored using temperature sensor, pressure sensor, gas sensor, and pH sensor. The sensor range for individual component could differ drastically. For example, the temperature range for the turbine is between 1.81 to 37.11 Celsius, and the temperature range for the boiler is between 540 to 570 Celsius. For ease of system development, we use a python program to simulate various sensors. For functional tests, we set up a testbed to mock-up a fossil power plant. The testbed consists of the temperature, pressure, gas, pH, air flow, liquid flow, vibration, and particle sensors with a RaspberryPi 3. A portion of the testbed is shown in Fig. 3.

3) WEB SERVICE

The sensors communicate with the sensing data processing and logging system by calling JSON-based Web services over the HyperText Transfer Protocol. The Web service is implemented using a Python-based framework called CherryPy.

4) SENSOR AUTHENTICATOR

When a new sensor is added to the system, the sensor must first enroll itself with the sensor authenticator. Every message submitted by a sensor is authenticated by the sensor authenticator. The message is dropped if the authentication fails.

5) DATA AGGREGATOR

The data aggregator is responsible to group the raw sensing data into chained blocks, and build a small amount of data for each block of raw sensing data to be placed on the public blockchain for safe-keeping. The data to be placed on the public blockchain serve for two purposes: (1) the data must contain a fingerprint for the block of a raw sensing data irrespective of the content of the raw data, and (2) the data must contain a meaningful summary of the block of raw sensing data. More details on the data aggregation mechanisms are provided in Section IV-B2.

6) DATASTORE

The raw sensing data are stored in a local MongoDB for development. A professionally managed database system with sufficient backup plan or commercial cloud storage should be used for practical deployment of the system.

7) BRIDGE TO BLOCKCHAIN

This component is to isolate the logistics on communicating with a public blockchain from other components in the system so that the system is not tied to any particular blockchain for flexibility and long-term evolvability of the system. To facilitate the search on the data placed on the public blockchain, a clustered index is built and stored both locally and the on the IPFS.

8) DATA VISUALIZER

This component provides a graphical user interface to fetch the raw data stored at the datastore or the aggregated data stored on the public blockchain. All such operations are read-only.

IV. SYSTEM DESIGN AND IMPLEMENTATION

In this section, we present the design and implementation of sensor identification and authentication, sensing data aggregation and logging, safe-keeping and data retrieval with public blockchain, and data visualization.

A. SENSOR IDENTIFICATION AND AUTHENTICATION

In this section, we present the details on the design of sensor identification and authentication mechanisms used in our system. Sensor identification and authentication consists of the sensor enrollment phase and the sensor authentication phase.

During the sensor enrollment phase, a form of physical security is used where a sensor is allowed to enroll with the system only when the sensor and the authenticator are physically close to each other. The most straightforward method to ensure physical security is to set up an ad-hoc WiFi network for the sensors and the authenticator, which is supported by all major operating systems (*i.e.,* Linux, Mac, and Windows), and it is usually supported by wireless sensors. By configuring an ad-hoc WiFi network with each sensor during the enrollment phase, the system assigns a static IP address to each sensor as shown in Fig. 4, which constitutes as additional access control for stronger security.

The detailed steps for sensor enrollment are illustrated in Fig. 5 and explained below:

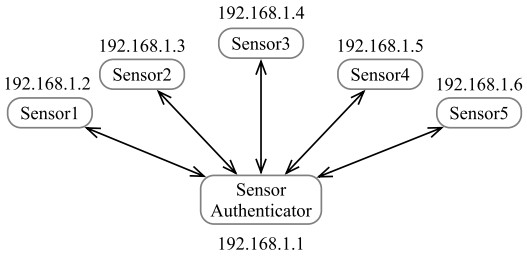


FIGURE 4. During the sensor enrollment phase, ad hoc WiFi networks are set up for the sensors and the sensor authenticator where each node is assigned a static IP address.

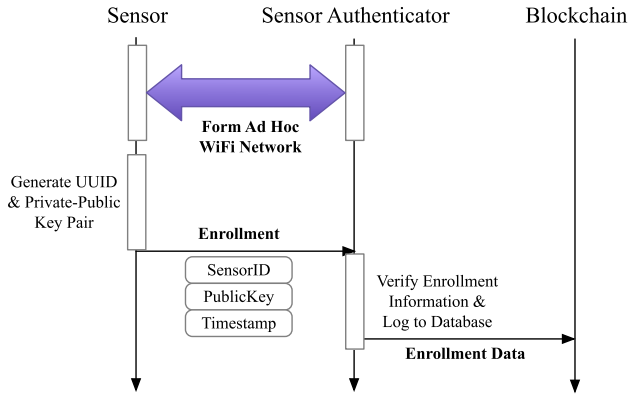


FIGURE 5. The sensor enrollment steps.

- The sensor node and the sensor authenticator are configured to form an ad hoc WiFi network, as shown in Fig. 4.
- Then, a universally unique identifier (UUID) and one pair of private-public keys are created. In our implementation, we used a python program to generate the UUID and the pair of keys, and save them in files. The UUID is used as the sensor identifier (ID). For each sensor, the UUID and the pair of keys are created only once. Specifically, the Elliptic Curve Cryptography (ECC) is used as the public key algorithm because the keys are much shorter than those for the traditional RSA algorithm. In ECC, a passphrase is used to generate the private key and the public key is derived from the private key. Both the ECC public key and the signature are much shorter than those produced by RSA.
- Next, the sensor generates and sends the sensor authenticator an enrollment message containing the sensor ID, the public key, and the timestamp of the current time. The sensor authenticator first checks for possible duplicate. Any duplicate request is ignored. The timestamp is also checked. Because all future sensing samples are timestamped, it is important for the sensors' clocks be synchronized with the system within a reasonable range. If the sensor is much too out of sync, an error notification will be generated so that the system administrator could fix the issue. If it is not a duplicate enrollment request and the timestamp is within the expected range, a new enrollment entry is inserted in the sensor enrollment table, and is stored in the datastore.
- The sensor authenticator would ask the bridge to the blockchain to generate a transaction including the enrollment data and submit the transaction to the public blockchain for safe-keeping. It is also possible to encapsulate the entire set of sensor enrollment data in a single transaction to be placed on the public blockchain after all sensors have finished enrolling with the system.

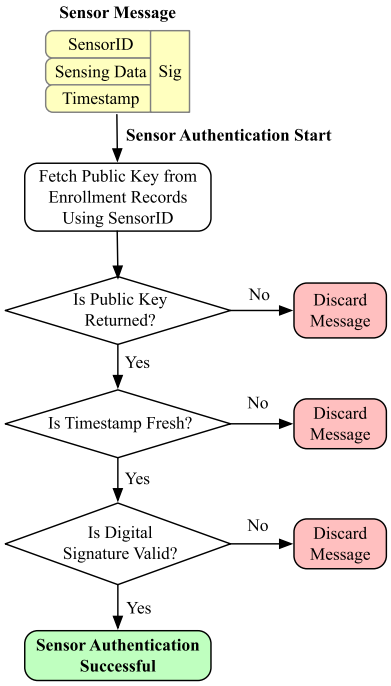


FIGURE 6. Steps in sensor authentication of a sensing message.

Once a sensor has been enrolled with the system, it will start collecting sensing data and submitting the data to the system. The submitted data will be authenticated by the sensor authenticator before they are processed and logged. The message containing the sensing data must include the sensor's sensor ID and the timestamp when the sample was taken, and the message must be digitally signed using the sensor's private key, as shown in Fig 6. Valid messages will be passed to the data aggregator for further processing and logging.

