View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Explore Bristol Research

Li, Y., Liu, W., Zhu, Y., Chen, H., Cheng, H., Chen, T., Hu, P., & Huan, R. (2021). Privacy-Aware Fuzzy Range Query Processing Over Distributed Edge Devices. *IEEE Transactions on Fuzzy Systems*. https://doi.org/10.1109/TFUZZ.2021.3059952

Peer reviewed version

Link to published version (if available):
10.1109/TFUZZ.2021.3059952

Link to publication record in Explore Bristol Research
PDF-document

This is the author accepted manuscript (AAM). The final published version (version of record) is available online via IEEE at https://doi.org/10.1109/TFUZZ.2021.3059952 . Please refer to any applicable terms of use of the publisher.

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available: http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2021.3059952, IEEE Transactions on Fuzzy Systems

Privacy-Aware Fuzzy Range Query Processing Over Distributed Edge Devices

Yinglong Li, Weiru Liu, Yihua Zhu, *Senior Member, IEEE,* Hong Chen, *Senior Member, IEEE,* Hongbing Cheng, Tieming Chen, Ping Hu, and Ruohong Huan

***Abstract*—Range query processing is a common edge computing and service in the Internet of Things, which can extract user-interest information from distributed edge devices. How to design lightweight privacy-preserving range query processing methods remains a challenging task. Existing secure range query approaches suffer from both high communication cost and long response time, which makes them unsuitable for edge computing over resource-constrained edge devices. In this paper, we propose two privacy-aware fuzzy query processing schemes based on fuzzy theory. Linguistic range variables, fuzzy overlap information and its recovery mechanism are introduced respectively. In addition, two distributed privacy-aware fuzzy range query processing algorithms are devised. Our approaches not only serve for privacy protection, but also aim to provide other optimal performances in terms of reliability, energy efficiency, and real-time response. Theoretical analysis and experimental evaluations based on real-world data sets validated our motivation.**

***Index Terms*—Range query, Fuzzy sets, Edge computing, Internet of Things, Privacy protection.**

I. Introduction

In the age of Internet of Things (IoT), services in pervasive edge environments [1]-[3] are expected to offer end-users better Quality-of-Services (QoS) than that in cloud environments. Various kinds of edge devices, including mobile phones, laptops, connected vehicles, smart cameras, and a range of sensor-equipped devices [4] have been deployed in the pervasive edge computing environments. These edge devices have limited communication, computing, and storage capacities. Especially their energy supply is always constrained and hard to recharge or replace, which makes the edge computing extremely challenging.

End-users often ask for edge computing and services [5]-[8] with good QoS, such as less communication cost, quick response time and good privacy protection. However, the resource-constrained edge devices cannot afford to provide edge services that meet users' QoS requirements. For example, a common privacy threat is the sensitive data leakage when an adversary eavesdrops or sniffs the message packets during the data transmission. To deal with such a privacy threat, a conventional way is to make both the readings of edge devices and users' queries encrypted. However, it is a challenge to process encrypted queries over encrypted data without knowing their actual values. Besides, for recovering the usability of sensor data and users' queries, massive data exchange and computation should be performed, leading to huge communication cost and time overhead. Therefore, it is significant to design lightweight methods for edge computing and services as to achieve good QoS under resource constraints.

Range query [9]-[15] is a common edge computing and service in IoT applications. Through range query processing, end-users can extract information from distributed edge devices (e.g., extracting health information from wearable health monitoring devices). Nevertheless, measurements and locations of edge devices, as well as fusion information are prone to be illegally acquired by adversaries, which brings great security risks. Therefore, more and more research efforts have been focused on the privacy protection problems of range queries [11], [15], [23], [25], whereas most of the existing methods still suffer from relatively complex computation and costly communication despite tremendous endeavors had been made. Resource-restricted edge devices are very sensitive to the time and energy overhead during edge computing and services, especially in real-time edge service scenarios, as such, users call for lightweight and service-tailored range query services.

In addition, existing query processing methods for IoT applications always rely on using raw sensing data, which is time-consuming and also causes massive data transmission. However, even a centralized processing manner cannot ensure accurate results due to imprecise raw sensory measurements (reasons include the equipment accuracy problems of edge devices, the impact of devices' deployed environments, and the interference during transmission). In many cases, end-users neither care about these raw sensory data nor the data format during in-network fusion, but interested fuzzy information, such as "How serious is the situation?" and "where

Manuscript received October 17, 2019; revised June 15, 2020 and November 20, 2020; accepted February 11, 2021. Day of Publication Month Day, 2021; date of current version Month Day, 2021. This work is supported by the Zhejiang Natural Science Foundation of China, under Grants LY19F020024, LY19F020032 and LY19F020023; the National Natural Science Foundation of China, under Grants 62072460, 62072407, 61802346, 61502421 and 61532021; Key R&D Project of Zhejiang Province under Grant 2021C01117; Zhejiang Postdoctoral research project under Grant ZJ2020089; the China Scholarship Council as well.

Y. Li, Y. Zhu, H. Cheng, T. Chen, P. Hu and R. Huan are with the school of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China. E-mail: {liyinglong, yhzhu, chenghb, tmchen, pinghu, huanrh}@zjut.edu.cn.

W. Liu is with the School of Computer Science, Electrical and Electronic Engineering, and Engineering Maths, University of Bristol,Bristol BS81UB, UK. E-mail: weiru.liu@bristol.ac.uk.

H. Chen is with the School of Information, Renmin University of China, Beijing 100872, China. E-mail: chong@ruc.edu.cn.

Color version of one or more of this fugures in this paper are available online at http://ieeeexplore.ieee.erg.
Digital Object Identifer ***

1063-6706 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY OF BRISTOL. Downloaded on March 30,2021 at 08:58:29 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2021.3059952, IEEE Transactions on Fuzzy Systems

is the approximate event location?". Fuzzy sets [28] provide a good foundation for analyzing and processing the imprecise and uncertain raw data of sensor systems in a robust and understandable way, which can bring lots of benefits in terms of intelligence, energy efficiency and privacy protection.

To tackle the aforementioned problems, we propose two novel lightweight privacy-preserving range query processing schemes for IoT edge services using fuzzy sets, which have not been studied to the best of our knowledge. Our contributions are summarized as follows:

1) Motivated by the idea of fuzzy sets, linguistic range variables and fuzzy overlap ratio labels are proposed respectively, and linguistic variables instead of raw sensory values are used for fuzzy range query processing, which benefits both privacy protection and energy efficiency.
2) Fuzzy range recovering mechanism as well as two distributed privacy-aware fuzzy range query processing schemes are introduced, which can provide intelligent edge services in two common IoT application scenarios.
3) Theoretic analysis and experimental evaluations using real-world data sets are performed to validate our motivation in terms of effectiveness and efficiency.

The remainder of this paper is organized as follows. Section II reviews the related work. In Section III we introduce the preliminaries such as the definition of fuzzy range query over edge devices. In Section IV, we propose data transformation methods using fuzzy sets and two privacy-aware fuzzy range query schemes. The performance analyses are discussed in Section V. Experimental evaluations are conducted in Section VI. Finally, Section VII summarises our conclusions and future work.

II. Related Work

There has been a considerable amount of literature [11]-[20], [23-27] dealing with privacy-preserving range queries over various edge devices. Wu et al in [12] proposed ServeDB supporting secure and scalable multidimensional range query on outsourced database. Though the proposed SVETree in ServeDB cannot be used directly when organizing multidimensional sensory data on edge IoT devices, its creative ideas have inspired our research, especially when dealing with multidimensional sensor data while conducting experiments. In [16], Sheng et al. firstly studied the problem of privacy-preserving range query and applied a bit map to represent the buckets that have data and broadcasts the bit map to the nearby nodes. Nevertheless, a compromised storage node is able to breach the integrity verification of the network easily. Shi et al proposed spatial-temporal crosscheck scheme in [17], in which bitmap index instead of hash check code was used for examine the integrity of the query results. However, [17] inherited the same weakness as that in [16], and the cost of communication between storage nodes and base stations is high. To address these drawbacks, Chen et al. proposed the SafeQ scheme based on prefix family and neighborhood chains in [19]. The basic idea of SafeQ is that the sensory data and queries are encoded via prefix, where each data is prefixed by a range. Although SafeQ is efficient, its costs are still relatively high. Kong et al. in [18] proposed a range query scheme that can successfully preserve the location privacy of the involved data requesters and vehicles, and protect the confidentiality of the sensed data. Chen et al. in [20] proposed a privacy-preserving range query technique. Through a data hidden technique, both sensory data and queries are encoded whilst storage nodes can correctly process encoded queries over encoded data.

Techniques such as Order-Preserving Encryption (OPE) [21], Order-Revealing Encryption (ORE) [21] and differential privacy [22] for secure range queries in database systems have been proposed. Although these techniques can achieve desired privacy protection relying on custom data structures of certain tasks in traditional database systems, they are still a bit costly in terms of communication and storage in resource-constrained edge-computing scenarios. Wang et al. in [23] designed a symmetric-key searchable encryption scheme that can support geometric range queries on encrypted spatial data, though there are some small drawbacks relevant to high cost. Yi et al. in [24] proposed an order preserving function-based secure range query scheme (QuerySec) and a link watermarking scheme to encode both sensor-collected data and sink issued queries. Compared with SafeQ, QuerySec scheme requires lower communication cost. In order to reduce the communication cost, Zhang et al. [25] presented an encoding-based scheme (ES-RQ) for secure IoT range query. Zeng et al. [26] proposed a privacy-preserving multi-dimensional range query protocol called PERQ, which not only achieves data privacy, but also considers collusion attacks, probability attacks and differential attacks. Dong et al. [27] proposed a privacy-preserving range query framework SecRQ. In SecRQ, a generalized inverse matrix and a distance-based query mechanism are adopted, which can provide privacy protection for sensitive data as well as integrity checking.

III. Preliminaries

In this section, we briefly introduce the definition of fuzzy sets, and then give the definition of fuzzy range query over edge devices. Besides, the network structure and privacy threat models are presented as well.

Fuzzy sets [28] which were first introduced by Lotfi A. Zadeh in 1965 provide a good basis for analyzing and processing the imprecise and uncertain data of complex systems in a robust and understandable way [29]-[33].

Definition 1: **Fuzzy Sets**, a fuzzy set F is a pair (X, μ), where X is a set and $\mu : X \rightarrow [0, 1]$ is a membership function. The reference set X is called a universe of discourse, and for each $x \in X$, the value $\mu(x)$ is called the grade of membership of x in (X, μ). Function $\mu = \mu_F$ is called the membership function of the fuzzy set $F = (X, \mu)$. The (crisp) set of all fuzzy sets on a universe X is denoted as $F(X)$.

For a finite set $X = x_1, \cdots, x_n$, the fuzzy set (X, μ) is often denoted by $\{\mu(x_1)/x_1, \cdots, \mu(x_n)/x_n\}$. Let $x \in X$. Then x is referred to as

- **not included** in the fuzzy set (X, μ) if $\mu(x) = 0$ (not a member)
- **fully included** if $\mu(x) = 1$ (full member)
- **partially included** if $0 < x < 1$ (fuzzy member)

Authorized licensed use limited to: UNIVERSITY OF BRISTOL. Downloaded on March 30,2021 at 08:58:29 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2021.3059952, IEEE Transactions on Fuzzy Systems

Note that $\mu(x) = 0.5$ is the greatest uncertainty point. In this paper, we will propose several fuzzy information description methods using the idea of membership function in subsequent privacy-aware fuzzy range query schemes.

