Differential Private Spatial Decomposition and Location Publishing Based on Unbalanced Quadtree Partition Algorithm

YAN YAN¹,², XIN GAO¹, ADNAN MAHMOOD², TAO FENG¹, AND PENGSHOU XIE¹
¹School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China
²Department of Computing, Faculty of Science and Engineering, Macquarie University, Sydney, NSW 2109, Australia
Corresponding author: Yan Yan (yanyan@lut.edu.cn)

This work was supported in part by the National Nature Science Foundation of China under Grant 61762059, Grant 61762060, and Grant 61862040, and in part by the China Scholarship Council under Grant 201808625040.

ABSTRACT Centralized publishing of big location data provides great convenience for various location-based interactive queries and services. Privacy protection of users’ location information is an indispensable issue in the security of big data applications. Partition publishing is an effective way to release statistical information of two-dimensional big location data. By combining with the differential privacy model, it can provide more accurate range counting query service on the premise of ensuring location privacy. In order to further improve the availability of location data subsequent to centralized publishing, this paper analyzes the primary noise sources of partition publishing and discusses the constraints among publishing errors, the spatial partition structure, and privacy budget allocation. An unbalanced quadtree partition algorithm based on regional uniformity is proposed. Accordingly, the gradient privacy budget allocation scheme and adjustment method are designed to ensure the effectiveness of the differential privacy model. Experimental comparison of the real-world datasets proves the advantages of the proposed algorithm in improving the querying accuracy of the published data.

INDEX TERMS Privacy preserving data publishing; location privacy; private spatial decomposition; differential privacy; unbalanced quadtree partition; gradient budget allocation.

I. INTRODUCTION
Centralized publishing of big location data primarily facilitates in providing a variety of location-based interactive queries and services. Massive sensors and intelligent terminals continuously collect all kinds of location data. After the integration, analysis, and processing of the centralized publishing platform, the statistics of location data are published according to different needs and patterns for scientific research, decision support, and public service. This kind of centralized publishing reflects the value of location information in a number of application domains, including but not limited to, intelligent transportation systems, location-based services, and location advertisement push [1], [2]. However, location information demands strong privacy as malicious attackers may use big data analysis software to collect, infer, and analyze the location information of specific users (e.g., where he has visited? what is his current location or even future trajectory?), thereby, resulting in the leakage of personal privacy information such as living habits, health status, hobbies, and economic conditions. In some serious cases, it may even endanger peoples’ property and life safety [3]–[5].

Partition publishing is widely used in location-based big data applications and provides the statistical information of location so that the users can query a number of other users within a certain geographic area or learn about the traffic conditions. Partition publishing of big location data only involves the statistical number of users in the region and hence the leakage risk of users’ real location is considerably reduced. By employing the differential privacy protection model [6], [7] to incorporate noise to the statistical values, the privacy preserving data publishing effect could be further improved. The publishing error of partition method (i.e., the difference between the actual statistical results and published results in a query area) directly affects the availability...
of published data. For instance, the real number of taxis in a certain area is 6, whereas, the published result depicts this as 15. Excessive errors may cause users to spend a lot of time waiting for the taxis that do not actually exist, subsequently leading to a serious decline of location-based service quality. Therefore, on the premise of not impairing the effect of location privacy protection, improving the accuracy of published data to a considerable extent is an important issue which has a direct impact on the big location data applications.

In fact, the publishing error of big location data transpires due to the error introduced by the noise disturbance as well as from the error caused by the uniform assumption of data distribution by the querying algorithm. The above errors are closely related to the partition structure and privacy budget allocation method. In order to explain the correlation between them in an intuitive manner, let’s consider the following instance with a grid-based partition structure. As depicted in Figure 1, the real statistical value of the region $P_5$ is 1. If a user submits a range counting query with a region of $Q_1$ (i.e., the red dashed box), it is easy to directly expose the location privacy corresponding to this particular region. However, the statistical result after incorporating noise disturbance based on the differential privacy model tends to be 2.69 which has a similar $K$-anonymous protection effect to the corresponding user ($K = 2.69$). Generally, the more detailed the partition structure is segregated and the larger the allocated differential privacy budget is (the smaller the corresponding noise disturbance), the more accurate is the range counting query result provided to the user. However, if the differential privacy noise is too small, the effect of location privacy protection cannot be achieved. For instance, the counting query in region $Q_1$ cannot protect the existence of the location point when the differential privacy noise drops to 0.1. If the spatial structure is over-partitioned, a large number of empty nodes (i.e., areas with a real count value of 0) would be generated. At this time, adding noise disturbance would introduce large noise error. For example, the real statistical value in region $Q_2$ (i.e., the blue dashed box) in Figure 1 is 1, whereas, the query result after incorporating differential privacy noise is 4.24 which has a large error and reduces the availability of published data.

Since the two-dimensional space cannot be accurately divided into a single location point, the partition method often needs to construct a certain spatial structure and assumes that the location point distributes uniformly in the sub-regions. The range counting query error caused by the uniform hypothesis estimation is referred to as the uniform assumption error and its size is closely related to the true distribution of location points and partition structure. If the real distribution of location point is relatively uniform, an excessive fine spatial partition structure would increase the noise error. On the other hand, if the real distribution of location point is not uniform, an excessive coarse spatial division would cause a large uniform assumption error. Figure 2(a) portrays a uniformly distributed area. The returned result is equal to $900 + N = 100 + N$ when the query area is $Q$. After further partition as depicted in Figure 2(b), the same query region returns a result of $100 + N$. In this example, detailed partition causes the increase of noise error. The unbalancedly distributed area and query region as shown in Figure 2(c) returns a result of $100 + N$ in the case of detailed division, whereas, the rough structure shown in Figure 2(d) returns a result of $100 + N = 100 + N$ within the same query region. In this case, the rough partition generates large uniform assumption errors.

The above examples fully demonstrate the impact of the spatial partition structure and privacy budget allocation on the availability of the published location data. To address these
issues, we design an unbalanced quadtree partition method and corresponding privacy budget allocation strategy to balance the noise error and uniform assumption error, which in turn helps to enhance the utility of published location data. The salient contributions of this paper are as follows:

- In order to balance the noise error and uniform assumption error that may occur during the partition process, we propose an unbalanced quadtree partition structure based on the regional uniformity condition and design the corresponding partition algorithm. According to the distribution density of the location information, the iterative partition can be carried out adaptively to reduce the negative effects caused by empty nodes and over-partition.