The detailed steps in sensor authentication for each sensing message are shown in Fig. 6. All messages sent by a sensor must include a sensor ID and the sensing event timestamp, and the message must be digitally signed. The sensor authenticator would first retrieve the public key using the sensor ID included in the message from the sensor enrollment table. If an entry is found, the corresponding public key is retrieved. The message is dropped immediate if no entry is found. Next, the sensor authenticator checks if the timestamp included in the message is fresh. If the timestamp is the same or older than the last seen timestamp from the same sensor, the message is discarded. This check is necessary to mitigate

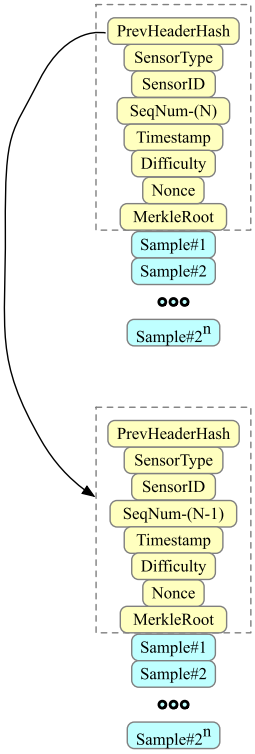


FIGURE 7. The raw sensing data are grouped into each block, and the blocks are chained together in a similar fashion to the blockchain.

replay attacks. If the message is fresh, the public key is then used to verify the digital signature. The message is discarded if the signature validation fails. The sensor authentication is successful only when the message has passed all these verification steps, and the message will be forwarded to the data aggregator for further processing and logging.

Note that the sensor enrollment data have low degree of privacy concern because they only contain a list of sensor IDs and the corresponding public keys. Adversaries who have obtained such information could not launch any meaningful attacks on the system. That said, such data may be encrypted to enhance the security of the system.

B. SENSING DATA AGGREGATION AND LOGGING

Valid sensing data are logged immediately at the local datastore, and are aggregated periodically. A key insight that we have learned from the blockchain design is to create a self-imposed barrier to modification of the logged data. Similar to Bitcoin, the local data are grouped into blocks and these blocks are chained together with some moderate proof-of-work. This design also aligns perfectly well with our plan to store a small amount of condensed sensing data on the blockchain for safe-keeping. For each block of raw sensing data, one transaction is constructed to be placed on the public blockchain.

1) RAW SENSING DATA LOGGING

One issue with traditional data storage is that the data can be modified with very little cost. While data corruption due to hardware failures can be easily detected by embedding checksums into the data, it is difficult to identify premeditated alternation of data because the checksum can be recomputed rather quickly to reflect the intentional changes to the data. That is why many publications have proposed to either store the data directly on the public blockchain, or store the fingerprint of the data on the public blockchain so that the data can be considered immutable, or any modification can be detected via the fingerprint placed on the blockchain.

Inspired by the blockchain design principle, we propose to store the raw sensing data in a data structure similar to blockchain. More specifically, as shown in Fig 7, each block includes 2^n number of raw sensing samples of taken by the same sensor, where n is a configurable parameter that determines the block size. The Merkle root is computed based on these 2^n samples by using the hash of each of the samples as the leaf nodes in the Merkle tree. The details have been described in our previous work [62]. The block header includes:

- `PrevHeaderHash`. This field contains the base64 encoded 32-byte-long (256-bit) hash value of the parent block produced with the SHA256 algorithm (using the python hashlib library).
- `SensorType`. This field contains the sensor type of the sensing data in this block.
- `SensorID`. This field contains the ID of the sensor that produced the sensing data in this block.
- `SeqNum`. This field is the sequence number assigned to this block. The first block has a sequence number 1. The *N-th* block would have a sequence number of N. The sequence number is equiavlent to the block height in blockchain.
- `Timestamp`. This is the timestamp when this block is created.
- `Difficulty`. This field indicates the difficulty level of the proof-of-work puzzle. More details will be provided next.
- `Nonce`. This field records the solution to the puzzle. The nonce is a 4-byte integer one would have to find such that the hash of the block header meets the difficulty target.
- `MerkleRoot`. This field contains the base64 encoded 32-byte Merkle root of the group of sensing samples in the current block.

The proof-of-work implementation is based on an open-source project at Github (https://github.com/mvarge/proof_of_work). We changed the implementation so that the nonce is 4-byte long (instead of 2-byte long) and added a function to verify the validity of the nonce. Different from the proof-of-work algorithm defined in Bitcoin, the difficulty field in our proof-of-work algorithm indicates the difficulty level instead of a specific number where the block header hash must be lower than. The difficulty level is the number of leading 0s in the block header hash. If the block header hash with a nonce contains the same as or larger than the number of leading 0s as indicated in the difficulty field, then the nonce

has met the difficulty level, and therefore, the proof-of-work puzzle is considered solved.

The inclusion of the proof-of-work in block creation for raw sensing data logging has the benefit of creating a barrier to modification of the logged data. Similar to the property of blockchain, the older the sensing data, the higher the barrier to change because it would take longer for the adversary to resolve all the proof-of-work puzzles in the blocks to be modified and all the following blocks. Given a chain of blocks of the raw sensing data, it would take $k \times \mu$ seconds on average to redo all the proof-of-work puzzles to modify a block that is k block deep, where μ is the mean block interval (*i.e.,* the time it takes to solve one proof-of-work puzzle given a difficulty level).

In proof-of-work based blockchain systems, the difficulty target for proof-of-work is adjusted dynamically to accommodate the change of hashing hardware power and the scale of the system while maintaining a pre-set block interval. For example, in Bitcoin, the block interval is set to be 10 minutes. Similarly, for sensing data logging, one would have to determine what block interval is appropriate. Obviously, this is application-dependent. Although a larger block interval would lead to a higher barrier to modification of the data, one must accommodate the arrival rate of the sensing data.

Before we end this subsection, *we note that the computation of the proof-of-work and the Merkle root is at the data aggregator, not at the sensor*. Hence, such computation does not place any burden on the sensors.

2) CONDENSING SENSING DATA

As we have mentioned in Section III, the condensed data must satisfy two requirements: (1) the data must contain a fingerprint for the block of a raw sensing data irrespective of the content of the raw data, and (2) the data must contain a meaningful summary of the block of raw sensing data. Required 1 is satisfied by including the Merkle root of the block of raw sensing data in the condensed sample. Requirement 2 is satisfied by taking a statistical analysis on the block of raw sensing data and including the result of the statistical analysis as part of the condensed data.

Fig. 8 shows the details of the condensed data and the relationship between the block of raw sensing data and the corresponding condensed data. The condensed data structure consists of the meta data, the Merkle root, and the statistical summary of the block of sensing samples. The meta data include sensor type, sensor ID, and sequence number, which are identical to the corresponding fields in the block header. The meta data are used to establish the correspondence between the block of raw sensing data and the condensed data for the block. For the statistical summary, we assume that the sensing values are one-dimensional scalar values (such as temperature). For multidimensional vector values, the statistical summary would either include every dimension, or the magnitude of the vector. The statistical summary includes the number of raw sensing samples in the block,

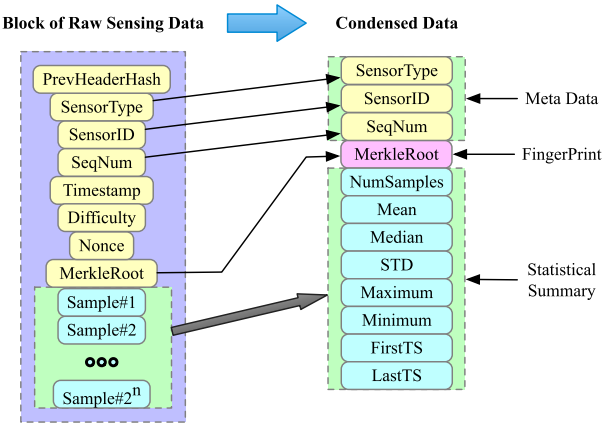


FIGURE 8. The detailed structure and content of the condensed sample, and the relationship between the block of raw sensing data and the condensed data.

the mean and the median values of the sensing samples, the standard deviation (*i.e.,* STD), the maximum value, the minimum value, the timestamp of the earliest raw sensing sample, the the timestamp of the most recent raw sensing sample in the block. The condensed data will be included in a transaction for a public blockchain so that the condensed data can be stored on the public blockchain for safe keeping.

C. SAFE-KEEPING AND DATA RETRIEVAL WITH PUBLIC BLOCKCHAIN

Typically, the only way to programmatically retrieve a particular transaction in the blockchain is through a transaction identifier (ID). However, this makes it impossible to query the blockchain for certain groups of transactions using some keywords. Previously, the IOTA distributed ledger system offered a tag mechanism as a form of primitive way for searching the data on the ledger, and we exploited this feature in an earlier version of our system [62]. Recently, IOTA introduced a more comprehensive indexation feature. Unfortunately, the feature is severely under-documented, and we could not figure out how to use the indexation feature in IOTA.

