Implementation of a Secure LoRaWAN System for Industrial Internet of Things Integrated With IPFS and Blockchain

Md. Shahjalal, *Student Member, IEEE*, Md. Mainul Islam, *Graduate Student Member, IEEE*, Md. Morshed Alam, *Graduate Student Member, IEEE*, and Yeong Min Jang, *Member, IEEE*

***Abstract*—Low-power, low-cost, and long-range connectivity for the Industrial Internet of Things (IIoT) networks are the key stipulations, nowadays. However, implementing a cost-effective, flexible, and feasible system considering server and networking security is still an open challenge. In this article, a complete end-to-end long-range wide area network (LoRaWAN) system has been demonstrated by implementing blockchain-based secure distributed data management, which is applicable in various secure IIoT applications. Dynamic data collected by multiple LoRa sensors are encrypted in a LoRa server, and the encrypted content is automatically stored in the InterPlanetary file system (IPFS) to ensure data confidentiality, integrity, and availability. To achieve data consistency, the content IDs collected from the IPFS are stored in the quorum blockchain with consortium setup using a smart contract. The consortium network is maintained by the Raft consensus algorithm employing seven nodes. The design architecture of the hardware used for both LoRa transmitting node and gateway has been described in comprehensive manners. The performance of the LoRaWAN system is analyzed by the received signal strength indicator, the communications range, and packet loss rate metrics in both line-of-sight and nonline-of-sight test systems. The data management scheme is implemented in Python, and the performance is evaluated in terms of transaction time and block size.**

***Index Terms*—Blockchain, Industrial Internet of Things (IIoT), long range, long-range wide area network (LoRaWAN), security.**

I. Introduction

INDUSTRIAL Internet of Things (IIoT) is generally a form of machine-to-machine communication that focuses on the collection of real-time machine monitoring, smart production, industry environment, logistic and quality control, and power management data [1]. IIoT aims to set a communication link between industrial machinery and control systems with real-time monitoring and maintenance for smart manufacturing. The demand for highly scalable and flexible IIoT networking systems is increasing due to the fourth industrial revolution (or Industry 4.0) [2]. In the past, the networking systems for industrial control and automation were only limited to wired topology with small-scale deployment and high cost. With the advantages of long-term reliability, cost effectiveness, scalability, and flexibility, the use of wireless technologies has been escalating greatly in recent years [3]. However, wireless personal and local area networks, i.e., WiFi, Bluetooth, BLE, and Zigbee, are not suitable for large IIoT deployment due to the low coverage and high power consumption. Long-term evolution for machines and narrowband-IoT (NB-IoT) are the two cellular IoT protocols currently vying for dominance in industrial applications. However, the essential infrastructure support and the licensed band increase the operational cost for them [4]. IIoT applications typically require a relatively low throughput per node, and therefore, the capacity is not the main concern [5]. Furthermore, cost/energy efficiency is a vital issue for massively connected IIoT networks. Long-range wide area network (LoRaWAN) is a promising candidate technology for low-power wide area networks, which uses a license-free spectrum, consumes less power, and provides longer battery life (15 years), compared with NB-IoT. Moreover, based on the frequency plan, it can support the bandwidth of 125, 250, or 500 kHz, which is higher than that of NB-IoT (180 kHz) [6]. LoRaWAN uses LoRa spread spectrum modulation technique [7], [8] and transmits packets of a small size over a long distance (5−15 km) consuming very low power and using relatively low-cost chipsets [9].

LoRaWAN entity is based on an advanced encryption standard (AES) that provides authenticated packet transmission and ensures data security between LoRa nodes and server. However, it cannot ensure the data integrity of the sensor data while storing them in a traditional MySQL database [10]. To achieve data integrity along with consistency, the integration of blockchain with the LoRaWAN system can play an important role. A blockchain is an immutable shared ledger, which keeps record of timestamped transactions and eliminates the need for a trusted third party (TTP). It arranges transactions in a binary or Merkle tree structure and stores them in chronological blocks across a distributed network. Because the blocks are connected to each other using cryptography and can be publicly verified, it is impossible to modify a transaction once it is stored in a block. Thus, this technology offers high resistance against data

Manuscript received 4 May 2021; revised 21 December 2021; accepted 1 May 2022. Date of publication 1 June 2022; date of current version 9 December 2022. This work was partly supported by the Technology Development Program of MSS under Grant S3098815 and the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program under Grant IITP-2022-2018-0-01396 supervised by the IITP (Institute for Information and Communications Technology Planning and Evaluation). *(Corresponding author: Yeong Min Jang.)*

Md. Shahjalal, Md. Morshed Alam, and Yeong Min Jang are with the Department of Electronics Engineering, Kookmin University, Seoul 02707, South Korea (e-mail: mdshahjalal26@ieee.org; mmorshed@ieee.org; yjang@kookmin.ac.kr).

Md. Mainul Islam is with the Department of Computer Science, Korea University, Seoul 02841, South Korea (e-mail: mainul.islam@ieee.org).

Digital Object Identifier 10.1109/JSYST.2022.3174157

alteration and ensures data integrity and transparency without needing a TTP or auditor [13]−[16]. Blockchains are decentralized, which means they hold data across a network of participants that work together to monitor and manage the network. The system as a whole cannot be owned by a single entity, rather everyone on the network owns it.

Besides many advantages, a downside of blockchains is that they require high storage capacity. For a higher transaction size, the size of a blockchain increases exponentially with the number of blocks. Therefore, massive data that increase the transaction size should not be included in transactions; rather, they can be stored in a separate data storage server and their metadata (i.e., URL and hash value) can be included in transactions to record on the blockchain. However, if the server is somehow offline, the transactions cannot be verified, and therefore, the system will lack transparency. This contradictory issue can be solved by adopting the InterPlanetary file system (IPFS) for off-chain data storage. IPFS is a distributed file system that enables peer-to-peer (P2P) file sharing worldwide while ensuring data integrity. It provides data integrity by using a multihash scheme that includes several cryptographic hash algorithms. However, IPFS cannot protect the data confidentiality of a file because the file becomes globally accessible via its content identifier (CID) after it is uploaded and pinned. To preserve data confidentiality, the file should be encrypted first and then its cipher text should be stored in IPFS node instead of plain text. In these ways, the combination of IPFS and blockchain can satisfy all the essential properties of data management, such as data confidentiality, integrity, verifiability, and consistency, in an IoT infrastructure.

