Received: 1 September 2021 | Revised: 15 December 2021 | Accepted: 7 January 2022

DOI: 10.1002/cpe.6965

SPECIAL ISSUE PAPER

WILEY

A privacy preserving data aggregation and query for metro passenger flow via mobile crowdsensing

Yuanyuan Zhang[1] | Zuobin Ying[2] | Bowen Zhao[3] | Chun Lung Philip Chen[1,4]

[1]School of Computer Science and Engineering, South China University of Technology, Guangzhou, China

[2]Faculty of Data Science, City University of Macau, Macau, China

[3]Guangzhou Institute of Technology, Xi'dian University, Guangzhou, China

[4]Pazhou Lab, Guangzhou, China

Correspondence
Chun Lung Philip Chen, School of Computer Science and Engineering, South China University of Technology, Guangzhou, China.
Email: philip.chen@ieee.org

Funding information
National Key Research and Development Program of China, Grant/Award Numbers: 2019YFA0706200, 2019YFB1703600; National Natural Science Foundation of China, Grant/Award Numbers: 61702195, 61751202, U1813203, U1801262, 61751205

Summary

With the popularity of mobile Internet networks and mobile terminal equipment, mobile crowdsensing (MCS) has become a popular and economical way of data collection, sharing, and query services. However, the existing privacy-preserving query schemes MCS-based less consider the sensing user's privacy leakage at data aggregation phase and querier's privacy leakage at query service phase simultaneously. In this article, we propose a novel privacy-preserving aggregation and query scheme applied in metro passenger flow via MCS, which achieves the metro passenger flow acquisition and query service and protects the sensing user's identity and location privacy as well as the querier's query privacy. We exploit Paillier cryptosystem and pseudonym mechanism to prevent the privacy leakage, and utilize secure kNN algorithm to protect the query privacy. Meanwhile, we achieve the sensing user accountability based on digital signature algorithm and credible management. The privacy analysis demonstrates that our scheme satisfies the privacy-preserving and security requirements. The performance evaluation and experimental result show the efficiency of our scheme in practice.

KEYWORDS

aggregation and query, metro passenger flow, mobile crowdsensing, privacy preserving

1 | INTRODUCTION

With the rapid development of mobile Internet technology and the popularization of mobile devices, the mobile crowdsensing (MCS) technology using mobile devices as sensing nodes has gradually become a popular and economical way of data collection.

The popularity of mobile terminal devices, especially smartphones which are enriched with multiple sensors, has made MCS viable in large-scale. MCS can be used to retrieve information in lots of areas, including traffic congestion, environment, weather, urban computing as well as any other information that collectively forms knowledge.[1-3] By analyzing a large number of positive or passive sensing data of mobile device users through specific model and algorithm, we can solve many important practical problems in our daily life.[4-6]

Realizing the huge potential of MCS, a number of applications have started to emerge. MCS applications bring great convenience to people's lives, especially in the smart city system. For instance, in scheme,[7] people can scan with Bluetooth device of smartphone to quickly estimate the crowd density in public places. Thiagarajan et al.[8] proposed a crowdsensing system called VTrack, which leverages the information collected by smartphone users to generalize the prediction of traffic flow. Koukoumidis et al.[9] introduced a novel software service called SignalGuru that relies solely on a collection of mobile phones to detect and predict the traffic signal schedule.

Looking at the current people's daily travel habits, the metro, as a major means of transportation in the city, is still the preferred travel mode for most people. However, due to the special operation environment of the metro station, its space and passenger capacity are limited. For passengers entering the metro station, under the circumstances of holidays, rush hours, adverse climate change, and emergencies around, the large numbers

of passengers flow into the subway station will delay the waiting time of passengers. Therefore, the metro station passenger flow query service has certain practicability for people taking the metro.

Nevertheless, the existing data aggregation and query schemes still have the limitations of complex encryption algorithms, high communication cost and additional data leakage, meanwhile, cannot be applied to resource-limited mobile devices for metro passenger flow. Meanwhile, the MCS applications require the sensing users consciously or unconsciously uploading the data related to the identity or location in the data aggregation phase, and providing the location or query keywords in the query phase to obtain the corresponding service, which may seriously leakage privacy of the users.[10-12] Privacy leakage affects the enthusiasm of users to participate in data sensing and even inhibits the innovation and integration of MCS applications in the new area.

Motivated by the aforementioned challenges, we propose a novel privacy-preserving data aggregation and query (PDAQ) scheme applied in metro passenger flow via MCS, which provides the data aggregation and query service. Meanwhile, PDAQ scheme protects the identity and location information of the sensing user, and prevents the query user's query privacy leakage. Specifically, we exploit the pseudonym mechanism to protect the identity and location, and verify and aggregate the data tuples based on Paillier cryptosystem. To prevent the query privacy leakage, we utilize secure kNN algorithm to encrypt the index vectors and the query keywords. Meanwhile, we achieve the sensing user accountability based on digital signature algorithm and credible management.

The main contributions of PDAQ are summarized as follows:

- We propose a PDAQ scheme, which achieves the metro passenger flow query and prevents the privacy leakage. We exploit the pseudonym mechanism to protect the identity and location, and verify and aggregate the data tuples based on Paillier cryptosystem. We utilize secure kNN algorithm to prevent the query privacy leakage. The sensing user accountability mechanism is achieved without revealing the identity information to the untrusted entities.
- Compared with the previous MCS applications, PDAQ attempts to protect querier's query privacy utilizing secure kNN algorithm, and the first scheme to combine the privacy-preserving of MCS with searchable encryption.
- We gave the security analysis under the threat model, the performance evaluation shows the time costs of each phase, and the experimental result demonstrated the efficiency of PDAQ scheme in practice.

