Secure Caching Scheme by Using Blockchain
for Information-Centric Network-Based Wireless Sensor Networks

Shintaro Mori

Department of Electronics Engineering and Computer Science, Faculty of Engineering, Fukuoka University
8-19-1 Nanakuma, Jonan-ku, Fukuoka 814-0180, Japan
E-mail: smori@fukuoka-u.ac.jp

Abstract This paper addresses a secure caching scheme for information-centric network (ICN)-based wireless sensor networks (WSNs). In order to achieve the above mechanism, we utilize both the public-key cryptography technique and the blockchain technique, which enable data to be safely gathered and the decentralized and cross-verified sensing data to be copied and stored. In addition, we propose a protocol design for introducing the proposed structure into an ICN-based WSN system. Furthermore, we formulate statistical models and demonstrate numerical results by performing computer simulations and hardware-based experiments.

Keywords: wireless sensor network, information-centric network, caching scheme, blockchain, public-key cryptography

1. Introduction

New wireless sensor networks (WSNs) and mobile ad hoc networks have received considerable attention for use in future wireless and mobile communication systems. Advanced WSNs change the way we live and work, through smart homes, healthcare, transportation, smart cities, smart agriculture (farming), advanced meteorological observation, and so on, and are expected to lead to a highly heterogeneous environment with a rich variety of devices, technologies, and services. WSN applications lead to a data-driven society, that is, a society in which people are interested in what data they receive rather than where they receive it from. Toward achieving such a perspective, several emerging frameworks have been designed on the basis of information-centric network (ICN) models instead of traditional host-centric (IP-based) network models [1]–[4]. In particular, investigation of the facilitative caching mechanism has made a major contribution to faster retrieval and network traffic reduction in order to gather and provide sensing data efficiently [5]–[7].

In ICN-based WSNs, in-network caching data should be shared as much as possible; however, duplicated data is privacy-sensitive in many cases [8]. In fact, near-future WSN systems are being increasingly frequently constructed and utilized in the daily living space; thus, the dissemination of secured data, such as life-logs and health data might give rise to serious security and privacy concerns. To avoid such issues, several studies [9]–[16] have investigated a new model of malicious attack and security prevention in the ICN scheme; however, there have been no investigations of the caching scheme in ICN-based WSNs and its behavior as far as we know.

In this paper, we focus on a novel secure caching mechanism for ICN-based WSNs and its important preliminary evaluations. Notably, the proposed scheme aims to as follows: an identification mechanism of the sensing data, an accumulation mechanism of the secured sensing data to the shared database, and an acquisition mechanism of the duplicate sensing data anywhere and anytime for users. To achieve these requirements, the proposed scheme adopts two key technologies: the public-key cryptography technique and the blockchain technique [17], [18]. In particular, the former is utilized to verify the source node identification and the integrity of caching data, whereas the latter is utilized to protect the shared database (i.e., distributed ledger) of caching data. Therefore, by combining these techniques, the proposed scheme has a significant advantage that once verified caching data is registered in the blockchain-based ledger, the stored caching data is trustable without reconfirming the public-key-cryptography-based certification when users try to acquire the sensing data.

The reason why the blockchain technique should be introduced into the proposed WSN environment is that the
blockchain mechanism can efficiently work in a decentralized environment without the need for a central authority among non-trusted intermediaries. On the other hand, the blockchain mechanism is not suitable for resource-constrained WSN devices since the mining task in the verification process requires burdensome computer calculations. To overcome this issue, in the related works [14]–[16], the mining task is performed on cloud servers; however, we are deeply concerned about ensuring the safety of the blockchain-based ledger against the well-known 51% attack—the blockchain can be illegally rewritten when a malicious miner group occupies over 50% or the larger due to the open and free mining environment. On the other hand, the proposed scheme can construct and provide an unopened (closed) mining network on a WSN system by using the overlay concept to solve the above problems of the necessity of computer resources and the 51% attack.

Consequently, our contributions are as follows:

• We propose a secure data collection scheme by using the public-key cryptography technique.
• We propose a decentralized sharing and cross-verification scheme by using the blockchain technique.
• We formulate a statistical model of the proposed mechanism.
• We evaluate and demonstrate the above contributions by performing computer simulations and hardware-based experiments.

The remainder of this paper is organized as follows. Section 2 discusses the related works. Section 3 describes the proposed scheme. Section 4 describes the statistical model of the proposed scheme. Section 5 reports and discusses the result of the computer simulations and hardware-based experiments, a summary and conclusion are given in Sec. 6.

2. Related Works

A major theme in current research on the caching mechanism of ICN systems [6], [7] is to achieve efficient content delivery while providing the closest and highest frequency available caching data to users. For example, Sourlas et al. [5] proposed an autonomic cache management architecture, in which a distributed manager makes the (re)placement decisions depending on the popularity and locality of content data while minimizing the total network traffic and response time. On the other hand, several researchers have investigated a new scheme for security-aware ICN systems that is a prototyping scratch and/or a promising blueprint [9]–[13]. Xie et al. [9] proposed a locality-disruption pollution attack to make a named data networking (NDN) system that is called the cache shield method, which is a traditional and familiar ICN model and more robust by increasing the attack cost. Conti et al. [10] proposed an optimization mechanism for the cache shield technique [9]. Li et al. [11] proposed and evaluated a lightweight mechanism for integrity verification and access control against poisoning attacks on an NDN system. Mauri et al. [12] and Guo et al. [13] proposed and analyzed a unified model to profile the attacker’s strategy for polluting an NDN framework.