- In order to further improve the querying accuracy of the published location data, we design a new privacy budget allocation scheme for the unbalanced quadtree partition structure. First, the geometric privacy budget allocation is performed according to the quadtree structure from the vertical direction. Subsequently, the privacy budget is adjusted for each node on the same layer according to their regional density in order to achieve a reasonable privacy budget allocation result.

- We conduct extensive experimental studies to demonstrate the effectiveness of our proposed unbalanced quadtree partition and budget allocation method in balancing the noise error and uniform assumption error as well as enhancing accuracy of counting queries and utility of released data.

The rest of the paper is organized as follows. Section II reviews state-of-the-art pertinent to private spatial decomposition methods. Section III delineates on some basic introduction of interactive query service of centralized data publishing and outlines fundamental definitions of differential privacy. Section IV details the unbalanced quadtree partition method and privacy allocation strategy. Section V reports a set of empirical studies and the results, whereas, Section VI concludes the paper.

II. RELATED WORK

A brief glimpse of the literature reveals that over the past few years, numerous methods have been proposed to protect the privacy of location information [8]–[13]. The privacy partition and publishing method focuses on improving the querying accuracy of published location data without compromising the user privacy. According to the two-dimensional characteristics of location information, the partition and publishing method usually adopts a grid structure or a tree structure. The Uniform Grid (UG) method proposed by Qardaji et al. [14] segregates the two-dimensional space into \(m \times m\) grids and incorporates differential privacy noise to the statistical results with the same budget in each grid. Although the UG algorithm is highly efficient, it does not consider the real distribution of location points which results in a large uniform assumption error. Wang et al. [15] employs a linear regression method to obtain the optimal solution for partition granularity based on the UG algorithm. The unit-merging strategy based on bucket sorting is used to allocate all the similar units into one partition which reduces the noise error by decreasing the noise incorporated to each unit. The Adaptive Grid (AG) method [14] first performs the UG partition process with the privacy budget of \(\alpha \epsilon\), where, \(\alpha\) is a scale factor within \([0,1]\) and \(\epsilon\) is the total privacy budget. Subsequently, each grid is adaptively segregated into \(m_2 \times m_2\) fine-grained units according to its density and finally the differential noise with the privacy budget of \((1 - \alpha)\epsilon\) is incorporated to each unit. The AG algorithm takes into account the influence of data distribution characteristics on the partitioned area, nevertheless, there is still room for reducing the noise error and uniform assumption error. Xiong et al. [16] uses contour plot to characterize the location distribution in spatial crowdsourcing which firstly partitions the entire area into some disjoint cells and then connects the cells with the same noisy count to form a larger region. Hien et al. [13] extends the AG method to achieve effective spatial crowdsourcing services while offering privacy guarantees to the workers. Zhou et al. [17] defines the intensity of privacy-preserving needs based on the divergence of each grid and proposes a differential privacy noise dynamic allocation algorithm based on the AG method.

The tree-based partition method can be classified into two categories, i.e., data-dependent partition and data-independent partition. The quadtree structure is the most common data-independent structure used to segregate two-dimensional space. The Quad-opt method proposed by Cormode et al. [18] employs a full quadtree structure and proposes the geometric privacy budget allocation scheme and post-processing method to improve the range counting querying accuracy of the published data. However, the independent partition structure still has the problem of large uniform assumption error. Wang et al. [19] proposes an unbalanced privacy budget allocation method which uses the Fibonacci series to allocate different privacy budgets to different nodes on each layer of the quadtree structure in order to obtain higher querying accuracy. The Quad-heu method proposed by Wu et al. [20] first carries out a h-layer full quadtree partition according to [18] and then performs the bottom-up adjustment and combination according to certain uniformity judgment to reduce the uniform assumption error and improve querying accuracy. The above data-independent partition methods primarily obtain a more reasonable partition level for quadtree structure (tree depth \(h\)) based on the experimental results. This method of determining the partition structure based on the posterior results severely limits the accuracy of privacy budget allocation and cannot meet the requirements of dynamic release of big location data. Zhang et al. [21] proposes the PrivTree method to decide whether to perform the quadtree partition by introducing a controllable deviation, thereby, eliminating the requirement of the pre-defined partition depth. However, it cannot obtain an ideal partition result on the actual big location
dataset of large volume and the accuracy of range query is limited. Data-dependent partition structures, Kd-trees, and R-trees [18] split nodes based on the actual distribution of location points. The Kd-tree partition method proposed by Inan et al. [22] reduces the problem of uniform assumption error. Nevertheless, the Kd-tree structure easily exposes the privacy information of the location points on the division line and, therefore, it is necessary to spend a part of the privacy budget to protect the median point. The Kd-hybrid partition method proposed by Cormode et al. [18] first performs some layers of partition via the data-dependent Kd-tree method and then further divide the results using the data-independent quadtree method, thereby, proving to have better range counting querying accuracy.

To address the location privacy concerns of workers’ during the task allocation in mobile crowdsensing, Yang et al. [23] proposes a partition method based on workers’ density. The proposed density-based partition method recursively segregates a region into four sub-cells with the maximum density difference from some initial partition points. The hybrid partition strategy proposed by Yan et al. [24] first performs adaptive meshing based on the distribution density of location points and then classifies the meshes based on the high-density thresholds and low-density thresholds. For the sparse areas, partition would be stopped to avoid introducing excessive noise errors. On the contrary, for dense areas, adaptive grid partition would be performed to improve publishing accuracy. Moreover, for areas with skewed distribution characteristics, the heuristic quadtree partition would be carried out to improve the availability of published data.

### III. PRELIMINARIES

For the statistical release of census information and medical data, the differential private dataset is sufficient to prevent the leakage of user’s existence in the published data [6], [7]. Different from these kinds of static big data, location data is not only large in scale but also fast in production and frequent in its update. Therefore, the release and use of big location data often transpires in the form of interactive query services, i.e., the user submits a query request (e.g., how many taxis are there in a range of 1 km from me?) to the data platform based on his location and the platform replies the query with differential statistical information pertinent to a specific area according to the real-time location dataset. Partition publishing method combined with differential privacy model is used to provide range counting query service under the premise of ensuring user’s location privacy. The publishing process and interactive querying service of centralized big location data is depicted in Figure 3. Terminals, devices, sensors, vehicles, and users from the Mobile Internet, Internet of Vehicles, Internet of Things, Sensor Networks, Social Networks, etc., constantly send location data in various forms, structures, and qualities to the centralized publishing platform. This process uses an honest model which assumes that all the data is sent to a reliable data publisher. In order to protect the privacy of location data generator, the identifier attribute that can uniquely identify the user is deleted to achieve anonymization of the original location data. However, there may be malicious users (i.e., attackers) among a large number of data receivers and users. By collecting big location data, leveraging advanced big data analysis and mining tools, and comprehensively employing the background knowledge obtained from other channels, attackers may identify specific users from the published data or even further obtain the users’ privacy information. Therefore, it is necessary to incorporate random perturbation to the released location statistical information with the help of the differential privacy model so that the attackers cannot identify a particular user regardless of the background knowledge.

