IEICE TRANS. COMMUN., VOL.E105–B, NO.11 NOVEMBER 2022

PAPER Special Section on Towards Management for Future Communications and Services in Conjunction with Main Topics of APNOMS2021

Performance and Security Evaluation of Table-Based Access Control Applied to IoT Data Distribution Method

Masaki YOSHII[†a)], ***Student Member***, **Ryohei BANNO**[††], ***Member***, ***and*** **Osamu MIZUNO**[†], ***Senior Member***

SUMMARY New services can use fog nodes to distribute Internet of Things (IoT) data. To distribute IoT data, we apply the publish/subscribe messaging model to a fog computing system. A service provider assigns a unique identifier, called a *Tag ID*, to a player who owes data. A *Tag ID* matches multiple IDs and resolves the naming rule for data acquisition. However, when users configure their fog node and distribute IoT data to multiple players, the distributed data may contain private information. We propose a table-based access control list (ACL) to manage data transmission permissions to address this issue. It is possible to avoid unnecessary transmission of private data by using a table-based ACL. Furthermore, because there are fewer data transmissions, table-based ACL reduces traffic. Consequently, the overall system's average processing delay time can be reduced. The proposed method's performance was confirmed by simulation results. Table-based ACL, particularly, could reduce processing delay time by approximately 25% under certain conditions. We also concentrated on system security. The proposed method was used, and a qualitative evaluation was performed to demonstrate that security is guaranteed.

key words: *fog computing, IoT data, publish/subscribe, ACL, security*

1. Introduction

Fog computing [1] has been proposed to apply preprocessing techniques, such as machine learning and data cleansing of Internet of Things (IoT) data on fog nodes between IoT devices and cloud servers [2], [3]. Figure 1 depicts an overview of fog computing. Cloud is deployed applications for IoT services. The fog node is part of the cloud processing; in the example of Fig. 1, it is part of the IoT data collection and part of the IoT data analysis process. Fog nodes analyze IoT data without sending it to the cloud, and the results are sent to service users. If the fog node is unable to process the data, the IoT data analyzed up to the halfway point will be sent to the cloud for analysis using the cloud's resources. The cloud sends the analysis results of the IoT data back to the user via the fog node. Furthermore, distributing IoT data through multiple fog nodes may result in new services based on IoT data.

In this paper, a player who provides services using IoT data is defined as "a service provider." A player who contracts with a service provider and issues their own IoT data is defined as "a user." Assume the players who provide the fog node are distinct. In that case, each player's ID naming rules for IoT data acquisition differ, so other players may not specify IoT data acquisition. It may be difficult to obtain data. As a result of resolving the ID for acquiring IoT data, the player who uses the IoT data can specify the data ID for the player who issues the data, allowing the IoT data to be distributed [4]. However, it is possible that a user's published data, including private information, could be delivered to a service provider without the user's knowledge via a fog node. To address this issue, we propose a table-based access control list (ACL) [5]–[7]. The user can prevent unintentional data distribution by specifying "permit/deny" data transmission permissions for each service provider in the table-based ACL.

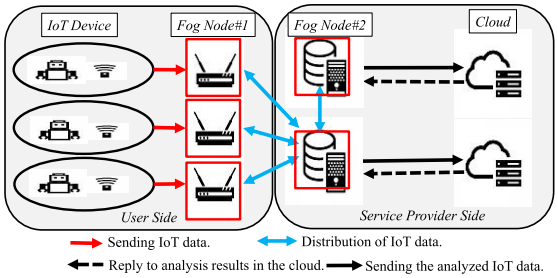


Fig. 1 Overview of fog computing.

In this study, we perform simulations to compare the average processing delay time between nodes before and after applying the proposed method. We also evaluate security qualitatively.

The remainder of this paper is organized as follows. Section 2 discusses related technologies and previous work. Section 3 introduces the research's issues and policies, as well as the proposed table-based ACL operation. Section 4 describes the simulation and its results. Section 5 discusses the qualitative evaluation of the security. Section 6 contains comparisons with related work. Section 7 contains conclusions and recommendations for future research.

2. Related Technologies and Prior Works

2.1 Fog Computing

Fog computing is the next-generation cloud computing technology proposed by Cisco Systems [8]. Fog computing places fog nodes close to a user. A fog node preprocesses IoT data, executes data cleansing, performs some cloud ap-

Manuscript received December 8, 2021.
Manuscript revised February 26, 2022.
Manuscript publicized May 27, 2022.

[†]The authors are with Kogakuin University Graduate School, Tokyo, 163-8677 Japan.
[††]The author is with Kogakuin University, Tokyo, 192-0015 Japan.
a) E-mail: y.masaki@ieee.org

DOI: 10.1587/transcom.2021TMP0007

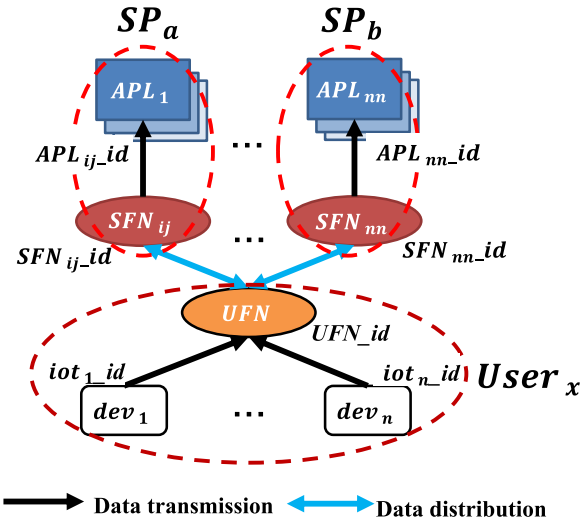


Fig. 2 New service model using fog computing.