On the other hand, there are proposals of WSN systems secured with the blockchain technique [14]–[16]. Zhang et al. [14] and Dorri et al. [15] proposed a secure home network system in order to share secure health data by using pervasive social network nodes, such as smartphones, tablet computers, and personal computers. Banerjee et al. [16] surveyed the existing security approaches in the Internet of Things domain, and categorized and discussed them into two types (reactive and proactive solutions). Moreover, Banerjee et al. [16] described an example in which the blockchain technique is applied to protect the bootloader of a sensor node (SN); when an SN device’s bootloader is managed in a remote environment, if the individual setup and initializing information are rewritten, the WSN system can be controlled by malicious attackers.

3. Proposed Scheme

3.1 Overview of the proposed scheme

To install the blockchain mechanism, most of the computational hardware resources should be assigned to the mining task; miners who participate in a verification task solve an exhaustive hash calculation. Hence, we cannot directly adopt a blockchain mechanism in low (poor)-performance WSN devices. In order to relax this limitation, the design of the proposed scheme is based on two separate layer structures: the WSN plane and the ICN plane, as shown in Fig. 1. There are three kinds of network nodes constituting the proposed scheme; SNs, cluster heads (CHs), and the coordinator. We assume that SNs are resource-constrained and inexpensive devices that CHs have sufficient hardware resources different from the SNs, and that the coordinator is a unique and special CH. By excluding resource-limited SN devices, the proposed scheme enables an ICN-based WSN to be satisfactorily maintained with the blockchain technique.

For a data flow, as shown in Fig. 1, SNs are scattered on the surface of the observation field, the distributed SNs are divided into several groups called clusters, and a CH is
allocated for every cluster in the WSN plane. An SN measures environmental monitoring data as sensing data, and the sensing data is wirelessly transmitted from the SN to the corresponding (dominant) CH. A CH gathers the sensing data as summarized sensing data, which consists of the individual internal cluster’s sensing data. Thus, CHs work similarly to typical fusion center nodes.

On the other hand, in the ICN plane, CHs and the coordinator provide an ICN mechanism, in which the summarized sensing data is shared as the caching data. Note that ICN mechanism providers can generally reply with not only the original content data but also the duplicated data at an individual cache memory. This is because the ICN principle does not distinguish between the original data and the copied caching data in terms of contents. In addition, the coordinator copies and stores the verified summarized sensing data into the cloud storage—in order to guarantee the compatibility of the legacy system, which will be described in the following paragraph.

For sensing data acquisition, as shown in Fig. 1, the proposed scheme assumes two kinds of users: ICN-available users and legacy users. We define users who can directly access and be available to the proposed ICN-based WSN system as ICN-available users, and we define other users as legacy users. When ICN-available users try to acquire the desired sensing data, they access the closest CH, and then the CH sends a message requesting of the target sensing data. After that, the required sensing data is providing by the CH that has the target data as with the legacy ICN procedure. On the other hand, legacy users cannot directly access CHs; hence, they connect to the cloud servers and acquire the desired sensing data from the cloud storage. The detailed procedure of sensing data acquisition is described in Sec. 3.4.

3.2 Secure sensing data collection scheme based on the public-key cryptography technique

Figure 2 shows the procedure of the proposed caching mechanism, in particular, how to gather and append sensing data. Beforehand, the CH generates a public key and private (secret) key based on a public-key cryptography algorithm, \( \Pi(G, E, D) \), similarly to the Rivest-Shamir-Adleman (RSA) scheme. As the three functions in \( \Pi(\cdot) \), let \( G(\cdot), E(\cdot), \) and \( D(\cdot) \) denote the generation function of public and private keys, the encryption encoding function, and the encryption decoding function, respectively. Let \( L = \{L_1, L_2, \ldots, L_M\} \) denote a set of \( M \) CHs and let \( S_i = \{S_{i1}, S_{i2}, \ldots, S_{iN_i}\} \) denote a set of \( N_i \) SNs in the \( i \) th cluster. For the \( j \) th SN in the \( i \) th cluster of \( S_{ij} (\in S_i; i = 1, 2, \ldots, M) \), the CH in the \( i \) th cluster of \( L_i \) generates the public key, \( \bar{PK}_{ij} \), and the private key, \( \bar{SK}_{ij} \), as follows:

\[
(\bar{PK}_{ij}, \bar{SK}_{ij}) = G(L_i \oplus S_{ij}) \tag{1}
\]

where the operator \( \oplus \) denotes any conjunction formula bit by bit. The private key, \( \bar{SK}_{ij} \), is distributed from \( L_i \) to \( S_{ij} \), and all public keys, \( \bar{PK}_{ij} \), for all SNs in the \( j \) th cluster are held by \( L_i \). In the same manner, for \( L_i \), the coordinator generates the public key, \( \bar{PK}_i \), and the private key, \( \bar{SK}_i \), using Eq. (2). Note that the private key, \( \bar{SK}_i \), is distributed from the coordinator to \( L_i \), and all public keys, \( \bar{PK}_i \), are shared among both the coordinator and all CHs.