**Definition 1 (ε-Differential Privacy):** For the sibling datasets, \( T_1 \) and \( T_2 \) (with only one different record), and any output, \( S \subseteq \text{Range}(K) \), if a query result obtained by the privacy protection algorithm \( K \) satisfies Eq. (1):

\[
\frac{\Pr[K(T_1) = S]}{\Pr[K(T_2) = S]} \leq e^\varepsilon
\]
then, the algorithm $K$ is said to satisfy $\varepsilon$ differential privacy [6], [7].

Eq. (1) indicates that by using $T_1$ and $T_2$ as the input of the privacy protection algorithm $K$, the probability of obtaining the same query result $S$ is very close. Therefore, it is difficult to determine whether a record belongs to $T_1$ or $T_2$ by observing the output of the publishing algorithm, thereby, providing privacy protection. The smaller the privacy budget $\varepsilon$ is, the lower is the probability of distinguishing the two datasets and the higher the degree of privacy protection. The differential privacy model has a natural matching with the release and protection requirements of centralized big location data. Its strict mathematical definition guarantees that the output probability distribution of the differential privacy protection algorithm is almost unchanged regardless whether a record $t$ exists in the dataset $T$. Therefore, even if the attacker obtains all sensitive data except the record $t$, he still cannot judge whether the record exists in the original dataset based on the output result.

Definition 2 (Laplace Noise Mechanism): Laplace noise can be added to achieve differential privacy protection of published data. The query request on the original dataset $T$ can be regarded as the value of a function $f$ acting on $T$. Therefore, the query result can be expressed as $f(T)$. In order to achieve $\varepsilon$-Differential Privacy protection, the output of the random algorithm $K$ is $K(T) = f(T) + \eta$, where, $\eta$ is a continuous random variable satisfying the Laplace distribution and its probability density function can be expressed as Eq. (2):

$$p(\eta) = \frac{1}{2\lambda}e^{-|\eta|/\lambda} \quad (2)$$

Theorem 1 (Sequential Composition [25]): Suppose that there is a set of algorithms, $\{K_1, K_2, \ldots, K_n\}$, wherein $K_i (1 \leq i \leq n)$ satisfies the $\varepsilon_i$ differential privacy on the dataset $T$. Then, the combination algorithm of the set of algorithms, $\{K_1, K_2, \ldots, K_n\}$, can achieve $\sum_{i=1}^{n} \varepsilon_i$ differential privacy on the same dataset $T$.

Theorem 2 (Parallel Composition [25]): Let's assume that the dataset $T$ can be divided into $n$ independent and disjointed subsets, $\{T_1, T_2, \ldots, T_n\}$. There is a set of random algorithms, $\{K_1, K_2, \ldots, K_n\}$, applied to the above subsets respectively, wherein $K_i (1 \leq i \leq n)$ satisfies the $\varepsilon$ differential privacy on the subsets $T_i$. Then, the sequence of $K_i$ provides $\varepsilon$ differential privacy on the dataset $T$.

Definition 3 (Noise Error): After incorporating noise to each area of the partitioned structure according to the Laplace noise mechanism, the difference between the original statistical result and published result is referred to as the noise error. Using $Q$ to represent the query area, $C(Q)$ and $C(Q^*)$ represent the original and noisy statistical value within this area, the noise error can be represented as per Eq. (3) as:

$$\text{NoiseErr} (Q) = |C(Q) - C(Q^*)| \quad (3)$$

Definition 4 (Uniform Assumption Error): Let’s suppose that $P_i (i = 1, 2, \ldots, m)$ is the sub-region intersected with the query region $Q$ in the partition structure and $r_i (i = 1, 2, \ldots, m)$ represents the proportion of the intersected area between query region $Q$ and the above subregions. Then, the uniform assumption error in query area $Q$ can be calculated from Eq. (4) as:

$$\text{UniAssErr} (Q) = \left| \sum_{i=1}^{m} r_i \cdot C(P_i) - C(Q) \right| \quad (4)$$

IV. PARTITIONING AND PUBLISHING OF BIG LOCATION DATA BASED ON UNBALANCED QUADTREE

When splitting the two-dimensional space from top to bottom, it is important to determine the conditions under which segmentation stops. Too many partition levels would generate a lot of empty nodes, thereby, introducing excessive noise errors. On the contrary, insufficient partition levels would reduce the accuracy of range counting query and lose the availability of published data. Therefore, the optimal space partition method should adaptively adjust the segmentation depth of the structure according to the distribution characteristics of the location points. The overall number of empty nodes should be minimized in order to avoid excessive noise errors.

A. UNBALANCED QUADTREE PARTITION ALGORITHM

In the full quadtree partition process, the original area is recursively divided into four equal regions via horizontal and vertical lines through the midpoint of each range. This is referred to as the data independent decomposition because the partition line would not reveal the exact median of the current area. However, the full quadtree partition method requires specifying the partition depth of the tree structure in advance. This kind of requirement is unrealistic for the data publishing system that treats different data sizes and characteristics. In this section, we propose the unbalanced quadtree partition method (refer to Algorithm 1). It employs the average split method of the full quadtree to the region, and, therefore, no extra budget needs to be allocated to protect the privacy of the partition line. Whether the current area needs to be partitioned is determined by its distribution uniformity. The four sub-areas are traversed according to the depth-first principle and the above judgment and partition process is repeated until the stop condition is satisfied. Figure 4 depicts the unbalanced quadtree partition results of the location dataset, Storage.\(^1\)

The stop condition for the unbalanced quadtree partition primarily include the following three situations:

- If the region is empty (i.e., no location point exist within the cell), the partition progress would stop in order to prevent incorporating too much noise;
- If the region is too small (i.e., less than the physical range of counting queries that users may put forward), the partition progress should also be stopped in order to avoid over-partition; and

\(^1\)http://www.infochimps.com/datasets/storage-facilities-by-neighborhood-2
The two-dimensional space covered by the big location points in two-dimensional space in a reasonable partition. In order to evaluate the uniformity distribution of one of the conditions for stopping the unbalanced quadtree partition structure, we learn from the method of digital error and uniform assumption error consequently improving the utility of the published data. Therefore, we take the uniformity of the local area as the primary of spatial partition and the utility of the published data. Therefore, we take the uniformity of the local area as the primary of spatial partition and one of the conditions for stopping the unbalanced quadtree partition. In order to evaluate the uniformity distribution of location points in two-dimensional space in a reasonable and accurate manner, we learn from the method of digital image processing and decompose the region from multiple directions such as vertical, horizontal, and diagonal. Regional uniformity can be judged from the following definition.