Definition 2: **Fuzzy Range Query Over Edge Devices**, it refers to extracting user-interested fuzzy information from distributed edge devices, where the readings fall into user's query range in a specified time window. Therefore, an user's range query usually contains three elements: query objects, a query time window, and a query range, which can be formalized as a tripe notation:

$$fuzzyR = (S, T^w, [lower, upper]^i),$$

where S is the set of sensor-equipped edge devices and T^w is the time window. $[lower, upper]^i$ is the user's query range on the i^{th} attribute readings over S, and the "*lower*" and "*upper*" is the lower bound and upper bound of a user's query range respectively. $fuzzyR$ is the fuzzy query results that meet the user's query ranges over edge devices S in time window T^w. It is worth mentioning that $[lower, upper]^i$ would be transformed using the idea of fuzzy sets in our subsequent fuzzy range query schemes.

Network Structure. There are two common network deployments for range query over edge devices. The first one is that edge devices are randomly deployed and these devices form a TAG [34] routing tree to an IoT gateway in a self organized way, shown as in Fig. 1a. In this network structure, the sensory measurements are collected and stored in the local edge devices. The second network structure is that storage devices are added to the first network, and this structure is usually called a two-tiered [16], [17], [25] network structure, as shown in Fig. 1b. Storage nodes might have more resources, or can be the normal edge devices. These storage nodes not only store the local sensor readings but also manage the local network topology, and eventually build data forward paths to the gateway, shown as in Fig. 1b. The first network structure is relatively simple, thus is easy to deploy and manage. The second one is more complicated, whereas it is more suitable for delay-sensitive edge systems because the storage nodes reduce the transmission paths.

Privacy Threats. Adversaries try to obtain sensitive data by sniffing or eavesdropping the message packets during the data transmission. As long as the sensed measurements or users' queries are transmitted in an easy-to-understand format or even using plaintext, these data and queries might be exposed to malicious adversaries. Therefore, *eavesdropping attack* is the main privacy threat that this paper focuses on. Besides, we partially consider the *compromise attack* of storage nodes or edge devices, since fully resist compromise attacks usually need specific access control or other protection measures, which deviates a bit from the main focus of this paper.

IV. Privacy-Aware Fuzzy Range Query Schemes

In this section, we propose two privacy-aware range query schemes: *local storage based privacy-aware range query* and *proxy storage based privacy-aware range query*, with regard to the two mentioned network structures of range query in Section III. The former is designed for edge computing and services with higher accuracy requirements, while the latter can be applied to the edge services with requirements of higher real-time response. In addition, our two schemes can work alongside with the ones using raw sensed data. Our algorithms as well as others can be embedded in the gateway, and any of these approaches can be chosen and be set as the default option according to users' requirements.

Before introducing the two aforementioned range query schemes, some preprocessing over sensed measurements and user's queries should be performed. We try to find a lightweight homomorphic preprocessing method for both sensory data and user's queries.

A. Sensed Data Transformation Using Fuzzy Sets

As discussed in Section I, it is a conventional way to have both the readings of edge devices and user's queries encrypted for the privacy protection purpose. However, the encryption and decryption processes are computationally complex, communication costly and time consuming, besides encryption destroys data usability, or pays a high cost for recovering data. To this end, a lightweight data transformation method using fuzzy sets is proposed, which paves a way to process range query directly on the transformed fuzzy data.

Non-Uniform Range Partition: The overall range $\Re$ of sensed measurements is non-uniformly divided into m sub-ranges $sub\Re$ based on the probability density function (e.g., $f(x)$ in subsequent (1)) of sensed measurements, so that the readings of edge devices fall into these sub-ranges approximately on average. Let $\Re$ be $[L, H]$, then $\Re$ could be divided into m sub-ranges $sub\Re^i (i = 1...m)$, such that:

$$1) \bigcup_{i=1}^{m} sub\Re^i = [L, H],$$
$$2) sub\Re^i \cap sub\Re^j = \emptyset, \forall i, j \in \{1, ..., m\}, i \neq j,$$
$$3) \int_{sub\Re^i} f(x)\mathrm{d}x = \int_{sub\Re^j} f(x)\mathrm{d}x, \forall i, j \in \{1, ..., m\}.$$

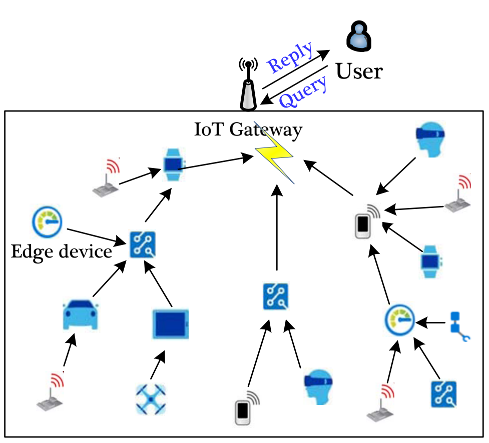
Generally speaking, readings of an edge device in an IoT application approximately follow a certain statistical distribution during a certain period of time, which reveals some characteristics of the usually spatio-temporal correlated sensory data. For example, the environmental sensed measurements such as *Temperature* and *Humidity* of a sensor device during a specific period of time usually approximately follow Gaussian distribution [35] with mean μ and standard deviation σ, of which probability density function ($f(x)$) is:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (1)$$

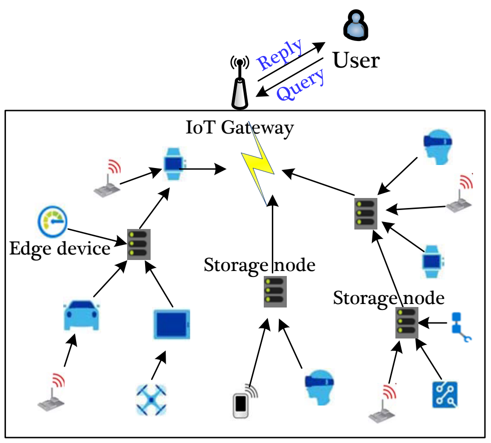
where μ and σ can be assigned based on historical data statistics or by a domain expert. According to the characteristics of Gaussian distribution, we know that the closer $sub\Re$ to the mean value μ, the smaller its interval size is.

1063-6706 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY OF BRISTOL. Downloaded on March 30,2021 at 08:58:29 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2021.3059952, IEEE Transactions on Fuzzy Systems



(a) Edge network with TAG routing tree



(b) Two tiered edge network with storage nodes

Fig. 1. Network structure of range query over edge devices.(a) Edge network with ordinary sensor devices, these devices form a TAG routing tree to IoT gateway in a self organized way. (b) Edge network with storage nodes, as well as a routing tree based on these storage nodes.

TABLE I
EXAMPLES OF LRV DEFINITION

$sub\Re^i(i = 1...m)$	LRVs	Interpretation
$[x_{m-1}, H]$	H(when m is 8)	...
...	...	...
$[x_2, x_3]$	C	*Very close to the* μ
$[x_1, x_2]$	B	*Close to the* μ
$[L, x_1]$	A	...

Note that, our range partition method does not only work on Gaussian distribution, other distributions even rectangular distribution can also produce non-uniform sub ranges using our method. Sensor readings of edge devices are usually spatio-temporal correlated or have some characteristics of a certain distribution, which benefits our sub range partitions. Therefore, to some extent, our range partition method is somewhat distribution independent.

Linguistic Range Variable (LRV). Motivated by the idea of fuzzy sets [28], we firstly calculate all the average distances (e.g., Euclidean distance) between $sub\Re^i(i = 1...m)$ and the mean value of the sensory data distribution. And then the distance could be the universe of discourse of fuzzy set "close to the mean value". We define a Linguist Range Variable (LRV) for each $sub\Re$, and each LRV has a semantic interpretation. Apparently, there are also m LRVs, and their interpretations are shown in Table I.

where m is specified according to the user's accuracy requirement. Generally speaking, a larger m benefits the higher accuracy of a query result, since the larger the m is, the more accurate the sub-range description is. However, a lager m brings more LRVs and more fuzzy overlap ratio labels, which makes the LRVs recovering more complex during range query processing. The range partition and LRV definition are embedded and shared in both edge devices and gateway. LRVs instead of raw values are used for later range query processing based on the idea of fuzzy sets.

B. Local Storage Based Privacy-Aware Fuzzy Range Query

In this section, we present a Local storage based Privacy-aware fuzzy Range query (denoted as LPRange) scheme. There are three major processes in LPRange. The first one is the overlap ratio labels definition based on the overlap ratio calculation between the partitions and the user's query range, followed by the fuzzy transformation of user's query range, both of which are performed in the gateway side. The third one is the query range recovery and query matching in edge devices.

Overlap Ratio Computing. Check the overlap between user's query range $\Re^u$ ([$lower$, $upper$]) and $sub\Re^i(i = 1...m)$, if there are overlaps (namely, $\max(\Re^u.lower, sub\Re^i.lower) \leq \min(\Re^u.upper, sub\Re^i.upper)$), then calculate the overlap ratios $\gamma^i(i = 1...m)$. There are three cases when there is an overlap between $\Re^u$ and $sub\Re^i(i = 1...m)$, which are shown in Fig. 2a, Fig. 2b, and Fig. 2c respectively.

Their corresponding ways to compute $\gamma^i(i = 1...m)$ are:

$$Case1 : \gamma^i = \frac{sub\Re^i.upper - \Re^u.lower}{sub\Re^i.upper - sub\Re^i.lower}, \quad (2)$$

$$Case2 : \gamma^i = \frac{\Re^u.upper - sub\Re^i.lower}{sub\Re^i.upper - sub\Re^i.lower}, \quad (3)$$

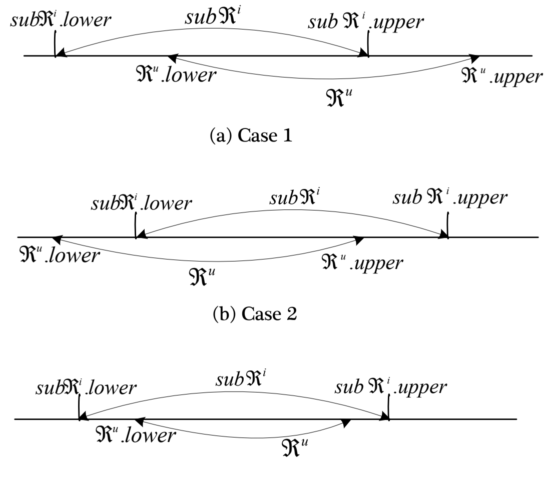
$$Case3 : \gamma^i = \frac{\Re^u.upper - \Re^u.lower}{sub\Re^i.upper - sub\Re^i.lower}. \quad (4)$$

Once the $\gamma^i(i = 1...m)$ values are obtained, these values are then transformed to Fuzzy Overlap Ratio (FOR) labels based on the idea of membership function [28] in fuzzy sets. We define five FOR labels to describe the noticeable overlap ratios $\gamma^i(i = 1...m)$. For example, the "*Extremely big*" label is used to denote any $\gamma^i(i = 1...m)$ which is larger than 0.95, and those $\gamma^i(i = 1...m)$ that are smaller than 0.25 are filtered without any transformation. More details about how to process such fuzzy transformation are demonstrated in Table II.