A decentralized LoRaWAN 1.0.2 architecture using was proposed in [17] based on passive roaming techniques. A smart contract is used to the join registry of end devices on the blockchain, while we mainly adopt blockchain for sensor data management. The limitation of their approach is the absence of digital signatures over media access control (MAC) messages. Because the originator of a LoRaWAN message cannot be identified cryptographically, their blockchain-based proposed solution is incomplete in this article. However, they addressed the limitation in [18] where the elliptic curve digital signature is used to add a nonrepudiation feature. They also verified the feasibility of their solution by providing a full-scale demonstration. Recently, blockchain has been proposed in LoRaWAN systems to enhance security and key management issues. In [19], smart contact and permissioned blockchain-based architecture for LoRaWAN key management were proposed to enhance the security and availability. A similar work was proposed in [20] with a new approach to update the root key. LoRaWAN has been creating opportunities for healthcare industries to provide long-range, secure communications using blockchain services. Froiz-Mìguez *et al.* [21] provided an implementation of LoRaWAN and blockchain-based health monitoring system for Industry 4.0 operators. The system collects health data from the sensors integrated into a wearable that monitors the operators' health and safety. The collected data are transmitted through LoRaWAN and directly stored in IPFS. Part of such data or data hashes is stored in a blockchain through a smart contract. However, a major concern with their system is that it is serverless

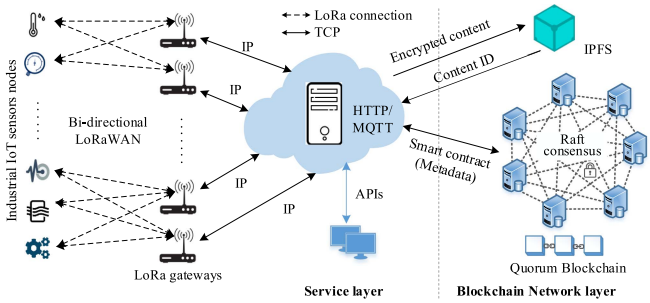

Fig. 1. Proposed end-to-end LoRaWAN scenario.

and stores operators' health data in a decentralized network without any encryption. As anyone in the IPFS network can view personal data, the system cannot ensure data confidentiality and, therefore, affects operators' privacy. Although our system does not focus on health data management, it is capable to handle this task by including health monitoring sensors without compromising users' privacy. This is because sensor data in our system are initially encrypted in a server, and then, the encrypted data are dispersed to the decentralized network ensuring data confidentiality.

In this article, we propose an IPFS and consortium blockchain-integrated LoRaWAN design to deploy a low power, secure, and trusted data framework in IIoT environment. The overall scenario is illustrated in Fig. 1. For simplicity, our demonstrated end-to-end LoRaWAN system contains three sensor nodes, one gateway, and a MySQL local server. It is worth mentioning that the number of sensor nodes can be increased up to thousands of nodes based on the requirements by incorporating several schemes mentioned in the recent studies [7], [22], [23]. However, this article focuses to establish an end-to-end connectivity by designing both the LoRa nodes and gateway and integrate the system with the proposed blockchain-enabled network server that would be compatible in the industrial scenarios. The implemented LoRaWAN connectivity has been tested in both line-of-sight (LOS) and nonLOS (NLOS) environments, considering multiple building obstacles. With an acceptable packet loss rate (PLR) of only 5%, a communication distance of 400 m can be achieved in a highly attenuated NLOS link. The performance of the proposed IPFS and blockchain-based LoRaWAN system is tested by making transactions on the quorum blockchain. The time required for encryption and upload to IPFS is 26.6 ms per data list and 49.3 ms per object, respectively. On average, 135.2 ms consensus time is required for the seven nodes.

The rest of this article is organized as follows. Section II introduces the LoRa and LoRaWAN systems along with the PHY and channel characteristics. Section III presents the demonstrated LoRaWAN system's hardware design, including both sensor node and gateway. This section also describes the proposed IPFS and blockchain-integrated network layer of the LoRaWAN system. Section IV analyzes the experimental results and performance of the proposed blockchain system. Finally, Section V concludes this article.

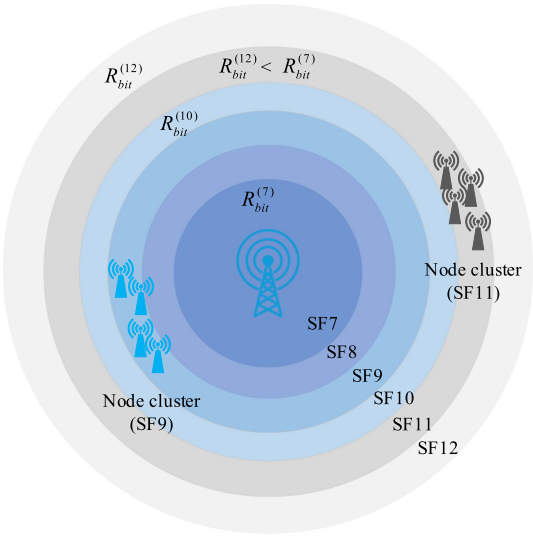

Fig. 2. LoRaWAN system bit rate comparison on different SFs.

II. Background

A. LoRa and LoRaWAN Overview

The LoRa physical layer (PHY) solution is a proprietary system that was patented by Semtech and further promoted by LoRa Alliance [10]. LoRa PHY is based on the chirp spread spectrum (CSS) modulation technique. CSS technique has been utilized for decades in space and military private networking purposes due to its large communication range, whereas LoRa uses it for low-cost commercial applications. The CSS technique comprises a combination of up-chirp and down-chirp symbols, which has the flexibility to utilize the entire available bandwidth. The communication range highly depends on the spreading factor (SF). SF is simply the duration of chirp. Higher SF takes a longer time for each chirp modulation and results in increasing time on-air (ToA). Therefore, a higher SF can significantly increase the range, however compensates the rate of transmission, as illustrated in Fig. 2. The bandwidth also has a significant role in adjusting the data rate. LoRa operates in the sub-GHz industrial, scientific, and medical band and generally uses three bandwidths: 125, 250, and 500 kHz. An increase in operating bandwidth desensitizes the receiver by increasing the probability of the addition of noise power in the channel. Therefore, for a fixed SF, the designer can choose either a narrow bandwidth by increasing the sensitivity but decreasing the transmission rate or a wider bandwidth for a higher transmission rate by negotiating the receiver sensitivity. LoRaWAN is a point-to-multipoint MAC protocol that uses Semtech's LoRa modulation scheme [10]. A LoRa system constitutes a star networking architecture that helps preserving battery lifetime when confirming long-range connectivity. This system consists of end sensor nodes, gateways, network servers, and user interfaces. The network servers act as the root in the star networking system. The sensor nodes transmit data using LoRaWAN protocol to multiple gateways. LoRaWAN determines how the radio waves communicate with gateways to do things, such as encryption and identification. These gateways convey the received data to the network server via some backhaul networks (i.e., cellular, Ethernet, or WiFi). The network server manages the network by filtering the redundant packets from multiple gateways and sending acknowledgement signals through the optimal gateway. The data reception from multiple gateways mitigates the issue of handover management for a mobile end node. LoRaWAN can be used for industrial applications or other private networks by generating LoRa synchronization keys. The gateways that lie within the private network can receive the data by matching it with the keys and forward it to the server.