**Definition 5 (Regional Uniformity):** For a given region $R$, let $C(D_1), C(D_2), \ldots, C(D_i)$ represent the statistical value of location points in the sub-areas after multiple direction partition. Set $\bar{C} = \frac{\sum_{i=1}^{m} C(D_i)}{m}$ to be the average statistical value of region $R$. If the ratio of the sum of absolute errors to the statistical value of the region satisfies Eq. (5), region $R$ can be judged as uniformly distributed.

$$\frac{\sum_{i=1}^{m} |C(D_i) - \bar{C}|}{\sum_{i=1}^{m} C(D_i)} \leq \theta$$

where, $i$ is the number of sub-areas after multiple direction partition and $\theta$ is the threshold.

**Algorithm 1 Unbalanced Quadtree Partition (UBQP)**

**Require:** Big location dataset ($T$); Regional uniformity threshold ($\theta$); and Minimum region size ($MinSize$).

**Ensure:** Depth of the quadtree ($h$); Spatial decomposition structure ($SDS$).

1: $h = 0$, $SDS = \Phi$
2: Current node $\leftarrow T$
3: Count $\leftarrow$ number of location points within the current node
4: $R \leftarrow$ size of the current node
5: if Count $\leq 1$ or $R \leq MinSize$ then
6:   $SDS = SDS \cup$ Current node
7: else
8:   if Regional uniformity of current node $\leq \theta$ then
9:     $SDS = SDS \cup$ Current node
10: else
11:   Carry out quadtree partition on the current node and separate current node into four sub-regions, $T = \{T_1, T_2, T_3, T_4\}$
12:   $h = h + 1$
13:   $SDS = SDS \cup \{T_1, T_2, T_3, T_4\}$
14:   for $T_i \in T$ do
15:     Current node $\leftarrow T_i$
16:     Go to step 3
17:   end for
18: end if
19: end if
20: return $h$, $SDS$

FIGURE 4. Unbalanced quadtree partition of Storage dataset.

- If the distribution of location point in a certain region is already uniform, the stop condition is also met. As analyzed in the Introduction section, if the regional distribution is uniform, then excessive fine partition structure would only increase the noise error.

As we analyzed in the introduction section, the uniformity of the regional distribution promotes a balance between noise error and uniform assumption error consequently improving the utility of the published data. Therefore, we take the uniformity of the local area as the primary of spatial partition and one of the conditions for stopping the unbalanced quadtree partition. In order to evaluate the uniformity distribution of location points in two-dimensional space in a reasonable and accurate manner, we learn from the method of digital image processing and decompose the region from multiple directions such as vertical, horizontal, and diagonal. Regional uniformity can be judged from the following definition.

**Definition 5 (Regional Uniformity):** For a given region $R$, let $C(D_1), C(D_2), \ldots, C(D_i)$ represent the statistical value of location points in the sub-areas after multiple direction partition. Set $\bar{C} = \frac{\sum_{i=1}^{m} C(D_i)}{m}$ to be the average statistical value of region $R$. If the ratio of the sum of absolute errors to the statistical value of the region satisfies Eq. (5), region $R$ can be judged as uniformly distributed.

$$\frac{\sum_{i=1}^{m} |C(D_i) - \bar{C}|}{\sum_{i=1}^{m} C(D_i)} \leq \theta$$

where, $i$ is the number of sub-areas after multiple direction partition and $\theta$ is the threshold.

**B. PRIVACY BUDGET ALLOCATION METHOD**

The two-dimensional space covered by the big location dataset $T$ can be divided into $n$ independent and non-intersected subsets, $\{T_1, T_2, \ldots, T_n\}$, by employing the above unbalanced quadtree partition method. Each subset, $T_i$ (where, $1 \leq i \leq n$) corresponds to a node in the unbalanced quadtree partition structure and satisfies $\forall_{i \neq j} T_i \cap T_j = \emptyset$, $\bigcup_{i=1}^{n} T_i = T$. According to the sequential composition and parallel composition characteristics (Theorem 1 and Theorem 2) of the differential privacy model, the simplest and most straightforward way to achieve $\varepsilon$ differential privacy protection on such a partition structure is to incorporate $\varepsilon$ budget Laplace noise to the statistical value of each subset (parallel composition). However, as the depth of partition increases, the querying accuracy of equal distribution of differential privacy budget would deteriorate significantly [18].

Cormode et al. [18] designed the geometric privacy budget allocation strategy for the full quadtree partition structure and proved that the querying accuracy of the geometric allocation method is better than the equal allocation method under the same partition level. Inspired by the geometric privacy allocation strategy, when the privacy budget increases from the root node to the leaf node on a layer-by-layer basis, the querying accuracy of the leaf node receives the optimal value. Therefore, we designed a sequence with increasing differences and applied it to the privacy allocation of the tree structure. What we have to point out is that the geometric allocation method is suitable for the full quadtree partition
structure with four equal child-nodes in every sub-tree. However, in the proposed unbalanced quadtree partition structure, some areas have not been further segregated primarily owing to their distribution characteristics and/or size. Therefore, it is necessary to adjust the privacy budget of these nodes without further segregation to ensure the effectiveness of the differential privacy protection model.

Definition 6 (Gradient Allocation): Suppose \( A = \{a_1, a_2, \ldots, a_n\} \) is a sequence with increasing differences with its generic term expressed as \( a_n = n(n + 1)/2 \). For the tree-based partition structure with depth \( h \) and total privacy budget \( \varepsilon \), the gradient privacy budget allocated to each layer can be ascertained via Eq. (6):

\[
\varepsilon_i = \frac{a_i}{\sum_{j=1}^{h+1} a_i} \varepsilon
\]  

wherein, \( i = 1, 2, \ldots, h + 1 \) indicates the partition depth of the tree structure, \( \varepsilon_1 \) denotes the gradient privacy budget allocated to the root node, and \( \varepsilon_{h+1} \) represents the gradient privacy budget of the deepest leaf node.