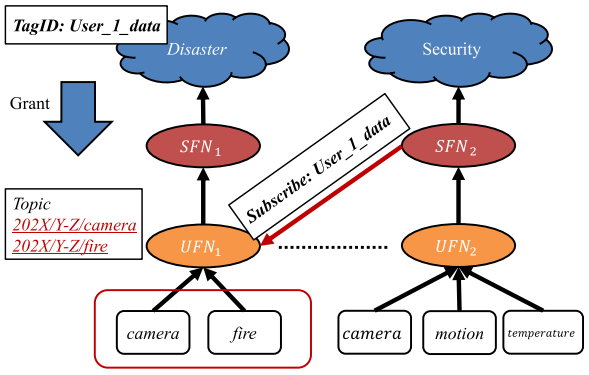


Fig. 3 Tag ID-based pub/sub model.

plication processes, and returns results to users. Consequently, network congestion on the cloud can be reduced. Fog computing is suitable for reducing the amount of traffic and latency.

This research assumes that the fog nodes are widely distributed and used to distribute data among players who have fog nodes.

2.2 Service Model Using Fog Computing

Two types of players are assumed in this study, i.e., service providers (SPs) and users. Figure 2 depicts a service model based on fog computing. An SP, as shown in Fig. 2, can run multiple applications. Each player, fog node, and IoT device (Dev) have IDs that are owned by the player. The SP also has a service fog node (SFN). The user enters into a contract with the SP and makes use of the IoT services. The IoT devices are the user's property. The Dev is linked to the user fog node (UFN) that the user has configured, and the data are sent to the UFN. The UFN sends data to the SFN of the SP with which the user has a contractual relationship.

Suppose SP_a does not have a contractual relationship with $User_x$ and SP_a has a contractual relationship with another SP, i.e., SP_b. SP_b and $User_x$ have a contractual relationship with each other. Then, IDs that point to $User_x$ can share with SP_a, and SP_a can use $User_x$'s data referring to their IDs. Thus, a new player is not needed for user ID resolution. When IDs are shared, the user is obligated to be notified, and the user is notified.

Users use their UFN to collect IoT data and use it for IoT services such as visualization of IoT data using business intelligence tools, anomaly detection of IoT devices using machine learning, smart home applications in home energy management system deployed in the UFN. Companies that use IoT devices, as well as individuals who own IoT devices and build service integration platforms such as IFTTT [9], are examples of users. SPs use the cloud to deploy their services. Users' IoT data with whom they have a contractual relationship be used to develop new IoT services and improve the quality of existing IoT services. Users can benefit from new IoT services as well as high-quality IoT services provided by service providers.

The specific data distribution method is described in Sect. 2.3.

2.3 Tag ID-Based Publish/Subscribe Messaging Model

The Publish/Subscribe messaging model (Pub/Sub model) [10] is used for IoT data distribution. Well-known Pub/Sub model protocols and systems include MQ Telemetry Transport [11] and Apache Kafka [12]. An ID called Topic, which is arbitrarily determined by the user, is used for data acquisition. The Topic can be set for each IoT device, and there are as many Topics as IoT devices.

Consequently, in the model depicted in Fig. 2, when an SP requests IoT data from the UFN, it must subscribe to as many Topics as IoT devices. Furthermore, because the user can arbitrarily determine the Topic ID, if the user changes the Topic ID and does not notify the SP, data acquisition for the changed Topic becomes impossible. Consequently, a new naming convention is required.

We propose the Tag ID-based Pub/Sub model. Figure 3 shows the operation of the proposed model for a disaster mitigation service. UFN_1 user has placed a surveillance camera and a fire alarm (labeled *camera* and *fire*). UFN_2 user has placed surveillance camera, motion sensor (labeled *motion*), and temperature sensor (labeled *temperature*). UFN_1 connected these devices. Topics "*202X/Y-Z/camera*" and "*202X/Y-Z/fire*" are for the surveillance camera and fire alarm, respectively. Next, UFN_1 is assigned the *Tag ID* "User_1_data" because it has a contractual relationship with the SP that operates the disaster mitigation service. The disaster mitigation service analyzes disaster information using fire alarms and camera images. UFN_1 manages and maintains the correspondence between the Topic and *Tag ID*. The security service uses motion sensors, surveillance camera images, and temperature sensors to provide users with detection services for illegal entry. The SP that operates the security service does not have a direct contractual relation-

ship with the user who has UFN_1. However, suppose there is a contractual relationship with the SP that operates the disaster mitigation service. In that case, The *Tag ID* assigned by the disaster mitigation service will be used. It can be shared and subscribed to by UFN_1. Thus, it is possible to solve the ID naming convention for sending Subscribe messages within the range of the contractual relationship.

Name resolution can reduce the number of data requests from the SP. Here, a table is used for ID conversion between Topic and *Tag ID*, and the elements that make up the table are Topic and *Tag ID*. When the number of Topics is N_{Tp}, the number of *Tag ID*s is N_{Tg}, and the number of rows in the table is N_{tb}. The relationship is expressed as follows:

$$N_{\mathrm{t}} = N_{\mathrm{Tp}} \times N_{\mathrm{Tg}} \tag{1}$$

The search time is $O(\log(N_{\mathrm{t}}))$ when binary search algorithm is used. If we use the hash search method, it will be $O(1)$. However, since the search efficiency of hash tables may deteriorate if the buffer size in a node is too small. Since hash tables are highly dependent on the hash algorithm to be implemented, we assume the binary search method for table search as the most popular algorithm in this study.

The search time of the table used to convert between *Tag ID* and Topic is used to calculate the average processing delay.

When viewed from the SP's perspective, the Tag ID-based Pub/Sub model assigns a single Tag ID to multiple Topics, which reduces the number of IDs that the SP must maintain. It is also possible to reduce the number of signals used for IoT data acquisition. From the user's perspective, because the UFN contains a table containing the correspondence between Topic and Tag ID, the user only needs to manipulate the table when changing the Topic of the IoT device, and the SP is not required to be notified of the Topic change.

Qualitative benefits can be expected from both user and service provider perspectives.