The following parts of this article are organized as follows: The related works are discussed in Section 2, then, Section 3 describes the system model, design goals and threat model, and also revisits the preliminaries in our work. In Section 4, we construct the proposed PDAQ scheme. The privacy security and the performance evaluation are demonstrated in Sections 5 and 6, respectively. Finally, Section 7 concludes this article.

2 | RELATED WORK

The studies at home and abroad focus on location privacy and query privacy have issued many results.

In order to protect the location privacy and achieve the data aggregation. Hoh et al.[13,14] and Gruteser and Grunwald[15] adopted the trusted parties to aggregate data, and used the space distortion technique to protect the user's location privacy. Gedik and Liu[16] and Krumm[17] proposed the privacy-preserving schemes to hide the user's sensitive location by the spatial and temporal subsampling techniques. Xiong et al.[12] proposed a light-weight secure data clustering scheme to achieve optimal balance between performance and privacy. However, the aforementioned schemes less consider the data verification in data aggregation phase, the malicious or dishonest users deliberately upload illegal data to affect the statistical results. To guarantee the correctness of uploaded data and prevent the illegal data contaminating the aggregation results, Popa et al.[18] proposed a privacy-preserving data legitimacy verification scheme, which used the blind signature algorithm and zero knowledge proof technique. When the legitimacy verification of the uploaded data fails, the data tuples will be discarded. For the malicious or dishonest users, Popa's[18] scheme did not consider the user accountability. Based on the trusted parties, Chen et al.[19] achieved data legitimacy verification and malicious user accountability. Chen's scheme utilized the Paillier cryptosystem and the multi-pseudonym mechanism to verify the uploaded data with privacy preserving.

In terms of the query privacy, the most existing schemes[20-23] are designed for the query location protection. Based on the trusted anonymity, Chow et al.[23] achieved the specific query location hiding via expanding the user's query range. However, the query accuracy is affected by the user's query range expanding. In order to guarantee the query accuracy in a certain range, Ghinita's scheme[24] costed heavy computational and communication overheads without the trusted anonymity. To reduce the overhead of computational and communication with the client, Niu et al.[25] proposed a novel scheme to prevent privacy leakage of the query location, which used K-anonymity algorithm. When receiving a query request from the query user, the cloud server need perform K times query computation, which increased an extra cost to the cloud server.

Nevertheless, the existing data aggregation and query schemes still have the limitations of complex encryption algorithms, high communication cost and additional data leakage, meanwhile, cannot be applied to resource-limited mobile devices for MCS applications.[26-28] Meanwhile, the

sensing user's privacy leakage at data aggregation phase and querier's privacy leakage at query service phase, the malicious user accountability need to consider.

3 | PROBLEM DESCRIPTION

In this section, we introduce the system model at first, then, give a threat model and design goals. At last, we give the preliminaries about Paillier cryptosystem.

3.1 | System model

Our system consists of four entities: sensing user (SU), cloud server (CS), trusted authority (TA), and query user (QU) are shown in Figure 1. We support a PDAQ service for metro passenger flow via MCS. The main two phases are data aggregation phase and data query phase. The task of data aggregation phase is to collect data and conduct aggregation statistics. The data query phase is to respond the metro passenger flow query requests.

TA is responsible for generating, distributing, and maintaining cryptosystem parameters. TA initializes the parameters of the whole system, publishes the public key and distributes the private key to the authenticated user. At the same time, TA keeps and manages the pseudonyms list of the registered user to realize the accountability of malicious users. Moreover, TA generates the trapdoors to respond the query requests from QU.

SU is in charge of processing and uploading the sensing data tuples to the CS. The location information is encrypted and uploaded automatically to the CS, which are obtained via the mobile device interacting with GPS, base station or WiFi.

CS has powerful storage and computing capabilities, which is in charge of aggregating and storing encrypted data tuples and matching ciphertext index and query trapdoor to provide the query service for the QU.

QU sends the query request to the CS, using query trapdoor with the required keyword. After receiving the query result, QU utilizes the corresponding key to decrypt the result.

3.2 | Threat model

TA is fully trusted among all entities participating in our PDAQ scheme and is infeasible to be compromised by an adversary. CS is semi-honest, namely, the honest-but-curious CS honestly performs the data aggregation and query but may try to infer the information about the location, identity and the query contents. Besides, we assume that there is no collusion between CS and users. The honestly authorized users will not disclose their private keys to others. Besides, the communication channels between TA and users are secure.