Considering that not all regions have equal partition depth in the unbalanced quadtree partition structure, some adjustment strategies should be adopted once the gradient privacy budget has been allocated. For the node, \( \varepsilon_i \), that meets the uniformity condition and stops partition, its privacy budget could be set in accordance to Eq. (7):

\[
\varepsilon_i^* = \sum_{j=i}^{h+1} \varepsilon_j
\]

Figure 5 depicts the schematic diagram of the unbalanced quadtree partition structure and privacy budget allocation portrayed in Figure 4. The black nodes represent the sub-regions that meet the uniformity condition and stop further partition.

According to the budget adjustment method, \( \varepsilon_4^* = \sum_{j=4}^{5} \varepsilon_j = \varepsilon_4 + \varepsilon_5 \), and \( \varepsilon_3^* = \sum_{j=3}^{5} \varepsilon_j = \varepsilon_3 + \varepsilon_4 + \varepsilon_5 \).

The above gradient privacy budget allocation strategy and adjustment method guarantee that each path from the leaf to the root node in the unbalanced quadtree partition structure satisfies \( \sum_{i=1}^{h+1} \varepsilon_i = \varepsilon \) differential privacy protection.

Algorithm 2 Differential Privacy Location Data Release Based on UBQP

Require: Depth of the quadtree \( (h) \); Spatial decomposition structure \( (SDS) \); Privacy budget \( (\varepsilon) \)
Ensure: Publishing dataset \( (T^*) \)

1: \( \) for \( i = 1 \) to \( h + 1 \) do
2: \( \varepsilon_i = \frac{\varepsilon}{\sum_{i=1}^{h+1} a_i} \times a_i \)
3: \( \) end for
4: \( \) for \( cell_i \in SDS \) do
5: \( \) if \( cell_i \) is not a partitioned node then
6: \( \varepsilon_i^* = \sum_{j=i}^{h+1} \varepsilon_j \)
7: \( \) end if
8: \( \) end for
9: \( \) for \( cell_i \in SDS \) do
10: \( \) \( Count_i \leftarrow \) number of location points within \( cell_i \)
11: \( \) \( N_i = Count_i + \text{Laplace} \left( \frac{\varepsilon_i^*}{\varepsilon_i^*} \right) \)
12: \( \) end for
13: \( \) return publishing dataset \( T^* = \{N_1, N_2, \ldots, N_m\} \)

C. PRIVACY PRESERVING PUBLISHING ALGORITHM FOR BIG LOCATION DATA

The differential privacy protection model and the statistical release of centralized big location data have the natural matching characteristics. The strict mathematical definition of the differential privacy model guarantees that the output probability distribution of the publishing algorithm is almost unchanged regardless of whether a user’s location record exists in the publishing data or not. Therefore, even if the malicious attacker has a large amount of background knowledge and obtained all other records except the user ones, he still cannot determine whether the user is in the querying area based on the range counting query results. Algorithm 2 depicts the big location data statistics information publishing process to achieve the above location privacy protection.

Theorem 3: By employing the proposed unbalanced quadtree partition method and gradient privacy budget allocation and adjustment scheme, for any query range \( Q \) proposed by the user, the published statistical information of big location data can provide \( \varepsilon \) differential privacy protection.
Proof: For the counting query within an arbitrary range \( Q \) proposed by the user, the following two situations are generally included:

Situation (1): The querying range \( Q \) is located within a certain node’s area after the unbalanced quadtree partition.

Let \( \epsilon \) represent the privacy protection intensity of this node’s area and assume that the node is located in the layer \( l \) of the unbalanced quadtree partition structure (\( l = 1 \) represents the root node and \( l = h + 1 \) represents the leaf node). According to the aggregation characteristics of differential privacy model (Theorem 1), the privacy protection intensity of this node is \( \epsilon Q = \sum_{i=1}^{l} \epsilon j \). If the layer \( i \) is not quadtree partitioned, then according to the privacy budget adjustment scheme, the privacy protection intensity of this node is \( \epsilon i = \sum_{j=1}^{l} \epsilon j = \epsilon_j \), and \( \epsilon Q = \sum_{i=1}^{l} \epsilon i = \epsilon_1 + \epsilon_2 + \cdots + \epsilon_l = \epsilon \), such that \( \epsilon Q = \epsilon \). If the region is quadtree partitioned in layer \( i \), then according to the gradient privacy budget allocation scheme, the privacy protection intensity of this node is \( \epsilon i = \sum_{j=1}^{l} \epsilon j = \epsilon_j \), and \( \epsilon Q = \sum_{i=1}^{l} \epsilon i = \epsilon_1 + \epsilon_2 + \cdots + \epsilon_l = \epsilon \), such that \( \epsilon Q = \epsilon \).

Situation (2): The querying range \( Q \) contains \( n \) different node regions (\( 2 \leq n \leq m \), where \( m \) is the total number of nodes).

Let \( \epsilon Q (i = 1, 2, \ldots, n) \) represent the privacy protection strength of different nodes’ areas. As the unbalanced quadtree partition method segregates the two-dimensional space covered by the location dataset into \( m \) independent and disjoint subsets, querying range \( Q \) achieves the privacy protection intensity \( \epsilon Q = \epsilon Q(j) \), differential privacy model (Theorem 2). For any of the sub-region, the privacy protection intensity is exactly the same as the above situation (1), i.e., \( \epsilon Q = \epsilon \), such that \( \epsilon Q = \epsilon \).

Combining the above two situations, the proposed algorithm can provide \( \epsilon \) differential privacy protection intensity for any querying range \( Q \) proposed by the user. □

V. EXPERIMENTAL ANALYSIS
In order to verify the effect of the unbalanced quadtree partition and publishing algorithm (UBQP-gra) on the statistical publishing of big location data, we compare and analyze the proposed algorithm with a number of classical quadtree-based and grid-based partition methods from the aspects of precision of range counting query, privacy budget allocation effect, and algorithm operational efficiency. The baseline methods include, but are not limited to, uniformed grid partition method (UG) [14], adaptive grid partition method (AG) [14], quadtree partition method with geometric budgeting (Quad-geo) [18], quadtree partition method with heuristic structural adjustment (Quad-heu) [20], and density-based quadtree partition method (DBP) [23].