B. LoRa PHY Parameters

For a specific channel and bandwidth, BW, different SFs $\{F \in \mathbb{Z} | 7 \leq F \leq 12\}$ can be allocated to each node. LoRa modulation allows the spreading of the spectrum by generating a chirp signal that continuously varies through the whole available bandwidth. The data collected from the sensor are chipped at a higher rate and modulated onto the chirp signal. The chip rate is defined as one chip being sent per second per Hz of bandwidth. Hence, for this system, the frequency bandwidth of this chirp is equivalent to the spectral bandwidth of the signal. This definition exists in the relationship between the chip rate and symbol rate. Chip rate is defined as $R_{\text{chip}} = 2^F \times R_{\text{sym}}$, where R_{sym} is the symbol rate defined as $R_{\text{sym}} = \frac{\text{BW}}{2^F}$. Thus, this relation gives

$$R_{\text{chip}} = \text{BW}. \tag{1}$$

The data can be corrupted by interference from other signals of the same bandwidth and SF. Therefore, LoRa includes a forward error correction (FEC) technique to minimize the bit error during packet reconstruction at the receiver. According to the FEC technique, redundant error correction bits (ECBs) are added along with the transmitted data. Adding more redundant bits to the payload eases restoring the original data. The proportion of adding ECB and the number of bits that actually carries the original data are determined by the coding rate. The coding rate that LoRa allows is represented as $R_{\text{coding}} = \frac{4}{4+n}$, where $\{n \in \mathbb{Z} | 1 \leq n \leq 4\}$. A PHY frame format is specified for the Semtech's LoRa transmitters and receivers [10]. It consists of a preamble, PHY-header (PHDR), PHY-payload, and payload cyclic redundancy check (CRC). The preamble is used to recognize the start of the frame and it includes a specified synchronization word (SyncWord). The LoRa SyncWord helps to isolate LoRa networks that uses the same frequency bands [25]. The PHDR can be set as optional and is transmitted using the R_{coding} of $4/8$. PHDR holds the necessary information of the payload size, used R_{coding} for the rest of the frame, and the presence of CRC. The actual data are carried by the payload section, and the amount of bits available for the data to be transmitted is determined by the SF and R_{coding} [26]. Table I represents the relation of data bits (DB) and ECB at different SF and R_{coding}. As depicted from this table, even though the increase in R_{coding} enhances data restoring accuracy at the receiver, it moderates the allocated bits to transmit actual data. This results in transmitting more data, which decreases battery lifetime. The number of symbols that makes up the PHY-payload and PHDR

TABLE I
EFFECT ON DB AT DIFFERENT SF AND R_{CODING}

SF	$R_{\text{coding}} = \frac{4}{5}$		$R_{\text{coding}} = \frac{4}{6}$		$R_{\text{coding}} = \frac{4}{7}$		$R_{\text{coding}} = \frac{4}{8}$	
	DB	**ECB**	**DB**	**ECB**	**DB**	**ECB**	**DB**	**ECB**
7	5.6	1.4	4.6	2.4	4	3	3.5	3.5
8	6.4	1.6	5.3	2.7	4.5	3.5	4	4
9	7.2	1.8	6	3	5.1	3.9	4.5	4.5
10	8	2	6.6	3.4	5.7	4.3	5	5
11	8.8	2.2	7.3	3.7	6.2	4.8	5.5	5.5
12	9.6	2.4	8	4	6.8	5.2	6	6

is given by

$$N_{\text{sym}} = 8 + \max\left(\frac{4(8N_{\text{pl}} - 4F + 28 + 16\text{CRC} - 20H)}{4(F - \text{DE})R_{\text{coding}}}, 0\right) \tag{2}$$

where N_{pl} is the number of actual payload bytes. The value of CRC will be 1 if CRC is enabled and 0, otherwise. H determines the presence of PHDR, which is 0 when the header is enabled and 1 when no header is present. The parameter DE represents the low data rate optimization status of a LoRa transceiver. $\text{DE} = 1$ when this parameter is enabled and $\text{DE} = 0$ for disabling condition. This parameter is mandatory in LoRa when using SFs of 11 and 12 with a bandwidth of 125 kHz or lower. The bit-rate R_{bit} at which a LoRa end-device transmits a signal on the specific bandwidth, SF, and R_{coding} is represented as

$$R_{\text{bit}} = F \times \frac{\frac{4}{(4+R_{\text{coding}})}}{\frac{2^F}{\text{BW}}}. \tag{3}$$

C. Channel Characteristics

Because LoRa is both bandwidth and frequency scalable, it can be used for both narrowband frequency hopping and direct sequence applications [10]. The chirp pulses used for LoRa modulation are relatively broadband, offering high immunity to multipath and fading irrespective of regional environments. Assuming Rayleigh fading channels, let h_i be the small-scale channel fading gain between an end device and the gateway, and $g(l_i)$ determines the path loss attenuation function, where l_i is the distance from the gateway. Following the Friis transmission equation for path loss, $g(l_i)$ can be represented as [27]

$$g(l_i) = \frac{\lambda}{(4\pi l_i)^{\alpha}} \tag{4}$$

where λ represents the carrier wavelength, and $\alpha \geq 2$ is the path loss exponent. The signal transmitted by a LoRa node passing through a chirp modulator can be designed as, $s(t) = \sqrt{\frac{2E_s}{T_s}}\cos[2\pi f_c t \pm \pi(u(\frac{t}{T_s}) - w(\frac{t}{T_s})^2)]$, where E_s is the energy of $s(t)$ in the symbol duration T_s, f_c represents the carrier frequency, and peak-to-peak frequency deviation and sweep width are denoted by u and w. Consider that the Rayleigh fading channel is modeled as a zero mean, independent, and circularly symmetric complex Gaussian random variable with unit variance. Thereby, the received signal at the gateway, $r(t)$ is the sum of the attenuated transmitted signal, interference, and noise [28]. It can be represented as

$$r_1(t) = g(l_1)h_1(t)s_1(t) + \sum_{k=2}^{N} \chi_k^F(t)g(l_k)h_k(t)s_k(t) + \eta(t) \tag{5}$$

where $\chi_k^F(t)$ is an indicator function whether a different node ($k \neq 1$) is transmitting a signal at the same time, frequency, and SF or not. $\eta(t)$ represents additive white Gaussian noise with zero mean and variance, $\sigma_c^2 = -174 + NF + 10\log_{10}(\text{BW})$ dBm, where NF denotes the receiver's noise figure and is fixed depending on hardware implementation.

The circular area covered by the SFs can be represented by distance ranges as

$$l_F = \left(\frac{P_{\max} A(f_c)}{q_F}\right) \tag{6}$$

where $P_{\max}$ is the maximal transmit power and $A(f_c) = (f_c^2 \times 10^{-2.8})^{-1}$ is the deterministic path loss term. The symbol, $q_F = -174 + NF + 10\log_{10}(\text{BW}) + \xi_F$ denotes the receiver's sensitivity of each SF, where ξ_F represents the signal-to-interference-noise ratio.

D. InterPlanetary File System

IPFS is a P2P file-sharing protocol, where each content is addressed by a unique CID unlike traditional location-based addressing. The CID is computed by first encoding the content and then hashing the encoded value with a multihash protocol that includes a number of hash functions. Once a file of any formats (e.g., jpg,. png,. pdf,. docx,. json, etc.) is uploaded to a local node, the node provides a 46-character CID of the file content. Any change made in the file content results in a different CID, which proves the tamper resistance of IPFS data. The uploaded file is distributed to the network in a permanent storage manner. A shared file cannot be deleted if all the nodes that store the file do not remove it from their local storage devices. The commands for uploading, viewing, and downloading an object are as follows:

1) upload object: IPFS add <filepath>;
2) view object: IPFS cat <CID>;
3) download object: IPFS get <CID>.