To avoid tying our system to a particular public blockchains, we decide to develop our own indexing mechanism. With this custom indexing mechanism, users of our system could perform data retrieval in a way similar to that of a traditional relational database systems. Considering the nature of the sensing data we log, our index would support the search terms based on the date, sensor type, and sensor ID, as shown in Fig. 9. Our indexing scheme can be considered as a form of clustered indexing [66].

A dictionary data structure is used to implement this clustered indexing scheme for each day in the form of

```
{"date": "2022-11-20",
"sensor_type":"temperature",
"sensor_id":"sensorID#1",
"transaction_ids":[txnid#1,txnid#2,...]}.
```

The value of the key "transaction_ids" is a list of transaction IDs, one per transaction submitted to the public blockchain.

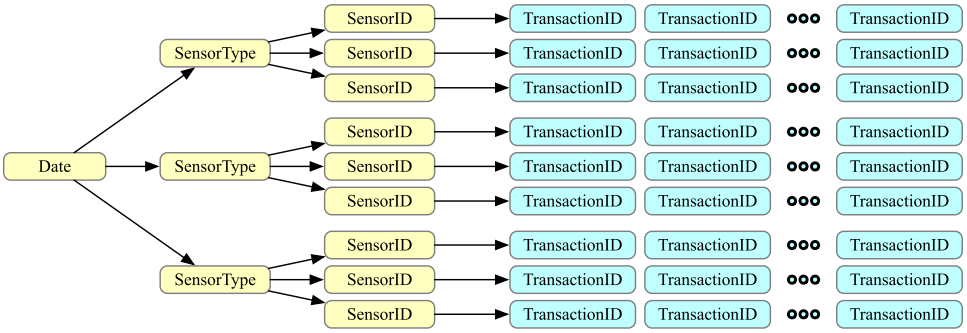


FIGURE 9. The indexing scheme supports three search terms: date, sensor type, and sensorID.

During each day, this index dictionary is updated every time a new transaction containing the condensed data is submitted to the public blockchain, and the dictionary is written to or updated in the local MongoDB for persistence of the records. At the beginning of a new day, the previous day's index is written to a JSON (JavaScript Object Notation) file, and uploaded to the IPFS for additional fault tolerance and security.

The index allows the search using any one or a combination of the three terms: "date", "sensor type", and "sensor ID". The range of dates can be easily accommodated by decomposing the range to a list of dates. A search for a particular date can be done via a simple lookup on the index for the corresponding dictionary with the searched date. Once a dictionary is found, then the list of transaction IDs can be located and used to query the public blockchain for the list of transactions issued on the particular date. A search for a single sensor type or a particular sensor ID would force a linear search of the index because all index dictionaries need to be fetched. If this type of search is undesirable, then additional indices could be used. The down side is the redundancy of the transaction IDs need to be stored. If the search terms include the date, then the search is done by the date first, which is efficient.

When it is necessary to locate the raw sensing data corresponding to a particular transaction stored on the public blockchain, the sequence number contained in the condensed data is used to locate the block of raw sensing data stored locally. Then, the meta data (*i.e.,* sensor type and sensorID) and the Merkle root are used to verify the integrity of the raw sensing data block, as shown in Fig. 8.

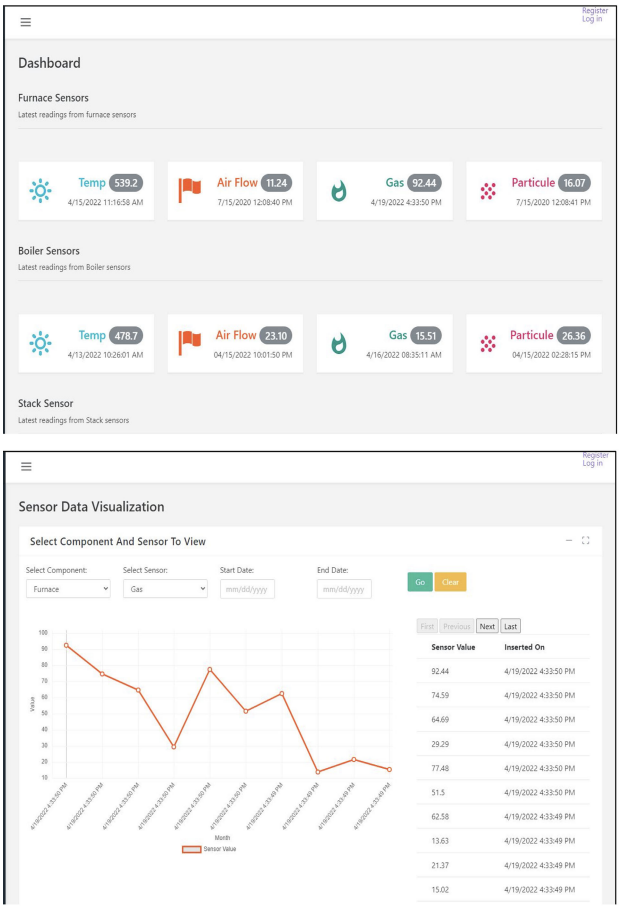


FIGURE 10. The graphical user interface for the visualization of the sensing data.

D. DATA VISUALIZER

The data visualizer in the system allows a user to view the current status of the system being monitored and visually inspect the sensing data, as shown in Fig. 10. The raw sensing data are queried directly, and the condensed data placed on the public blockchain are queried via the index.

V. EXPERIMENTAL RESULTS AND DISCUSSION

Most of the experiments are carried out using an iMac-27 with core-i5 CPU and 64GB RAM. The experiment with the IOTA blockchain is with its test network deployed on the Internet. Table 2 outlines the individual experiment and related parameters.

A. RAW SENSING DATA AGGREGATION AND LOGGING

We first explore suitable levels of difficulty for the proof-of-work (PoW). We intentionally choose to use a single-threaded implementation to search for the right nonce to solve the proof-of-work puzzle. More specifically, four nested `for` loops are used to search for the solution. In each loop, the

TABLE 2. Individual experiments and related parameters.

Catagory	Individual Experiment	Parameters
Raw sensing data aggregation & logging	Proof-of-work (PoW)	Difficulty levels 1-6, PoW latency pdf, & PoW latency mean
	Data aggregation	Block size; latency on computing Merkle root; sampling rate; number of sensors; sampling rate
Aggregated data storage and retrieval	Work with IOTA	Transaction creation and fetch latency
	Work with Ethereum	Transaction creation and fetch latency
Public key cryptography operations	ECC operation Latency	Raspberry Pi Zero, 1, 2, and 3; key length is set at 192-bit long
Scalability of the system	Throughput demand	Sensor sampling rate set at 10Hz; total number of sensors varies between 10 to 1,000; different aggregation levels

full range of each byte, *i.e.,* between 0 and 255 (inclusive), is iterated. At the innermost loop, the four bytes are concatenated together as the nonce in the block header, and the block header containing the proposed nonce is hashed. Then, the hash value of the block header is checked with respect to the number of leading 0s. If the number of leading 0s is the same or greater than the difficulty level, then a solution is found. This single-threaded implementation can be easily made parallel by using 256 threads (*i.e.,* for all the possible values in the outermost byte) to search for a solution concurrently. Considering that data from all sensors will need to be aggregated into blocks, a single processing node could potentially aggregate for dozen's of sensors, which means dozen's instances of proof-of-work would have to be executed concurrently. The single-threaded implementation is more appropriate to help determine an appropriate difficulty level based on the need of the application.

We experimented with six levels of difficulty, from 1 to 6, to explore appropriate levels for sensing data logging. The probability density function for the time it takes to solve the proof-of-work puzzle (termed as PoW latency) for each of the difficulty levels are shown in Fig. 11. The density values as shown in the vertical axis in Fig. 11 are larger than 1 for levels 1, 2, 3, and 4. One might expect that the values to be lower than one. However, what matters is that the area below the density curve should sum up to be one. It is possible for the vertical axis to have values of greater than one.