2.4 Research Objectives and Problems

The objective of this research is to provide highly convenient services by distributing IoT data among multiple users and multiple service providers. The three problems are as follows:

1) No unanticipated private data leakage:
   Inadvertent distribution of private data may be detrimental to the user.
2) Scalability must be ensured for the proposed IoT data distribution method:
   SPs need to understand how they affect the quality of IoT services, such as processing latency and bandwidth usage; due to the nature of IoT services, scalability is an important indicator of service quality.
3) The method must be able to guarantee the assumed security risks:
   Since IoT data including private data is used in IoT services, risks other than unexpected private data leakage must be addressed.

1) is expected to be solved by the mechanism proposed in Sect. 3. 2) is evaluated by simulation in Sect. 4 to confirm the effectiveness of the proposed method. 3) describes the security risks assumed in Sect. 5 and discusses possible solutions for each.

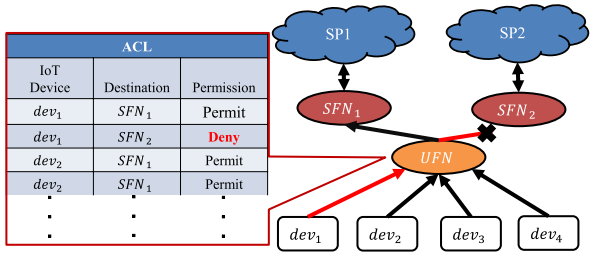
3. Table-Based Access Control List

3.1 Table-Based Access Control List Overview

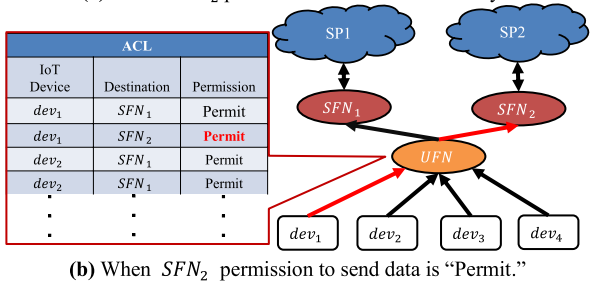
The Tag ID-based Pub/Sub model can resolve ID naming conventions and reduce the number of Subscribes within the scope of contractual relationships. Assume, conversely, that a Tag ID is assigned to the Topic of the IoT device that publishes private data. In this case, when a user subscribes to a topic with a *Tag ID*, private data may be distributed in ways that the user does not intend. To register Topics of IoT devices that publish private data that are inconvenient to distribute, an ACL is required. Consequently, we propose a table-based ACL that works with the Tag ID-based Pub/Sub model.

Next, we describe how the table-based ACL operates. Tables are assumed to compose databases. The elements that make up the table-based ACL are the IoT device, the data destination, and the "permit/deny" of data transmission. If data transmission of dev_1 is not permitted for SFN_2 (Fig. 4(a)), UFN will not transmit the data to SFN_2. If data transmission of dev_1 to SFN_2 is permitted (Fig. 4(b)), the UFN will transmit the data to SFN_2. Here, the number of SFNs is N_{s}, the number of Publishers is N_{p}, and the number of rows in the table is N_{t}. The relational expression is as follows:

$$N_{\mathrm{t}} = N_{\mathrm{s}} \times N_{\mathrm{p}} \tag{2}$$

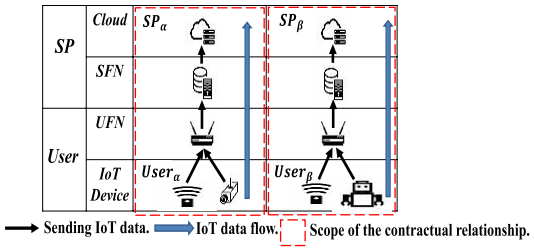


(a) When SFN_2 permission to send data is "deny."

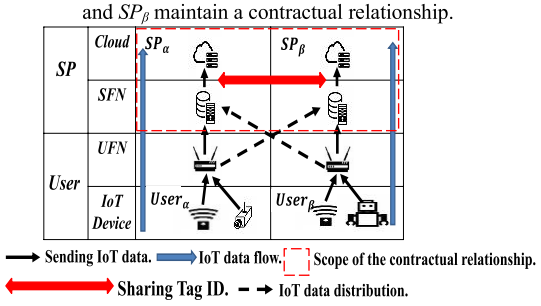


(b) When SFN_2 permission to send data is "Permit."

Fig. 4 Table-based ACL operation overview.



(a) $User_\alpha$ and SP_α maintain a contractual relationship and $User_\beta$ and SP_β maintain a contractual relationship.



(b) SP_α and SP_β maintain a contractual relationship with each other.

Fig. 5 Example of the correspondence between each player.

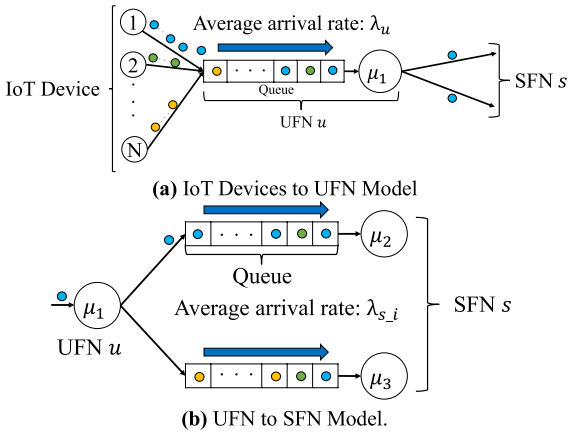
If the binary search method is used in the table search, the search time is $O(\log(N_t))$. If we use the hash search method, it will be O (1). As mentioned in Sect. 2.3, we use the binary search method for table search to keep search efficiency.

3.2 Security Model

Figure 5 shows the example of the correspondence between players. In Fig. 5(a), $User_\alpha$ and SP_α maintain a contractual relationship, and $User_\beta$ and SP_β maintain a contractual relationship.

In this case, the SP that assigned the Tag ID does not share the Tag ID with other SPs. Consequently, acquiring IoT data using the Tag ID of the other SP is not possible. It is only possible to acquire IoT data from users who have a contractual relationship with one another. The user must include "Permit" in the table-based ACL of the UFN among the "Permit/Deny" of IoT data transmission that the user allows to be transmitted among the IoT devices that the user owns. Set "Permit" for nonpermitted IoT data transmissions.