In Table II, the partition of $\gamma^i(i = 1...m)$ is performed non-uniformly, the $\gamma^i(i = 1...m)$ below 25% can be filtered,

1063-6706 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY OF BRISTOL. Downloaded on March 30,2021 at 08:58:29 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2021.3059952, IEEE Transactions on Fuzzy Systems



(c) Case 3

Fig. 2. Three cases of overlaps between $\Re^u$ and $sub\Re^i$. (a) Case 1: when $sub\Re^i.upper \leq \Re^u.upper$. (b) Case 2: when $sub\Re^i.upper) \geq \Re^u.upper$. (c) Case 3: when $\Re^u \subseteq sub\Re^i$.

TABLE II
EXAMPLES OF FOR LABELS AND ENCODINGS

$\gamma^i(i=1...m)$	FOR labels	Encodings
>95%	*Extremely big*	101
85%∼95%	*Very big*	100
70%∼85%	*Big*	011
50%∼70%	*Medium*	010
25%∼50%	*Small*	001
<25%	NULL(be filtered)	NULL

where NULL* means "*without FOR label and encodings*". More FOR labels are used to describe higher overlap ratios, such as the top rows of the first column in Table II have higher γ^i than the ones of the bottom rows. This strategy can improve the accuracy of range queries. Fixed-length encodings shown in the third column in Table II instead of FOR labels can be used in data transmission for energy-saving and privacy protection. In addition, variable length encodings, such as Huffman encodings, are recommended to encode the FORs when the number of FOR labels is relatively large, because in this case variable length encodings are beneficial to reduce the total encoding bits and thus trim down communication cost, which might be useful in communication cost sensitive IoT applications.

Fuzzy Transformation of Users' Range Queries. Firstly, the overlap ratios of $\Re^u$ and $sub\Re^i(i=1...m)$ should be calculated, and then the overlap ratios that are larger than the threshold are transformed into corresponding LRVs and FOR labels based on the definitions of LRV and FOR in Table I and Table II respectively. We provide an algorithm (Algorithm 1) to describe how to do such fuzzy transformation in details. LRVs are used for fuzzy transformation of sensitive queries, while FOR labels are used for approximately recovering LRVs to users' ranges, both of which contribute to privacy protection, as well as energy efficiency and real-time response.

TABLE III
EXAMPLES OF RANGE PARTITIONS AND LRV DEFINITION

$sub\Re^i(i=1...m)$	LRVs
...	...
[15, 23)	D
[11, 15)	B
[9, 11)	A
[5, 9)	C
[-3, 5)	E
...	...

TABLE IV
EXAMPLES OF FUZZY TRANSFORMATION OF $\Re^u$

$\Re^u$	LRVs	FOR labels
[12, 15)	B	*Big*
[13, 18)	B, D	*Medium, Small*
[-6, 5.5)	E	*Extremely big*
[6, 10)	C, A	*Big, Medium*
...	...	...

Algorithm 1. Fuzzy Transformation of $\Re^u$

```
INPUT: $\Re^u$ ($[lower, upper]$)
OUTPUT: LRVs and FOR labels
1. for $i$: 1 to $m$ do
2.     if max($\Re^u.lower, sub\Re^i.lower$)
           $\leq$ min($\Re^u.upper, sub\Re^i.upper$) then
3.         Calculate $\gamma^i$ using (2) or (3) or (4);
4.         if $\gamma^i > thrd$ then // $thrd$ is 25% here
5.             Return $LRV(sub\Re^i)$ and $FOR(\gamma^i)$;
6.         endif
7.     endif
8. endfor
```

In algorithm 1, $LRV(sub\Re^i)$ is the LRV corresponding to $sub\Re^i$, as is shown in Table I. $FOR(\gamma^i)$ is the FOR label corresponding to γ^i as shown in Table II. $thrd$ is the ratio threshold that determines whether $LRV(sub\Re^i)$ is a user's LRV . $thrd$ influences the accuracy of users' queries and it can be assigned according to their accuracy requirements.

Examples of Fuzzy Transformation of $\Re^u$. We give some examples showing how Algorithm 1 works. The examples of non-uniform range partition and LRV definition are shown in Table III.

Taking a user's query range ($\Re^u$ is [12, 15]) for example, there is only one sub-range [11, 15) overlapping with $\Re^u$. The overlap ratio γ between $\Re^u$ and [11, 15) is 75%, therefore its FOR label is "*Big*" according to Table II, and its LRV is '*B*' according to Table III shown in Table IV. When $\Re^u$ is [-6, 5.5], there is only one sub-range [-3, 5) overlapping with $\Re^u$, and the overlap ratio γ between $\Re^u$ and [-3, 5) is bigger than 95%, therefore the FOR label is "*Extremely big*", as is shown in the fourth row of Table IV. When $\Re^u$ is [6, 10], there are two sub-ranges overlapping with $\Re^u$, which are [5, 9) and [9, 11). Their overlap ratios are 75% and 50%, and then the FOR labels are "*Big*" and "*Medium*", as is shown in Table IV.

After transforming users' queries into LRVs and FOR labels in a gateway, the LRVs and FOR Labels instead of users' raw queries are sent to edge devices. An ideal case is that users' queries match the partitions completely, which is very ben-

1063-6706 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY OF BRISTOL. Downloaded on March 30,2021 at 08:58:29 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2021.3059952, IEEE Transactions on Fuzzy Systems

eficial for high accuracy. However this is extremely difficult because users' queries vary in different scenarios, which makes it difficult for having fuzzy information precisely describe the actual users' queries. Therefore, we provide a recovery mechanism for fuzzy queries in order to approximately recover users' actual queries.

Recovering of Users' Queries. In an edge device, when it receives k LRVs and FOR labels from a gateway, the fuzzy user's query range can be recovered as $\Re'$. There are two cases to compute $\Re'$, one of which is that when there is only one LRV and FOR label ($k = 1$), the lower and upper bound of $\Re'$ can be calculated as:

$$\Re'.lower = sub\Re(LRV).lower + (sub\Re(LRV).upper - sub\Re(LRV).lower) \times \frac{(1 - avg(\gamma(FOR)))}{2}, \quad (5)$$

$$\Re'.upper = sub\Re(LRV).upper - (sub\Re(LRV).upper - sub\Re(LRV).lower) \times \frac{(1 - avg(\gamma(FOR)))}{2}, \quad (6)$$

where $sub\Re(LRV)$ is the sub-range of the LRV, and $avg(\gamma(FOR))$ is the average overlap ratio of the FOR label. $avg(\gamma(FOR))$ can be defined as: $avg(\gamma(FOR)) = (\gamma(FOR).upper + \gamma(FOR).lower)/2$, where $\gamma(FOR).upper$ is the upper bound of γ (overlap ratio range partition) and $\gamma(FOR).lower$ is the lower bound of γ. Take Table II for example, $avg(\gamma(Big)) = (70\% + 85\%)/2 = 77.5\%$, namely, 0.775.

Another case is that when k is bigger than 1, lower and upper bound of $\Re'$ can be computed as:

$$\Re'.lower = sub\Re(LRV^1).upper - (sub\Re(LRV^1).upper - sub\Re(LRV^1).lower) \times avg(\gamma(FOR^1)), \quad (7)$$

$$\Re'.upper = sub\Re(LRV^{k-1}).upper + (sub\Re(LRV^k).upper - sub\Re(LRV^k).lower) \times avg(\gamma(FOR^k)). \quad (8)$$

Examples of $\Re^u$ Recovering. When there are two LRVs and two corresponding FOR labels, which are "(B, *Medium*)" and "(D, *Small*)", as is shown in Table IV. Then the recovered query range is [12.6, 18] according to (7) and (8). Comparing with the true user's query range $\Re^u$([13, 18]), there might be 8% false positives in query results. When $\Re^u$ is ([12, 15]), there is only one LRV and one FOR label, which is "(B, *Big*)". Then the recovered query range is [11.45, 14.55] using (5) and (6), which results in 18.3% false positives and 15% false negatives respectively.

LPRange Processing. The main process of LPRange can be described as follows: Firstly, User's range request $\Re^u$ is transformed to LRVs and FOR labels in the gateway. According to the LRVs and FOR labels received in edge devices, the user's fuzzy query range can be recovered as $\Re'$ using (5). Then each edge device locally checks whether its measurements fall into $\Re'$, and return the encrypted results that meet user's query request. More details are shown in Algorithm 2.

Algorithm 2. LPRange Processing

```
INPUT: LRVs and FOR labels in Query Message(QM)
OUTPUT: Fuzzy range query results
1.  Gateway floods QM with $k$ LRVs and FOR labels
      // in a distributed way
2.  for each edge device $i$ receives the QM do
3.     if $k > 1$ then
4.        $\Re' = [\Re'.lower, \Re'.upper]$ using (7) and (8);
5.     else if $k == 1$ then
6.        $\Re' = [\Re'.lower, \Re'.upper]$ using (5) and (6);
7.        endif
8.     endif
9.     if $readings(i) \in \Re'$ then
10.       Return encrypted results to the gateway;
            // usually locations
11.    endif
12. endfor
```

In the query results returning phase of Algorithm 2, the "encrypted results" usually means encrypted locations or global location identifications. If encryption is needed, a lightweight Diffie-Hellman [34] key exchange protocol might be a good choice. The time overhead of Algorithm 2 is discussed in later Subsection V.D.

Running Example of LPRange. Firstly, an user's query $\Re^u$ (take the example (13, 18] based on Table II, Table III and Table IV) is transformed to LRVs and FOR labels ("$B, D, Medium, Small$") in a gateway (as shown in Fig. 1). Secondly, the gateway floods QM with "$B, D, Medium, Small$" (3-bit FOR encodings actually) to edge devices in a distributed way. Then range recovery is performed after receiving QM in each edge device. The example "$B, D, Medium, Small$" was recovered as (12.6, 18], namely (15-0.6*4, 15+8*0.375] using (7) and (8). Finally, the query results (mainly locations) are sent to the gateway and eventually to the users. There were two false positives but no false negative in LPRange in this experimental scenario.

C. Proxy Storage Based Privacy-Aware Fuzzy Range Query

In LPRange scheme, query message flooding consumes massive time and energy. In addition, the privacy of sensed measurements stored in edge devices are fragile due to the possible edge device compromise. Therefore, we propose another secure range query scheme using proxy storage nodes, called Proxy storage based Privacy-aware Range query (PPRange), in order to improve the privacy protection and real time performance.

Fuzzy Transformation and Storage of Devices' Measurements. When there is a sensed measurement in an edge device, it is transformed to a LRV using the methods of range partition and the LRV definition discussed in Subsection IV.A. Then the LRV is sent to the nearby storage node. There is a table in each storage node for storing the received LRVs, as shown in Table V. In the example of Table V, the storage node receives three

Authorized licensed use limited to: UNIVERSITY OF BRISTOL. Downloaded on March 30,2021 at 08:58:29 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2021.3059952, IEEE Transactions on Fuzzy Systems

TABLE V
An Example of Storage Table in a Proxy Storage Node

$Timewindows$	LRVs	Locations
Sampling period 1	B	2(ciphertext format)
Sampling period 1	D	5
Sampling period 1	E	3
Sampling period 2	A	5
Sampling period 2	C	2
...	...	...