Fig.1 Overview of the proposed network model

Fig.2 Procedure of signal processing for collecting, gathering, and sharing the sensing data among SNs, CHs, and the coordinator
\[
(\mathcal{F}_i, \mathcal{S}_i) = G(L_i) 
\] (2)

As shown in Fig. 2, in the stage ‘\(\bigcirc\) Gathering sensing data as summarized sensing data,’ \(S_{i,j}\) produces the sensing data, \(D_{i,j}\), that encapsulates the environmental monitoring data. At the SN device, the sensing data is encryption-encoded by using the SN’s private key, and the signed sensing data is sent to the CH, which is given by

\[
(D_{i,j} | \mathcal{S}_i) = E(D_{i,j}, \mathcal{S}_i) 
\] (3)

where the operator \((\cdot,\cdot)\) means signed data, which is composed of the sensing data and a signature based on the public-key cryptography mechanism. Note that the signed data is encoded by using \(E(\cdot)\), and \(E(\cdot)\) has two arguments: the plaintext data and the private key.

At the CH device, the signed sensing data received from the SN is encryption-decoded by using the SN’s public key as follows:

\[
\hat{D}_{i,j} = D(\, D_{i,j} | \mathcal{S}_i) 
\] (4)

Here \(\hat{D}_{i,j}\) denotes the sensing data recovered from \(S_{i,j}\). \(D(\cdot)\) has two arguments: the signed data (ciphertext) and the public key. If the received sensing data can be successfully decoded, the CH concludes that the received sensing data is generated by \(S_{i,j}\) and is not altered through wireless transmission. This guarantees the authenticity and integrity of the sensing data based on the principle of the public-key cryptography technique as with an electronic signature mechanism. In addition, the CH device accumulates the sensing data into the cache memory, and the stored sensing data is assembled as summarized sensing data. In the \(i\)th cluster, \(L_i\) compiles and generates the summarized sensing data, \(\mathcal{D}_i\), expressed in the form

\[
\mathcal{D}_i = \{ \hat{D}_{i,v_{si}}, \hat{D}_{i,v_{si}}^2, \ldots, \hat{D}_{i,v_{si}}^K \} 
\] (5)

where \(K\) denotes the number of sensing data per summarized sensing data.

In the stage ‘\(\bigcirc\) Spreading summarized sensing data,’ similar to the case of sensing data at the SN device, the summarized sensing data is encryption-encoded by using the CH’s private key as follows:

\[
(\mathcal{D}_i | \mathcal{S}_i) = E(\, \mathcal{D}_i, \mathcal{S}_i) 
\] (6)

The signed summarized sensing data is broadcast among the CHs and the coordinator, then the received summarized sensing data is encryption-decoded using Eq. (7).

\[
\mathcal{D}_i = D(\, \mathcal{D}_i | \mathcal{S}_i) 
\] (7)

The correctly recovered summarized sensing data is stored into both the coordinator’s and the CH’s cache memories as caching data for providing the ICN mechanism.

Note that the public-key encryption technique is utilized to provide unfabricated and identified sensing data between SNs and CHs; that is, the unencoded summarized sensing data is appended to the blockchain-based ledger. Therefore, users need not verify the caching data by using the private key. On the other hand, malicious attackers can tamper with the stored caching data in this situation. To solve this problem, the proposed scheme maintains the stability of the caching data’s shared ledger by using the blockchain technique, which is described in the next section.

### 3.3 Decentralized sharing and cross-verification scheme based on the blockchain technique

Figure 3 shows the structure of the blockchain-based ledger in the proposed scheme. As shown in Fig. 3, the elemental block in the blockchain-based ledger has four parts: the hash value of the previous block, the summarized sensing data, the nonce token, and the hash value of the current block. To validate the summarized sensing data, the miner CH participates in the mining task. Therefore, in the stage ‘\(\bigcirc\) Sending request of mining task,’ as shown in Fig. 2, the coordinator selects any summarized sensing data as the next certification candidate block, then it sends a mining request to all CHs.

In the mining task, in the case of verification for the \(n\)th candidate block, all CHs should calculate a hash value of the \(n\)th block, \(h_n \mid x\), using Eq. (8) as soon as possible,

\[
h_n \mid x = H(h_{n-1} \oplus \mathcal{D}_n \oplus l_n) 
\] (8)

where \(H(\cdot)\) denotes the hash function, \(\mathcal{D}_n\) denotes the
summarized sensing data in the $n$th block, and $I_n$ denotes the temporal random value, called nonce token, of the $n$th block, respectively. Note that hash values, such as $h_{n|\kappa}$ and $h_{n-1|\kappa}$ in Eq. (8), must satisfy the condition that the bit sequence is followed by a long string of zeros with $\kappa$ bits from the most significant bit (MSB). In other words, the force and difficulty of the mining task can be adaptively and flexibly changed via the parameter $\kappa$, i.e., the proposed scheme can be applied to various scenarios.