The mean PoW latency in seconds for different levels are illustrated in Fig. 12. As can be seen, the mean PoW latency is linearly increasing in the log scale with respect to the difficulty level. For difficulty level 4, the mean PoW latency is $0.37s$. For difficulty level 5, the mean PoW latency is $4.50s$. For difficulty level 6, the mean PoW latency is $97.31s$. These levels are good candidates depending on the number of sensors and the number of processing nodes to perform sensing data logging. Lower difficulty levels incur too little work and therefore, are not sufficient towards the goal of creating a barrier to change. As expected, the time it takes to verify a proof-of-work solution is in the order of $10^{-}5s$, which is sharply less than the PoW latency.

Next, we investigated the impact of different levels of aggregation of the raw sensing data in terms of the block size and the time it takes to compute the Merkle root for each block of raw sensing data. The block size would be linearly increasing with the number of sensing samples included in the block. However, the size of the aggregated data to be placed on the public blockchain is independent from the block size of the raw sensing data. Naturally, the ratio of the block of

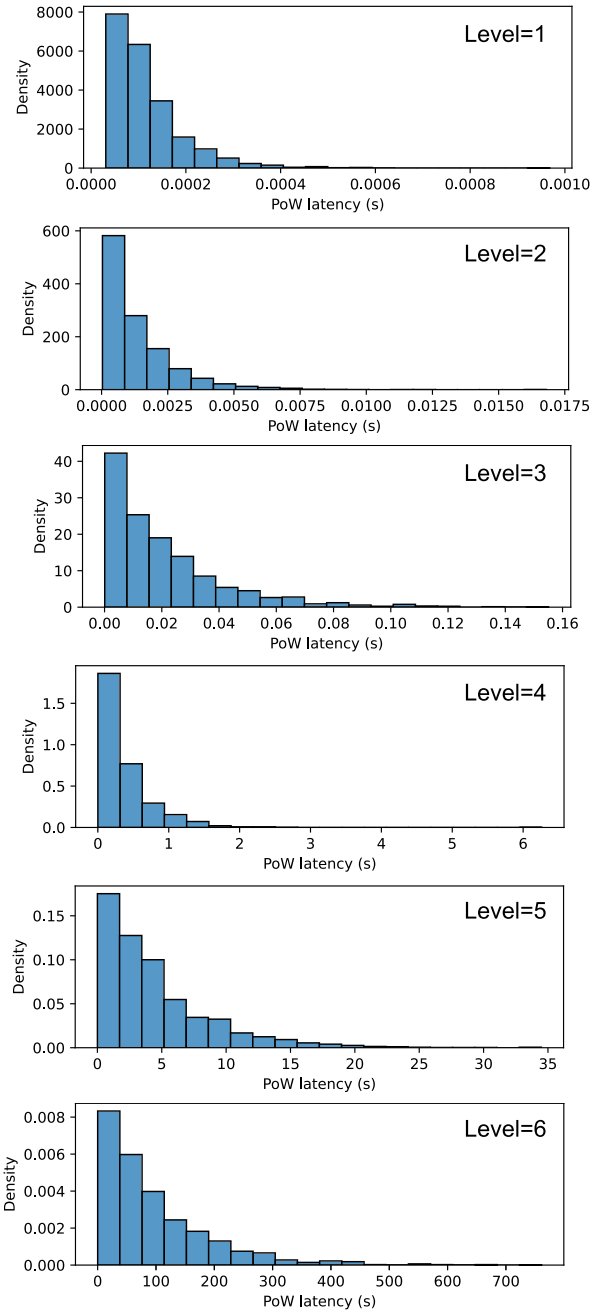


FIGURE 11. Probability density functions for the PoW latency for six difficulty levels.

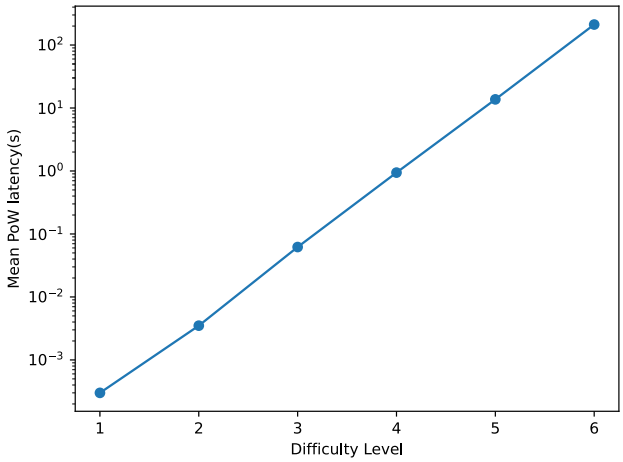


FIGURE 12. The mean PoW latencyl with respect to difficulty level. Please note the log scale in the vertical axis.

raw sensing data versus the aggregated data linearly increases with the number of samples included in the block. A greater level of aggregation would increase the scalability of our system in that it would reduce the fraction of sensing data placed on the public blockchain. This in turn would reduce the demand on the throughput of the public blockchain. In the experiment, we considered seven aggregation levels, from each block including 1024 (*i.e.,* 2^{10}) samples, to 2048 (*i.e.,* 2^{11}), 4096 (*i.e.,* 2^{12}), 8192 (*i.e.,* 2^{13}), 16384 (*i.e.,* 2^{14}), 32768 (*i.e.,* 2^{15}), 65536 (*i.e.,* 2^{16}), and 131072 (*i.e.,* 2^{17}) samples. The results are shown in Fig. 13. As can be seen in Fig. 13(a), the time it takes to compute the Merkle root grows to about 120 seconds for a block with 131072 samples. This latency is on par with level 6 proof-of-work PoW latency. Also as shown in Fig. 13(c), the compression ratio is on the order of 10,000.

To establish a guideline for the right difficulty level and the block size, we need to additionally consider the sampling rate of the sensors and the number of sensors used in the system. At low sampling rate and smaller number of sensors, a smaller block may be used. At high sampling rate and/or high number of sensors, a larger block size is necessary to ensure that the throughput demand on the public blockchain is low. The level of proof-of-work difficulty depends on the target desirable block interval B. The target block interval B is proportional to the number of samples included in the block s, but inversely proportional to the sampling rate r, *i.e.,* $B = s/r$. Furthermore, given n total sensors used in the system, n transactions will be created for the public blockchain per block interval. Therefore, the throughput demand T for the public blockchain is determined by Eq. 1.

$$T = \frac{n}{B} = \frac{n}{s/r} = \frac{nr}{s} \tag{1}$$

Given a predefined throughput demand, sampling rate, and the number of sensors, the block size in terms of the number of samples required can be derived by transforming Eq. 1

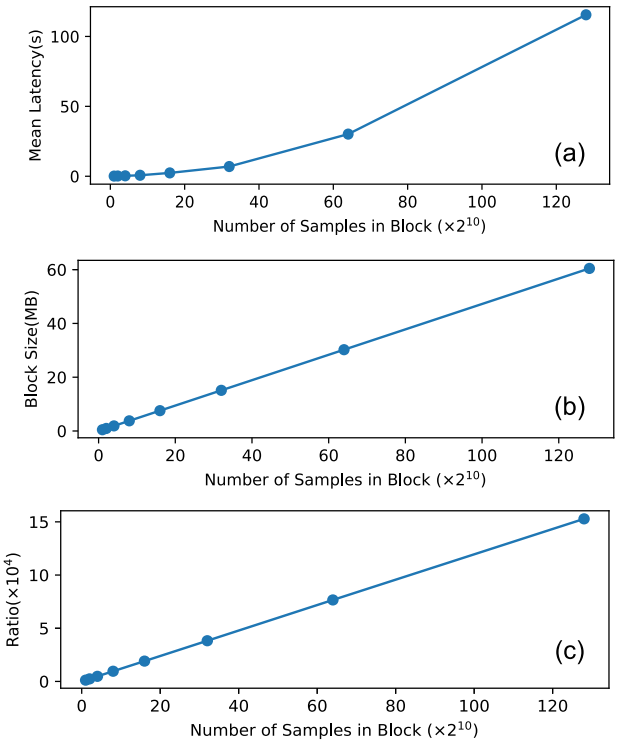


FIGURE 13. The mean latency in computing the Merkle root (a), the block size (b), and the ratio of the aggregated data versus the block of raw sensing data (c).

into Eq. 2:

$$s = \frac{nr}{T} \tag{2}$$

If T is set to be no more than 0.1 transactions per second, the number of sensors to be 100, and the sampling rate be 1Hz, the minimum number of samples per block will be $100 \times 1/0.1 = 1000$. We then round this minimum number of samples to the nearest power of 2, which is 1024. The target block interval is $B = s/r = 1024/1 = 1024$ seconds. In this case, the PoW difficulty level of 6 can be safely used without impacting the need for logging the raw sensing data. While performing the PoW calculation for one block, the new samples submitted by the sensor will be temporarily logged in a table of the datastore. While building the next block, the temporarily logged samples will be retrieved. Once the new block has been logged in the database, the temporarily logged samples can be purged. Given the same sampling rate, the larger block size, the higher PoW difficulty level may be used for better security. Furthermore, at this level of block interval, the latency for computing the Merkle root is negligible (less than 0.3 seconds).