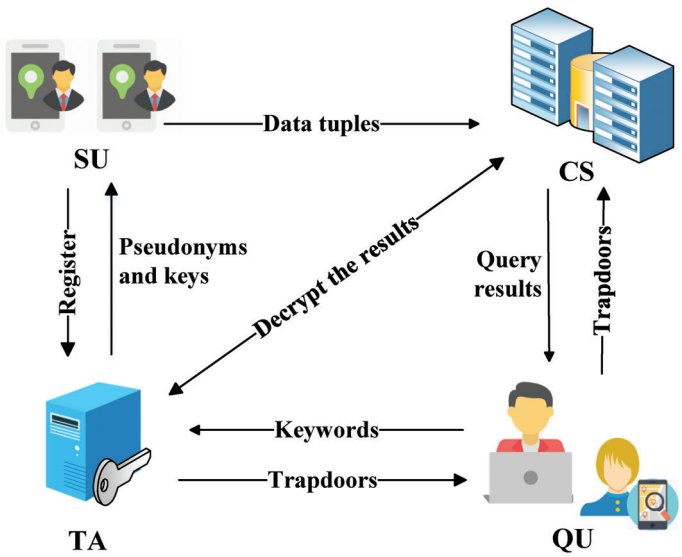


FIGURE 1 System architecture

Based on the above assumptions, we consider the following threat model:

- CS can obtain the encrypted tuples, indexes and query trapdoors from SU and QU. CS learns more information to infer the plaintext and ciphertext pairs from the query requests and search results.
- An internal adversary attempts to disclose the identity of SU and access the SU's location.
- An internal adversary attempts to tamper with the uploaded data tuples. The malicious user attempts to upload the invalid data to interfere with the aggregation result.
- An external adversary could eavesdrop the communications to get the transmitted information.

3.3 | Design goals

To enable the PDAQ for metro passenger flow under the aforementioned threat model, PDAQ should satisfy the five aspects of identity anonymity, location privacy, data integrity, query privacy, and traceability, which are described as follows:

- *Identity anonymity* . The identities of users participating in our scheme should be protected, and the adversaries should not be able to obtain the sensitive information involved in the uploaded data from the users.
- *Location privacy*. The location information of users participating in our scheme should be protected. The sensitive location information should not be disclosed to the CS and other entities.
- *Data integrity*. The uploaded data tuples should be verified to guarantee the data integrity and validity.
- *Query privacy*. Since the query requests and results are all available to the semi-honest CS and adversaries, our proposed protocol should achieve the query indexes, trapdoors and results privacy preservation. CS and adversaries will not infer and predict any query content plaintexts.
- *Accountability*. Our proposed scheme could track the identity of a malicious user who uploads the invalid data to interfere the aggregation result.

3.4 | Preliminaries

Paillier cryptosystem is an asymmetric cryptographic algorithm with additive homomorphism property. The details are as follows:

Key generation. The two large prime numbers p and q are selected randomly, $N = p \times q$ and $\lambda = \text{lcm}(p-1, q-1)$, where $\text{lcm}(x, y)$ is the least common multiple of x and y. Then, $g \in Z^*_{N^2}$ is selected randomly, which satisfies $\gcd(L(g^\lambda \bmod N^2), N) = 1$, where $L(x) = \frac{(x-1)}{N}$, $\gcd(x, y)$ is the greatest common divisor of x and y. (N, g) is the public key and the private key is λ.

Encryption. A number $r \in Z^*_N$ is selected randomly, the plaintext $m \in Z_N$ is encrypted with the public key (N, g) as:

$$c = E(m) = g^m r^N \bmod N^2. \tag{1}$$

Decryption. Given the ciphertext $c \in Z^*_{N^2}$, the corresponding plaintext m can be obtained as follows:

$$m = D(c) = \frac{L(c^\lambda \bmod N^2)}{L(g^\lambda \bmod N^2)} \bmod N. \tag{2}$$

Homomorphic property. For any plaintext $m_1, m_2 \in Z_N$, and random number $r_1, r_2 \in Z_N$, the two homomorphic properties can be described as:

$$E(m_1) = g^{m_1} r_1^N \bmod N^2, \ E(m_2) = g^{m_2} r_2^N \bmod N^2. \tag{3}$$

$$E(m_1) \cdot E(m_2) = g^{m_1+m_2} (r_1 r_2)^N \bmod N^2 = E(m_1 + m_2) \bmod N^2. \tag{4}$$

4 | CONSTRUCTION OF THE PDAQ SCHEME

In this section, the proposed PDAQ scheme includes four phases, namely: system initialization, registration, data aggregation, and data query. Note that the data flows among four entities are shown in Figure 1. TA initializes the system parameters to generate the system keys and the related lists. SU registers with the identity to TA, and obtains public–private key pair, the pseudonym information, and the station list. The location information

of SU is encrypted and uploaded automatically to the CS, which is obtained from the mobile device interacting with GPS, base station, or WiFi. After receiving the uploaded tuples, CS verifies the legitimacy of the pseudonym and signature information. Upon successful validation, CS obtains and stores the aggregation results by interacting with TA. QU sends the query request to the CS, using query trapdoor with the required keyword from TA. CS calculates the similarity between the query trapdoor and the ciphertext index tree node to find the best matching node. According to the path of passenger flow information contained in the node, CS returns the query result to QU.

We list the notations frequently used in PDAQ and their descriptions in Table 1.

4.1 | System initialization

TA produces necessary system parameters and key pairs, the following steps are executed.

Step 1: TA generates Paillier public–private key pair (PK_P, SK_P), then generates RSA signature public–private key pair (PK_S, SK_S).

Step 2: TA randomly generates a vector S with n bit and two invertible matrices $\{M_1, M_2\}$ of $(n+n') \times (n+n')$ as the secure kNN private key SK, where n' is the extended virtual dimension, and adding the virtual dimension is to resist the statistical attack.

Step 3: TA establishes a pseudonym list, denoted by P_List. Each record R_i in P_List contains the following information:

$$R_i = \left(PI_i, \sigma_{SK_P}, PK_{PI_i}, SK_{PI_i}\right), \tag{5}$$

where PI_i is the pseudonym identity, PK_{PI_i}, SK_{PI_i} are the public key and private key of the pseudonym identity, and σ_{SK_P} is the digital signature generated with the private key SK_P.