For the blockchain-based certification strength, miners find a suitable pair of the nonce token, $I_n$, and the hash value, $h_n$, by changing $I_n$ by brute force and by round-robin competition. Depending on the characteristics of the computer’s operation and manipulation, it may be difficult to calculate this pair by using the inverse hash function, $\mathcal{H}^{-1}(\cdot)$, where

$$I_n \neq \mathcal{H}^{-1}(h_{n-1|\kappa} \oplus \mathbb{W}_n \oplus h_{n|\kappa}) \tag{9}$$

When the mining task has been finished, the winner miner CH broadcasts the completion message of the mining task with the calculated pair of the nonce token and the hash value for the current verified block to other losing miner CHs and the coordinator in the stage ‘\textcircled{4} Reporting end of mining task,’ as shown in Fig. 2. After receiving the above completion message, the losing miner CHs forcefully finish the mining task, then all CHs and the coordinator append the verified block in the blockchain-based ledger on the cache memories (and the cloud servers). Note that in Fig. 2, the four stages are performed by event-driven operation in parallel.

### 3.4 Procedure by which users acquire the sensing data

In this section, we describe how users acquire sensing data from the blockchain-based ledger. As mentioned above, the proposed scheme assumes two kinds of users: ICN-available users and legacy users. To guarantee backward compatibility, legacy users can obtain the desired sensing data from the blockchain-based ledger stored in the cloud servers. As with common frameworks, we construct a query mechanism so that an application server replies to the legacy user’s and coordinator’s requests with a result retrieved from the blockchain-based ledger. Therefore, the application server works as the gateway and provides an HTTP request and response interface.

For ICN-available users, as shown in Fig. 4, the proposed scheme executes request, inquiry, and response transactions to acquire sensing data as with a typical ICN-based system. In the stage ‘\textcircled{1} Request of sensing data to CH’s broker,’ as shown in Fig. 4, an ICN-available user asks about required sensing data based on search tag information for the broker of the closest CH, then the broker searches for the target sensing data in the blockchain ledger of its cache memory in the stage ‘\textcircled{2} Inquiry to blockchain.’ If the target sensing data is found, the CH’s broker replies with the requested sensing data in the stage ‘\textcircled{3} Response to users;’ otherwise, the CH’s broker sends another request message to the coordinator’s broker in the stage ‘\textcircled{4} Request of sensing data to coordinator’s broker.’

When the coordinator’s broker receives the above message, it makes an inquiry to the application server in the cloud servers, as mentioned above, in the stage ‘\textcircled{5} Inquiry to blockchain’ in order to search for the requested sensing data in the blockchain-based ledger in the cloud storage. Then, the coordinator’s broker replies with the response to the ICN-available users via the intermediate CH’s broker in the stage ‘\textcircled{6} Response to users.’

### 3.5 Protocol design and system architecture

Figure 5 shows the system architecture of the proposed scheme. As the system components, as shown in Fig. 5, the proposed scheme consists of three hardware devices: an SN device, a CH device, and the coordinator device, and data and signal flows are classified into three logical blocks: the part for sensing data collection, the part for the mining task, and the part for the sensing data inquiry. In the rest of this section, we describe the design of the above mechanism in detail.

Regarding the signal procedure for sensing data collection, in the SN device, environmental monitoring data, such as temperature, humidity, atmospheric air
pressure, or atmospheric gas density, is measured. The observed values are stored in the SN’s buffer and the buffered data are packetized as sensing data. In the manner described in Sec. 3.2, the sensing data are encryption-encoded by using a distributed SN’s secret key, then the encoded sensing data are wirelessly transmitted to the CH device through the RF transmission module. In the CH device, for the data flow from SN devices, the received signal is encryption-decoded by using the SN’s public key, and the correctly recovered data is stored in the CH’s buffer. When the CH’s buffer accumulates over $K$ units of sensing data, the $K$ stored sensing data are encapsulated as summarized sensing data. The summarized sensing data is encryption-encoded by using the CH’s secret key, and the encoded summarized sensing data are broadcast to other CH devices including the coordinator device. The verified block is appended to the blockchain-based ledger in the CH’s cache memory of the blockchain and is also stored in the cloud servers through the coordinator’s interface with the Internet. Note that the capacity of the cache memory in the CH device and the coordinator device is not infinite; thus, in the proposed scheme, the cached data should be deleted in order of age using the first-in-first-out algorithm. However, the proposed scheme can remove the above limitation by using another logical line, which is constructed on the basis of cloud storage and the broker.

Regarding the procedure for the sensing data inquiry, in the described in Sec. 2.4, the CH’s broker searches for the required sensing data in the CH’s cache memory of the blockchain to satisfy the request of sensing data from an ICN-available user. If the cache memory of the blockchain of the CH device does not have the requested sensing data, the CH’s broker makes an inquiry based on another request message to the coordinator’s broker. In this situation, if the CH’s broker receives the request message from the other CH’s broker, the request messages are forwarded by the ad hoc networking method. On the other hand, the coordinator’s broker accesses the blockchain ledger on the cloud server via the interface with the Internet in the coordinator device, and then the obtained requested message is delivered via the CH’s broker to the ICN-available user.
4. Statistical Model

In this section, we describe the statistical model for collecting and delivering sensing data from/to the blockchain-based ledger. This section provides the models of caching data generation and verification and formulates the benchmark workable conditions and the response time for caching data acquisition.