Figure 5(b) shows the case where SP_α and SP_β maintain a contractual relationship with each other. The Tag ID is shared between both SPs in this case, and the user is notified that the Tag ID has been shared. In this case, the user will provide IoT data to SPs that do not have a contractual relationship with each other if it is determined that by distributing IoT data to them, the user will be able to use higher quality services and new services. The user registers Permit in the UFN's table-based ACL among the destinations and "Permit/Deny" of IoT device data that may be provided. SPs can subscribe to users without a direct contractual relationship by using shared Tag IDs. Unauthorized IoT data will not be transmitted.



(a) IoT Devices to UFN Model

(b) UFN to SFN Model.

Fig. 6 Modeling for queuing theory analysis.

3.3 Modeling by Queuing Theory

To evaluate the processing delay time from the IoT device to the UFN and from the UFN to the SFN, we performed simulations with/without table-based ACL. We measure the processing delay time because it is necessary to comprehend the impact of data transmission control on each fog node when using the table-based ACL. Processing delay time and bandwidth usage are critical indicators for understanding how they affect the quality of IoT services. It is a critical quality metric for fog nodes and IoT service users. It is also an evaluation metric for service providers when it comes to service implementation [13].

Before performing the simulation, we analyze the operation of IoT data distribution using the queuing theory. Fig. 6 shows the model from the IoT device to the SFN.

In this study, we assume eight types of IoT devices, including temperature sensors, surveillance cameras, and vibration detectors. These IoT devices publish event-driven data, or they publish IoT data at decided intervals. However, in the service model of this study, IoT data are distributed to service providers without any contractual relationship with users who own devices. In this case, IoT devices, like temperature, humidity, and light intensity sensors, will be set a certain threshold value to send data. Then, IoT data publish when the observed data exceed the threshold. On the other hand, the Research Institute of Industrial Safety reports that data generation from fire alarms, smoke detectors, and vibration detectors, among others, have a Poisson distribution for the interval [14].

In this paper, we assume that the IoT data for IoT devices are generated randomly and we use Poisson distribution for the simulation as the first step.

To analyze the behavior of various IoT service use cases, simulations using the IoT traffic model with uniform and beta distributions [15] and comparing with the $M/M/1$ model are future studies.

Here, the number of IoT devices is N, the number of UFNs is one, and the number of SFNs is two. Publications

of IoT data will follow a Poisson distribution. The arrival rate of the IoT data published from an IoT device to UFN u is λ_u [calls/unit time]. The queue in the UFN stores the called data. The average wait time of the data stored in the queue is denoted W_u. The UFN processes the stored data on a first-come, first-served basis. The relationship between the average processing rates μ_u, λ_u, and W_u is given by the following equation using Little's theorem [16]:

$$\rho = \frac{\lambda_u}{\mu_u} \tag{3}$$

$$W_u = \frac{1}{\mu_u - \lambda_u} - \frac{1}{\mu_u}[\text{unit time}] \tag{4}$$

Here, ρ ($\rho < 1.0$) indicates the load factor of the fog node. The reciprocal of μ_u is the average processing rate of the UFN. Thus, the average processing time T_u generated in the UFN is as follows:

$$T_u = \frac{1}{\mu_u}[\text{unit time}]. \tag{5}$$

Next, consider the case of calling IoT data from the UFN to the SFN. The data follow the Poisson distribution, as with IoT devices. The number of installed SFNs is K. Let μ_{s_i} be the average processing rate of the i-th SFN. If the average arrival rate of IoT data transmitted from UFN u to the i-th SFN s_i is λ_{s_i}, the equation relating to the average processing rate of SFNs is expressed as follows:

$$\rho = \frac{\lambda_{s\_i}}{\mu_{s\_i}} (i = 1, 2, \cdots, K). \tag{6}$$

The queue in the SFN stores the called data. The average queuing delay time W_{s_i} of the i-th SFN can also be expressed by the following equation by Little's theorem, as in Eq. (4).

$$W_{s\_i} = \frac{1}{\mu_{s\_i} - \lambda_{s\_i}} - \frac{1}{\mu_{s\_i}}[\text{unit time}] \tag{7}$$

The average processing time of the i-th SFN T_{s_i} is as follows:

$$T_{s\_i} = \frac{1}{\mu_{s\_i}}[\text{unit time}]. \tag{8}$$

Since each SFN has a queue, the section between UFN and SFN can be represented as an $M/M/1$ model.

4. Simulation Evaluation

4.1 Simulation Overview

We add the Tag ID-based Pub/Sub model operation and the proposed table-based ACL method to the NS-3 network simulator [17]. Table 1 shows the simulation parameters, and Fig. 7 shows the simulation topology. The topology is a star topology. It is assumed that there are 96 IoT devices, one UFN, and two SFNs.

There are 96 Topics, which corresponds to the number of IoT devices. One *Tag ID* is assigned to each of the 12 Topics. Sect. 2.3 describes the Topics and Tag ID.

Table 1 Simulation parameters.

Parameter	Value
Simulator	NS3(ns-3.29)
Model	M/M/1
Average processing time	1, 2, …, 10 [ms]
Average Waiting Time	Exponential Distribution
Average Arrival Interval	Poisson Distribution
Service Discipline	FCFS
Number of Publishers	96
Number of Brokers	1
Number of Subscribers	2
Number of Topics	96
Number of Tag IDs	8
Simulation Topology	Star topology
Number of Trials	30

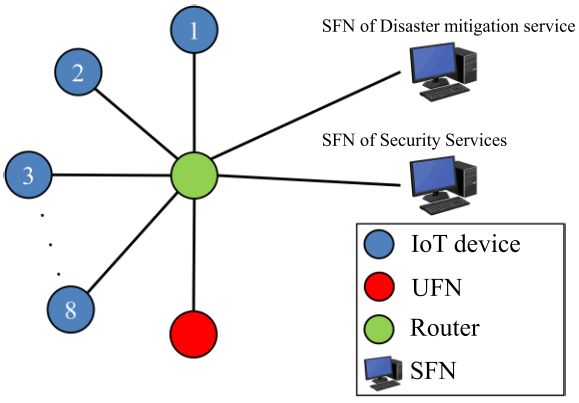


Fig. 7 Simulation topology.