LRVs (*B*, *D*, and *E*) in sampling period 1, and their locations are recorded as well.

Fuzzy Transformation of Users' Range Queries. Fuzzy transformation of users' queries in PPRange is partially similar to the one of LPRange, but there are some differences. The similar parts are the overlap calculation and LRV transformation which can be described as follows. Using the overlap ratio computing methods in Subsection IV.B, the overlap ratios of users' queries and $sub\Re^i(i = ...m)$ can be calculated. When a overlap ratios $\gamma^i(i = ...m)$ is larger than a filter threshold, its corresponding LRV should be included in query message. For example, in Table IV, when $\Re^u$ is [-8, 5.5], the overlap ratio of $\Re^u$ and $sub\Re(E)$ is greater than 1, thus '*E*' is one of the LRVs in query message; whilst the overlap ratio of $\Re^u$ and $sub\Re(C)$ is 12.5% and is less than the threshold "25%", and thus '*C*' is not included in query message. The main difference is that there is no FOR labels in the query message of PPRange, because there is no range recovery of users' queries in PPRange.

Main Process of PPRange. Firstly, a gateway sends a query message to those proxy storage nodes. LRVs are the major data in query message. When storage nodes receive the query message, those locations whose LRVs match user's LRVs are encrypted and sent to the gateway. More details of the PPRange processing are shown in Algorithm 3.

```
Algorithm 3. PPRange Processing
INPUT: LRVs in Query Message (QM)
OUTPUT: Fuzzy range query results
1. Gateway sends QM with $k$ LRVs
2. for each storage node $i$ receives QM do
       // in a distributed way
3.     $i$ checks its storage table, and returns the encrypted
       locations whose LRVs match the user's LRVs;
4. endfor
```

In Algorithm 3, due to the distributed processing, the computational complexity of Algorithm 3 is $O(k{*}L_t)$, where k is the number of LRVs in a query message, and L_t is the table length of storage nodes. Actually, data transmission consumes most of the time in Algorithm 3.

Running Example of PPRange. Firstly, when there is a sensed measurement in an edge device, it is transformed to a LRV and is sent to its nearby storage node (storage table, as shown in Table V). Secondly, a user's query $\Re^u$ is transformed to LRVs only. Take the same $\Re^u$ (13, 18] and LRV definition in Table III for instance, its transformed LRVs were '*B*' and '*D*', and were sent to all the storage nodes. Finally storage nodes check their storage tables, and return the LRV-matched results. Actually, the executed query range was (11, 23] in this example, and there were only false positives due to the enlarged executed ranges, namely, (11, 13] and (18, 23].

V. Performance Analysis

In this section, the performance analysis in terms of privacy protection, accuracy, communication cost and response time of our two proposed schemes are studied.

A. Privacy Protection

Observation 1. *It is difficult for adversaries to infer users' actual queries or actually executed queries both in LPRange and PPRange, which makes our schemes have desirable capabilities of privacy protection.*

Both in LPRange and PPRange, users' queries are sent in the following format: $< LRV^1, \cdots, LRV^k, FORLabel^1, \cdots, FORLabel^k >$. It is extremely difficult to learn the actual queries ($\Re^u.lower$ and $\Re^u.upper$]) from LRV^i and $FORLabel^i$ $(i = 1, \cdots, k)$. Our LRVs are irregular characters based on non-uniform partitions. Table I shows some single-character examples mainly for an easy-to-understand purpose. Actually, Any character on the keyboard (including 128 characters) can be used as a single-character LRV. There is no correlations among these LRVs, nor any correlations between these LRVs and users' query range boundaries. Since users' range boundaries are neither discrete nor fixed, it is very difficult to infer both of them, as well as their correlations even by brute-force search. Besides, When a LRV is an irregular multi-digit string, brute-force search becomes more difficult.

There are also no correlations between FOR labels and their encodings.If longer bit encodings were used, the brute-force search space would become very large. For example, m FOR Labels and n-bit encodings results in search space being $O(m * 2^n)$, and as n increases, the search space grows geometrically. By the way, the rise of n increases the computational cost and communication overhead. Fortunately, in most practical applications, users may not need tough demand on security / privacy protection, so the number of FOR encodings does not need to be very large, which helps improving the overall performances (energy-efficiency, real time and privacy protection). All the definitions of LRVs, FOR Labels and their encodings are not involved in data transmission, which makes the adversaries impossible to know the actually executed queries in our two schemes by eavesdropping or sniffing attacks.

Observation 2. *To some extent, the privacy protection performance of PPRange is better than LPRange.*

In PPRange, The raw sensing data in edge devices can be deleted after fuzzy transformation to LRVs, which can prevent sensitive raw sensory measurements from leaking when edge devices are compromised. Besides, due to the LRVs information being stored in storage nodes, the privacy leakage risk when storage nodes being compromised in PPRange is

1063-6706 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY OF BRISTOL. Downloaded on March 30,2021 at 08:58:29 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2021.3059952, IEEE Transactions on Fuzzy Systems

also decreased. However, the raw sensor readings in LPRange cannot be deleted, because they are needed for matching the recovered queries. Therefore, LPRange cannot resist the compromise of edge devices, which makes its privacy protection capability weaker than that of PPRange.

B. Accuracy

In LPRange, errors mainly come from both the fuzzy transformation of $\Re^u$, and its recovery stage. During the transformation from $\Re^u$ to *LRVs* and *FOR Labels*, All the parts of $\Re^u$ whose overlap ratios (γ^i) of $\Re^u$ and $sub\Re^i$ ($i = 1, \cdots, m$) below $thrd$ are filtered out, resulting in a total missed ranges: $Err.Tr = \sum_{i=1}^{m} \int_{\gamma^i * sub\Re^i} f(x)\mathrm{d}x$, and it reaches the maximum error when $\gamma^i = thrd$. In fact, the actual range of $thrd$ varies when $sub\Re^i$ changes. In the stage of query recovery, our recovery methods (such as Eq.(5) and Eq.(6) when $k = 1$ or Eq.(7) and Eq.(8) when $k > 1$) make the recovered range either enlarged or reduced, because the actual range boundaries of $\Re^u$ might be on the left of $avg(\gamma(FOR^i))$, or on the right of it (x-axis). Let $\triangle sub\Re^i$ and $\nabla sub\Re^i$ represent the amount of missed and enlarged range with respect to $sub\Re^i$, which is determined by our recovery mechanism (such as γ^i and $avg(\gamma(FOR^i))$). Then the errors (both false negatives and false positives) in the query recovery are $\int_{\triangle sub\Re^i} f(x)\mathrm{d}x + \int_{\nabla sub\Re^i} f(x)\mathrm{d}x$ ($k = 1$) and $\sum_{i=1}^{k} \int_{\triangle sub\Re^i} f(x)\mathrm{d}x + \sum_{i=1}^{k} \int_{\nabla sub\Re^i} f(x)\mathrm{d}x$ ($k > 1$).

And then the total error rate (τ) of LPRange can be calculated approximately when $k = 1$ and $k > 1$ respectively as follows.

$$\tau_{k=1} = \frac{\int_{\triangle sub\Re^i} f(x)\mathrm{d}x + \int_{\nabla sub\Re^i} f(x)\mathrm{d}x + Err.Tr}{\int_{\Re^u} f(x)\mathrm{d}x}$$

,

$$\tau_{k>1} = \frac{\sum\limits_{i=1}^{k} \int_{\triangle sub\Re^i} f(x)\mathrm{d}x + \sum\limits_{i=1}^{k} \int_{\nabla sub\Re^i} f(x)\mathrm{d}x + Err.Tr}{\int_{\Re^u} f(x)\mathrm{d}x}$$

.

In PPRange, errors mainly come from the fuzzy transformation of $\Re^u$ and such errors are similar to the ones of LPRange, namely, $Err.Tr = \sum_{i=1}^{m} \int_{\gamma^i * sub\Re^i} f(x)\mathrm{d}x$, and it reaches the maximum error when $\gamma^i = thrd$. When $k = 1$, the possibly enlarged range is $sub\Re^i - \Re^u$ due to no range recovery in PPrrange, and the possible false positives are $\int_{sub\Re^i - \Re^u} f(x)\mathrm{d}x$. When $k > 1$, the possibly enlarged range is $\sum_{i=1}^{k} (sub\Re^i - \Re^u)$ and the possible false positives are $\sum_{i=1}^{k} \int_{sub\Re^i - \Re^u} f(x)\mathrm{d}x$.

Therefore the total error rate of LPRange can be calculated approximately when $k = 1$ and $k > 1$ respectively as follows.

$$\tau_{k=1} = \frac{\int_{sub\Re^i - \Re^u} f(x)\mathrm{d}x + \sum_{i=1}^{m} \int_{\gamma^i * sub\Re^i} f(x)\mathrm{d}x}{\int_{\Re^u} f(x)\mathrm{d}x},$$

$$\tau_{k>1} = \frac{\sum_{i=1}^{k} \int_{sub\Re^i - \Re^u} f(x)\mathrm{d}x + \sum_{i=1}^{m} \int_{\gamma^i * sub\Re^i} f(x)\mathrm{d}x}{\int_{\Re^u} f(x)\mathrm{d}x}$$

.

C. Communication Cost

In wireless communications, data transmission accounts for most of the communication cost. For example, transmitting one bit can consume as much energy as running several thousand instructions on a sensor's CPU [35]. Besides, data transmission is time-consuming and has direct impact on real time performance.

In LPRange, most of the data transmission occurs in query message floodings and query result returnings. The major data structure is (LRVs, FOR labels). When there are k LRVs and k FOR labels in a query message, the main data transmission $D_{queryTran}$ in query message flooding is:

$$D_{queryTran} = N \times k \times (sizeof(LRV) + sizeof(FOR)),$$

where $sizeof()$ means the number of bits, and N is the number of edge devices.

In a query result returning phase, the major data transmission D_{ret} is:

$$D_{ret} = \sum_{i=1}^{U} hop(i) \times sizeof(enLoc),$$

where U is the number of query results, $hop(i)(i = 1...U)$ is the hop count of the i^{th} query result, and $enLoc$ is the encrypted location of a query result.

Therefore, the total data transmission D_{tran} of LPRange is:

$$\begin{aligned} D_{tran} &= D_{queryTran} + D_{ret} \\ &= N \times k \times (sizeof(LRV) + sizeof(FOR)) \\ &+ \sum_{i=1}^{U} hop(i) \times sizeof(enLoc). \end{aligned}$$

In PPRange, query messages only need to be sent to those storage nodes, therefore, the major data transmission of a query message flooding $D_{queryTran}$ is:

$$D_{queryTran} = \sum_{i=1}^{V} hop(i) \times k \times sizeof(LRV),$$

where V is the number of storage nodes, and k is the number of LRVs in a query message. $hop(i)(i = 1...V)$ is the hop count of the i^{th} storage node. The data transmission in a query result returning phase D_{ret} is:

$$D_{ret} = \sum_{i=1}^{W} hop(i) \times num(i) \times sizeof(enLoc),$$

where W is the number of storage nodes having query results, and $num(i)$ is the number of query results in the i^{th} storage node.