To monitor a fast changing physical system, higher sampling rate such as 10Hz may be required. If the throughput demand remains to be 0.1 transactions per second and the number of sensors remain to be 100, the minimum number of samples per block will be $100 \times 10/0.1 = 10000$. The closest power of two is 16384. The block interval is then $16384/10 = 1638.4$ seconds. The latency on computing

the Merkle root is less than 3 seconds. Hence, the impact is negligible. This block interval allows the use of a relatively high PoW difficulty level of 6.

A large fast changing physical system may require the use of 1000 sensors with 10Hz sampling rate. Given the same throughput demand at 0.1 transactions per second, the minimum number of samples per block will be $1000 \times 10/0.1 = 100000$. The closest power of two is 131072. The block interval is then $131072/10 = 1310.72$ seconds. The latency on computing the Merkle root is about 120 seconds. While this is on par with the PoW latency for difficulty level of 6, the combined delay will not negatively impact the collection of sensing samples as we have elaborated previously.

B. AGGREGATED DATA STORAGE AND RETRIEVE WITH BLOCKCHAIN

To demonstrate the support for generic blockchain platforms, we experimented with two public blockchain systems, one is IOTA distributed ledger, and the other is Ethereum. The IOTA public distributed ledger uses a direct acyclic graph to store the transactions, which is referred to as Tangle. The current version of the IOTA network is referred to as Shimmer (https://api.testnet.shimmer.network). The testing is done using the testbed running the Shimmer protocol. For Ethereum, we used the Web3.py python library from Ethereum. To facilitate Ethereum application development, Web3.py offers an `EthereumTesterProvider`, which provides a simulated Ethereum node with relaxed permission and pseudo coins for experimentation.

A very interesting feature of IOTA is that a client that submits a transaction to the network must perform a moderate amount of proof-of-work, which is a clever design to mitigate attacks from spammers (particularly considering that no transaction fee is collected). This requirement limits the number of transactions that can be submitted by any client. The probability density function of the transaction submission latency and the transaction fetch latency (obtained by doing 2000 transaction submissions and 2000 transaction fetch operations) are shown in Fig. 14(a) and (b), respectively. As can be seen from Fig. 14(a) and Fig. 12, the transaction submission latency for IOTA is similar to level 4 of proof-of-work. The average transaction submission latency is 1.08s. The proof-of-work is not required for transaction fetch operation. That is why the transaction fetch latency is much shorter (0.19s).

The Ethereum transaction contains a data field, which can be used to store the aggregated data. Unlike IOTA, which explicitly supports data-only transactions and does not charge a transaction fee, an Ethereum transaction is designed to transfer coins from one account to another, and a transaction fee (in the form of gas as part of the transaction) is required for the transaction to be accepted and executed [18]. In our experiment, to store the aggregated data of about 400 bytes, the gas required is 41000. According to the Ethereum gas price tracking website (https://ethgasprice.org/), 41000 gas

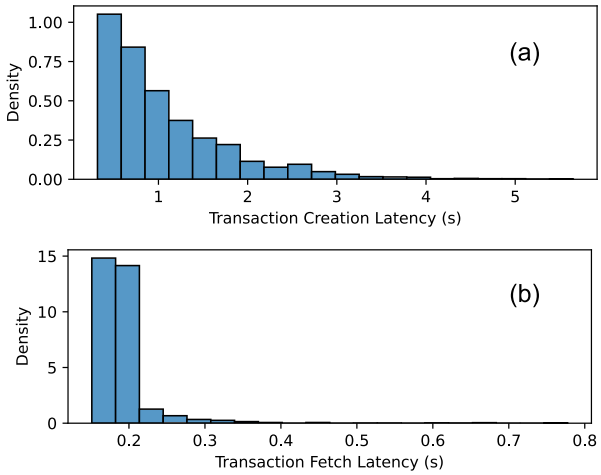


FIGURE 14. The probability density function of the transaction submission latency (a) and the transaction fetch latency (b).

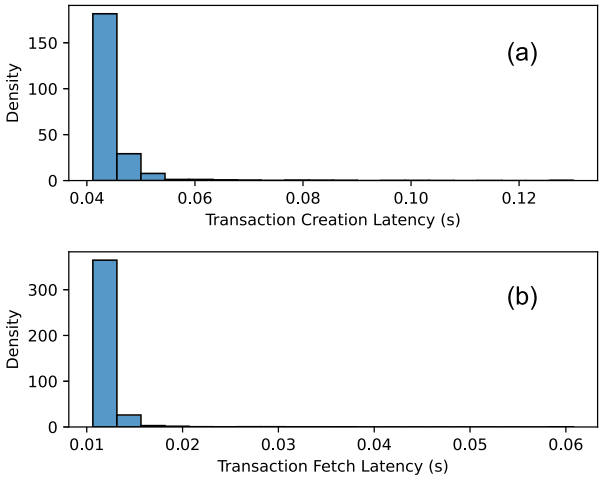


FIGURE 15. The probability density function of the transaction submission latency (a) and the transaction fetch latency (b).

is roughly about US $0.80. Over a long time, the cost for storing data on Ethereum could be significant. This fact further motivates the use of higher aggregation levels. The transaction creation time and the fetch time with Ethereum is much shorter, as shown in Fig. 15, because we used a simulated testing environment. As expected, the transaction creation time is much longer than the time to fetch the transaction.

C. PUBLIC KEY CRYPTOGRAPHY OPERATIONS ON RASPBERRY PI

For sensor identification and authentication, we do require that the sensors support the use of elliptic curve cryptography (ECC) operations. As we have acknowledged previously, this would preclude the use of microcontroller-based sensors and IoT devices. However, many ARM-based devices are powerful enough to do such public key operations. The Raspberry Pi line of products are good candidates for use with our system. The latency measurements results for doing ECC

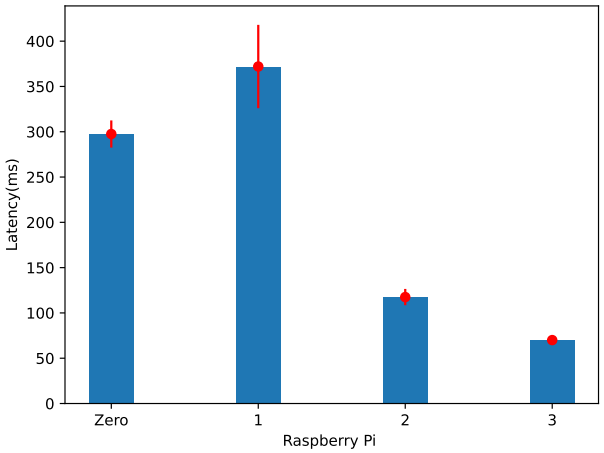


FIGURE 16. The latency in ECC operation on Raspberry Pi Zero, 1, 2, and 3.