The SP's cloud is excluded from this simulation because the evaluation's goals are the queuing delay bandwidth usage between the IoT device and the SFN, as well as the average processing delay time.

The simulation time is 3600 s, and the number of simulations is 30. The formulas used to evaluate the average processing delay time are as follows:

$$T_{du} = D_d + D_u + D_t \tag{9}$$

$$T_{uf} = D'_u + T_{s\_i} + D'_t \tag{10}$$

Here, T_{du} is the sum of the data transmission delay from the IoT device to the UFN and each node's processing time. D_d is the IoT device's processing time. D_u denotes the UFN's processing time, which is the sum of the conversion process time from Topic to *Tag ID* and T_u in Eq. (5). D_t is the data transmission delay time from the IoT device to the UFN. T_{uf} is the sum of the data transmission delay time from the UFN to the SFN and each node's processing time. D'_u is the processing time in the UFN required to send data to the SFN. T_{s_i}, shown in Eq. (8), is the sum of the processing time, including *Tag ID* to the Topic conversion and the table-based ACL procedure. D'_t is the transmission delay time

Table 2 IoT devices and Tag IDs.

IoT Device	Tag ID
1) Temperature sensor	Temp
2) Humidity sensor	Humid
3) Light intensity sensor	Light
4) Surveillance camera	Image
5) Fire alarm	Fire
6) Smoke detector	Smoke
7) Thermography	Thermography
8) Vibration detector	Vibration

when data are transmitted from the UFN to the SFN.

As table-based ACLs reduce the amount of IoT data received by the SFN, the average bandwidth usage near the SFN is also expected to decrease. The evaluation Eq. (11) is shown below:

$$B_s = \frac{Data_{all}}{T_{sim}} \text{ [Mbps]} \tag{11}$$

B_s is the average bandwidth usage, $Data_{all}$ is the total amount of IoT data received, and T_{sim} is the simulation time.

4.2 Simulation Scenario

The simulation scenario assumes disaster mitigation services and security services and that the user has a contract with both services. It is assumed that the user has eight types of IoT devices and relationship *Tag ID*s, as shown in Table 2. Devices 1)–4) publish data primarily used for disaster mitigation services, and devices 5)–8) publish data mainly used for security services. The SFN possessed by the disaster mitigation service assigns four *Tag ID*s to the UFN.

It is assumed that the SPs have contractual relationships with each other, and the *Tag ID* is shared among the SPs and is known. Image and video data are collected by the surveillance camera. The fire alarm continuously collects fire-related data in the event of a disaster (Thermal and smoke sensing data, etc.). Under normal conditions, i.e., non-disaster conditions, the alarm collects ambient temperature data regularly. During a fire, the smoke detector collects data on the smoke produced by the fire. Under normal circumstances, it collects data on the intensity of ambient light. The thermography camera gathers thermal image data from the environment. The vibration detector collects acceleration data generated by vibration.

At the start of the simulation, the UFN randomly determines the "permit/deny" setting for data transmission in the table-based ACL. Because different users may determine which device publishes private data, the setting is determined at random. The data transmission parameter is "permit" when publishing data from Devices 1)–4) to the SFNs owned by disaster mitigation and security services. When the data from Devices 5)–8) is published to the security service's SFN, the parameter is also "permit." When each SP subscribes to data from a different SP, the parameters are assigned at random.

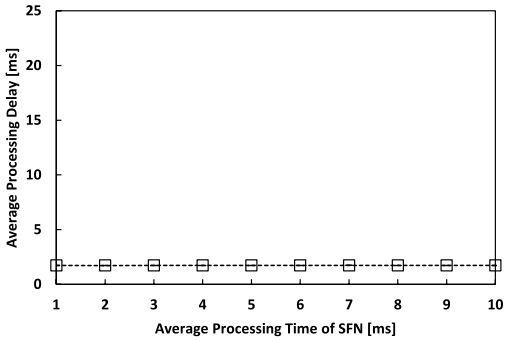


Fig. 8 Average processing delay time from the IoT device to the UFN.

Devices 1), 2), 3), and 8) publish data to the UFN at an average interval of 1 s. Devices 4)–7) publish the data to the UFN at an average interval of 60 s. In disaster service, Devices 5) and 6) publish data at an average interval of 1 s from the simulation elapsed time of 2000 s. The message size of Device 4) is from 10000 to 35000 bytes, and Device 7) is from 15000 bytes to 18000 bytes. Other devices publish data from 100 to 300 bytes.

To increase the service fog node load, the average processing time is increased by 0.001 s from 0.001 to 0.01 s inclusive and perform the simulation. The average processing time of user fog nodes was fixed at 0.001 s. Simulations are performed 30 times at each utilization ratio.

A scatter diagram is used to display the average value and variance of the acquired processing delay time. Noteworthily, the average processing delay time only refers to information that has increased exponentially because it is heavily influenced by the measurement environment and the actual machine.

4.3 Simulation Results and Discussion

Figure 8 shows the average processing delay time from the IoT device to the UFN. Because the use of table-based ACL does not affect the average processing delay between IoT and UFN, the results are shown only after the use of table-based ACL. The average processing delay times are arranged almost horizontally at each average processing time. The UFN is unaffected because the SFN's average processing time is altered.

Figure 9 shows the average bandwidth usage of SFNs with/without table-based ACLs: Table-based ACL reduces the average bandwidth by 23% on SFN1 and 30% on SFN2, respectively. Because table-based ACLs can reduce the number of IoT data received in SFNs.

Figure 10 is the queuing delay time of SFNs. The vertical axis represents the queuing delay, and the horizontal axis represents the configured SFNs' average processing time. Table-based ACLs reduced the queuing delay in SFN1 and SFN2 by approximately 23% and 55%, respectively. In addition, table-based ACL reduces IoT data queuing. This is due to a reduction in the number of IoT data waiting in the

SFN queue. Without table-based ACLs, the queueing delay tends to increase in proportion to the average processing time of SFNs; with table-based ACLs, the queue delay increases gradually.