Therefore the total data transmission D_{tran} of PPRange is:

$$\begin{aligned} D_{tran} &= D_{queryTran} + D_{ret} \\ &= \sum_{i=1}^{V} hop(i) \times k \times sizeof(LRV) \\ &+ \sum_{i=1}^{W} hop(i) \times num(i) \times sizeof(enLoc). \end{aligned}$$

1063-6706 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY OF BRISTOL. Downloaded on March 30,2021 at 08:58:29 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2021.3059952, IEEE Transactions on Fuzzy Systems

D. Response Time

Compared with the time overhead on data transmission, the computing time can be ignored. Considering the distributed data transmission in both query messages flooding and results returning phases, the time cost of a query message flooding $T_{queryTran}$ depends on the largest hop count of edge devices, namely,

$$T_{queryTran} = \max(hop(i)) \times T(hop), (i = 1...N),$$

where N is the number of edge devices and $hop(i)$ is the hop count of the i^{th} edge device. $T(hop)$ is the time overhead of sending and receiving a data packet between two neighboring nodes.

The time overhead of a query result returning T_{ret} is:

$$T_{ret} = \max(hop(j)) \times T(hop), (j = 1...M),$$

where M is number of edge devices having a query reply and $hop(j)(j = 1...M)$ is the hop count of the j^{th} edge device having the query reply.

If there is an encryption for query results, then the time overhead of encryption T_{enc} should be considered, and T_{enc} is determined by a specific encryption technique used. Therefore, the time overhead of LPRange $Time$ is:

$$\begin{aligned} Time &= T_{queryTran} + T_{ret} + T_{enc} \\ &= [\max(hop(i)) + \max(hop(j))] \times T(hop) + T_{enc}, \\ &(i = 1...N, j = 1...M). \end{aligned}$$

Beware of that the data transfer from edge devices to storage nodes usually occurs before a user's query starts, which has no impact on any real time performance. In Algorithm 3, the query message transmitting and query results returning also perform in a distributed way. Therefore, the maximum hop count of storage nodes affects the response time directly. The time overhead of a query message transmitting $T'_{queryTran}$ is:

$$T'_{queryTran} = \max(hop(i)) \times T'(hop), (i = 1...V),$$

where V is the number of storage nodes, $hop(i)$ is the hop count of the i^{th} storage node, and $T'(hop)$ is the time overhead of sending and receiving a data packet between two neighboring storage nodes

The time overhead of a query result returning T'_{ret} is:

$$T'_{ret} = \max(hop(j)) \times T'(hop), (j = 1...W),$$

where W is the number of storage nodes having a query reply, and $hop(j)$ is the hop count of the j^{th} storage having a query reply.

Therefore, the total time overhead of PPRange $Time'$ is:

$$\begin{aligned} Time' &= T'_{queryTran} + T'_{ret} + T'_{enc} \\ &= [\max(hop(i)) + \max(hop(j))] \times T'(hop) + T'_{enc}, \\ &(i = 1...V, j = 1...W). \end{aligned}$$

where T'_{enc} is the time overhead of possible encryption of query results in PPRange.

VI. Experimental Evaluation

In this section, we evaluate our approaches using OMENT++ [38], where OMNET++ is a discrete event and component-based C++ simulation library and framework, primarily for building network simulators. Our experimental evaluations are based on both real-world data set LUCE [39] and Melbourne Urban Environments data set [40]. The real-world data set LUCE that we had downloaded before consists of location information and sensor data (*node ID*, *Temperature*, *Humidity*, *Solar Radiation*, *Wind Speed*, etc.) of 88 valid sensor nodes. Due to the disfunction of some edge devices or the influence of devices' surrounding environments, small part of the measurements in LUCE was missing that made the data sets somewhat incomplete. Therefore, we synthesized a small amount of data based on the spatio-temporal correlation of the real measurements to complement the missing data. Those 88 nodes were deployed in a 450m*300m edge region. When a gateway's location was at (0, 40) and the communication range was 80m, these nodes formed a TAG routing tree connecting to the gateway, shown as in Fig. 3a. We also set up a 12 storage nodes topology over the network deployment, shown as in Fig. 3b.

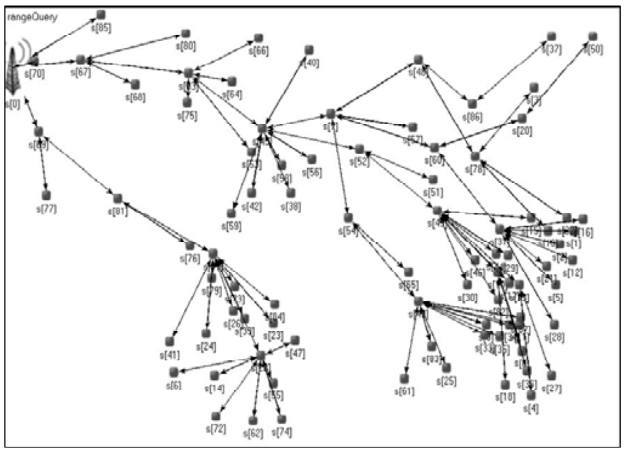
In addition to the data set LUCE, the real-world data set of Melbourne Urban Environments (Sensor readings, with temperature, light, humidity every 5 minutes at 9 locations (trial, 2014 to 2015)) [40] was also used to evaluate our schemes. In this data set, the environmental sensors, measuring *light level*, *humidity* and *temperature*, have been deployed at Fitzroy Gardens and Library at the Dock. The data collected assist the Urban Landscapes branch to better understand and communicate the impact of canopy cover for urban cooling. In our experiments, users try to find the locations where the edge device's measurements match the user's query ranges with the privacy protection on sensor data, user's queries, and locations of edge devices. We evaluated our two schemes in terms of accuracy, energy consumption, and real time performance, as well as performed comparisons with some of the existing methods.

A. Accuracy

We studied the accuracy of LPRange and PPRange with nine ($m = 9$) partitions of possible readings (*Temperature*: [-40, 60]). The range partitions and FOR labels shown in Table III and Table II respectively were used in our experiments. We conducted the experimental evaluations with three different $\Re^u$: [12, 15], [13, 18], and [-6, 5.5] and their accuracy performances are shown in Table VI, Table VII, and Table VIII, where $\#(false\_positives)$, $\#(false\_negatives)$, and $\#(true\_results)$ mean the numbers of false positives, false negatives, and true results respectively.

In Table VI, There are one false positive and one false negative in LPRange, and there are two false positives without false negative in PPRange when $\Re^u$ is in [12, 15]. The user's fuzzy query "(*B*, *Big*)" was recovered to be [11.45, 14.55] in LPRange, which was close to the true user's query range contributing more accurate query results. However there is no range recovering in PPRange, and the actual query range was

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2021.3059952, IEEE Transactions on Fuzzy Systems


(a) A network without storage nodes

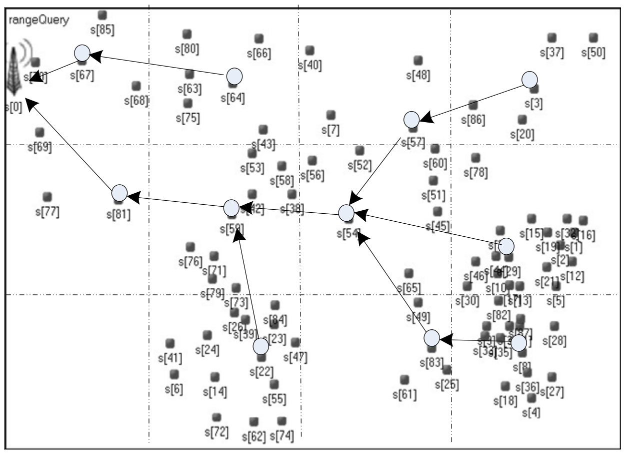

(b) A network with 12 storage nodes

Fig. 3. Real-world network with 88 valid edge devices. (a) Edge network composed ordinary sensor devices, these devices connected to IoT gateway hop-by-hop. (b) Edge network with 12 storage nodes formed a TAG routing tree based on these storage nodes.

TABLE VI
THE NUMBER OF ERRORS WHEN $\Re^u$ WAS [12, 15]

	LPRange	PPRange
$\#(false\_positives)$	1	2
$\#(false\_negatives)$	1	0
$\#(true\_results)$	14	14

TABLE VII
THE NUMBER OF ERRORS WHEN $\Re^u$ WAS [13, 18]

	LPRange	PPRange
$\#(false\_positives)$	2	9
$\#(false\_negatives)$	0	0
$\#(true\_results)$	21	21

TABLE VIII
THE NUMBER OF ERRORS WHEN $\Re^u$ WAS [-6, 5.5]

	LPRange	PPRange
$\#(false\_positives)$	0	2
$\#(false\_negatives)$	3	3
$\#(true\_results)$	22	22

[11, 15], which enlarged the query range, thus leading to more false negatives in PPRange.

When $\Re^u$ was [13, 18], there were two and nine false positives in LPRange and PPRange respectively, while both had no false negatives, as shown in Table VII. In LPRange, the user's fuzzy query "(*B*, *Medium*, *D*, *Small*)" was recovered to be [12.6, 18], which guarantees a better accuracy performance. While in PPRange, the actual range query performed was [11, 23], which brought some false positives. From these experimental results, it is not difficult to know that FOR labels below "*medium*" or with a rough range partitions enlarge the query range in PPRange since there is no range recovering in PPRange.

When $\Re^u$ was [-6, 5.5], there were three false positives in LPRange and PPPRange respectively, as shown in Table VIII. The actual query ranges performed in LPRagne and PPRange were [-2.9, 4.9] and [-3, 5.5] respectively, both of which were similar due to a good match between the sub-range of '*E*' and $\Re^u$.

TABLE IX
THE AVERAGE ACCURACY USING LUCE DATA SET

	LPRange	PPRange
$accuracy$	87.8%	71.9%

From the above experiments, we can see that LPRange has good and stable accuracy performance due to the range recovery of FOR information, while PPRange has relatively lower accuracy compared with LPRange. If users' queries in PPRange approximately fit the partitions, the accuracy of PPRange is similar with that of LPRange's, otherwise, its accuracy is lower and less stable than the one of LPRange due to the lack of range recovery in PPRange. Therefore, the accuracy of PPRange is somewhat users' queries dependent. However, the response time of PPRange is much less than LPRange's, which will be discussed later.

We also conducted experiments with two other partitions of *Temperature* measurements when m was 7 and 11 respectively based on the same aforementioned three different user's queries using the data collected in different ten periods, and computed the average $accuracy$ using the following definition: $accuracy = 1 - (\#(false\_positives) + \#(false\_negatives))/\#(true\_results)$.

The average $accuracy$ of LPRange over ten experimental runs was about 87.8% and the average $accuracy$ of PPRange was approximate 71.9%, as is shown in Table IX. LPRange has a better average accuracy performance than that of P-PRange, because there is a query range recovery mechanism in LPRange, while PPRange does not, which always enlarge its query ranges.