4.1 Model of caching data generation

Let $\lambda$ denote the number of caching data generated per unit time, which has a Poisson distribution with its probability density function (PDF) given by

$$p(x; \lambda) = \lambda^x e^{-\lambda} / x!$$

where $\lambda$ denotes the mean value of $\lambda$. Let $\overline{N} (= \sum_{i=1}^{M} N_i / M)$ denote the average number of SNs per cluster and $v$ denote the number of sensing data produced per SN per unit time; hence, $\overline{\lambda}$ can be calculated as follows

$$\overline{\lambda} = vN/K$$

As defined in the previous section, $M$ denotes the number of clusters and $K$ denotes the number of sensing data per summarized sensing data. As shown by Eq. (11), the parameter, $\overline{\lambda}$, also depends on the block size. In general, in the blockchain technique, the block size can be unconditionally decided, and is typically adjusted not only with consideration of the amount of inter-content data and the authentication time but also a reward for the winning miner. In fact, in the Bitcoin system [17], the block size is set as up to 1 Mbyte.

4.2 Model of caching data verification

Let $\mu$ denote the number of candidate blocks approved per unit time, which has a Poisson distribution with its PDF is given by

$$p(x; \mu) = \mu^x e^{-\mu} / x!$$

where $\mu$ denotes the mean value of $\mu$.

For an approved block, generally no one trusts a newly verified block until several approvals blocks have been appended in the blockchain, even if the miner CHs have verified the block. This is because only the longest blockchain is trusted; in other words, the shorter branched blockchains are ignored in the typical blockchain rule. Notably, we assume that the users accept the $n$th block when $\varepsilon$ blocks are appended after the $n$th block; hence, the user will accept the $n$th block when the $(n + \varepsilon)$th block has been approved. In this case, we can calculate $\overline{\mu}$ as

$$\overline{\mu} = 1/eT_{\text{min}}$$

where $T_{\text{min}}$ denotes the average time of the mining task per block. Note that $T_{\text{min}}$ is decided with consideration of the capability of the mining CH device and the parameter $\kappa$ in Eq. (8) used to adjust the difficulty of the mining task.

In practical usage, the users can highly trust the $n$th block by setting a large value for $\varepsilon$, i.e., polluted blocks are difficult to replace by malicious attackers, whereas the approval time for users to accept blocks increases owing to the large $\varepsilon$. In fact, for Bitcoin [17], $\varepsilon$ is set as 6 and $T_{\text{min}}$ is set as approximately 10 min.

4.3 Benchmark formulation of workable conditions

In this section, we formulate the available throughput, certification delay, and response time as the benchmark conditions for the proposed scheme to effectively work. If $\rho$ denotes the average availability throughput, the proposed scheme can always work under the following.

$$\rho = \overline{\lambda} / \overline{\mu} \leq 1$$

On the other hand, regarding the certification delay, we can calculate the average latency from the arrival of a candidate block to the appending of the verified block into the blockchain ledger, $\delta$, based on Eq. (15). Note that in Eq. (15), to avoid complex analysis, we ignore the processing time of the control signal transmissions among CHs and the coordinator.

$$\delta = 1/(\overline{\mu} - \overline{\lambda})$$

In addition, regarding the response time, the proposed scheme cannot deliver the old sensing data from the blockchain ledger in the CH’s cache memory. Therefore, letting $\eta$ denote the probability that the requested sensing data is stored in the CH’s cache memory, we calculate the average response time for the sensing data acquisition, $\tau$, as follows:

$$\tau = \eta \cdot \overline{T_U} + (1 - \eta) \cdot (\overline{T_U} + \overline{T_F} + \overline{T_I})$$

where $\overline{T_U}$, $\overline{T_F}$, and $\overline{T_I}$ denote the average round-trip times between the ICN-available user and the CH, between the CH and the coordinator, and between the coordinator and
the cloud server, respectively. In the next section, we demonstrate the effectiveness of the proposed scheme based on τ as the benchmark.

5. Numerical Results

In this section, we provide numerical results for the statistical model discussed in Sec. 4. In particular, we illustrate the generation of caching data (Sec. 4.1) and the benchmark response time (Sec. 4.3) by performing computer simulations in low-power wide-area (LPWA) network [19], [20]-based WSNs. In addition, we demonstrate the verification of caching data (Sec. 4.2) by performing hardware-based experiments.

5.1 Evaluation environments of computer simulation and hardware-based experiment

In the computer simulations, as shown in Fig. 6, we assume that SNs are randomly scattered, CHs are deployed in the lattice (grid) pattern on the surface of a square observation area, and the coordinator is located at the corner of the observation area. Individual SNs generate sensing data at equal time intervals and send the sensing data to CH in accordance with Japanese low-power long-range (LoRa) wide area network regulation [21]; the parameter settings of the media access control (MAC) layer and physical (PHY) layer are shown in Table 1. In addition, to determine the packet reachability, we calculated the received signal-to-noise ratio (SNR) in the manner described in Sec. 5.2.

On the other hand, in the hardware-based experiments, based on our related study [22], we developed and implemented the CH device by using a familiar single-board computer of a Raspberry Pi 3 device [23] with the default Raspbian kernel version 4.9. For the exhaustive hash function in the mining task, we select the message digest 5 (MD5) method [24] and the first-generation secured hash algorithm (SHA-1) method [25]; these algorithms are widely utilized for general purposes. We implemented the calculation software of the mining task by using C++ language on the testbed device. We used the CLX C++ library [26] to calculate the hash function and we compiled and linked an executable binary program by using the GNU GCC compiler version 6.3.0 [27]. To measure the processing time, we used the method function (or member function) of the ‘std::chorno’ class library in the latest C++ standard template library (STL).