Figure 11 shows the average processing delay from UFN to SFN. Comparing the results before and after applying table-based ACL, SFN1 reduces the average processing delay by approximately 10%; SFN2 reduces the average processing delay by approximately 25%. Because the queuing delay is added to the Eq. (10) to calculate the average processing delay, the average processing delay is reduced as well. The average processing delay increased exponentially with/without the application of table-based ACL by increasing the average processing time of SFN. When table-based ACLs are used, the average processing time gradually increases. According to the simulation results, table-based ACL can reduce the average queuing delay and processing delay time in SFN, as well as the average bandwidth usage. We did not use network topologies like the BA model or the Waxman model in this paper because our goal is to understand the trend of the pure average processing delay in the proposed method. When the topology is complex, packet transmission delays are expected to be significant. For the same reason, two SFNs are used to compare the two models fundamentally. Furthermore, increasing the number of SFN nodes increases the number of rows in the table-based ACL table, which increases the table search time, which increases the average processing delay.

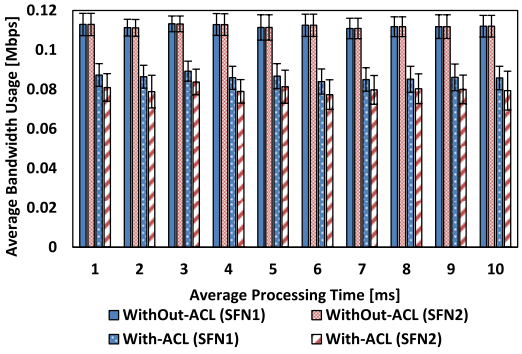


Fig. 9 Average bandwidth usage.

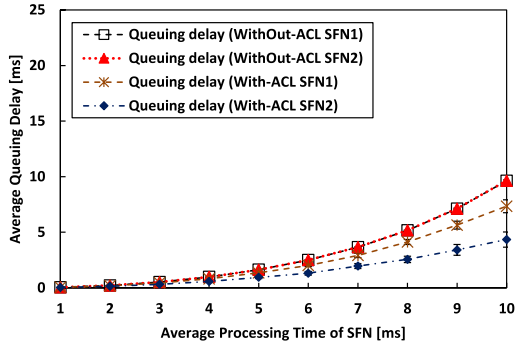


Fig. 10 Average queuing delay time from the SFN.

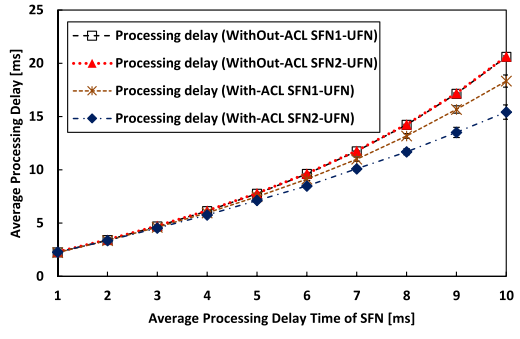


Fig. 11 Average processing delay time from the UFN to the SFN.

5. Qualitative Evaluation of Security

5.1 Possible Security Threats

We analyze the security of the entire system by applying the proposed method. The analysis of security threats considers the case where security measures using table-based ACL are not taken.

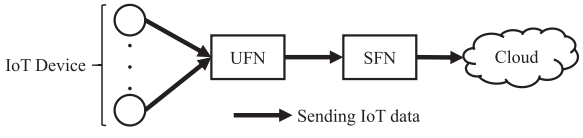
Figure 12 shows the system model for qualitative evaluation. From the user side to the SP, the order is from UFN to SFN, and from SFN to cloud. From the cloud to the user's side, evaluation is performed in the order of SFN from the cloud used by the SP and SFN to UFN.

(i) User side to the Service Provider side.

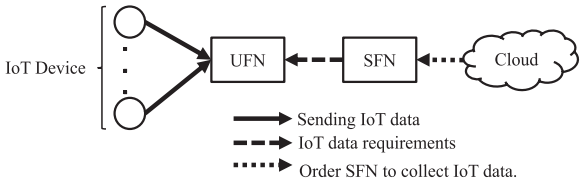
1) UFN to SFN: The UFN sends data to the SFN after the SFN subscribes. If the data are tampered with, the user's quality of service may be degraded.
2) SFN to Cloud: The SFN receives IoT data from multiple UFNs and sends the data to the SP's cloud. As the number of UFNs increases, the number of received packets increases; thus, the load in the SFN increases [18].

(ii) Service Provider side to the user side.

1) Cloud to SFN: The cloud issues a command to the SFN



(a) User's side to Service Provider side.



(b) Service Provider side to User's side.

Fig. 12 System model for qualitative evaluation.

to request IoT data. Eavesdropping of the SFN communication line from the cloud causes a threat [19].
2) SFN to UFN: SFNs subscribe to IoT data from UFNs by *Tag ID*. The *Tag ID* is a threat if leaked to a malicious actor. Additionally, as the number of IoT devices increases, the number of IoT data items sent to UFN and SFN increases. Congestion near an SFN leads to transmission delay.

5.2 Security Risk of IoT Data Distribution Method

This section discusses the qualitative evaluation of the security of the IoT data distribution method. The security considerations for threats mentioned in Sect. 5.1 are as follows:

(i) User side to the cloud.

1) UFN to SFN: The threat of eavesdropping and data tampering between UFN and SFN can be dealt with by applying an IPsec VPN [20], [21] between the UFN and SFN. The load on the SFN can also be reduced because the number of packets sent to the SFN can be reduced by applying the proposed table-based ACL.
2) SFN to the cloud: By distributing the SFNs, the traffic load during IoT data collection can be expected to decrease. Thus, it is also expected to reduce the traffic load near the cloud server.

(ii) Service Provider side to user side.