We further evaluated how the number of range partitions (m) influences the accuracy of our schemes LLRange and PPRange. We divided the sensed measurement *Temperature* ([-40, 60]) into 7, 9, and 11 sub-ranges, as shown in Table X.

Based the above three range partitions, we conducted three

1063-6706 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY OF BRISTOL. Downloaded on March 30,2021 at 08:58:29 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2021.3059952, IEEE Transactions on Fuzzy Systems

TABLE X
AN EXAMPLE OF THREE PARTITIONS

$m = 7$	(36, 60], (20, 36], (12, 20], (8, 12], (0, 8], (-16, 0], (-40, -16]
$m = 9$	(39, 60], (23, 39], (15, 23], (11, 15], (9, 11], (5, 9], (-3, 5], (-19, -3], (-40, -19]
$m = 11$	(40.5, 60], (24.5, 40.5], (16.5, 24.5], (12.5, 16.5], (10.5, 12.5], (9.5, 10.5], (7.5, 9.5], (3.5, 7.5], (-5.5, 3.5], (-21.5, -5.5], [-40, -21.5]

TABLE XI
THE AVERAGE ACCURACY USING RANDOM DATA

	LPRange	PPRange
accuracy	89.2%	67.9%

queries ([13, 18], [12, 15], and [-6, 5.5]) respectively. The experimental results are shown in Fig. 4. Learned from Fig. 4a, Fig. 4b, and Fig. 4c, m had a certain degree of influence on the accuracy of our schemes. The smaller the m is, the larger the number of errors (false positives or false negatives) is, and thus the lower the accuracy is. For example, as shown in Fig. 4b and Fig. 4c, when m was 7, the number of errors of both LPRange and PPRange were relatively larger, while there were fewer errors when m was 9 and 11. However, the value m and the accuracy were not proportional. For instance, the number of errors was bigger when m was 11 than that when m was 9, as shown in Fig. 4b. This phenomenon also happened in Fig. 4a, the number of errors was the smallest when m was 7. In fact, the accuracy of our scheme is not only related to m, but also related to the uses' queries themselves. When there are better matches between sub-ranges and users' queries, the recovered query range of LPRange is more accurate, as well as the executed queries in PPRange are closer to the initial users' queries, both of which contribute to a higher accuracy of our schemes.

In order to test the influence of data distribution on accuracy, we synthesized random data (such as random temperature data over [-50, 50]) uniformly for 87 valid nodes in LUCE. Then we made three groups of uniform partitions (m is 10, 15 and 20 respectively) and carried out experiments over six query ranges ([-7, 7], [-5, 5], [-5, 10], [5, 25] , [10, 18], [31, 36]), of which the experimental results are shown in Table XI.

In our experiments, we find that the data distribution (random data) has a negligible impact on LPRange's accuracy (89.2% in Table XI vs. 87.8% in Table IX), and its impact is even less important than that of subrange partitions (*m*). Actually, both of them have little influence on the accuracy of LPRange due to the range recovery of FOR in LPRange, making the executed query ranges in LPRange close to the ones using raw data. The influence of data distribution on PPRange's accuracy is also small, while the number of range partitions has a certain influence on PPRange's accuracy. Generally speaking, the larger the *m* is, the more accurate the PPRange is. This is because PPRange has no range recovery mechanism, and the larger *m* makes users' queries fit partitions better.

B. Communication Cost

Existing secure range queries are usually based on bucket techniques [14, 15, 17, 18, 22]. We strive to make a fair comparison with the existing methods, though it is difficult to do that because the ideas and techniques used are essentially different. For instance, our non-uniform partition method produces no empty buckets, which makes it a bit different from the existing ones. In addition, fuzzy data transformation and recovering are used in our approaches, which did not appear in the existing methods.

In our experiments, LRV was a 8 bits CHAR, and FORs were 3-bit fixed-length encodings. Sensed measurements such as *Temperature*, *Humidity*, and encrypted values in an encryption approach were all 32-bit FLOAT values (due to the 32-bit simulation platform). In bucket-based methods, the sizes of an encrypted bucket, a bucket tag and an authentication code are 16-bits, 8-bits and 32-bits respectively. For simplicity, other information such as HEAD, CHECK CODE in a message packet were not considered in the comparisons.

We conducted extensive experiments to evaluate our two schemes (LPRange and PPRange) in terms of data transmission using real-world data set LUCE based on five query ranges: $\Re^1$ ([12, 15]), $\Re^2$ ([13, 18]), $\Re^3$ ([-6, 5.5]), $\Re^4$ ([6, 10]), and $\Re^5$ ([0, 15]). We also made experimental comparisons among our two schemes, the naive method using raw data and the bucket technique using encrypted data. The communication cost (data transmission) performances are shown in Fig. 5.

In Fig. 5a, the amounts of data transmission of LPRange and of PPRange were similar, while both of them were much less than the one of naive method using raw data. In LPRange the amount of data transmission of a query message flooding is much bigger than the one of PPRange. Nevertheless, there is some communication overhead on storage in PPRange, while LPRange does not, which offsets the data transmission in LPRange. The raw data $[lower, upper]$ of a query range in the naive method is much larger than the LRVs and FOR Labels used in our two schemes, which leads to the larger data transmission of the naive method. In the experiments of query range $\Re^5$ ([0, 15]), four LRVs and FOR labels were required to describe the query range due to the large interval of its $[lower, upper]$, which makes the data transmission of the query flooding in LPRange increase significantly, While in PPRange, query messages only need to be transmitted to those storage nodes, which makes its data transmission grow slightly as shown in Fig. 5b. The data transmission of the bucket technique using encrypted data was much larger than the that of PPRange, as is shown in Fig. 5b, because both of its data transmission for key management and its data storage for bucket descriptions are larger than that of PPRange.

We also evaluated how gateway's locations influence the data transmission. When the gateway moved to the location of (260, 200) (shown as in Fig. 6a and Fig. 6b respectively), the communication cost (data transmission) performances are

1063-6706 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY OF BRISTOL. Downloaded on March 30,2021 at 08:58:29 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2021.3059952, IEEE Transactions on Fuzzy Systems

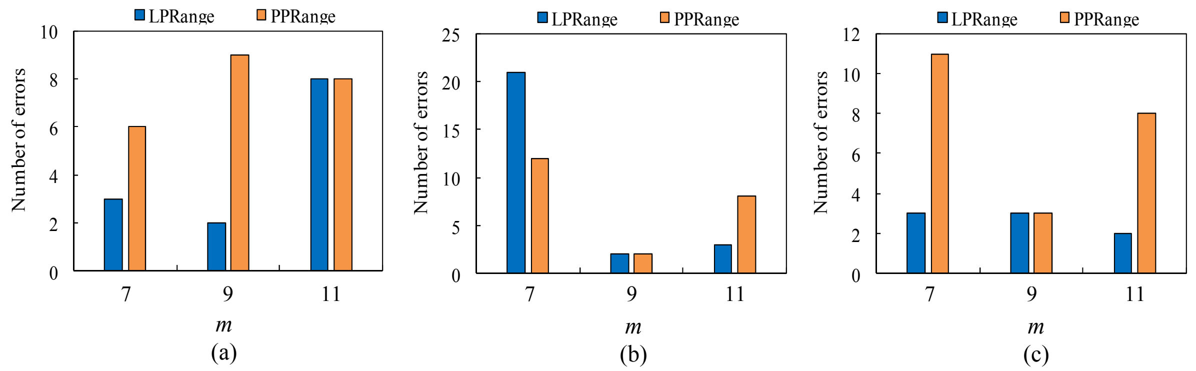


Fig. 4. Average accuracy comparison with different m in three user's queries over ten runs. (a) The average number of errors (both false positives and false negatives) when $\Re^u$ was [13, 18]. (b) The average number of errors when $\Re^u$ was [12, 15]. (c) The average number of errors when $\Re^u$ was [-6, 5.5]

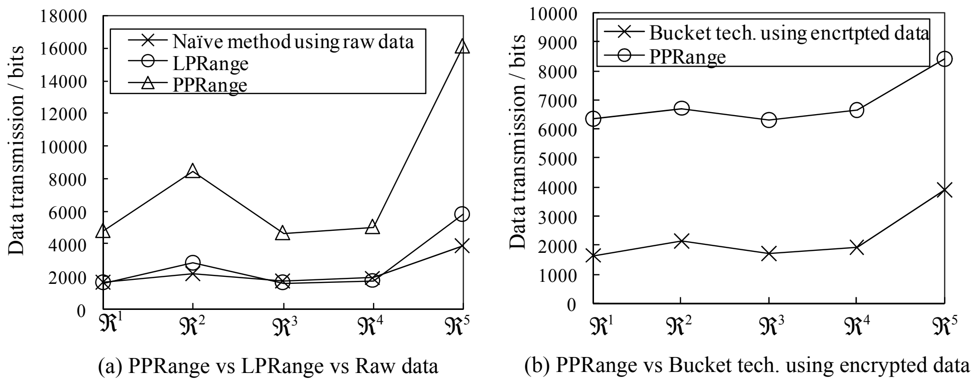


Fig. 5. Communication Cost comparison when gateway located at (0, 40). (a) Data transmission comparison between our two schemes. (b) Data transmission comparison between PPRange and bucket technique using encrypted data.

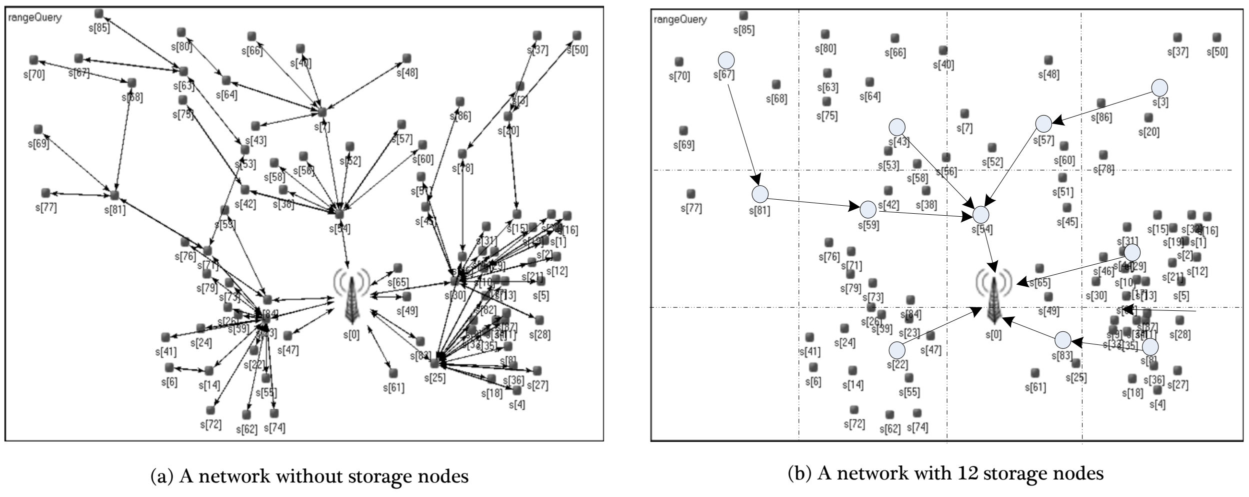

rangeQuery
(a) A network without storage nodes
(b) A network with 12 storage nodes

Fig. 6. Edge network of LUCE when gateway located at (260, 200). (a) Edge network deployment with ordinary sensor devices, these devices connected to a IoT gateway in a hop-by-hop way. (b) Edge network with 12 storage nodes and a storage nodes-based routing tree.

shown in Fig. 7.