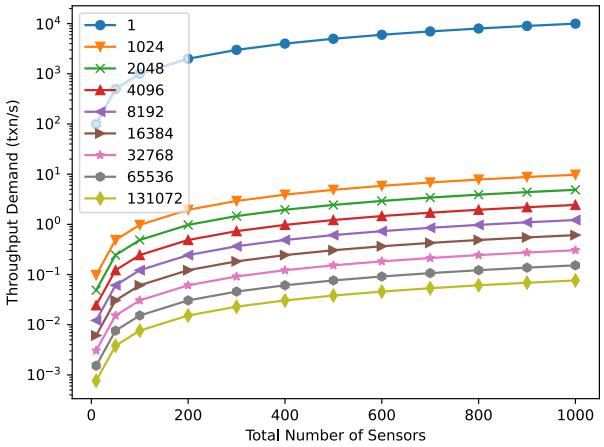


FIGURE 17. The throughput demand with respect to the total number of sensors in the system operating at 10Hz sampling rate for different aggregation levels.

operations on the Raspberry Pi Zero, 1, 2, and 3 have been fully documented in an excellent paper [67]. We verified with a Raspberry Pi 3 and the result is consistent with what has been reported in that paper. For the benefit of the readers, we report the results from [67]. In our implementation, we used the default ECC algorithm, which has a key length of 192 bits. The latency for EEC operations is largely determined by the key length and varies very little for different algorithm parameters. For the 192-bit key length, the latency for Raspberry Pi 3 varies between 69ms to 71ms, the latency for Raspberry Pi 2 varies between 113ms to 122ms, the latency for Raspberry Pi 1 varies between 349ms to 395ms, and the latency for Raspberry Pi zero varies between 290ms to 305ms. These data show that to support 10Hz sampling rate, a Raspberry Pi 3 or more powerful device must be used. If a 1Hz sampling rate is desirable, then any Raspberry Pi line products with ARM CPU may be used. This result is illustrated in Fig. 16.

D. SCALABILITY OF THE PROPOSED SYSTEM

The sensing data aggregation mechanism is designed to drastically increase the scalability of the sensing data processing and logging systems. To facilitate the discussion, we define the scalability in terms of the throughput demand on the public blockchain (*i.e.,* output from our system) in response to the total number of samples generated by the sensors per unit of time (*i.e.,* input to the system). By using increasing levels of aggregation for higher sensing data arrival rate, our system can ensure that the throughput demand on the public blockchain remains to be at a low level. This can be understood by looking at the formula provided in Eq. 1 as part of Section V-A. The throughput demand T can remain to be under a desirable level as long as the number of samples included in each block s increases with the total number of samples generated by the sensors nr. To illustrate the scalability of our system, we assume a 10Hz sampling rate at each sensor, and show the throughput demand on the public blockchain in response to various number of sensors used in the system with and without aggregation in Fig. 17. In Fig. 17, each curve corresponds to a different aggregation level where the legend denotes the number of samples for each block. As we have argued in the related work section II-C, in existing solutions, the sensors would directly interact with the blockchain systems, which corresponds to the no-aggregation case (*i.e.,* the top curve with legend of 1). Without any aggregation, the number of transaction per second is the same as the product of the total number of sensors and the sampling rate. If the total number of sensors is 1,000, then the throughput demand would be $1,000 \times 10 = 10,000$ transactions per second. This throughput demand exceeds the capability of any public blockchain. In contrast, with an aggregation level of 131,072 samples per block, the throughput demand remains below 0.1 transactions per second for 1,000 sensors.

VI. CONCLUSION

In this article, we presented the details of the design and implementation of a secure system for wireless sensing data processing and logging. The system incorporates several innovative blockchain-inspired mechanisms. The system also depends on the use of a public blockchain for security attainment. An import lesson that we learned while developing this system is that not only the obvious benefits of public blockchain, such as data immutability, can be used as the foundation for building various secure systems, the mechanisms introduced as part of the public chain technology can be employed to strengthen the security of the system, even for off-chain operations such as sensor authentication and data storage. Indeed, the sensor identification and authentication mechanism in our system was inspired by the user authentication mechanism commonly used in public blockchain systems. Similarly, the self-protection mechanism for the logged raw sensing data was inspired by the proof-of-work algorithm and the block chaining mechanism in Bitcoin. The use of a Merkle root as a fingerprint for a block of raw sensing data is also inspired by Bitcoin.

That said, we did not simply adopt these blockchain mechanisms naively in our system. We introduced additional mechanisms to make the system secure and usable for the particular purpose of wireless sensing data processing and logging. For example, the user authentication mechanism in blockchain by itself cannot mitigate identity-based attacks such as Sybil and spoofing attacks. We addressed this issue by introducing an ad-hoc network formation between the sensors and the sensor authenticator where each sensor is manually assigned a static IP address. In this setup, the sensors can communicate with the sensor authenticator directly without going through any third-party entity such as an access point, which could make the system vulnerable. The grouping of raw sensing data into blocks, and adding proof-of-work and block chaining in the block formation are meant to protect the local raw sensing data from arbitrary data in a very short amount time without being detected. Our work demonstrates the benefit of this scheme for local data security beyond reaching decentralized consensus. Furthermore, we introduced additional mechanisms to facilitate the construction of condensed data, to enable efficient search for the condensed data placed on the blockchain (with clustered indexing), and to quickly locate the corresponding raw sensing data block given a particular condensed sample.

REFERENCES

[1] T. Qiu, N. Chen, K. Li, M. Atiquzzaman, and W. Zhao, "How can heterogeneous Internet of Things build our future: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2011–2027, 3rd Quart., 2018.

[2] W. Zhao, C. Jiang, H. Gao, S. Yang, and X. Luo, "Blockchain-enabled cyber-physical systems: A review," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4023–4034, Mar. 2021.

[3] H. A. Abdul-Ghani and D. Konstantas, "A comprehensive study of security and privacy guidelines, threats, and countermeasures: An IoT perspective," *J. Sensor Actuator Netw.*, vol. 8, no. 2, p. 22, Apr. 2019.

[4] I. Butun, P. Osterberg, and H. Song, "Security of the Internet of Things: Vulnerabilities, attacks and countermeasures," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 616–644, 1st Quart., 2020.

[5] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: Application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82721–82743, 2019.

[6] W. Viriyasitavat, L. D. Xu, Z. Bi, and D. Hoonsopon, "Blockchain technology for applications in Internet of Things—Mapping from system design perspective," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8155–8168, Oct. 2019.

[7] X. Wang, X. Zha, W. Ni, R. P. Liu, Y. J. Guo, X. Niu, and K. Zheng, "Survey on blockchain for Internet of Things," *Comput. Commun.*, vol. 136, pp. 10–29, Feb. 2019.

[8] B. Alotaibi, "Utilizing blockchain to overcome cyber security concerns in the Internet of Things: A review," *IEEE Sensors J.*, vol. 19, no. 23, pp. 10953–10971, Dec. 2019.

[9] W. Zhao, S. Yang, and X. Luo, "On threat analysis of IoT-based systems: A survey," in *Proc. IEEE Int. Conf. Smart Internet Things (SmartIoT)*, Aug. 2020, pp. 205–212.

[10] J. Sengupta, S. Ruj, and S. D. Bit, "A comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT," *J. Netw. Comput. Appl.*, vol. 149, Jan. 2020, Art. no. 102481.

[11] U. Guin, P. Cui, and A. Skjellum, "Ensuring proof-of-authenticity of IoT edge devices using blockchain technology," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Jul. 2018, pp. 1042–1049.

[12] U. Javaid, M. N. Aman, and B. Sikdar, "BlockPro: Blockchain based data provenance and integrity for secure IoT environments," in *Proc. 1st Workshop Blockchain-Enabled Networked Sensor Syst.*, Nov. 2018, pp. 13–18.

[13] L. Aniello, B. Halak, P. Chai, R. Dhall, M. Mihalea, and A. Wilczynski, "Anti-BLUFf: Towards counterfeit mitigation in IC supply chains using blockchain and PUF," *Int. J. Inf. Secur.*, vol. 20, pp. 445–460, Jun. 2021.

[14] S. P. Mohanty, V. P. Yanambaka, E. Kougianos, and D. Puthal, "PUFchain: A hardware-assisted blockchain for sustainable simultaneous device and data security in the internet of everything (IoE)," *IEEE Consum. Electron. Mag.*, vol. 9, no. 2, pp. 8–16, Mar. 2020.