Step 4: TA randomly divides the metro stations into m groups, denoted by g_j is the group number, and the metro stations denoted by $s_{j,i}$, where $j \in \{1, 2, \ldots, m\}, i \in \{1, 2, \ldots, h\}$. Namely, the station group $g_j = (s_{j,1}, s_{j,2}, \ldots, s_{j,h})$. Then, all the station groups are stored in the station list denoted as S_List.

4.2 | Registration

SU needs to register with the identity to TA. After that, TA sends PK_P, R_i and S_List to the SU via a secure channel. Subsequently, TA encrypts and stores the mapping between the SU's identity and the record R_i in relation map list denoted as R_List.

In our PDAQ scheme, the pseudonym has a certain term of validity, if the time is expiration, SU must to register a new pseudonym from TA.

To ensure the security, P_List, S_List, and R_List are encrypted and stored in TA.

TABLE 1 Notations and descriptions

Notations	Descriptions
(PK_P, SK_P)	The public–private key pair
(PK_S, SK_S)	The RSA signature public–private key pair
$SK = \{M_1, M_2\}$	The secure kNN private key
P_List	The pseudonym list
S_List	The station list
R_List	The relation map list
M_list	The mistake list
g_j	The station group
$s_{j,i}$	The metro station
LI_j	The location tag
$W = \{W_1, W_2, \ldots, W_k\}$	The key set
I	The index tree
Γ	The encrypted index tree
T_w	The trapdoor

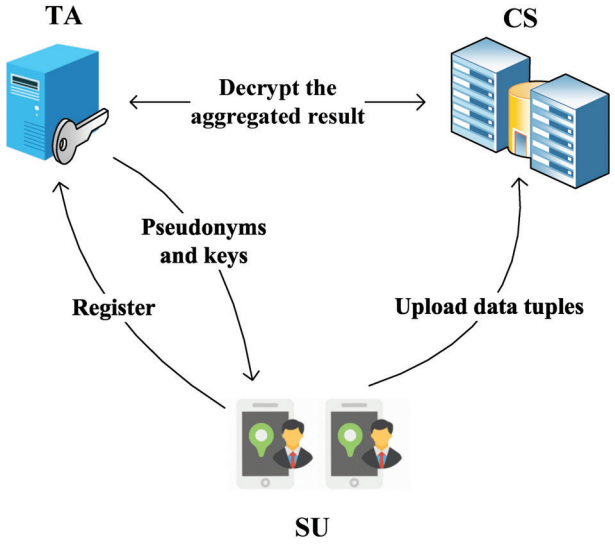


FIGURE 2 Data aggregation phase

4.3 | Data aggregation

The data aggregation is shown in Figure 2.

The mobile sensing device autonomously obtains the SU's current location through positioning, and located to the relevant station group g_j. Then, the location tag is generated as, $LI_j \in g_j$, $LI_j = (z_{j,1}, z_{j,2}, z_{j,3}, \ldots, z_{j,h})$, where $z_{j,i}$ can only be represented by 1 or 0. "1" represents SU is in this location range. In practice, the Guangzhou metro stations are as the example to describe. Suppose that a SU is located in "Xilang" station and the station group, $h = 10$, g_2=("Kecun," "Yuancun," "Chigang," "Xiaobei," "Dashadi," "Shiqiao," "Dashi," "Nanzhou," "Guanzhou," "Xilang"). Then the location tag $LI_2 = (0, 0, 0, 0, 0, 0, 0, 0, 0, 1)$.

Then, SU encrypts each $z_{j,i}$ in LI_j with PK_P to obtain E_{LI_j}, where $E_{LI_j} = (E(z_{j,1}), E(z_{j,2}), \ldots, E(z_{j,10}))$. Then it randomly chooses a pseudonym R_i or R'_i as the SU's identity and signs the E_{LI_j} with the private key SK_{PI_i} of the chosen pseudonym, that is

$$\sigma(E_{LI_j})_{SK_{PI_i}} = Sign_{SK_{PI_i}}(E_{LI_j}). \tag{6}$$

SU uploads the tuple to the CS within the time interval. Each tuple is generated as:

$$\left\{PI_i, \sigma_{SK_P}, PK_{PI_i}, g_j, E_{LI_j}, \sigma(E_{LI_j})_{SK_{PI_i}}\right\}. \tag{7}$$

Upon receiving the uploaded tuples, CS verifies σ_{SK_P} with the public key PK_P and verifies the $\sigma(E_{LI_j})_{SK_{PI_i}}$ with PK_{PI_i}. If CS finds that the SU's pseudonym has expired, the corresponding tuple is discarded. If verification is successful, the CS verifies the legitimacy of E'_{LI_j} as follows:

$$\begin{cases} IIIE(sum) = E\left(\sum_{i=1}^{h} z_{j,i}\right) \bmod N^2 = \prod_{i=1}^{h} E(z_{j,i}) \bmod N^2, \\ E(rsum) = \left(\sum_{i=1}^{h} (r_{j,i} z_{j,i})\right) \bmod N^2 = \prod_{i=1}^{h} E^{r_{j,i}}(z_{j,i}) \bmod N^2, \end{cases} \tag{8}$$

where $z_{j,i}$ is the member of LI_j which is only 0 or 1, $r_{j,i}$ is the random number generated by the CS, $r_{j,i} \in Z_N$, $i \in \{1, 2, \ldots, h\}$, $j \in \{1, 2, \ldots, m\}$. Then, the CS sends $E(sum)$ and $E(rsum)$ to the TA. TA decrypts it with private key SK_P and returns the decryption result to the CS.