E. Raft Consensus

Raft is a family of crash fault tolerance of the consensus algorithm. Raft networks often feature an odd number of nodes since having an even number of nodes provides no benefit. Each node in a replicated state machine (server cluster) can be in one of three states: leader, candidate, or follower, as illustrated in Fig. 3. Under typical circumstances, a node can remain in any of the three states listed previously. Any request to the follower node is forwarded to the leader node; only a leader can interact with the client. To become a leader, a candidate seeks for votes. Only the candidate or the leader receives responses from a follower. The Raft algorithm divides time into short terms of arbitrary length to maintain these server statuses. Each term is designated by an ever-increasing number, referred to as term number, as shown in Fig. 4. Every node keeps track of this term number,

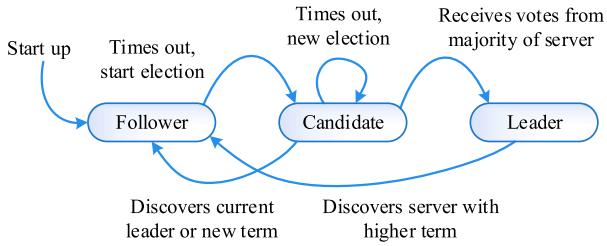

Fig. 3. Server states of Raft consensus.

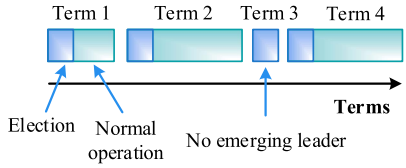

Fig. 4. Time division in Raft consensus.

which is passed between nodes during connections. Every term begins with an election to elect a new leader. To gain a majority, the candidates solicit votes from other server nodes (followers). If the candidate receives the required majority, it becomes the current leader for the remainder of the term. When there is no majority, the scenario is known as a split vote, and the term ends with no leader. As a result, a term can only have one leader. Raft employs randomized election timeouts to ensure that split votes are uncommon and swiftly resolved. Election timeouts are determined at random from a fixed interval (e.g., 150–300 ms) to prevent split votes in the first place. This distributes the servers so that only one will time out in most circumstances; it wins the election and sends heartbeats before any other servers do. Split votes are handled using the same approach. At the start of an election, each candidate resumes their randomized election timeout and waits for that timeout to expire before starting the next election; this decreases the possibility of another split vote in the current election.

Unless there are pending transactions, the Raft consensus does not mint blocks. Because no empty blocks containing zero transactions are minted, this can save a lot of space, especially when the transaction load is low. Another benefit of utilizing Raft over the Istanbul Byzantine fault tolerance (IBFT) is the faster block time. The leader mints a block within 50 ms of receiving the transaction (as is the default configuration), and moving a proposed block through the Raft cluster and collecting majority acknowledgments is a quick operation. Average block times are consistently subseconds under the most common network settings.

While the IBFT consensus algorithm can tolerate an f number of faulty nodes in $3f + 1$ number of total nodes, Raft can tolerate an f number of faulty nodes in $2f + 1$ number of total nodes. A faulty node means the node can propose a false block that contains invalid transactions, which may result in an incorrect blockchain if other nodes are not conscious about the malicious block. If there are f number of dishonest nodes in the network, the network must have $(2f + 1 - f)$ or a majority number of honest nodes to prevent a malicious block from being added to the blockchain. Therefore, Raft is less expensive to run because it requires fewer nodes to be fault tolerant, but it assumes that there are no adversary nodes on the network. IBFT is more expensive to run since it requires more nodes to stay fault tolerant, but it assumes the network has adversarial nodes.

Algorithm 1: Proposed Algorithm For Storing Dynamic Sensor Data In IPFS And Quorum Blockchain.

Input: Sensor data D, secret key Sk, public key Pk, and smart contract Sc

Define data pointer: $L \leftarrow 0$
Define transaction index: $i \leftarrow 0$
Define the number of data per transaction: $N \leftarrow 160$
while True **do**
  Insert D in the LoRa server
  Read server data D_s
  if len$(D_s) - L \geq N$ **then**
    List N new data: $D_n = D_s[L : L + N]$
    Encrypt D_n: $E =$ Sk + map(D_n)
    Create a dictionary d:
    d[“Tx index”] $= i$
    d[“About”] $=$ “LoRa Sensor Data”
    d[“Type”] $=$ “Encrypted”
    d[“Ciphertext”] $= E$
    Write a json file: json.dump$(d,$ “Tx$_i$.json”)
    Upload the file to IPFS: Res $\leftarrow$`IPFS add 'Tx`$_i$`.json'`
    Collect the file CID: CID $=$ Res[“Hash”]
    Define the file URL:
    URL $\leftarrow$ “http://ipfs.io/ipfs/”+ str(CID)
    Compute a timestamp: $t =$ Datetime.now()
    Compute the transaction: $T = [i,$ CID, URL, $t]$
    Sign the transation: Signature $=$ sign(Sk, T)
    Insert Signature and Pk in T:
    $T = T +$ [Signature, Pk]
    Make a transaction to quorum blockchain:
    `Sc.functions.newData(`T`).transact()`
  end if
end while

III. Demonstration of LoRaWAN System

A. Hardware Design

The hardware of LoRa system comprises LoRa transmitting nodes and LoRa gateways. Because LoRa is considered as a low-power and low-cost IoT solution, the peripheral devices used along with the LoRa module should be of low cost and considerably less power consuming [29]. For our demonstration, we have considered three LoRa transmitting nodes and one gateway. The goal is to fully develop end-to-end communication for multiple nodes and network server using one gateway. Each node is connected to a sensor that collects required testing data. The three nodes are designed to collect data from temperature, humidity, and dust sensors, respectively. DHT11 is used for

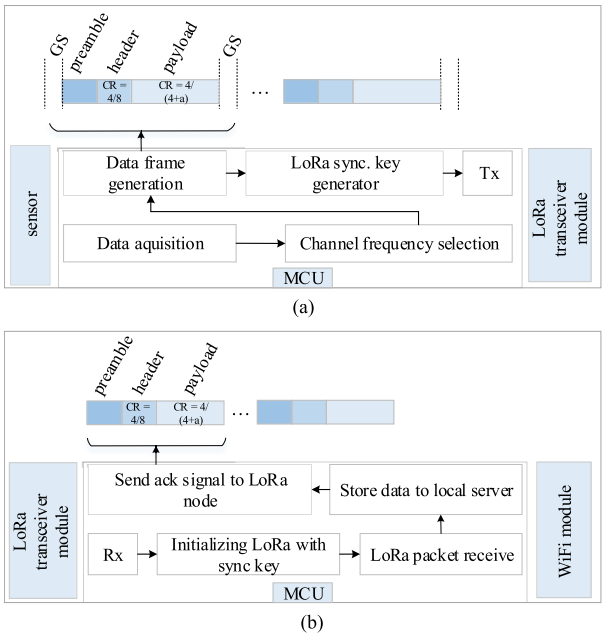

Fig. 5. Design considerations of LoRa: (a) sensor node and (b) gateway.