1) Cloud to SFN: It is possible to deal with the threat of eavesdropping on the communication path from the cloud to SFN with existing technologies, such as VPNs.
2) SFN to UFN: In this research, since IoT data destinations are registered in table-based ACL, data are not transmitted even if a third party subscribes by *Tag ID*.

We consider the security of the players. First, we assume the transmission of personal information from the user to the SP. Because the table-based ACL is applied to the UFN, private data transmission to the SP can be reliably suppressed. Consequently, confidentiality can be guaranteed between contractually bound parties.

For the suppression of data transmission from the user to the SP, suppressed data transmission by table-based ACL can reduce the load on the fog node owned by the SP, which leads to availability.

Next, we consider data transmission from a user-owned IoT device to an SP. Since new IoT data are continually issued and transmitted, SPs can obtain new data. Further integrity can be ensured by performing data cleansing with fog nodes.

6. Related Works

Wardana and Perdana [22] proposed the application of authentication servers and tokens for authentication in IoT systems. There are no security measures against eavesdropping in IoT systems using MQTT.

They also pointed out that the integrity and confidentiality of the data on the subscriber side can be affected. An authentication server that issues tokens and manages and verifies the secure payload in the MQTT broker is their proposal. Furthermore, SSL certificates are used with the MQTT protocol to secure communication. Their work differs from ours in that they proposed a method to secure the MQTT protocol itself. Our research focuses on a new service model based on fog computing, as well as the security of IoT data distribution schemes based on the service model. Furthermore, their work involves an authentication server, which may complicate the service model and make contractual relationship management difficult when assuming actual IoT services. The service model in our work is straightforward because it comprises only two parties: the service provider and the user.

Schmitt et al. [23] discussed the dynamic management of bridges between different MQTT brokers. A bridge is a feature that allows multiple MQTT brokers to share Subscribe messages. They add a new feature to the open-source MQTT broker known as mosquitto [24]: a topic for broker A to make a bridge connection to broker B, and a topic for broker B to remove its bridge connection. From the standpoint of hacker attacks, these two topics would expose the addition or removal of unnecessary bridges. Consequently, we use an ACL file to limit access to only specific users.

In comparison with our study, the file system requires the privileged user to add the ACL configuration file by himself, and it is assumed that the information of the newly added ACL configuration file is written in the program. Because the table of the table-based ACL is assumed to be made up of a database, the user with a UFN can operate the ACL made up of the database. In terms of simplicity, our research is excellent. However, because the ACL configuration file can be created by privileged users, it is considered superior in terms of ACL scalability, such as access to the broker only during a specific period and access control only for limited users. Another point of distinction is that the ACL in [23] restricts access to a small number of users because, from the standpoint of hacker attacks, bridging and deleting unnecessary brokers is undesirable. Our research aims to protect the distribution of IoT data which may be detrimental to the users of IoT devices.

Bhatt et al. [25] proposed an Attribute-Based Access Control (ABAC) model [26] for AWS IoT [27]. Here, the attribute information refers to information such as the IP address of the IoT device and the owner's affiliation and age. They implement policy decision points (PDP) and policy enforcement points (PEP). They assumed an oil refinery and they measured the processing power required to implement it on AWS IoT. They assumed specific evaluation scenarios such as devices in an oil refinery and actions taken by workers in the oil refinery. In this case, information to be registered need to mediate among players.

On the other hand, we aim to apply the proposed method for generic use cases that are connected to various IoT de-

vices, it is necessary to share the attribute information of IoT devices among the players and register them in the access control system. In addition, the attribute information and the access control policies can be tied to players. Users can register information including the destination of the IoT data and the transmission "Permit/Deny" information in the table-based ACL of the UFN, because such information depends on users themselves.

7. Conclusion

Fog computing, which is the next-generation cloud technology for IoT data distribution, may create new services. Consequently, we proposed a Pub/Sub model based on Tag IDs that assigns a single *Tag ID* to multiple topics. Because multiple topics are associated with *Tag ID*s in the Tag ID-based Pub/Sub model, private data can be transmitted at the same time. To address this issue, we proposed a table-based ACL in which "permit/deny" for IoT data transmission is registered in the table.

Simulation results indicated that it is possible to reduce the average processing delay time up to approximately 25% using the proposed method. The average bandwidth usage was also reduced by up to 30%. It was discussed that existing techniques and table-based ACLs can be used to support the security of the proposed method.

The problems described in Sect. 2.4 were solved, and the purpose of this study is expected to be achieved.

In the future, we will consider that in the proposed model, registering "permit/deny" in ACL depends on the user, which they may find annoying. More complex topologies and simulations with more nodes will be future work.

Acknowledgments

JSPS KAKENHI Grant Number JP18K11276 supported part of this research.

References

[1] M. Yannuzzi, R. Irons-Mclean, F. van Lingen, S. Raghav, A., Somaraju, C. Byers, T. Zhang, A. Jain, J. Curado, D. Carrera, O. Trullols, and S. Alonso, "Toward a converged OpenFog and ETSI MANO architecture," Proc. 2017 IEEE Fog World Congress (FWC), pages 6, Nov. 2017. DOI: 10.1109/FWC.2017.8368535

[2] K. Ashton, "That 'Internet of Things' thing," RFID Journal, vol.22, no.7, pp.97–114, June 2009.

[3] D. Wu, D.I. Arkhipov, E. Asmare, Z. Qin, and J.A. McCann, "UbiFlow: Mobility management in urban-scale software defined IoT," Proc. IEEE Conf. on Computer Communications, pp.208–216, April 2015. DOI: 10.1109/INFOCOM.2015.7218384

[4] M. Yoshii, K. Kimura, Y. Amano, and O. Mizuno, "IoT data sharing method applying publish/subscribe type protocol to fog computing," 2020 IEICE General Conference, B-7-6, March 2020 (in Japanese).

[5] M. Yoshii, R. Banno, and O. Mizuno, "Application of access control list in IoT data distribution method using fog computing," 2020 IEICE Communications Society Conference, BS-4-2, Sept. 2020 (in Japanese).