The CS verifies $(sum, rsum)$, if the result satisfies:

$$\begin{cases} sum = 1, \\ rsum \in \{r_{j,i}\} \end{cases} \tag{9}$$

the E'_{LI_j} is legitimate.

Then CS put E'_{LI_j} and its corresponding station group g_j into the set $S = \{y|y = (g_j, E_{LI_j}), 1 \le j \le m\}$. Otherwise, the CS discards the tuple and records the PI_i in the mistake list, denoted as M_list. If a pseudonym is recorded in the M_list multiple times, TA will tack the identity of SU based on R_List.

Algorithm 1. Aggregation statistics

input: $S = \{y|y = (g_j, E_{LI_j}), 1 \le j \le m\}$
output: *the aggregation results and the corresponding station tags*
1: create the big group $G = \langle G_j|1 \le j \le m\rangle$, and the subgroup G_j correspond to g_j.
2: **for all** $y \in S, 1 \le j \le m$ **do**
3: **if** $y.g_i$ correspond to G_j **then**
4: put E_{LI_j} in G_j;
5: **end if**
6: **end for**
7: **for** $G_j \in G, 1 \le j \le m$ **do**
8: **for** $1 \le i \le 10$ **do**
9: $Result_{G_{j,i}} = \prod E(z_{j,i}) \, mod \, N^2$.
10: **end for**
11: Send the result report $(g_i, Result_{G_{j,i}})$ of the subgroup to TA.
12: where $Result_{G_{j,i}} = (Result_{G_{j,1}}, Result_{G_{j,2}}, \dots, Result_{G_{j,h}})$.
13: **end for**
14: TA decrypts the result of each subgroup.
15: TA finds the F_d corresponding to the station in g_j.
16: **return** $F_{G_{j,i}}$ and $result_{G_{j,i}}$.

The CS groups the validated data E'_{LI_j}, aggregates the data of each group respectively, and sends the aggregation result of subgroup to TA. The aggregation statistics phase is illustrated in Algorithm 1.

4.4 | Data query

TA regards the metro stations as the key set $W = \{W_1, W_2, \dots, W_k\}$, and generates the tag set denoted by $F = \{F_d|0 < d \le n\}$. Each tag F_d represents a keyword, that is (F_d, W_k).

Then, TA constructs the encrypted index tree I. The structure of each node u in I is $u = (ID, D_i, P_l, P_r, F_d)$. ID refers to the identity of node u, and P_l, P_r refer to the left and right subtrees of node u, respectively. The vector D_i of leaf nodes is an n-dimensional vector generated from F_d, where each element $D_i[k]$ refers to whether the keyword W_k is corresponding to the F_d, which represents by 1 or 0. In order to improve the security, we extend D_i into $n + n'$ dimensions and each extended element $D_i[n + j], j = 1, \dots, n'$, is set as a random number ξ_j. Then TA divides D_i into two vectors $\vec{D}_i'$ and $\vec{D}_i^{\varepsilon}$ according to the vector S. If $S[j] = 0$, $\vec{D}_i'[j]$ and $\vec{D}_i^{\varepsilon}[j]$ will be set equal to $D_i[j]$; Otherwise, $\vec{D}_i'[j]$ and $\vec{D}_i^{\varepsilon}[j]$ will be set as two random values whose sum equals to $D_i[j]$. Finally, Γ is built where the node u stores the encrypted index vectors $ED_u = \{M_1^{-1}\vec{D}_i', M_2^{-1}\vec{D}_i^{\varepsilon}\}$.

TA uploads Γ and the set $F$to the CS.

We adopt secure kNN algorithm to encrypt index tree and query vector. The data query phase is shown in Figure 3.

The QU sends the keyword W_k to TA. TA generates trapdoor T_w for QU. First, TA generates a vector Q of $(n + n')$-dimensions, where each element $Q[j]$ corresponds to the keyword W_k. When $W_k = W_k \in W$, the $Q[j]$ is set to 1, and others are set to 0. Each extended virtual dimension element $Q[n + j], j = 1, \dots, n'$ is set to a random number of 0 or 1. Next, TA divides Q into two vectors $\vec{Q}'$ and $\vec{Q}^{\varepsilon}$ according to the vector S. The principle of splitting is that $\vec{Q}'[j]$ and $\vec{Q}^{\varepsilon}[j]$ are equal to $Q[j]$ if vector $S[j] = 1$; If vector $S[j] = 0$, the sum of $\vec{Q}'[j]$ and $\vec{Q}^{\varepsilon}[j]$ is equal to $Q[j]$. Finally, the trapdoor T_w is obtained by computing $T_w = \{M_1^T\vec{Q}', M_2^T\vec{Q}^{\varepsilon}\}$. Then TA sends $T_w$to the QU.