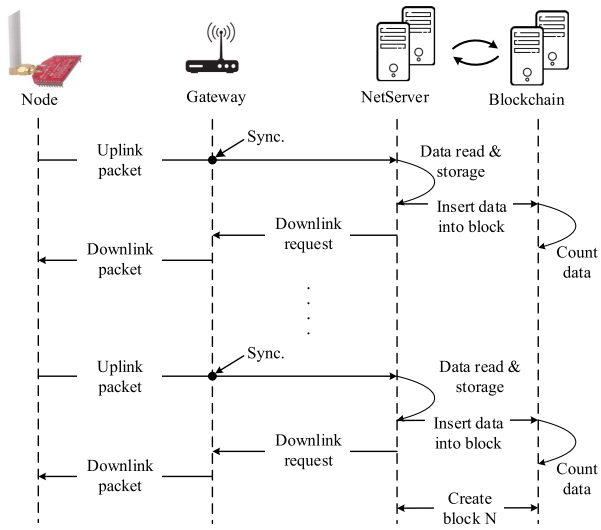

Fig. 6. Illustration of the data workflow through the network.

measuring the temperature and humidity data and PPD42NS is used for collecting dust concentration in the air.

The designed architecture of a LoRa transmitting node is shown in Fig. 5(a). It consists of a sensor unit, microcontroller unit (MCU), and LoRa module. Any kind of industrial sensor module can be used depending on the IIoT sensing requirements. The sensor is connected to the MCU so that the sensed data can be read and processed by the MCU. The next few tasks, such as the assignment of channel frequency, data frame generation, and allotment of the LoRa SyncWord, are performed in the MCU. The channel frequency is selected based on the type of LoRa module and the geographical region. Semtech's LoRa module SX1276/77/78/79 can be programmed as 137 to 1020 MHz transceiver [30]. The components inside the data packets, such as the preamble, PHDR, PHY-payload, R_{coding}, and receive window duration, are also programmed in the MCU. As soon as the sensor data acquisition and data frame construction are completed, the MCU initializes the LoRa module to transmit that packet. The total time regarding those data frame constructions is defined as transmitting ToA. This depends on the SFs while performing PHY-payload modulations. In our demonstration, we used well-known, low-power, 32 kB ATmega328 MCU and SX1278 as the LoRa transceiver module. This transceiver provides 168 dB maximum link budget and high sensitivity down to -148 dBm with the support of different modulations, such as ON–OFF keying, LoRa, frequency-shift keying, and minimum-shift keying [30].

The designed structure of a LoRa gateway is almost similar to the LoRa sensor node, as shown in Fig. 5(b). A similar LoRa transceiver module is used to receive the transmitted signal from any known LoRa sensor node. The MCU that is deliberate to process the received signal, initially matches the SyncWord to make sure an authorized communication. Thereafter, the target sensing data are decoded from the received LoRa packet. A WiFi module is installed to make a connection to the network server using internet protocol and store the sensing data. If necessary, a downlink LoRa message can be sent as an acknowledgment, after a successful data appending to the network server. In that case, a deliberate receive window will be created immediately after the uplink message has been transmitted from the LoRa sensor node to receive the downlink LoRa message from the network server, as shown in Fig. 5(a).

B. *Security Concerns in LoRa Connectivity*

According to the LoRaWAN specification [10], LoRaWAN security adopts the AES cryptographic primitive to provide double-layer (i.e., network and application layers) security by distributing two personalized session keys. First, a unique 128-bit network session key (called NwkSKey) is shared between the end devices and the network server to prove the packets' integrity and verify the sensor nodes' authenticity. Second, a unique 128-bit application session key (called AppSKey) is distributed to the application server for the encryption and decryption of sensor data [31]. Therefore, the network layer security scheme is responsible for ensuring sensor nodes' authenticity and packets' integrity in the network, and the application layer security scheme is responsible for keeping confidentiality by protecting the application server from third-party threads and attacks. Those LoRaWANs' inherent security needs to be accompanied by secure implementation and secure deployment of these schemes to maintain the protocol's built-in security mechanisms. In our demonstration, the network level security is provided by assigning dynamic LoRa SyncWord along with the actual data. Generally, the preamble of the LoRa PHY frame determines the start of the frame. We have used the preamble to store the SyncWord symbol, which is randomly generated and allocated to each node to ensure the authentication before receiving any transmitted data from the LoRa nodes. The assignment of the SyncWord is done by the associated MCU in the transmitting node. As depicted in Fig. 6, when the uplink LoRa packet reaches the LoRa gateway, only the authenticated gateway will be able

to track the signal and reroute it to the network server. This authentication process is performed by matching the SyncWord that will only be known to the network provider. The application level security is provided by implementing blockchain in the LoRa server. The following section gives a detailed description of the proposed blockchain-based server security system.

C. Blockchain Network Layer

As shown in Fig. 1, the network layer includes IPFS for encrypted sensor data storage, and quorum blockchain is used for storing the metadata CID of the encrypted content via a smart contract. Raft consensus is performed connecting seven nodes to manage the consortium blockchain. We choose quorum blockchain for the network layer because it supports consortium blockchains, providing several promising features, such as a number of consensus algorithms and ability to reconfigure access control for ensuring smart contract security and transaction privacy. Although the blockchain itself can ensure data integrity, the main purpose of using IPFS in our system is to reduce size of the blockchain as well as network overhead.

Algorithm 1 demonstrates how dynamic sensor data are processed in IPFS and quorum blockchain to ensure data confidentiality, integrity, and consistency. First, dynamic sensor data D are stored in the LoRa server. If the number of new data stored in the server is greater than or equal to a certain number N defined by the system, the new data are listed and encrypted using the server's private key Sk. The encrypted content and its metadata are arranged in dictionary format and written in a json file. Then, the json file is uploaded and pinned to a local node. IPFS provides a CID of the file, which is a unique address by which the uploaded file can be accessed through the IPFS network. Upon collecting the CID, a transaction T is formed, including the transaction index i, CID, URL of the uploaded file, and a timestamp t. T is signed with Sk and the signature is inserted into T. Finally, the transaction is recorded on the quorum blockchain using the following smart contract (simplified version):

```
pragma solidity >=0.4.16 <0.8.4;
contract LoRaWAN{
  struct SensorData{
    uint256 tx_index;
    string cid;
    string url;
    string timestamp;
    bytes signature;
    bytes public_key;}
  mapping(string => SensorData) Data;
  string[] public SensorDataList;
  function newData(
  uint256  _tx_index,
  string memory _cid,
  string memory _url,
  string memory _timestamp,
  bytes memory _signature,
  bytes memory _public_key)
  public{
    Data[_cid].cid = _cid;
    Data[_cid].url = _url;
    Data[_cid].timestamp = _timestamp;
    Data[_cid].signature = _signature;
    Data[_cid].public_key = _public_key;}
  SensorDataList.push(_cid);
}
```

TABLE II
LoRa Design Parameters

Parameters	Values
Carrier frequency and bandwidth	433 MHz, 125 kHz
Spreading factor	9
Packet size	7 – 11 bytes
Coding rate	4/5
Transmit power	14 dBm
Modulation method	LoRa (based on CSS)
PHY header	Enabled
LoRa module	SX1278

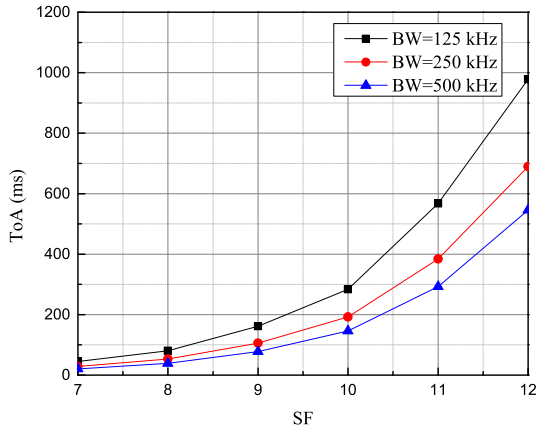

Fig. 7. ToA dependency on bandwidth and spreading factor.