The amount of data transmission of our two schemes in Fig. 6 did not change much, because the query results were the same due to the unchanged data set and user's queries. The changed network topology had some impact on the number of transmission hops. The numbers of relay hops both in the query message transmitting and the query result returning were reduced to some extent, causing a small decline of data transmission. In fact, both the number and the locations of query results have big influence on the amount of data transmission, because these two factors directly affect the number of returned packets and the total relay hops.

Authorized licensed use limited to: UNIVERSITY OF BRISTOL. Downloaded on March 30,2021 at 08:58:29 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2021.3059952, IEEE Transactions on Fuzzy Systems

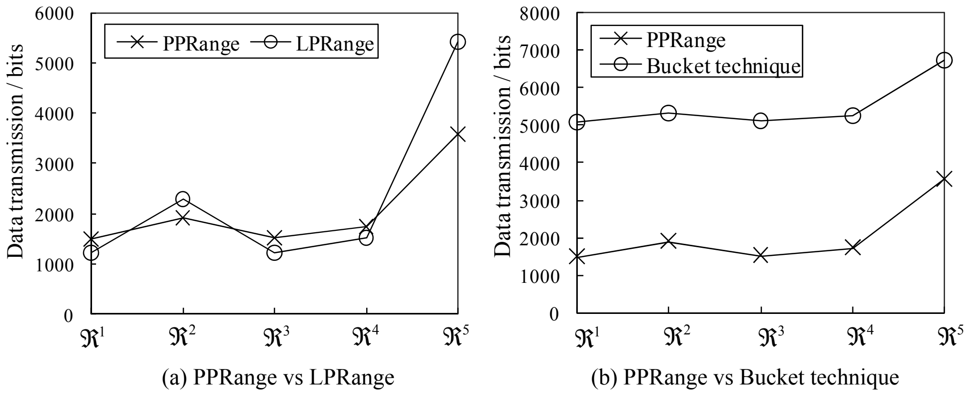


Fig. 7. Communication Cost comparison when gateway located at (260, 200). (a) Data transmission comparison between our two schemes. (b) Data transmission comparison between PPRange and conventional bucket techniques.

C. Response Time

In wireless communications, one hop of data packet transmission consumes much more time than the computation in a sensor. Therefore, the total number of transmission (relay) hops during query processing directly influences the response time. In PPRange the data transfer of data storage usually happens before a query starts, therefore, the time overhead on data storage should not be considered when evaluating the real time performance. In addition, the query message flooding in LPRange as well as the results returning in LPRange and PPRange are all performed in a distributed way. Therefore, the largest relay hops affect the real time performance. We conducted experiments using three query ranges ($\Re^1$([12, 15]), $\Re^2$([13, 18]), and $\Re^3$([-6, 5.5])) over two networks with 12 and 20 storage nodes respectively. The average numbers of relay hops of LPRange, PPRange and Encryption way over 12 and 20 storage nodes based networks are shown in Fig. 8a and Fig. 8b respectively.

In Fig. 8, the total number of relay hops of LPRange was bigger than the one of PPRange, since query messages have to be transmitted to each sensor device, and the node farthest from the gateway determines the time overhead due to the distributed transmission. The storage nodes help P-PRange shortening the longest paths, which helps reducing the response time of PPRange. In the 20 storage nodes based network, more storage nodes resulted in more relay hops in PPRange than that of 12 storage nodes based network. In our three range queries, there were results coming from the edge networks both in LPRange and PPRange, that is why the total number of transmission hops varied slightly in the three range queries. The numbers of total relay hops of Encryption both in Fig. 8a and Fig. 8b were much larger than that in our schemes, considering the distributed transmission as well. This is because more data packets were transmitted for the key management during encryption and decryption.

We further evaluated the influence of a gateway's location on the number of total relay hops. When the gateway moved to location at (260, 200), the response time performance of LPRange and PPRange are shown inFig. 8c and Fig. 8d respectively. After the gateway moving to (260, 200), the average number of network routing hops declined, as shown in Fig. 3 and Fig. 6. As a result, the average number of relay hops of both LPRange and PPRange decreased to some extent, as shown in Fig. 8c and Fig 8d. Actually, the locations of query results have a relatively large impact on the number of relay hops. From Fig. 8, we know that PPRange has shorter response time than that of LPRange, which can be used in the delay-sensitive and fault-tolerant IoT applications.

We further evaluated our schemes using another real-world data set Melbourne Urban Environments (MUE) [40]. There are 9 edge devices with 56,571 measurements of *temperature*, *light* and *humidity* every 5 minutes at 9 locations from 2014 to 2015. We conducted experimental evaluations over 1-D (*temperature*), 2-D (*temperature* and *light*) and 3-D (*temperature*, *light* and *humidity*) queries based on the measurements between 7.30am and 11.45am on Feb. 28, 2015 of data set MUE. In these experiments, users expected to find the records (mainly the time and locations) meeting their query ranges. The experimental results based on aforementioned data set (MUE) are shown in Fig. 9.

In Fig. 9a, the average accuracies of 1-D queries were the highest in LPRange, PPRange and the method using raw data comparing with those 2-D and 3-D experiments. This is because that both false positives and false negatives in 1-D queries usually increase the probabilities of false positives and false negatives in later 2-D and 3-D queries. The average accuracy performances of 1-D range queries using data set MUE were similar to that of 1-D queries using data set LUCE.

The average data transmission dropped as the number of dimensions declined, as is shown in Fig. 9b. The query results became fewer when more dimensions were involved in queries, since more dimensions involved in a query means more strict conditions were needed to filter the sensor readings. A smaller number of edge devices also reduces the data transmission when other query conditions are the same, such as the same dimension and sensor measurements. The average numbers of relay hops of LPRange and PPRange using data set MUE were smaller than that of our two schemes when LUCE used since there were less edge devices in MUE than the ones in LUCE. And the average numbers of relay hops of PPRange were smaller than that of LPRange, as shown in

Authorized licensed use limited to: UNIVERSITY OF BRISTOL. Downloaded on March 30,2021 at 08:58:29 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2021.3059952, IEEE Transactions on Fuzzy Systems

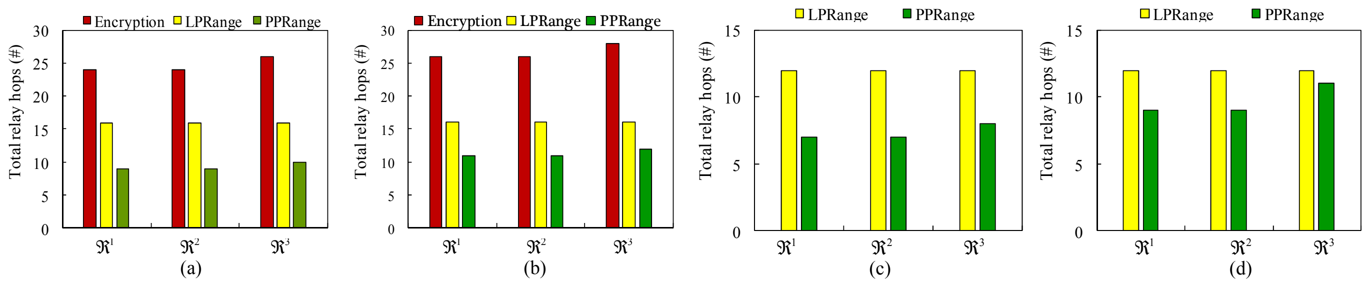


Fig. 8. Comparison of total number of relay hops among LPRange, PPRange and Encryption method using data set LUCE. (a) 12 storage nodes based edge network when gateway located at (0, 40). (b) 20 storage nodes based edge network when gateway located at (0, 40).(c) 12 storage nodes based edge network when gateway located at (260, 200). (d) 20 storage nodes based edge network when gateway located at (260, 200).

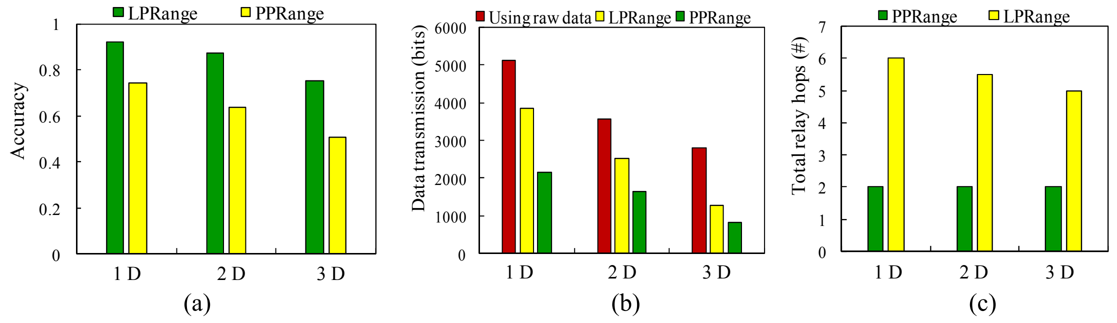


Fig. 9. Performance comparisons over multi dimensional queries using real-world data set Melbourne Urban Environments. (a) Comparison of average accuracies over 1-D (*temperature*), 2-D (*temperature+light*) and 3-D (*temperature+light+humidity*) range queries. (b) Comparisons of average data transmission over multi dimensional range queries. (c) Comparisons of average numbers of relay hops over multi dimensional range queries.

Fig. 9c, due to the proxy storage in PPRange.

VII. Conclusion

This paper addresses the qualities (such as energy efficiency, real-time response and privacy protection) of range query services over edge devices, and proposes two privacy-aware fuzzy range query schemes using fuzzy sets. Non-uniform range partitions, linguistic range variables and fuzzy overlap ratio labels are introduced respectively. Linguistic variables instead of raw sensory values are used for range query processing, which benefits privacy protection, real-time response and energy efficiency. Besides, a fuzzy range recovering mechanism as well as two distributed privacy-aware fuzzy range query algorithms are devised, which can provide good edge range query services in two common IoT application scenarios. Extensive evaluations validate our motivation in terms of reliability, real time, and energy efficiency. Though LPRange and PPRange are service-tailed for range queries, the idea of fuzzy data transformation and fuzzy information processing can be used by other information extraction applications.

In the future, we aim for systematic query processing in mobile edge networks that might confront sophisticated malicious attackers.

Acknowledgment

The authors are grateful to the anonymous reviewers and editors for their valuable suggestions, which have helped to improve the manuscript significantly.

References

[1] S. Khalifa, G. Lan, M. Hassan, A. Seneviratne, and S.K. Das, "HARKE: Human activity recognition from kinetic energy harvesting data in wearable devices," *IEEE Transactions On Mobile Computing*, Vol.17, No.6, pp.1353-1368, 2018.

[2] M.Z.A. Bhuiyan, G. Wang, J. Wu, J. Cao, X. Liu, and T. Wang, "Dependable structural health monitoring using wireless sensor networks," *IEEE Transactions On Dependable And Secure Computing*, Vol.14, No.4, pp. 1353-1368, 2017.