Upon receiving the trapdoor T_w, the QU sends to the CS for query. The CS performs the depth-first search algorithm to query result, which is illustrated in Algorithm 2.

$$\begin{aligned} RT_u &= ED_u \cdot T_w \\ &= \left\{ M_1^T\vec{D}_i', M_1^T\vec{D}_i^{\varepsilon} \right\} \cdot \left\{ M_1^T\vec{Q}', M_1^T\vec{Q}^{\varepsilon} \right\} \end{aligned}$$

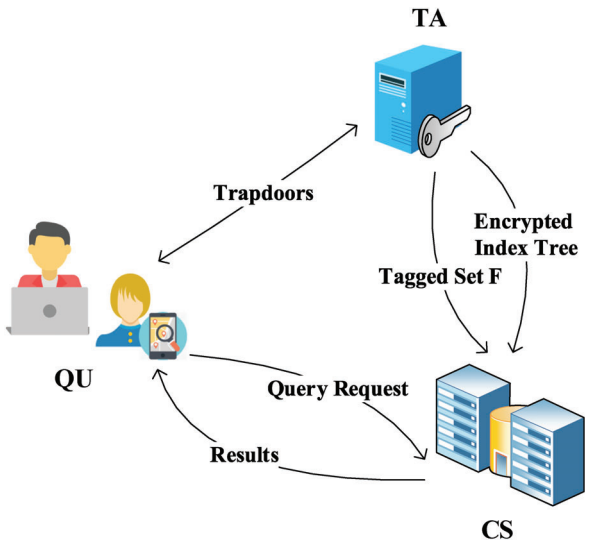


FIGURE 3 Data query phase

Algorithm 2. Depth-first search

input: root node u
output: the F_d in node u with the highest score

```
if the node u is not a leaf node then
    if RT_u(D_u, Q)> high score then
        highscore = RT_u(D_u, Q);
        DFS(u.lchild);
        DFS(u.rchild);
    else
        return 0
    end if
else
    if RT_u(D_u, Q)> high score then
        highscore = RT_u(D_u, Q), highFID = u.FID;
    end if
end if
return high FID.
```

$$= \vec{D_i}' \cdot \vec{Q}' + \vec{D_i}^{\varepsilon} \cdot \vec{Q}^{\varepsilon}$$
$$= D_i \cdot Q + \sum \xi_v,\ v \in \{j \mid Q[n+j] = 1\}. \tag{10}$$

RT_u refers to the similarity score between the trapdoor and each tree node u. RT_u is calculated based on Equation (10). Upon obtaining the high FID, the CS queries the newest record named high FID and returns the result to the QU.

5 | SECURITY AND PRIVACY ANALYSIS

In this section, we discuss the privacy security of the system under the threat model (defined in Section 3.2).

Identity anonymity. The identities of users participating in our scheme are protected, and the adversaries are unable to obtain the sensitive information involved in the uploaded data from the users. In our PDAQ scheme, the SU uses the pseudonym instead of their identities. The random generation mechanism guarantees the pseudonym unlinkability by proving that CS and adversaries cannot distinguish differences between any two pseudonyms from the same identities. Moreover, the pseudonym has a certain term of validity, if the time is expiration, SU must register a new

pseudonym from TA. The randomness and unlinkability of pseudonym mechanism further improve the security of hiding the user's real identity and prevent the statistical analysis attack of the adversary.

Location privacy. In our PDAQ scheme, the station group number g_j and the location tag LI_i are encrypted by the Paillier public key to obtain the data tuples and uploaded to the CS. CS couldn't establish a connection between the station and station group number g_j. We can use the security proof method often used in cryptography to define the concept of location privacy. Set up a security game between a challenger and an adversary, in which the adversary can obtain all the data tuples from the CS.

A challenger sets the security parameter λ of the Paillier encryption algorithm and generates the Paillier private key and public key pair. Then, the challenger sends the Paillier public key to an adversary.

The adversary randomly selects two different location tags LI_1, LI_2, since the location tag $LI_j = (z_{j,1}, z_{j,2}, \ldots, z_{j,h})$ where $z_{j,i}$ can only be represented by 1 or 0, then sends LI_1 and LI_2 to the challenger.

The challenger runs Paillier encryption algorithm to obtain E_{LI_0} and E_{LI_1}, and sends the result to the adversary. A fair random bit b is selected and the challenger computes and sends E_{LI_b} to the adversary.

The adversary outputs its best guess for b' and wins this game if $b = b'$.

$\mathcal{A}$ outputs $b' \in \{0, 1\}$ and if $b' = 0$, $\mathcal{A}'$ outputs $\eta = 0$, thus ξ is a valid tuple; otherwise, $\mathcal{A}'$ outputs $\eta = 1$, thus ξ is random. Let $Win(\mathcal{A}, \lambda) := Pr[b = b']$ be the probability that adversary wins this game.

For any polynomial time adversary $\mathcal{A}$, $Win(\mathcal{A}, \lambda) \leq 1/2 + \epsilon$, where ϵ is a negligible number.

Proof. In the case of $\eta = 1$, $\mathcal{A}$ gets no information about b, therefore, we have $Pr[b' \neq b | \eta = 1] = 1/2$. When $b' \neq b$, $\mathcal{A}'$ outputs a random guess η', thus we have $Pr[\eta' = \eta | \eta = 1] = 1/2$. In the case $\eta = 0$, $\mathcal{A}$ gets a valid ciphertext. In this case, its advantage is ϵ and thus we have $Pr[b' = b | \eta = 0] = \epsilon + 1/2$. Let σ_1 be the event that $\mathcal{A}'$ solves the Paillier encryption algorithm with the random guess $Pr[\sigma_1] = 1/2$. The advantage of $\mathcal{A}'$ in solving the Paillier encryption algorithm is $\epsilon = 1/2 Pr[\eta' = \eta | \eta = 0] + 1/2 Pr[\eta' = \eta | \eta = 1] - 1/2$. Therefore, ϵ is a negligible number. ∎

Based on the aforementioned analysis, any two location tags encrypted are indistinguishable for $\mathcal{A}$, the location information of users participating in our scheme is protected.