Experimental location datasets include Checkin,\(^2\) which provides location information from the social network site, Gowalla; facility location information set, Storage, and location information dataset, Landmark,\(^3\) of 48 states in the United States provided by Infochimps; and Taxi record dataset, Yellow_tripdata,\(^4\) provided by the New York City Taxi Management Committee. The querying range \( Q \) is set with six different sizes (as shown in Table 1) and the distribution status of the experimental datasets is shown in Figure 6 (in the case of equal longitude, the distance is about 1,113 meters per 0.01 degree, whereas, in the case of equal latitude, the distance is about 1,000 meters per 0.01 degree).

A. ANALYSIS OF RANGE COUNTING QUERY ACCURACY
The accuracy of range counting query refers to the comparison of the relative error between the original and published counting value within the querying range \( Q \). The definition of relative error is shown in Eq. (8):

\[
RE(Q) = \frac{|C^*(Q) - C(Q)|}{\max(C(Q), \rho)}
\]

where, \( C(Q) \) represents the query result obtained on the original dataset and \( C^*(Q) \) denotes the result received from the published dataset. To prevent the denominator from being zero, we have set \( \rho = 0.001 \times |T| \), where \( |T| \) is the size of the dataset.

During the experiments, the AG algorithm uses the parameter \( \alpha = 0.5 \) to determine how to split the privacy budget between the two levels, the Quad-geo algorithm and 2http://snap.stanford.edu/data/loc-gowalla.html
3http://www.infochimps.com/datasets/storage-facilities-by-landmarks
4http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml
Y. Yan et al.: Differential Private Spatial Decomposition and Location Publishing Based on Unbalanced Quadtree Partition Algorithm

TABLE 1. Experimental dataset and querying area information.

| Parameter   | Storage | Landmark | Checkin | Yellow_tripdata |
|-------------|---------|----------|---------|-----------------|
| Total data amount | 8938    | 870052   | 1000000 | 12504938        |
| Coverage     | 60×40   | 60×40    | 360×150 | 1×1             |
| q1          | 1.25×0.625 | 1.25×0.625 | 6×3     | 0.064×0.004     |
| q2          | 2.5×1.25  | 2.5×1.25  | 12×6    | 0.008×0.008     |
| q3          | 5×2.5    | 5×2.5    | 24×12   | 0.016×0.016     |
| q4          | 10×5     | 10×5     | 48×24   | 0.032×0.032     |
| q5          | 20×10    | 20×10    | 96×48   | 0.064×0.064     |
| q6          | 40×20    | 40×20    | 192×96  | 0.128×0.128     |

FIGURE 7. Comparison of querying accuracy on Storage dataset.

FIGURE 8. Comparison of querying accuracy on Landmark dataset.

The Quad-heu algorithm adopt the same partition level, i.e., \( h = 8 \). The differential privacy model incorporates Laplace noise with privacy budgets of \( \varepsilon = 0.1, 0.5, \) and \( \varepsilon = 1.0 \), respectively. Each type of querying area is randomly generated for 1,000 times and the average of the relative error is ascertained.

The relative error of range counting query on different datasets and privacy budgets is depicted in Figure 7-10. Under the premise of the same scale of differential privacy budget, the Yellow_tripdata dataset with a concentrated distribution of location points and the highest density achieves a higher querying accuracy. The querying accuracy of the Checkin dataset falls second which has a partial concentrated distribution of location points. The dense and sparse areas of the Landmark dataset are relatively scattered and the querying accuracy of this dataset is slightly lower in contrast to the previous two. The Storage dataset with the most sparsely distributed location points presents higher querying errors. The primary reason for this phenomenon is that when the location points are scattered and sparsely distributed, they are more likely to generate some empty nodes, and therefore, introduce more noise errors during the partition process. When the location points are concentrated and uniformly distributed, the uniform assumption estimation can better distribute noise.
error to each sub-unit consequently reducing the overall error. Also, observing the experimental results on the same dataset, we find that the relative errors of various algorithms gradually decrease with an increase in the privacy budget. This is because, with constant sensitivity, the increase of differential privacy budget reduces the incorporated Laplace noise, and hence, the deviation between real data and published result is reduced. From the aspect of distribution characteristics of the querying accuracy, the relative errors are maximized for queries of the middle sizes. In particular, the maximizing points are $q^3$ for Landmark dataset, whereas, $q^4$ for the Storage, Checkin, and Yellow_tripdata dataset respectively. This is primarily owing to the fact that when the querying range is small, fewer subregions have been included, and therefore, the overall querying error is relatively small. When the querying range is too large, the large blank areas cause the query to return low true count, and, therefore, the overall querying error is also relatively low.

From the effects of various partition methods, the UG partition algorithm has a larger querying error on various datasets and privacy budgets. The foremost reason is that the data-independent grid structure introduces more uniform assumption error and noise error because of the negligence of specific distribution of the location points. The AG algorithm performs the adaptive grid partition on the second layer based on the UG algorithm, which to some extent, offsets the errors introduced by the uniform partition on the first layer and improves the querying accuracy. The Quad-geo partition algorithm also adopts the data-independent partition strategy. It performs well for large-scale queries on sparse datasets but has poor effects if the dataset is densely distributed or unevenly distributed. The Quad-heu algorithm has a good effect on the dataset with a small querying range and relatively uniform distribution. When the local regions are uniformly distributed, the whole area would also be uniform, and thus, the bottom-up adjustment of Quad-heu algorithm would help to reduce the noise error of the sparse regions in the Quad-geo partition process. However, when the querying range is large or the distribution of datasets is different, the adjustment strategy of Quad-heu algorithm becomes unfeasible. The density-based partition method, DBP, realizes the dynamic adaptive adjustment of spatial partition with the distribution of location points. This consequently facilitates in reducing the uniform assumption error of dense regions and the noise error of sparse regions. However, it does not pay any attention to the privacy budget allocation. All the regions after partition are allocated the same privacy budget $\varepsilon$. The unbalanced quadtree partition algorithm proposed in this paper realizes the adaptive partition structure through the combination of regional uniformity judgment and quadtree
partition method. By improving privacy budget allocation and adjustment strategies, a fairly accurate counting query effect is achieved. The UBQP-gra method achieves the optimal range counting query precision on different datasets and privacy budget intensities.

**B. ANALYSIS OF PRIVACY BUDGET ALLOCATION EFFECT**

Table 2 depicts the comparison of privacy budget allocation methods and privacy protection intensities for all the selected partition and publishing algorithms. Together with the experimental results of Figure 7-10, it is not difficult to determine that assigning the same differential privacy budget (i.e., equal allocation method) to all the partition regions in the grid-base structure or the tree-based structure cannot achieve the accurate regional querying accuracy.