[15] A. S. Patil, R. Hamza, A. Hassan, N. Jiang, H. Yan, and J. Li, "Efficient privacy-preserving authentication protocol using PUFs with blockchain smart contracts," *Comput. Secur.*, vol. 97, Oct. 2020, Art. no. 101958.

[16] A. K. Das, S. Kalam, N. Sahar, and D. Sinha, "UCFL: User categorization using fuzzy logic towards PUF based two-phase authentication of fog assisted IoT devices," *Comput. Secur.*, vol. 97, Oct. 2020, Art. no. 101938.

[17] W. Yan, N. Zhang, L. L. Njilla, and X. Zhang, "PCBChain: Lightweight reconfigurable blockchain primitives for secure IoT applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 10, pp. 2196–2209, Oct. 2020.

[18] W. Zhao, *From Traditional Fault Tolerance to Blockchain*. Hoboken, NJ, USA: Wiley, 2021.

[19] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[20] S. Popov and Q. Lu, "IOTA: Feeless and free," *IEEE Blockchain Tech. Briefs*, Jan. 2019. [Online]. Available: https://blockchain.ieee.org/technicalbriefs/january-2019

[21] V. Buterin. (2013). *Ethereum White Paper*. [Online]. Available: https://ethereum.org/en/whitepaper/

[22] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22328–22370, 2019.

[23] W. Zhao, S. Yang, and X. Luo, "On consensus in public blockchains," in *Proc. Int. Conf. Blockchain Technol.*, Mar. 2019, pp. 1–5.

[24] U. Bodkhe, D. Mehta, S. Tanwar, P. Bhattacharya, P. K. Singh, and W.-C. Hong, "A survey on decentralized consensus mechanisms for cyber physical systems," *IEEE Access*, vol. 8, pp. 54371–54401, 2020.

[25] M. U. Hassan, M. H. Rehmani, and J. Chen, "Differential privacy techniques for cyber physical systems: A survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 746–789, 1st Quart., 2019.

[26] R. Gupta, S. Tanwar, F. Al-Turjman, P. Italiya, A. Nauman, and S. W. Kim, "Smart contract privacy protection using AI in cyber-physical systems: Tools, techniques and challenges," *IEEE Access*, vol. 8, pp. 24746–24772, 2020.

[27] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Gener. Comput. Syst.*, vol. 82, pp. 395–411, May 2018.

[28] N. Tariq, M. Asim, F. Al-Obeidat, M. F. Farooqi, T. Baker, M. Hammoudeh, and I. Ghafir, "The security of big data in fog-enabled IoT applications including blockchain: A survey," *Sensors*, vol. 19, no. 8, p. 1788, 2019.

[29] W. Zhao, "On blockchain: Design principle, building blocks, core innovations, and misconceptions," *IEEE Syst., Man, Cybern. Mag.*, vol. 8, no. 4, pp. 6–14, Oct. 2022.

[30] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, "Applications of blockchains in the Internet of Things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1676–1717, 2nd Quart., 2019.

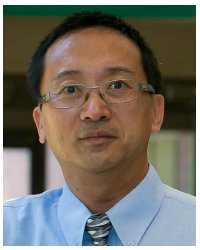
[31] E. L. Macedo, E. A. de Oliveira, F. H. Silva, R. R. Mello, F. M. Franca, F. C. Delicato, J. F. de Rezende, and L. F. de Moraes, "On the security aspects of Internet of Things: A systematic literature review," *J. Commun. Netw.*, vol. 21, no. 5, pp. 444–457, Oct. 2019.

[32] H.-N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for Internet of Things: A survey," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8076–8094, Oct. 2019.

[33] F. H. Pohrmen, R. K. Das, and G. Saha, "Blockchain-based security aspects in heterogeneous Internet-of-Things networks: A survey," *Trans. Emerg. Telecommun. Technol.*, vol. 30, no. 10, p. e3741, Oct. 2019.

[34] M. A. Ferrag, M. Derdour, M. Mukherjee, A. Derhab, L. Maglaras, and H. Janicke, "Blockchain technologies for the Internet of Things: Research issues and challenges," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2188–2204, Apr. 2019.

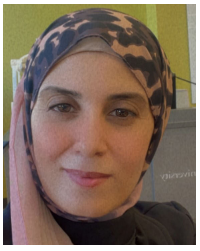
[35] F. Casino, T. Dasaklis, and C. Patsakis, "A systematic literature review of blockchain-based applications: Current status, classification and open issues," *Telematics Inform.*, vol. 36, pp. 55–81, Mar. 2018.
[36] M. J. Meirino, M. P. Méxas, A. D. V. Faria, R. P. Méxas, and G. D. Meirelles, "Blockchain technology applications: A literature review," *Brazilian J. Oper. Prod. Manag.*, vol. 16, no. 4, pp. 672–684, 2019.
[37] I. Mistry, S. Tanwar, S. Tyagi, and N. Kumar, "Blockchain for 5G-enabled IoT for industrial automation: A systematic review, solutions, and challenges," *Mech. Syst. Signal Process.*, vol. 135, Jan. 2020, Art. no. 106382.
[38] A. V. Barenji, Z. Li, W. Wang, G. Q. Huang, and D. A. Guerra-Zubiaga, "Blockchain-based ubiquitous manufacturing: A secure and reliable cyber-physical system," *Int. J. Prod. Res.*, vol. 58, no. 7, pp. 2200–2221, 2020.
[39] Y. J. Qu, X. G. Ming, Z. W. Liu, X. Y. Zhang, and Z. T. Hou, "Smart manufacturing systems: State of the art and future trends," *Int. J. Adv. Manuf. Technol.*, vol. 103, nos. 9–12, p. 3751, Aug. 2019.
[40] A. S. Musleh, G. Yao, and S. M. Muyeen, "Blockchain applications in smart grid—Review and frameworks," *IEEE Access*, vol. 7, pp. 86746–86757, 2019.
[41] P. Fraga-Lamas and T. M. Fernández-Caramés, "A review on blockchain technologies for an advanced and cyber-resilient automotive industry," *IEEE Access*, vol. 7, pp. 17578–17598, 2019.
[42] T. M. Fernandez-Carames and P. Fraga-Lamas, "A review on the application of blockchain to the next generation of cybersecure industry 4.0 smart factories," *IEEE Access*, vol. 7, pp. 45201–45218, 2019.
[43] J. Xie, H. Tang, T. Huang, F. R. Yu, R. Xie, J. Liu, and Y. Liu, "A survey of blockchain technology applied to smart cities: Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2794–2830, 3rd Quart., 2019.
[44] T. M. Fernández-Caramés and P. Fraga-Lamas, "Towards the internet of smart clothing: A review on IoT wearables and garments for creating intelligent connected e-textiles," *Electronics*, vol. 7, no. 12, p. 405, 2018.
[45] G. Aceto, V. Persico, and A. Pescape, "A survey on information and communication technologies for industry 4.0: State-of-the-art, taxonomies, perspectives, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3467–3501, 4th Quart., 2019.
[46] L. D. Xu, E. L. Xu, and L. Li, "Industry 4.0: State of the art and future trends," *Int. J. Prod. Res.*, vol. 56, no. 8, pp. 2941–2962, 2018.
[47] H. Rathore, A. Mohamed, and M. Guizani, "A survey of blockchain enabled cyber-physical systems," *Sensors*, vol. 20, no. 1, p. 282, Jan. 2020.
[48] A. Islam, A. Al Amin, and S. Y. Shin, "FBI: A federated learning-based blockchain-embedded data accumulation scheme using drones for Internet of Things," *IEEE Wireless Commun. Lett.*, vol. 11, no. 5, pp. 972–976, May 2022.
[49] J.-H. Lee, "BIDaaS: Blockchain based ID as a service," *IEEE Access*, vol. 6, pp. 2274–2278, 2017.
[50] Q. Lin, H. Yan, Z. Huang, W. Chen, J. Shen, and Y. Tang, "An ID-based linearly homomorphic signature scheme and its application in blockchain," *IEEE Access*, vol. 6, pp. 20632–20640, 2018.
[51] L. Li, J. Liu, L. Cheng, S. Qiu, W. Wang, X. Zhang, and Z. Zhang, "CreditCoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 7, pp. 2204–2220, Jul. 2018.
[52] C. Lin, D. He, X. Huang, K. K. R. Choo, and A. V. Vasilakos, "BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0," *J. Netw. Comput. Appl.*, vol. 116, pp. 42–52, Aug. 2018.
[53] C. Lin, D. He, X. Huang, M. K. Khan, and K.-K. R. Choo, "A new transitively closed undirected graph authentication scheme for blockchain-based identity management systems," *IEEE Access*, vol. 6, pp. 28203–28212, 2018.
[54] K. Fan, Y. Ren, Y. Wang, H. Li, and Y. Yang, "Blockchain-based efficient privacy preserving and data sharing scheme of content-centric network in 5G," *IET Commun.*, vol. 12, no. 5, pp. 527–532, Mar. 2018.
[55] N. Z. Aitzhan and D. Svetinovic, "Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams," *IEEE Trans. Depend. Sec. Comput.*, vol. 15, no. 5, pp. 840–852, Oct. 2018.
[56] K. Kanemura, K. Toyoda, and T. Ohtsuki, "Design of privacy-preserving mobile bitcoin client based on γ-deniability enabled Bloom filter," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–6.
[57] Q. Wang, B. Qin, J. Hu, and F. Xiao, "Preserving transaction privacy in bitcoin," *Future Gener. Comput. Syst.*, vol. 107, pp. 793–804, Jun. 2020.
[58] C. Yang, X. Chen, and Y. Xiang, "Blockchain-based publicly verifiable data deletion scheme for cloud storage," *J. Netw. Comput. Appl.*, vol. 103, pp. 185–193, Feb. 2018.
[59] P. Otte, M. De Vos, and J. Pouwelse, "TrustChain: A sybil-resistant scalable blockchain," *Future Gener. Comput. Syst.*, vol. 107, pp. 770–780, Jun. 2020.
[60] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," Lightning Labs, Lighting Network, Tech. Rep., 2016. [Online]. Available: https://lightning.network/lightning-network-paper.pdf
[61] R. C. Merkle, "A digital signature based on a conventional encryption function," in *Proc. Conf. Theory Appl. Cryptograph. Techn.* Cham, Switzerland: Springer, 1987, pp. 369–378.
[62] W. Zhao, S. Yang, and X. Luo, "Secure hierarchical processing and logging of sensing data and IoT events with blockchain," in *Proc. 2nd Int. Conf. Blockchain Technol.*, Mar. 2020, pp. 52–56.
[63] C. K. Cockrum. (2009). *Implementation of An Elliptic Curve Cryptosystem on An 8-Bit Microcontroller*. [Online]. Available: https://cockrum.net/Implementation_of_ECC_on_an_8-bit_microcontroller.pdf
[64] Y. Nanjo, M. A. A. Khandaker, T. Kusaka, and Y. Nogami, "Efficient pairing-based cryptography on raspberry Pi," *J. Commun.*, vol. 13, no. 2, pp. 88–93, 2018.
[65] W. Zhao, H. Upadhyay, and L. Lagos, "Design and implementation of a blockchain-enabled secure sensing data processing and logging system," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2021, pp. 386–391.
[66] L. Aronovich and I. Spiegler, "CM-tree: A dynamic clustered index for similarity search in metric databases," *Data Knowl. Eng.*, vol. 63, no. 3, pp. 919–946, 2007.
[67] J. Brychta, "Benchmarks with points on elliptic curves," in *Proc. 25st Conf. Student Eeict.*, 2019, pp. 520–524. [Online]. Available: http://hdl.handle.net/11012/186727