Data integrity. In data aggregation phase, an adversary may intercept and tamper with the SU's location tag E_{LI_j} in the communication channel to affect the aggregation results. However, each SU signs the encrypted location tag E_{LI_j} before uploading, so that the tampered encrypted location tag does not pass through the TA's validation and cannot affect the aggregation results. Moreover, each SU in our scheme has two pseudonyms, before uploading the data tuples, SU randomly chooses one of the pseudonyms. Although CS can obtain the total number of data tuples corresponding to all pseudonyms, CS couldn't distinguish which two pseudonyms belong to the same user. Therefore, it is difficult for CS to obtain the total number of data tuples uploaded by a specific user. The data integrity and aggregation results are protected.

Query privacy. In the data query phase, the QU sends the trapdoor to CS. In order to improve the security, we extend D_i into $n + n'$ dimensions, and choose a random number ξ_j for each extended element $D_i[n + j], j = 1, \ldots, n'$, then the trapdoor is encrypted with the private key of secure kNN algorithm. Therefore, the corresponding query trapdoor generated by the same keyword is different. When the F_d mapping relationship with keywords is confidential, it is difficult for CS and adversary to infer the QU's query content based on the query request.

6 | PERFORMANCE ANALYSIS

The SU is performed on a mobile smart device with 2 GHz CPU, Android 7.1.2., 32 GB RAM. The CS and TA are performed on the personal computers with 3.40GHz Intel(R) Core(TM) and 16 GB RAM. The scheme is implemented in BigInteger library and security library.

To achieve the semantic security of Paillier cryptosystem, we set N to 1024 bits. We use the Guangzhou metro station as an experimental case, the total number of stations in Guangzhou metro is 175, we set the key dimension n of the secure kNN to 175, and the extended dimension n' to 17, the dimension S is 192.

The experiment measures the performance of four phases. The cost in the data aggregation phase is mainly the time of collecting data and performing aggregation statistics. The performance of the data query phase is mainly reflected in the query cost, including the computational cost to generate the query trapdoors and the time to obtain the query results.

We have implemented the scheme client application on android smartphone. SU and QU can use this client to sense data and query the metro passenger flow, respectively.

6.1 | Performance of initialization and registration phase

We evaluate the execution time in initialization and registration phase. Table 2 gives the computational overhead of the pseudonym list initialization in TA, and the breakdown into the various operations of the system initialization process are shown in Table 3. We can see that the factors that affect the performance of initialization phase are the generation of the pseudonym list.

TABLE 2 The computational overhead of the pseudonym list initialization

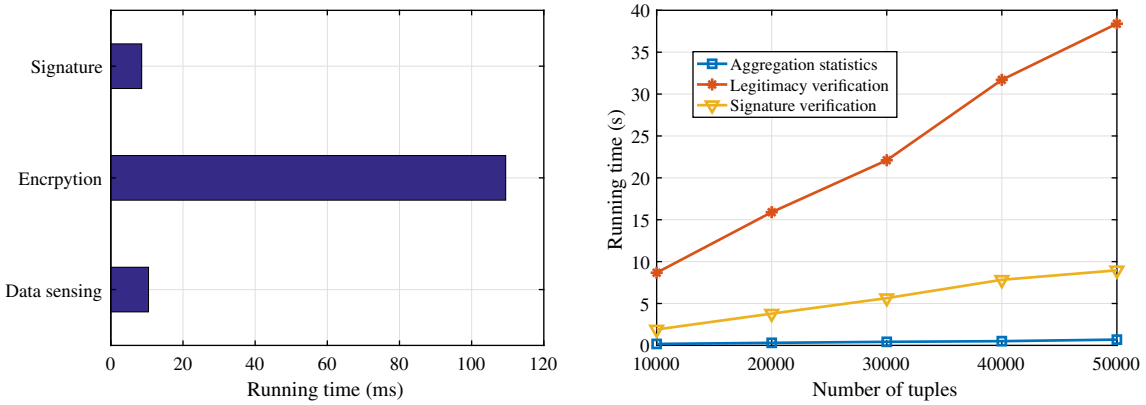
Initialization	Overhead
Setup without initializing the pseudonym list	0.412 s
Setup with initializing 1000 records of the pseudonym list	30.362 s
Setup with initializing 5000 records of the pseudonym list	151.75 s
Setup with initializing 10^4 records of the pseudonym list	296.51 s

TABLE 3 The computational overhead of system initialization

Key generation of Paillier	Key generation of TA	Station list generation
0.061 s	0.085 s	0.003 s

TABLE 4 The computational overhead of SU in registration phase

Registration overhead	Upload overhead
33.41 ms	128.45 ms



(A) The cost of SU in data upload process (B) The cost of CS in data aggregation phase

FIGURE 4 Performance of data aggregation phase. (A) The cost of SU in data upload process. (B) The cost of CS in data aggregation phase

Balance the relation between the cost and the number of records, we can set the initial number of the records to 5000, when TA initializes the pseudonym list. Then we create the corresponding thread to generate the new records during the operation of the system.

Table 4 shows the overhead of SU in the registration and the upload process. The time of the registration process is 33.41 ms and the latency of the upload process is 128.45 ms. The overhead spent on registration and upload process is low and can be basically ignored.