[3] M.G.R. Alam, M.S. Munir, M.Z. Uddin, M.S. Alam, T.N. Dang, and C.S. Hong, "Edge-of-things computing framework for cost-effective provisioning of healthcare data," *Journal of Parallel and Distributed Computing*, Vol.123, pp. 54-60, 2019.

[4] K. Wang, Y. Wang, Y. Sun, S. Guo, and J. Wu, "Green industrial Internet of Things architecture: An energy efficient perspective," *IEEE Communication Magazine*, Vol.54, No.12, pp.48-54, 2016.

[5] M. Mangia, F. Pareschi, R. Rovatti, and G. Setti, "Low-cost security of IoT sensor nodes with rakeness-based compressed sensing: Statistical and known-plaintext attacks," *IEEE Transactions on Information Forensics and Security*, Vol.13, No.2, pp.327-340, 2018.

[6] V.P. Illiano, L. Munoz-Gonzalez, and E.C. Lupu, "Don't fool Me!: Detection, characterisation and diagnosis of spoofed and masked events in wireless sensor networks," *IEEE Transactions On Dependable And Secure Computing*, Vol.14, No.3, pp.279-293, 2017.

[7] Z. Xiao , J.J. Yang, M. Huang, L. Ponnambalam , X. Fu, and R.S.M. Goh, "QLDS: A novel design scheme for trajectory privacy protection with utility guarantee in participatory sensing," *IEEE Transactions on Mobile Computing*, Vol.17, No.6, pp.1397-1410, 2018.

[8] J. Kang , R. Yu, X. Huang, and Y. Zhang, "Privacy-preserved pseudonym scheme for fog computing supported internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, Vol.19, No.8, pp.2627-2637, 2018.

[9] R. Xu, B. Palanisamy, and J. Joshi, "QueryGuard: Privacy-preserving latency-aware query optimization for edge computing," In *proc. of 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications / 12th IEEE International Conference On Big Data Science And Engineering*, 2018.

1063-6706 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY OF BRISTOL. Downloaded on March 30,2021 at 08:58:29 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2021.3059952, IEEE Transactions on Fuzzy Systems

[10] J. Xu, H. Lu, and H. Guting, "Range queries on multi-attribute trajectories," *IEEE Transactions on Knowledge and Data Engineering*, Vol.30, No.6, pp.1206-1211, 2018.

[11] A. Alnemari, C.J. Romanowski, and R.K. Raj, "An adaptive differential privacy algorithm for range queries over healthcare data," In *Proc. of IEEE International Conference on Healthcare Informatics*, 2017.

[12] S. Wu, Q. Li, G. Li, D. Yuan, X. Yuan, and C. Wang," ServeDB: secure, verifiable, and efficient range queries on outsourced database,"In *Proc. of IEEE 35th International Conference on Data Engineering*(IEEE ICDE) 2019.

[13] H. Zhu, X. Yang, B. Wang, and W.C. Lee, "Range-based nearest neighbor queries with complex-shaped obstacles," *IEEE Transactions on Knowledge and Data Engineering*, Vol.30, No.5 pp.963-977, 2018.

[14] B. Hore, S. Mehrotra, M. Canim, and M. Kantarcioglu, "Secure multidimensional range queries over outsourced data," *The VLDB Journal*, Vol.21, No.3, pp.333-358, 2012.

[15] R. Lu, "A new communication-efficient privacy-preserving range query scheme in fog-enhanced IoT," *IEEE Internet of Things Journal*, Vol.6, No.2, pp.2497-2505, 2019.

[16] B. Sheng, and Q. Li, "Verifiable privacy-preserving range query in two-tiered sensor networks," In *Proc. of the 27th IEEE International Conference on Computer Communications* (IEEE INFOCOM), 2008.

[17] J. Shi, R. Zhang, and Y. Zhang, "A spatiotemporal approach for secure range queries in tiered sensor networks," *IEEE Transactions on Wireless Communications*, Vol.10, pp.264-273, 2011.

[18] Q. Kong, R. Lu, M. Ma, H. Bao, "Achieve location privacy-preserving range query in vehicular sensing," *Sensors*, Vol. 17, No. 1829, pp.1-16, 2017.

[19] F. Chen, and A.X. Liu, "Privacy and integrity-preserving range queries in sensor networks," *IEEE/ACM Transaction on Networks*, Vol.20, No.6, pp.1774-1787, 2012.

[20] P. Chen, W. Zeng, Y. Zhu, and Y. Gu, "Secure range query based on privacy-preserving function in two-tiered sensor networks," In *Proc. of International Conference on Cyber Security of Smart Cities*, 2016.

[21] D. Bogatov, G. Kollios, and L. Reyzin, "A Comparative Evaluation of Order-Revealing Encryption Schemes and Secure Range-Query Protocols," *PVLDB*, Vol.12, No.8, pp.933-947, 2019.

[22] G. Cormode, T. Kulkarni, D. Srivastava, "Building Hierarchical Histograms Under Local Differential Privacy," *PVLDB*, Vol.12, No.10, pp.1126-1138, 2019.

[23] B. Wang, M. Li, and H. Wang, "Geometric range search on encrypted spatial data," *IEEE Trans. Inf. Forensics Secur.*, Vol.11, No.4, pp.704-719, 2016.

[24] Y. Yi, R. Li, F. Chen, A.X. Liu, and Y. Lin, "A digital watermarking approach to secure and precise range query processing in sensor networks," In *Proc. of the 32nd IEEE Conference on Computer Communications* (INFOCOM), 2013.

[25] X. Zhang, L. Dong, H. Peng, H. Chen, D. Li, and C. Li, "Achieving efficient and secure range query in two-tiered wireless sensor networks," In *Proc. of the IEEE/ACM International Symposium on Quality of Service*, 2014.

[26] J. Zeng, L. Dong, Y. Wu, H. Chen, C. Li, and S. Wang, "Privacy-Preserving and Multi-Dimensional Range Query in Two-Tiered Wireless Sensor Networks," In *Proc of GLOBECOM*, 2017.

[27] L. Dong, X. Chen, J. Zhu, H. Chen, K. Wang, and C. Li, "A secure collusion-aware and probability-aware range query processing in tiered sensor networks," In *Proc. of the IEEE 34th Symposium on Reliable Distributed Systems* (IEEE SRDS), 2015.

[28] https://en.wikipedia.org/wiki/Fuzzy_set

[29] X.X. Zhang, L.R. Zhao, H.X. Li, and S.W. Ma, "A novel three-dimensional fuzzy modeling method for nonlinear distributed parameter systems," *IEEE Transactions on Fuzzy Systems*,Vol.27, No.3, pp.489-501, 2019

[30] J.A. Morente-Molinera, J. Mezei, C. Carlsson, and E. Herrera-Viedma, "Improving supervised learning classification methods using multigranular linguistic modeling and fuzzy entropy," *IEEE Transactions on Fuzzy Systems*,Vol.25, No.5, pp.1078-1089, 2017.

[31] Y. Li, H. Chen, M. Lv, Y. Li, and Y. Li, "Extracting semantic event information from distributed sensing devices using fuzzy sets," *Fuzzy Sets and Systems*, Vol.337, pp.74-92, 2018.

[32] C. Zhang, J. Hu ; J. Qiu, W. Yang, H. Sun, and Q. Chen, "A novel fuzzy observer-based steering control approach for path tracking in autonomous vehicles," *IEEE Transactions on Fuzzy Systems*, Vol.27, No.2, pp.278-290, 2018.

[33] H. Zhang, J. Han, Y. Wang, and X. Liu, "Sensor fault estimation of switched fuzzy systems with unknown input," *IEEE Transactions on Fuzzy Systems*, Vol.26, No.3, pp.1114-1124, 2018.

[34] S. Madden, M.J. Franlin, J.M. Hellerstein, and W. Hong, "TAG: A tiny aggregation service for ad-Hoc sensor networks," In *Proc. of the 5th symposium on Operating Systems Design and Implementation*, 2002.

[35] M. Perron, and P. Sura, "Climatology of non-Gaussian atmospheric statistics," *Journal of Climate*, Vol.26, No.3, pp.1063C1083, 2013.

[36] J.A. Buchmann, *Introduction to Cryptography*, 2nd ed., Springer Science & Business Media, 2013, pp. 190C191.

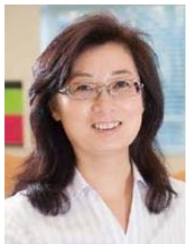
[37] G. Anastasi, M. Conti, M.D. Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Networks*, Vol.7, No.3, pp.537C568, 2009.

[38] OMNET++, http://www.omnetpp.org, last accessed 10/9/2017.

[39] LUCE, http://sensorscope.epfl.ch/index.php/Environmental_Data, last accessed: May 7, 2011.

[40] Melbourne Urban Environments Data Set, https://data.melbourne.vic.gov.au/Environment/Sensor-readings-with-temperature-light-humidity-ev/ez6b-syvw/data, last accessed: Nov. 23, 2018.

Yinglong Li received the PhD degree in computer science from Renmin University of China, Beijing, China, in 2014, and the BS and MS degrees in computer science from Jiangxi Normal University, Nanchang, China, in 2003 and 2006 respectively. Currently, he is an associate professor in school of computer science and technology, Zhejiang University of Technology, as well as an academic visitor in University of Bristol, UK. His research interests include intelligent edge-computing and privacy protection in Internet of Things (IoT).



Weiru Liu holds Chair of Artificial Intelligence (AI) at the University of Bristol, and is the Associate Dean for TQEC (Research and Enterprise), following her role as the Engineering Faculty Research Director between August 2017-March 2020. She is a member of UK EPSRC ICT Strategic Advisory Team (SAT) and a Co-Director for the EPSRC Centre for Doctoral Training in Future Autonomous and Robotic Systems: Towards Ubiquity (FARSCOPE-TU), 2019-2028. Prior to joining the University of Bristol in 2017, she held the Chair of AI at Queen's University Belfast (QUB), and was the Director of Research for the Knowledge and Data Engineering Research Cluster for 6 years. Her research interests include: data-driven intelligent autonomous systems; event modelling, reasoning and correlation in uncertain environments in sensor networks; and information fusion under uncertainty, with a wide range of applications such as security, healthcare, robotics. She has published over 200 peer-reviewed papers, and chaired several international conferences.

Yihua Zhu received his BS degree in mathematics from Zhejiang Normal University, Zhejiang, China, in 1982; his MS degree in operation research and cybernetics from Shanghai University, Shanghai, China in 1993; and his PhD degree in computer science and technology from Zhejiang University, Zhejiang, China, in 2003. He is a professor at Zhejiang University of Technology, Hangzhou, China. He is a member of China Computer Federation Technical Committee on Sensor Network. His current research interests include information dissemination, stochastic modeling and analysis, power management, mobility management for wireless networks, and network coding. He has served as technical program committee members in the international conferences ICC, WCNC, GlobeCom, etc. He is the recipient of the Best Paper Award of Chinacom 2008. He has published more than 130 research papers in proceedings and journals including IEEE Transactions on Wireless Communications, IEEE Transactions on Vehicular Technology, etc..

Authorized licensed use limited to: UNIVERSITY OF BRISTOL. Downloaded on March 30,2021 at 08:58:29 UTC from IEEE Xplore. Restrictions apply.