IV. Experimental Results

A prototype hardware is developed and implemented to evaluate the performance of the proposed blockchain-integrated LoRa network. The system is deployed in the university campus to test the complete LoRa connectivity, systems' received signal strength indicator (RSSI), LoRa PLR over distances considering building blockage, and the performance of blockchain in the network server. The three LoRa transmitting nodes are installed at three different places on the campus; node 1 is mounted at the rooftop, and nodes 2 and 3 are located inside our department building. Monopole helical antennas of gain 10 dBi and an operating frequency range from 400 to 470 MHz are used at both the transmitting nodes and the gateway, which propagates signal in an omnidirectional manner. A frequency band of 433 MHz is set for both the transmitter and the gateway to communicate between them. The gateway forward the received data to the network server using WiFi. A local personal computer is used as MySQL data server and served as the network server.

The parameters and devices considered for network development are summarized in Table II. Channel bandwidth of 125 kHz and SF of 9 are taken for the LoRa transmitting nodes to transmit packets. In Fig. 7, dwell time or the ToA has been represented for different SFs. The figure also shows the change of ToA with three different bandwidths, such as 125, 250, and 500 kHz, which depicts that a longer time will be needed if the signal is transmitted at higher SF and lower bandwidth. However, higher

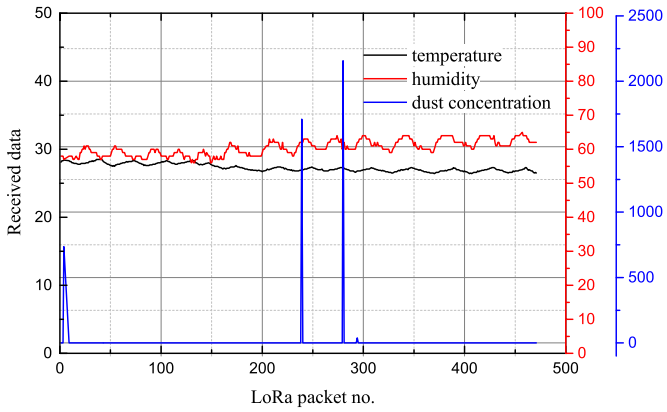

Fig. 8. Received data from the LoRa sensor nodes.

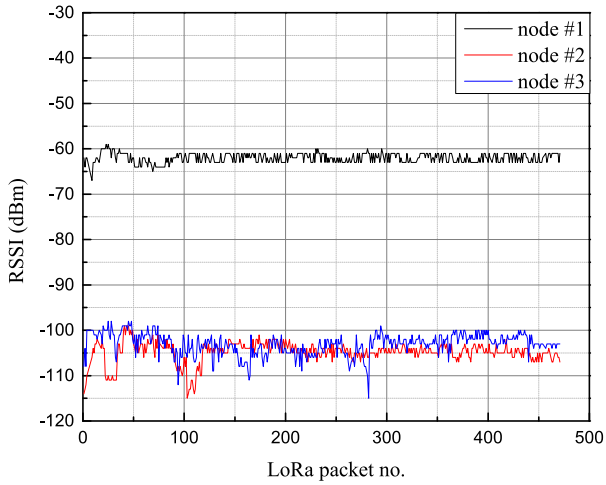

Fig. 9. RSSI status of the LoRa sensor nodes.

SFs are used in such applications where higher coverage is given greater priority than the transmission delay.

The received data from the three LoRa sensor nodes (i.e., temperature, humidity, and dust concentration) are shown in Fig. 8. Following the duty cycle limitations, a slotted ALOHA scheme [32] has been adopted to increase the permissible nodes for a particular channel. Owing to the SFs being orthogonal to each other, the number of supported end devices can be increased by applying the time-slotted channel hopping technique. This is calculated as $n_{\text{nodes}} = n_{\text{channels}} \times n_{\text{slots}}$, where n_{nodes} is the maximum number of nodes per minute, n_{channels} is the number of allocated channels, and n_{slots} represents the number of slots created per minute. The LoRa gateway is installed at a location from where the sensor nodes are kept at three different distances. From the GW, the node 1 is located at a distance of 100 m at the rooftop, whereas the nodes 2 and 3 are located at a distance of 300 and 400 m, respectively, with a greater path loss than node 1. Because the nodes 2 and 3 are located inside the building, this greater path loss is not only for the longer communication distance but also for attenuation of concrete walls. Fig. 9 illustrates the RSSI from the three LoRa transmitting nodes. The LoRa

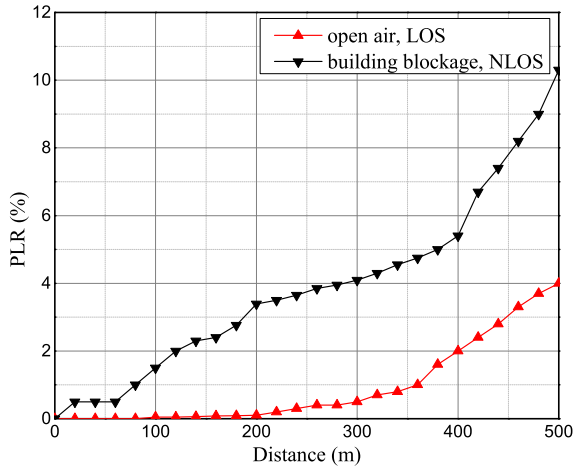

Fig. 10. LoRa PLR at different distances considering LOS and NLOS communication links.

packets from the node 1 has higher RSSI values (on average of −62 dBm) than the nodes 2 and 3. The figure also shows that the difference in RSSI values for the nodes 2 (on average of −103 dBm) and 3 (on average of −106 dBm) is very low. This is because, after a certain distance is reached, the RSSI decreases slowly with the increase in communication distance [33].

The LoRa connectivity has been tested at various communication distances to observe the rate of successful packet reception. Because this manuscript focuses on developing a private LoRaWAN for IIoT, a combination of multistory buildings and open spaces has been considered as the test system, where multiple attenuations have occurred due to walls and other obstacles. The PLR is represented in Fig. 10 with the change of communication distance, considering both the open-air LOS link and the NLOS link with multiple building blockages. Our system was tested at a maximum communication distance of 500 m, considering the IIoT scenario. The figure shows that within a circular coverage area of 0.125 km^2, the LOS LoRa connectivity can provide a successful packet reception rate of nearly 100%. However, in the case of attenuation due to multiple building blockages (breaking the LOS link), the PLR increases up to 4% in a similar coverage area. Practically, the industrial building area is not that large, and the amount of PLR experienced is quite acceptable on the way to deploying LoRaWAN for IIoT.