In order to further compare the impact of different privacy budget allocation methods on the same partition structure, the proposed unbalanced quadtree partition algorithm is combined with uniform privacy budget allocation method (UBQP-uni), geometric privacy budget allocation method (UBQP-geo), and gradient privacy budget allocation method (UBQP-gra) respectively. The uniform privacy budget allocation method assigns the total budget evenly to the nodes of each partition according to the depth of the partition structure (i.e., \( \varepsilon_i = \varepsilon / (h + 1) \)), wherein \( h \) represents the depth of the partition tree. The geometric and gradient privacy budget allocation methods make a gradual increase in the budget intensity with the increase of the partition depth from the root node to the leaf node. For the geometric privacy allocation method, the privacy budget available to each layer can be expressed as \( \varepsilon_i = 2^{-\frac{h}{2}} \times \varepsilon \times \frac{3^{2}\varepsilon^{-1}}{2^{h+1}} \).

The gradient privacy budget allocation method calculates the available privacy budget for each layer according to Eq. (6).

Figure 11 depicts the comparison of querying accuracy for different budget allocation methods on different datasets. To be fair, the same partition structure, total privacy budget, and absorption adjustment method are maintained on the same dataset. It can be observed that the uniform privacy budget allocation method received the largest relative error under queries of various sizes. The proposed gradient privacy budget allocation method achieves better querying accuracy in contrast to the uniform privacy budget allocation method and geometric privacy budget allocation method.

**C. ANALYSIS OF ALGORITHM EFFICIENCY**

The operational efficiency of the partition and publishing algorithm primarily compares the overall time of constructing partition structure and incorporating differential privacy noise to form the publishing data. The performance of this part directly affects the feasibility of the algorithm in the application environment of dynamic release of big location data. However, the specific running time of a partition and publishing algorithm is affected by hardware conditions, software environment, coding efficiency, and a number of other factors. Therefore, we compare the efficiency of different algorithms by analyzing their time complexities as depicted in Table 3.

The data-independent partition algorithms without considering data distribution characteristics are more time-saving in contrast to the data-dependent partition algorithms. The UG algorithm is typically data-independent. The partition process of the grid structure only needs to scan the input data once. Assuming that the location dataset contains \( n \) coordinate points, the time complexity of the UG algorithm is approximately \( O(n) \). Based on the UG partition for the first-level, the AG algorithm performs the second layer of adaptive partition based on the density of each grid. The overall algorithm requires to scan the input data for two times, so the time complexity of the AG algorithm is approximately \( O(n + n) \approx O(n) \). The Quad-geo algorithm performs a recursive full quadtree partition on the input dataset. The overall time complexity of the algorithm is about \( O(n \log(n)) \).

---

**TABLE 2. Experimental dataset and querying area.**

| Algorithm | Privacy budget allocation method | Privacy protection strength |
|-----------|----------------------------------|-----------------------------|
| UG        | Equal allocation                 | \( \varepsilon \)            |
| AG        | Hierarchical allocation          | \( \varepsilon \)            |
| Quad-geo  | Geometric allocation             | \( \varepsilon \)            |
| Quad-heu  | Geometric allocation             | \( \varepsilon \)            |
| DBP       | Equal allocation                 | \( \varepsilon \)            |
| UBQP-uni  | Uniform allocation               | \( \varepsilon \)            |
| UBQP-geo  | Geometric allocation             | \( \varepsilon \)            |
| UBQP-gra  | Gradient allocation + Absorption adjustment | \( \varepsilon \) |

**TABLE 3. Comparison of time complexity of tree-based partition algorithms.**

| Algorithm | Time complexity |
|-----------|-----------------|
| UG        | \( O(n) \)      |
| AG        | \( O(n) \)      |
| Quad-geo  | \( O(n \log(n)) \) |
| Quad-heu  | \( O(n \log(n)) \) |
| DBP       | \( O(hn) \)    |
| UBQP      | \( O(hn) \)    |
where, \( n \) is the total number of samples of the big location dataset. The Quad-heu algorithm adopts the same quadtree partition structure and geometric budget allocation scheme as the Quad-geo algorithm. In addition, it needs to scan all the nodes from the bottom up to adjust and combine adjacent uniform nodes so that the overall time complexity of the Quad-heu algorithm is slightly higher than the Quad-geo algorithm, i.e., about \( O(n \log(n) + 4^{h+1} \approx O(n \log(n)) \). The DBP algorithm and the proposed UBQP algorithm all depend on the specific distribution state of big location dataset, and therefore, the time complexity of these algorithms are all related to the partition depth of quadtree. In the worst case scenario, for a big dataset with a total number of \( n \) samples and \( h \) partition depth, the time complexity of the above partition algorithms are all approximately \( O(h n) \).

### VI. CONCLUSION

Reasonable and effective privacy protection algorithms can improve the accuracy of queries and analysis of published data on the premise of ensuring users’ privacy as well as for realizing the balance between data privacy and availability. The partition and publishing method based on the differential privacy model is an efficacious way to protect the privacy of the centralized statistics release of big location data. In order to further improve the querying accuracy of published data, we analyze the restrictive relationships among publishing error, spatial partition structure, and privacy budget allocation method. The unbalanced quadtree partition method is proposed to iteratively split the two-dimensional space according to the distribution characteristics of big location dataset, thereby, solving the problem of determining the stop conditions for top-down space partition. At the same time, it facilitates in avoiding the noise error introduced by excessive empty nodes and reduces the uniform assumption error. The gradient privacy budget allocation scheme and absorption adjustment method designed for the unbalanced quadtree partition structure ensures that the differential privacy model is implemented on all the partition regions. Experimental comparisons with other partition algorithms and privacy budget allocation strategies on actual location datasets prove
the advantages of the proposed algorithm in enhancing the querying accuracy of published data.

REFERENCES

[1] M. Z. Ge, H. Bangui, and B. Buhnova, "Big data for Internet of Things: A survey," Future Gener. Comput. Syst., vol. 87, pp. 601–614, Oct. 2018.

[2] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," IEEE Trans. Intell. Transp. Syst., vol. 20, no. 1, pp. 383–398, Jan. 2019.

[3] A. R. Beresford and F. Stajano, "Location privacy in pervasive computing," IEEE Pervis. Comput., vol. 2, no. 1, pp. 46–55, Jan. 2003.