[6] M. Yoshii, R. Banno, and O. Mizuno, "Performance evaluation of table-based access control list applied to IoT data distribution method using fog computing," Proc. IEICE International Conference on Emerging Technologies for Communications (ICETC), pages 4, E1-1, Dec. 2020. DOI: 10.34385/proc.63.E1-1

[7] M. Yoshii, R. Banno, and O. Mizuno, "Evaluation of table-based access control in IoT data distribution method using fog computing," IEICE Commun. Express, vol.10, no.10, pp.822–827, Oct. 2021. DOI: 10.1587/comex.2021XBL0134

[8] CISCO, "Fog Computing," https://www.cisco.com/c/m/ja_jp/solutions/internet-of-things/iot-system-fog-computing.html, (accessed 19 Feb. 2022).

[9] IFTTT, https://ifttt.com/ (accessed 12 Feb. 2022).

[10] P.T. Eugster, P.A. Felber, R. Guerraoui, and A.M. Kermarrec, "The many faces of publish/subscribe," ACM Comput. Surv., vol.35, no.9, pp.114–131, June 2003.

[11] OASIS, "Message queuing telemetry transport (MQTT) TC," https://www.oasis-open.org/committees/mqtt, (accessed 12 Feb. 2022)

[12] J. Kreps, N. Narkhede, and J. Rao, "Kafka: A distributed messaging system for log processing," ACM SIGMOD Workshop on Networking Meets Databases, page 6, 2011.

[13] K. Takahashi, H. Nakazato, and K. Kanai, "End-to-end response time evaluations in centralized and distributed iot data sharing model," IEICE Technical Report, CAS2019-100, Feb. 2020.

[14] S. Hanayasu, "A study on the time intervals between accidents (5)," Research Report of the Research Institute of Industrial Safety, RIIS-RR-89, 1989 (in Japanese).

[15] X. Jian, X. Zeng, Y. Jia, L. Zhang, and Y. He, "Beta/M/1 model for machine type communication," IEEE Commun. Lett., vol.17, no.3, pp.584–587, March 2013. DOI: 10.1109/LCOMM.2013.012213.122637

[16] J.D.C. Little, "A proof for the queueing formula: L=λW," Oper. Res., vol.9, pp.383–387, 1961. DOI: 10.1287/opre.9.3.383

[17] ns-3 Network Simulator, https://www.nsnam.org/, (accessed 13 Feb. 2022).

[18] T. Mori, Y. Utsunomiya, X. Tian, and T. Okuda, "Optimal design of fog-cloud computing system," IEICE Technical Report, IN2017-73, Jan. 2018 (in Japanese).

[19] S. Kashima, M. Ueno, T. Aso, and T. Miyazawa, "Consideration about deriving the security requisites for secure IoT services," IEICE Technical Report, IN2016-20, June 2016 (in Japanese).

[20] O. Mizuno, "Cyber security measures for building systems," J. Institute of Electrical Installation Engineering of Japan, vol.39, no.6, pp.314–317, June 2019 (in Japanese). DOI: 10.14936/ieiej.39.314

[21] B. Gleeson, A. Lin, J. Heinanen, G. Armitage, and A. Malis, "A framework for IP based virtual private networks," IETF RFC 2764, Feb. 2010.

[22] A.A. Wardana and R.S. Perdana, "Access control on internet of things based on publish/subscribe using authentication server and secure protocol," Proc. 10th International Conference on Information Technology and Electrical Engineering (ICITEE 2018), pp.118–123, July 2018. DOI: 10.1109/ICITEED.2018.8534855

[23] A. Schmitt, F. Carlier, and V. Renault, "Data exchange with the MQTT protocol: Dynamic bridge approach," Proc. IEEE VTC2019, Spring, pages 5, April 2019. DOI: 10.1109/VTCSpring.2019.874633

[24] Eclipse Mosquitto, "An open source MQTT broker," https://mosquitto.org/ (accessed 12 Feb. 2022).

[25] S. Bhatt, T.K. Pham, M. Gupta, J. Benson, J. Park, and R. Sandhu, "Attribute-based access control for AWS internet of things and secure industries of the future," IEEE Access, vol.9, pp.107200–107223, July 2021. DOI: 10.1109/ACCESS.2021.3101218

[26] V.C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone, "Guide to attribute based access control (ABAC) definition and considerations," NIST, NIST Special Publication 800-162, Jan. 2014. DOI: 10.6028/NIST.SP.800-162

[27] Amazon Web Service, "What is AWS IoT," https://docs.aws.amazon.com/ja_jp/iot/latest/developerguide/what-is-aws-iot.html (accessed 16 Feb. 2022).

Masaki Yoshii received his B.S. degree in Information Communication Engineering from Kogakuin University, Tokyo, Japan, in 2020. He is currently a master's course student at the Graduate school of Engineering, Kogakuin University. His research interests include IoT data distribution methods. He is a student member of IEEE, IEICE, and DBSJ.

Ryohei Banno received a Bachelor of Engineering and Master of Information Science and Technology degrees from Hokkaido University in 2010 and 2012, respectively, and a Ph.D. degree in Science from the Tokyo Institute of Technology in 2018. From 2012 to 2018, he was a researcher with NTT Network Innovation Laboratories. From 2018 to 2020, he was a researcher with the Tokyo Institute of Technology. Since 2020, he has been an Assistant Professor at Kogakuin University. His research interests include distributed systems and the IoT. He received the Outstanding Paper Award from IPSJ in 2015, the Inoue Research Award for Young Scientist from Inoue Foundation for Science in 2020, and the Funai Research Award from FFIT in 2020. He is a member of IEEE, IEICE, and IPSJ.

Osamu Mizuno received B.S. and M.S. degrees in Applied Electronics Engineering from Tokyo Institute of Technology in 1983 and 1985, respectively. He received a Ph.D. in Global Information Telecommunication from Waseda University, Tokyo in 2008. From 1985–2009, he worked in the research and development of network service systems for the Nippon Telegraph and Telephone Corporation. He joined Kogakuin University, Tokyo, in 2009. His research interests include IoT service systems and ID distribution. He is a member of IEEE, IEICE, IPSJ, and IEEJ.