The proposed IPFS and blockchain-integrated data management scheme are implemented in Python, and its performance is evaluated by performing experiment on an AMD Ryzen 7 4700 U processor (speed: 2 GHz, RAM: 16 GB). Table III demonstrates the latency analysis of Algorithm 1 and the memory consumption of transactions. The encryption time per data list is 26.6 ms, which can be reduced by employing a better-configured processor. The time required to upload each object (json file) to IPFS is 49.3 ms, where the size of each file is 34 kB. The average consensus time for the seven nodes is 135.2 ms. The total sum of all the latency is 205.8 ms, which is the average

TABLE III
PERFORMANCE EVALUATION OF BLOCKCHAIN NETWORK LAYER

Parameter	Value
Encryption time per data list	26.6 ms
IPFS storage time per object	49.3 ms
Size per IPFS object	34 kB
Number of consensus nodes	7
Average consensus time	135.2 ms
Average transaction time (total latency)	205.8 ms
Transaction size	1.70 kB
Number of transactions per block	1
Block size	3.07 kB

time required to conduct a single transaction. The sizes of each transaction and block are 1.70 and 3.07 kB, respectively.

We utilized the quorum blockchain that offers higher performance and privacy. It also supports alternative consensus mechanisms from which we choose the Raft algorithm as it offers faster transactions, which is a crying need of industrial sectors. As IPFS is a free storage system, there are no concerns about memory consumption and storage capacity in the system. The size of sensor data does not affect the performance of the blockchain network because only metadata (i.e., IPFS CIDs) are stored in the blockchain, whose size is constant and content independent. However, a massive amount of data will increase the time required to perform encryption on the server and upload the encrypted data to IPFS. To solve this issue, a high-performance processor can be employed in the server's computer or high-speed hardware technology, such as field-programmable gate array can be adopted to boost the encryption speed.

V. CONCLUSION

A demonstration of a secure LoRaWAN solution for IIoT applications has been presented in this article. The indispensable properties of LoRa PHY and MAC parameters and channel properties were analyzed. A LoRaWAN test system was developed, and the requirements for hardware design and development were provided. In our demonstration, supporting the device/network-level security, a randomized SyncWord authentication technique was developed. Furthermore, blockchain for industrial LoRaWAN solution was proposed and implemented, corroborating the application-level server security. The performance of the developed system was evaluated by the use of several experiments in an indoor and outdoor environments. The PLR was recorded at about 5% at 400 m communication distance, considering NLOS link. The rate can vary depending on the attenuation coefficient throughout the path. The proposed IPFS and blockchain-based data management scheme was tested by performing transactions on the quorum blockchain. It was found that the proposed scheme is capable to store encrypted dynamic sensor data in IPFS and metadata (IPFS CIDs) in the blockchain automatically through a Python program on the LoRa server integrated with the smart contract provided.

Moreover, a number of research issues can be addressed in future, such as evaluating the performance of the LoRaWAN for large number of nodes, verifying the blockchain network for increased nodes, and further minimization of the block transaction delay and storage overhead.

REFERENCES

[1] C. M. da Costa and P. Baltus, "Design methodology for industrial Internet-of-Things wireless systems," *IEEE Sens. J.*, vol. 21, no. 4, pp. 5529–5542, Feb. 2021.

[2] Y. Liao, E. de Freitas Rocha Loures, and F. Deschamps, "Industrial Internet of Things: A systematic literature review and insights," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4515–4525, Dec. 2018.

[3] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the Internet of Things and industry 4.0," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 17–27, Mar. 2017.

[4] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, opportunities, and directions," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.

[5] B. Reynders, Q. Wang, P. Tuset-Peiro, X. Vilajosana, and S. Pollin, "Improving reliability and scalability of LoRaWANs through lightweight scheduling," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1830–1842, Jun. 2018.

[6] Modulation & data rate, The Things Network. Accessed: Nov. 2021. [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/modulation-data-rate/

[7] J. Navarro-Ortiz, S. Sendra, P. Ameigeiras, and J. M. Lopez-Soler, "Integration of LoRaWAN and 4G/5G for the industrial Internet of Things," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 60–67, Feb. 2018.

[8] M. Rizzi, P. Ferrari, A. Flammini, and E. Sisinni, "Evaluation of the IoT LoRaWAN solution for distributed measurement applications," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 12, pp. 3340–3349, Dec. 2017.

[9] M. J. Faber *et al.*, "A theoretical and experimental evaluation on the performance of LoRa technology," *IEEE Sens. J.*, vol. 20, no. 16, pp. 9480–9489, Aug. 2020.

[10] LoRaWAN™ 1.1 specification, Fremont, CA 94538, USA, Oct. 2017. [Online]. Available: https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf

[11] Y. Zhang, C. Xu, X. Liang, H. Li, Y. Mu, and X. Zhang, "Efficient public verification of data integrity for cloud storage systems from indistinguishability obfuscation," *IEEE Trans. Inf. Forensic Secur.*, vol. 12, no. 3, pp. 676–688, Mar. 2017.

[12] J. Ni, Y. Yu, Y. Mu, and Q. Xia, "On the security of an efficient dynamic auditing protocol in cloud storage," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 10, pp. 2760–2761, Oct. 2014.

[13] 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.

[14] T. M. Fernández-Caramaés and P. Fraga-Lamas, "A review on the use of blockchain for the Internet of Things," *IEEE Access*, vol. 6, pp. 32979–33001, 2018.

[15] R. Li, T. Song, B. Mei, H. Li, X. Cheng, and L. Sun, "Blockchain for large-scale Internet of Things data storage and protection," *IEEE Trans. Serv. Comput.*, vol. 12, no. 5, pp. 762–771, Sep./Oct. 2019.

[16] W. Liang, Y. Fan, K. -C. Li, D. Zhang, and J. -L. Gaudiot, "Secure data storage and recovery in industrial blockchain network environments," *IEEE Trans. Ind. Informat.*, vol. 16, no. 10, pp. 6543–6552, Oct. 2020.

[17] A. Durand, P. Gremaud, and J. Pasquier, "Resilient, crowd-sourced LP-WAN infrastructure using blockchain," in *Proc. 1st Workshop Cryptocurrencies Blockchains Distrib. Syst.*, New York, NY, USA, 2018, pp. 25–29.

[18] A. Durand, P. Gremaud, and J. Pasquier, "Decentralized LPWAN infrastructure using blockchain and digital signatures," *Concurr. Comput. Pract. Exp.*, vol. 32, no. 12, pp. 6543–6552, Jun. 2020.

[19] V. Ribeiro, R. Holanda, A. Ramos, and J. J. P. C. Rodrigues, "Enhancing key management in LoRaWAN with permissioned blockchain," *Sensors*, vol. 20, no. 11, May 2020, Art. no. 3068.

[20] M. Tan, D. Sun, and X. Li, "A secure and efficient blockchain-based key management scheme for LoRaWAN," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2021, pp. 1–7.

[21] I. Froiz-Míguez, P. Fraga-Lamas, J. Varela-Barbeito, and T. M. Fernández-Caramés, "LoRaWAN and blockchain based safety and health monitoring system for industry 4.0 operators," *Proceedings*, vol. 42, no. 1, Nov. 2019, Art. no. 77.