[4] D. Kifer and A. Machanavajjhala, "No free lunch in data privacy," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2011, pp. 193–204.

[5] J. Krumm, "A survey of computational location privacy," Pers. Ubiquitous Comput., vol. 13, no. 6, pp. 391–399, Aug. 2009.

[6] C. Dwork, "Differential privacy," in Proc. 33rd Int. Colloq. Automata, Lang. Program., 2006, pp. 1–12.

[7] C. Dwork, "Differential privacy: A survey of results," in Proc. Int. Conf. Theory Appl. Models Comput., 2008, pp. 1–19.

[8] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Optimal geindistinguishable mechanisms for location privacy," in Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS), 2014, pp. 251–262.

[9] E. Fung, G. Kellaris, and D. Papadias, "Combining differential privacy and PIR for efficient strong location privacy," in Proc. Int. Symp. Spatial Temporal Databases, 2015, pp. 295–312.

[10] T. Harä, A. Suzuki, M. Iwata, Y. Arase, and X. Xie, "Dummy-based user location anonymization under real-world constraints," IEEE Access, vol. 4, pp. 673–687, 2016.

[11] X. Yi, R. Pautet, E. Bertino, and V. Vardhanarajan, "Practical approximate k nearest neighbor queries with location and query privacy," IEEE Trans. Knowl. Data Eng., vol. 28, no. 6, pp. 1546–1559, Jun. 2016.

[12] B. Liu, W. Zhou, T. Zhu, H. Zhou, and X. Lin, "Invisible hand: A privacy preserving mobile crowd sensing framework based on economic models," IEEE Trans. Veh. Technol., vol. 66, no. 5, pp. 4410–4423, May 2017.

[13] H. To, G. Ghinita, L. Fan, and C. Shahabi, "Differentially private location protection for worker datasets in spatial crowdsourcing," IEEE Trans. Mobile Comput., vol. 16, no. 4, pp. 934–949, Apr. 2017.

[14] W. Qardaji, W. Yang, and N. Li, "Differentially private grids for geospatial data," in Proc. IEEE 29th Int. Conf. Data Eng. (ICDE), Apr. 2013, pp. 757–768.

[15] J. Wang, R. Zhu, S. Liu, and Z. Cai, "Node location privacy protection based on differentially private grids in industrial wireless sensor networks," Sensors, vol. 18, no. 2, pp. 410–424, 2018.

[16] P. Xiong, L. Zhang, and T. Q. Zhu, "Reward-based spatial crowdsourcing with differential privacy preservation," Enterprise Inf. Syst., vol. 11, no. 10, pp. 1500–1517, Nov. 2017.

[17] G. Zhou, S. Qin, H. Zhou, and D. Cheng, "A differential privacy noise dynamic allocation algorithm for big multimedia data," Multimedia Tools Appl., vol. 78, no. 3, pp. 3747–3765, Feb. 2019.

[18] G. Cormode, C. Procopiuc, D. Srivastava, E. Shen, and T. Yu, "Differentially private spatial decompositions," in Proc. IEEE 28th Int. Conf. Data Eng. (ICDE), Apr. 2012, pp. 20–31.

[19] J. Wang, S. Liu, Y. Li, H. Cao, and M. Liu, "Differentially private spatial decompositions for geospatial point data," China Commun., vol. 13, no. 4, pp. 97–107, Apr. 2016.

[20] Y. Wu, Q. Lu, J. Cai, and X. Wang, "Differential privacy two-dimensional data partitioning publication algorithm based on quad-tree," J. Huazhong Univ. Sci. Technol. (Natural Sci. Ed.), vol. 44, no. 3, pp. 99–104, 2016.

[21] J. Zhang, X. Xiao, and X. Xie, "PrivTree: A differentially private algorithm for hierarchical decompositions," in Proc. Int. Conf. Manage. Data (SIGMOD), 2016, pp. 155–170.

[22] A. Inan, M. Kantarcioglu, G. Ghinita, and E. Bertino, "Private record matching using differential privacy," in Proc. 13th Int. Conf. Extending Database Technol. (EDBT), 2010, pp. 123–134.

[23] M. Yang, T. Zhu, Y. Xiang, and W. Zhou, "Density-based location preservation for mobile crowdsensing with differential privacy," IEEE Access, vol. 6, pp. 14779–14789, 2018.

[24] Y. Yan, X. Hao, and L. Zhang, "Hierarchical differential privacy hybrid decomposition algorithm for location big data," Cluster Comput., vol. 22, no. S4, pp. 9269–9280, Jul. 2019.

[25] F. D. McSherry, "Privacy integrated queries: An extensible platform for privacy-preserving data analysis," in Proc. 35th SIGMOD Int. Conf. Manage. Data (SIGMOD), 2009, pp. 19–30.

YAN YAN received the Ph.D. degree in control theory and control engineering from the Lanzhou University of Technology, China. She is currently an Associate Professor with the School of Computer and Communication, Lanzhou University of Technology. She has been an Academic Visiting Scholar with Macquarie University, since 2019. Her research interests include privacy preserving data publishing, differential privacy, and information hiding. She is a member of the IEEE and the China Computer Federation.

ADNAN MAHMOOD is currently an Associated with the Department of Computing, Macquarie University, Sydney, NSW, Australia. Before moving to Macquarie University, he spent a considerable number of years in the diverse academic and industrial settings of Republic of Ireland, South Korea (Republic of Korea), Malaysia, Pakistan, and People’s Republic of China. His research interests include software-defined networks, network functions virtualization, intelligent transportation systems, the Internet-of-vehicles, trust management, and next generation heterogeneous wireless networks. He also serves on the Technical Program Committees and Editorial Boards for several reputed international conferences and journals, respectively.

XIN GAO received the B.Eng. degree from Harbin Normal University, in 2013. She is currently pursuing the master’s degree with the School of Computer and Communication, Lanzhou University of Technology, China. Her research interests include privacy preservation data publishing, information security, and dynamic clustering. She is also a Student Member of the China Computer Federation.

TAO FENG received the Ph.D. degree in computer architecture from Xidian University, in 2008. He is currently a Full Professor and a Ph.D. Supervisor with the Lanzhou University of Technology. His main research interests include information security, provable security of protocol security, wireless network security, and sensor network security. He is also a member of the China Computer Federation and the China Cryptography Federation.

PENGSHOU XIE is currently a Full Professor with the Lanzhou University of Technology. His major research fields include security of the Internet of Things, location-based services, and information security. He is also a member of the China Computer Federation.

VOLUME 8, 2020 104787