5.2 Radio propagation model and calculation of SNR

When we utilize the binary phase shift keying (BPSK) method, the bit error probability, $p_b$, under a Rayleigh fading environment can generally be theoretically calculated [28] as

| Table 1 Simulation parameters |
|--------------------------------|
| Terms                        | Values                  |
| Observation area (square)     | 20 km x 20 km           |
| Packet length                 | Sensing data            |
| Sense data                    | 100 bytes               |
| Summarized sensing data       | 100,000 bytes           |
| SN Node number                | 1,000-100,000           |
| Trans. interval               | 1.200 s (20 min.)       |
| CH Station number             | 25                      |
| Antenna height                | h = 50 m                |
| Antenna gain                  | 3.53 dBi                |
| SN Transmission power         | $P_{TX} = 20$ mW        |
| Antenna gain                  | 0 dBi                   |
| MAC layer Protocol            | pure ALOHA              |
| Channel number                | 15                      |
| Trans. time                   | 4 s                     |
| Nuner of max retrans.         | 3                       |
| Backoff time                  | 30 s                    |
| PHY layer Modulation method   | BPSK                    |
| Error control coding          | NA                      |
| Radio frequency               | 920 MHz                 |
| ($f_c = 0.326$ m)             |
| Channel model                 | Rayleigh fading         |
| Radio propagation             | Erceg’s model [29]      |
| Parameters of Erceg’s model   | $\alpha = 3.6, b = 0.005,\ c = 20.0, \ e = -0.59,\ d_0 = 100$ m, $\sigma_{\mu} = 8.2, \sigma_{\nu} = 1.6$ |
| Required packet error probability | 1% ($p_b = 10^{-5}$)  |
where $\gamma$ denotes the SNR, and the relationship between $\gamma$ and the received radio power, $P_{RX}$, is given by

$$\gamma = \frac{P_{RX}}{k_B T_0}$$  \hspace{1cm} (18)$$

where $k_B \approx 4.0 \times 10^{-21}$ W/Hz denotes Boltzmann’s constant and $T_0 = 1,600$ K denotes the system device’s absolute temperature. Letting $\ell$ denote the packet length, the packet error probability, $p_e$, is given as

$$p_e = 1 - \left(1 - p_b\right)\ell^t$$  \hspace{1cm} (19)$$

Consequently, using Eqs. (17)–(19), we can calculate the required signal power for correct decoding as shown in Fig. 7, for which the simulation parameters were obtained from our computer simulation results.

On the other hand, at the receiver side, $P_{RX}$ in Eq. (18) is calculated by the link budget formula, which is given by

$$P_{RX} = P_{TX} - L_{TX} + G_{TX} - L_P + G_{RX} - L_{RX} \text{ (dB)}$$  \hspace{1cm} (20)$$

where $P_{TX}$ denotes the transmission radio power, $L_{TX}$ and $L_{RX}$ denote the power losses due to the physical circuit and impedance mismatching, and $G_{TX}$ and $G_{RX}$ denote the antenna gains at the transmitter side and receiver side, respectively. In addition, $L_P$ in Eq. (20) denotes the radio propagation loss and can be generally represented as

$$L_P(d) = \alpha + 10 \cdot \beta \cdot \log_{10}(d) + \sigma,$$  \hspace{1cm} (21)$$

where $d$ denotes the distance between the transmitter and receiver sides and $\sigma$ denotes the shadowing variation.

In this paper, when we use the model of Erceg et al. [29], $\sigma$ has the normal distribution of $\mathcal{N}(\mu_\sigma, \sigma_\sigma^2)$, and $\alpha$ and $\beta$ can be calculated as

$$\begin{cases}
\alpha = 20 \log_{10}(4\pi d_0/\lambda_c) \\
\beta = (a - bh + c/h) + \epsilon z
\end{cases}$$  \hspace{1cm} (22)$$

where $h$ denotes the antenna height at the receiver side, $\lambda_c$ denotes the carrier radio wavelength, and the constants, $a, b, c, d_0$, and $\epsilon$ are decided on the basis of the surrounding environment. $z$ is a random variable with the normal distribution of $\mathcal{N}(0,1)$. The above parameter settings are shown in Table 1.

### 5.3 Simulation results for caching data generation

Figure 8 shows the average number of summarized sensing data generated per hour, $\overline{\lambda}$, versus the number of SNs where the numerical results are given in Sec. 4.1. When the number of SNs is less than 75,000, $\overline{\lambda}$ monotonically increases with the number of SNs. This is because $\nu, M,$ and $K$ in Eq. (11) are constant values, meaning that $\overline{\lambda}$ is dominated by the number of SNs per cluster, $N$. On the other hand, when the number of SNs is greater than 75,000, $\overline{\lambda}$ decreases. In the computer simulation, in the MAC layer, the pure ALOHA scheme is utilized as a simple multiple access method. In this scheme, data is sent if a node has data to send, collisions occur when new data is released while any node is transmitting, and both of their data (i.e., current and new transmission data) are lost. Therefore, $\overline{\lambda}$ reaches a maximum of 88.4 when the number of SNs is 75,000, above which $\overline{\lambda}$ decreases because of packet collision in the MAC layer. The probability of
MAC collision versus the number of SNs is also shown in Fig. 8.