WENBING ZHAO (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of California at Santa Barbara, Santa Barbara, in 2002.

He joined Cleveland State University (CSU), Cleveland, OH, USA, in 2004, as an Assistant Professor. He is currently a Full Professor with the Department of Electrical Engineering and Computer Science, CSU. He is the author of two research monographs titled *Building Dependable Distributed Systems* (Wiley-Scrivener, 2004) and *From Traditional Fault Tolerance to Blockchain* (Wiley-Scrivener, 2021). He has published over 200 peer-reviewed journals and conference papers. His current research interests include dependable distributed computing and human-centered computing.

Dr. Zhao has served on the organizing and technical committees of numerous IEEE conferences. He has received the 2018 and 2020 IEEE Access Outstanding Associate Editor Award. He is an Associate Editor of IEEE Access.



IZDEHAR M. ALDYAFLAH received the bachelor's degree (Hons.) in computer engineering from Al-Balqa Applied University, in 2010, and the M.S. degree in electrical engineering from Cleveland State University (CSU), in 2019. She is currently pursuing the Ph.D. degree with the Department of Electrical Engineering and Computer Science. She has been a Teaching Assistant and a Research Assistant with CSU. Her research interests include cybersecurity, blockchain consensus, smart contract, blockchain-enabled applications, machine learning, deep learning, and artificial intelligence. She is a member of the Eta Kappa Nu International Honor Society.

PRANAV GANGWANI (Graduate Student Member, IEEE) received the M.S. degree in computer engineering from Florida International University (FIU), in 2020, where he is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering. He is currently a Graduate Research Assistant on the Department of Energy National Energy Technology Laboratory (DOE-NETL) Project with the Applied Research Center (ARC). His role is to build the entire blockchain platform to securely store and retrieve sensor data collected from a fossil fuel power plant and ensure the authenticity of the data. His current research interests include big data, blockchain, artificial intelligence, and cyber security.

SANTOSH JOSHI received the B.S. degree in computer science engineering from Karnatak University, India, and the master's degree in engineering management from Florida International University (FIU), where he is currently pursuing the Ph.D. degree in computer engineering. He has 18 years of experience in academia and IT industry in applied research and implementation of technology solutions designed to address complex problems. He has expertise in application development, maintenance, and technical project management. He is currently a Research Specialist II with the Applied Research Center (ARC), FIU. He has published multiple research papers in various journals and conferences. He worked on multiple research projects into the database implementations, data warehousing, data analysis, artificial intelligence, machine learning, and big data analytics in the cyber security domain. He is currently working on multiple research projects funded by the Department of Defense (DOD)—Test Resource Management Center (TRMC), Department of Energy (DOE) Environmental Management (EM); and DOE—National Energy Technology Laboratory (NETL).

HIMANSHU UPADHYAY is currently an Associate Professor with the College of Electrical and Computer Engineering, Florida International University. He brings more than 30 years of experience in applied artificial intelligence/machine learning, big data, cybersecurity, information technology, management, and engineering to his role, serving as a Co-Principal Investigator for multimillion—dollar cybersecurity and applied artificial intelligence projects for the Department of Defense and Defense Intelligence Agency. He is also responsible for knowledge/waste management, artificial intelligence, and big data research projects for the Department of Energy's Office of Environmental Management. He has published multiple articles in cybersecurity, machine learning, deep learning, big data, knowledge/nuclear waste management, and service-oriented architecture. His current research interests include applied artificial intelligence, machine learning, deep learning, big data, cyber analytics/visualization, cyber forensics, malware analysis, and blockchain.

LEONEL LAGOS received the B.S. degree in aerospace engineering from the University of Florida, in 1991, and the master's degree in mechanical engineering and the Ph.D. degree in environmental engineering from Florida International University (FIU), USA, in 1996 and 2007, respectively.

He is currently an Associate Professor with the Moss Department of Construction Management and the Director of Research at the Applied Research Center, FIU. He is also a Principal Investigator with the Department of Energy (DOE)—FIU Cooperative Agreement. He is the Program Director of the DOE—FIU Science and Technology Workforce Development Program.

Dr. Lagos is an Active Member of the DOE's Energy Facilities Contractors Group (an advisory group supporting DOE's environmental restoration mission) and serves as a Group Member for EFCOG's D&D and the Facility Engineering Working Group. He serves as an active Program Advisory Committee (PAC) Member for the Waste Management Symposia Organization. He also serves as the Executive Committee Member for the American Nuclear Society's Robotics and Remote Systems Division.