[22] D. Saluja, R. Singh, L. K. Baghel, and S. Kumar, "Scalability analysis of LoRa network for SNR based SF allocation scheme," *IEEE Trans. Ind. Informat.*, vol. 17, 10, pp. 6709–6719, Oct. 2021, doi: 10.1109/TII.2020.3042833.
[23] A. Mahmood, E. Sisinni, L. Guntupalli, R. Rondon, S. A. Hassan, and M. Gidlund, "Scalability analysis of a LoRa network under imperfect orthogonality," *IEEE Trans. Ind. Informat.*, vol. 15, no. 3, pp. 1425–1436, Mar. 2019.
[24] J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Low overhead scheduling of LoRa transmissions for improved scalability," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3097–3109, Apr. 2019.
[25] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of LoRa: Long range & low power networks for the Internet of Things," *Sensors*, vol. 16, no. 9, Sep. 2016, Art. no. 1466.
[26] Forward Error Correction & Coding Rate., Zaandam, Netherlands, Accessed: Dec. 2020. [Online]. Available: https://www.mobilefish.com/download/lora/lora_part14.pdf
[27] O. Georgiou and U. Raza, "Low power wide area network analysis: Can LoRa scale?," *IEEE Wireless Commun. Lett.*, vol. 6, no. 2, pp. 162–165, Apr. 2017.
[28] A. Hoeller, R. D. Souza, O. L. Alcaraz Lopez, H. Alves, M. de Noronha Neto, and G. Brante, "Analysis and performance optimization of LoRa networks with time and antenna diversity," *IEEE Access*, vol. 6, pp. 32820–32829, 2018.
[29] H. H. R. Sherazi, L. A. Grieco, M. A. Imran, and G. Boggia, "Energy-efficient LoRaWAN for industry 4.0 applications," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 891–902, Feb. 2021, doi: 10.1109/TII.2020.3042833.
[30] Wireless RF. Semtech Int'l AG, Beijing, Accessed: May 2022. [Online]. Available: https://www.semtech.com/products/wireless-rf
[31] K. Tsai, F. Leu, L. Hung, and C. Ko, "Secure session key generation method for LoRaWAN servers," *IEEE Access*, vol. 8, pp. 54631–54640, 2020.
[32] Z. Ali, S. Henna, A. Akhunzada, M. Raza, and S. W. Kim, "Performance evaluation of LoRaWAN for green Internet of Things," *IEEE Access*, vol. 7, pp. 164102–164112, 2019.
[33] Q. Zhou, K. Zheng, L. Hou, J. Xing, and R. Xu, "Design and implementation of open LoRa for IoT," *IEEE Access*, vol. 7, pp. 100649–100657, 2019.

Md. Shahjalal (Student Member, IEEE) received the B.Sc. degree in electrical and electronic engineering from the Khulna University of Engineering and Technology, Khulna, Bangladesh, in May 2017, and the M.Sc. degree in electronics engineering in August 2019 from Kookmin University, Seoul, South Korea, where he is currently working toward the Ph.D. degree in electronics engineering.

He has authored or coauthored more than 50 technical papers and patents. His research interests include optical wireless communications, wireless security, nonorthogonal multiple access, Internet of Things, low-power wide-area network, and 6G mobile communications.

Dr. Shahjalal was the recipient of the Academic Excellence during his M.Sc. from Kookmin University.

Md. Mainul Islam (Graduate Student Member, IEEE) received the B.Sc. degree in electrical and electronic engineering from the Khulna University of Engineering and Technology, Khulna, Bangladesh, in April 2018 and the M.Sc. degree in electronics engineering from Kookmin University, Seoul, South Korea, in February 2021.

He is currently a Full-Time Researcher with Intelligent Blockchain Engineering Laboratory, Korea University, Seoul, South Korea. His research interests include blockchain, central bank digital currency, cryptography, and data security.

Mr. Islam was the recipient of the Academic Excellence Award.

Md. Morshed Alam (Graduate Student Member, IEEE) received the B.Sc. degree in electrical and electronic engineering from the Khulna University of Engineering and Technology, Khulna, Bangladesh, in May 2018, and the M.Sc. degree in electronics engineering from Kookmin University, Seoul, South Korea, where he is currently working toward the Ph.D. degree.

He completed exchange program on power system from the University of Porto, Porto, Portugal, on September 2017. He has authored or coauthored more than 25 technical articles and patents. His research interests include energy management system, optimization and control system, smart grid, micro-grid, renewable energy, machine learning and deep learning algorithms, low-power wide-area network, and optical wireless communications.

Mr. Morshed was the recipient of the Academic Excellence Award during his M.Sc. from Kookmin University.

Yeong Min Jang (Member, IEEE) received the B.E. and M.E. degrees in electronics engineering from Kyungpook National University, Daegu, South Korea, in 1985 and 1987, respectively, and the Ph.D. degree in computer science from the University of Massachusetts, Boston, MA, USA, in 1999.

From 1987 to 2000, he was with the Electronics and Telecommunications Research Institute, Daejeon, South Korea. Since 2002, he has been with the School of Electrical Engineering, Kookmin University, Seoul, South Korea, where he was the Director of the Ubiquitous IT Convergence Center from 2005 to 2010. Since 2010, he has been the Director of the LED Convergence Research Center, Kookmin University, and since 2018, he has been the Director of the Internet of Energy Research Center, Kookmin University, Seoul, South Korea, and since 2021, the Director of the AI Mobility Research Institute, Kookmin University. He has coauthored more than 500 technical papers and holds more than 120 patents. His research interests include AI mobility, the Internet of Energy, IoT platform, AI platform, cloud platform, optical wireless communications, optical camera communication, 5G/6G mobile communications, and Internet of Things.

Dr. Jang is a Fellow of the Korean Institute of Communications and Information Sciences (KICS), Seoul, South Korea. From 2006 to 2014, he was an Executive Director with KICS. In 2019, he was the President of KICS. He was the recipient of the Young Scientist Award from the Korean Government from 2003 to 2006 and Irwin Jacobs Award in 2018. From 2007 and 2008, he was also the Founding Chair of the KICS Technical Committee on Communication Networks. he was/has been the Steering Chair of the MultiScreen Service Forum from 2011 to 2019, Society Safety System Forum from 2015 to 2021, and the ESG Convergence Forum, since 2022. He was also the Chairman of the IEEE 802.15 Optical Camera Communications Study Group in 2014 and the IEEE 802.15.7 m Optical Wireless Communications Task Group from 2015 to 2019, and successfully published IEEE 802.15.7-2018 and ISO 22738:2020 standard. Since 2020, he has been the Chairman of IEEE 802.15.7a Higher Rate and Longer Range OCC TG. He has organized several conferences and workshops, such as the International Conference on Ubiquitous and Future Networks from 2009 to 2017,International Conference on ICT Convergence from 2010 to 2016, International Workshop on Optical Wireless LED Communication Networks from 2013 to 2016, International Conference on Information Networking in 2015, and International Conference on Artificial Intelligence in Information and Communication since 2019. He is also the Editor-in-Chief of ICT Express (indexed by SCIE).