5.4 Experimental result for verification procedure

Figure 9 shows the average processing time versus the length of the bit sequence followed by a long string of zeros from the MSB in Eq. (8) when the block length is set as 100 kbytes and the nonce token length is set as 32 bits as with Bitcoin [17]. In addition, the average processing time is calculated from the measurement values of 100 trials using the hardware-based testbed. As a result, the average processing time increases with \( \kappa \). This is because the computational effort becomes large when the hash value’s condition becomes strict as \( \kappa \) increases. On the other hand, there is no significant difference between the MD5 and SHA-1 algorithms. In addition, the curves of the average processing time increase as a step function rather than monotonically.

In Sec. 5.3, for the most serious situation, we find that \( \lambda \) takes a maximum of 88.4; that is, the required \( \mu \) is also calculated to be 88.4 for the case of in Eq. (14). Hence, from Eq. (13), the average required processing time is 40.7 s per block. As shown in Fig. 9, for the above scenario, the parameter \( \kappa \) should be 14 for the MD5 algorithm and 12 for the SHA-1 algorithm.

5.5 Computer simulation results for benchmark of response time

To illustrate the effectiveness of the proposed scheme, we evaluate the response time from requesting the sensing data to providing the target data, which is formulated in Sec. 4.3. In the computer simulation, we ignore the parameters \( T_D \) and \( T_I \) in Eq. (16) due to their constant values. Figure 10 shows the probability that the requested sensing data is stored in the CH’s cache memory, \( \eta \), using the definition in Eq. (16) when the caching data are wirelessly transmitted under the ideal scenario, i.e., no packet error loss and the maximum data transfer rate in the LoRa WAN specifications (Gaussian frequency shift keying method (GFSK), 250 kbit/s). The numerical results are presented as average values over 10,000 trials.

It is found that when the proposed scheme works most efficiently (i.e., \( \eta = 1 \)), the average response time becomes zero since the response packet can be provided in the CH’s cache memory. On the other hand, when the proposed scheme is not an available scenario (i.e., \( \eta = 0 \)), i.e., in the traditional environment, the average response time becomes 176 s. Note that not only the mining task but also the ICN mechanism is provided in the cloud server in the related studies [14], [15]; therefore, in the conventional scheme, we can assume that there are no ICN-available users, in other words, all users are legacy users. In a practical environment, the caching data are not necessarily available anywhere and anytime in the CH’s cache memory. However, the proposed scheme can improve the average response time by 36.1%, 75.1%, and 95.9% at \( \eta = 0.2, 0.5, \) and 0.7 in comparison with the worst-case condition (without using the proposed mechanism), respectively.

5.6 Discussion

In this section, we showed that the proposed scheme could ensure a secure caching scheme for ICN-based WSNs in terms of secure data handling and decentralized management. The proposed scheme achieves the former requirement by using the public-key cryptography technique, by which the proposed scheme can guarantee
that the (summarized) sensing data are generated by the appropriate legal nodes and that no one can tamper with the collected sensing data. On the other hand, in order to realize the latter requirement, the proposed scheme utilizes the distributed ledger framework by using the blockchain technique, by which any nodes can be appended to any secured sensing data for a shared database based on cross-verification. In addition, in the proposed scheme, the users can obtain the target sensing data by accessing the blockchain ledger, even if they do not have the public key. Therefore, the proposed scheme can provide a suitable design for ICN-based WSNs, in which any user can utilize any available copied secured data from any location. Note that it is necessary to provide sufficient resources for verification in practical implementation. To resolve this issue, the proposed scheme can use a two-tier structure and separate the functionalities between the WSN and ICN.

On the other hand, the proposed scheme must satisfy two technical conditions in the blockchain technique to avoid certification delay and the 51% attack. Regarding the certification delay, there should be a sufficiently long delay before accepting and appending a new block into the blockchain ledger. As a result, the CHs and the coordinator cannot deliver the latest sensing data from the blockchain ledger. The detailed condition and discussion are illustrated by using the statistical model in Sec. 4. Regarding the 51% attack, if we adopt the blockchain technique, the blockchain-based ledger can be illegally operated when the group of the malicious miner CHs occupies over 50% of the ledger. On the other hand, due to the managed blockchain scheme, the proposed scheme has no effect on the 51% attack problem.

6. Conclusions

In this paper, we focused on a secure caching mechanism for ICN-based WSNs. In particular, we proposed a secure data collection scheme by using the public-key cryptography technique and we proposed a decentralized sharing and cross-verification scheme by using the blockchain technique. Moreover, we constructed a protocol design and we proposed a procedure for signal processing in order to realize the proposed mechanism. Furthermore, we formulated as in abstract statistical models and demonstrated numerical results by performing computer simulations and hardware-based experiments. In the future, we should adopt the proposed mechanism to the practical use and optimize the individual environment.

Acknowledgment

This work was partly supported by the Telecommunications Advancement Foundation (TAF).

References

[1] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher and B. Ohlman: A survey of information-centric networking, IEEE Commun. Mag., Vol.50, No.7, pp.26-36, 2012.

[2] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros and G. C. Polyzos: A survey of information-centric networking research, IEEE Commun. Surv. Tutorials, Vol.16, No.2, pp.1024-1049, 2014.

[3] C. Liang, F. R. Yu and X. Zhang: Information-centric network function virtualization over 5G mobile wireless networks, IEEE Network, Vol.29, No.3, pp.68-74, 2015.

[4] M. Amadeo, C. Campolo, J. Quevedo, D. Corujo, A. Molinaro, A. Iera, R. L. Agüiar and A. V. Vasilakos: Information-centric networking for the Internet of things: Challenges and opportunities, IEEE Network, Vol.30, No.2, pp.92-100, 2016.

[5] V. Sourlas, L. Gkatzikis, P. Flegkas and L. Tassiulas: Distributed cache management in information-centric networks, IEEE Trans. Network and Serv. Manag., Vol.10, No.3, pp.286-299, 2013.

[6] X. Wang, M. Chen, T. Taleb, A. Ksentini and V. C. M. Leung: Cache in the air: Exploiting content caching and delivery techniques for 5G systems, IEEE Commun. Mag., Vol.52, No.2, pp.131-139, 2014.

[7] M. Zhang, H. Luo and H. Zhang: A survey of caching mechanisms in information-centric networking, IEEE Commun. Surv. Tutorials, Vol.17, No.3, pp.1473-1499, 2015.

[8] E. G. A.-Allah, H. S. Hassanein and M. Zulkernine: A survey of security attacks in information-centric networking, IEEE Commun. Surv. Tutorials, Vol.17, No.3, pp.1441-1454, 2015.

[9] M. Xie, I. Wijdaja and H. Wang: Enhancing cache robustness for content-centric networking, Proc. IEEE Int. Conf. Computer Commun. (INFOCOM’12), pp.2426-2434, Mar. 2012.

[10] M. Conti, P. Gasti and M. Teoli: A lightweight mechanism for detection of cache pollution attacks in named data networking, Computer Networks, Vol.57, No.16, pp.3178-3191, 2013.

[11] Q. Li, X. Zhang, Q. Zheng, R. Sandhu and X. Fu: LIVE: Lightweight integrity verification and content access control for named data networking, IEEE Trans. Info. Forensics and Security, Vol.10, No.2, pp.308-320, 2015.

[12] G. Mauri, R. Raspadori, M. Gerlay and G. Verticale: Exploiting information centric networking to build an attacker-controlled content delivery network, Proc. 14th IFIP Annual Mediterranean Ad-Hoc Networking Workshop (MED-HOC-NET’15), pp.1-6, June 2015.
[13] H. Guo, X. Wang, K. Chang and Y. Tian: Exploiting path diversity for thwarting pollution attacks in named data networking, IEEE Trans. Info. Forensics and Security, Vol.11, No.9, pp.2077-2090, 2016.
[14] J. Zhang, N. Xue and X. Huang: A secure system for pervasive social network-based healthcare, IEEE Access, Vol.4, pp.9239-9250, 2016.
[15] A. Dorri, S. S. Kanhere and R. Jurdak: Towards an optimized blockchain for IoT, IEEE/ACM Second Int. Conf. Internet of Things Design and Implementation (IoTDI’17), pp.173-178, Apr. 2017.
[16] M. Banerjee, J. Lee and K. R. Choo: A blockchain future to Internet of Things security: A position paper, J. Digital Commun. and Networks, 2017.
[17] S. Nakamoto: Bitcoin: A peer-to-peer electronic cash system, Tech. Rep., 2008.
[18] K. Christidis and M. Devetsikiotis: Blockchains and smart contracts for the Internet of things, IEEE Access, Vol.4, pp.2292-2303, 2016.
[19] U. Raza, P. Kulkarni and M. Sooriyabandara: Low power wide area networks: An overview, IEEE Commun. Surv. Tutorials, Vol.19, No.2, pp.855-873, 2017.
[20] H. Wang and A. O. Fapojuwo: Survey of enabling technologies of low power and long range machine-to-machine communications, IEEE Commun. Surv. Tutorials, Vol.19, No.4, pp.2621-2639, 2017.
[21] LoRa alliance: https://www.lora-alliance.org/ (retrieved: Feb. 2018).
[22] S. Mori: Fundamental analysis for blockchain-based secured caching scheme for information-centric network-based wireless sensor network, Proc. 2018 RISP Int. Workshop on Nonlinear Circuits, Commun., and Signal Process. (NCSP’18), pp.517-520, Mar. 2018.
[23] Raspberry Pi: https://www.raspberrypi.org/ (retrieved: Feb. 2018).
[24] MD5 algorithm: IETF RFC 1321.
[25] SHA-1 algorithm: IETF RFC 3174.
[26] CLX C++ library: https://github.com/clown/clx (retrieved: Feb. 2018).
[27] GNU GCC compiler: https://gcc.gnu.org/ (retrieved: Feb. 2018).
[28] J. G. Proakis: Digital Communications, McGraw-Hill, 2008.
[29] V. Erceg, L. J. Greenstein, S. Y. Tjandra, S. R. Parkoff, A. Gupta, B. Kulic and A. A. Julius: An empirically based path loss model for wireless channels in suburban environment, IEEE J. Sel. Areas in Commun., Vol.17, No.7, pp.1205-1211, 1999.

Shintaro Mori received his B.S., M.S., and Ph.D. degrees from Kagawa University in 2007, 2009, and 2014, respectively. Since April 2014, he has been with the Department of Electronics Engineering and Computer Science, Faculty of Engineering, Fukuoka University, Japan, where he is currently an assistant professor. His research interests include cross-layer design and wireless sensor networks. He is a member of IEEE, IEICE, ISSJ, and RISP.

(Received October 30, 2017; revised February 8, 2018)