6.2 | Performance of data aggregation phase

We evaluate the execution time in data aggregation phase. Data upload is a process frequently performed by SU, we give the detailed experiment of data upload, as shown in Figure 4A. The figure shows the cost of SU in the data upload, which includes data sensing, data encryption, and data signature. The time of data sensing has already included the location time and the generation time of the location tag. The most of the overhead comes from the data encryption, while SU spends less time on data sensing and data signature.

When receiving the uploaded tuples, CS need to perform the signature verification, legitimacy verification for E_{LI_j} and E_{sum}, and implement the aggregation statistics. The cost of signature verification includes the cost to verify the signature of the pseudonym, the signature of E_{LI_j} and the signature of E_{sum}. Figure 4B shows the overhead of CS to perform the above processes. We can see that the runtime of each process increases linearly with the number of tuples, among which CS spends the most time on legitimacy verification of E_{LI_j} and E_{sum}, and spend the minimum time executing aggregation statistics.

In order to demonstrate the performance of CS, we implement many simultaneous upload processes to the CS to measure its throughput, are shown in Table 5. We tested the throughput in the same experimental platform with scheme.[18] As seen in Table 5, if there is no need to verify the data legitimacy, the number of packets that can be processed by each thread of the CS in 1 min is 1098 packets more than that of the scheme.[18] When all the uploaded data legitimacy needs to be validated, the packet processing efficiency of per thread within 1 min is about 10 times faster than that of scheme.[18] In short, the CS's performance of our system in sensing phase is more efficient than that of the scheme.[18]

However, the CS may have some verification errors when performing the legitimacy verification process, as shown in Table 6. As seen from the table, when the range of random numbers is larger, the error rate is smaller, but the average time to verify an upload is larger. In order to weigh down the error rate and verification time, we set the range of random numbers to 1000.

6.3 | Performance of the data query phase

In data query phase, we tested the average query time of QU, the performance of CS and TA in handling multi-querier's query requests. Table 7 gives the computational overhead of index generation in TA, including the initializing F_d and building index tree Γ. The query time of QU can be divided into the time to acquire trapdoor and obtain the query results. The query time of QU is shown in Figure 5, we can see that the overhead of getting the results is higher than that of getting trapdoor, and the total query time is about 80 ms. It demonstrates that our system can respond quickly to QU's query requests.

TABLE 5 The throughput of CS

Schemes	Throughput without legitimacy verification	Throughput with legitimacy verification
Scheme[18]	2400 uploads/thread/min	170 uploads/thread/min
PDAQ	3498 uploads/thread/min	1835 uploads/thread/min

TABLE 6 The error rate of legitimacy verification

Range of random number	Error rate	Average time of verification
100	2.89%	0.05 s
1000	0.23%	0.455 s

TABLE 7 The computational overhead of index generation

Initializing F_d	Building index tree Γ
0.09 s	0.172 s

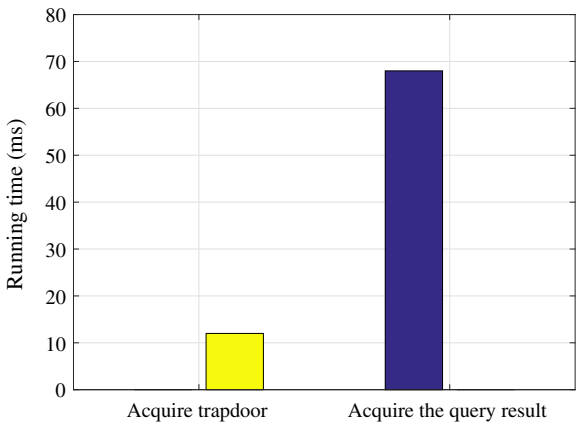


FIGURE 5 The query time of QU

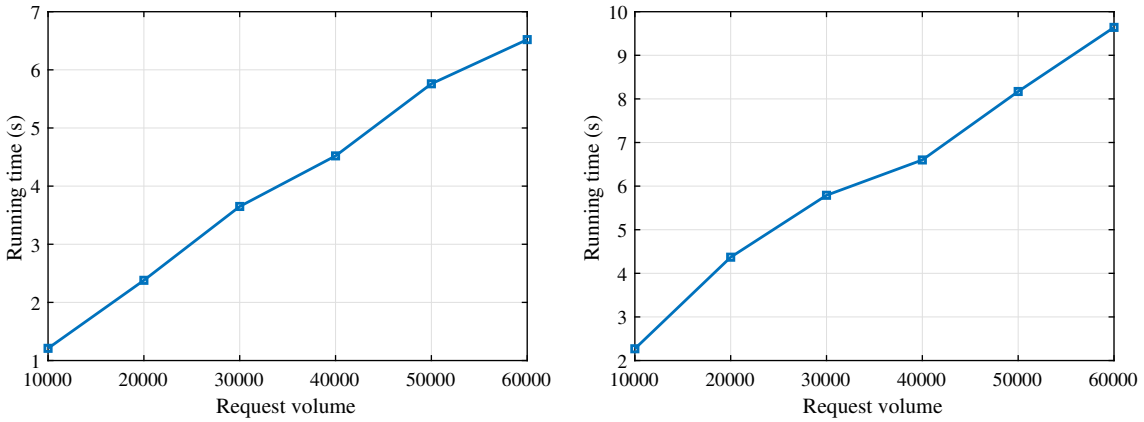


FIGURE 6 Performance of data query phase. (A) The cost of trapdoor generation. (B) The cost of responding to query requests

The cost of TA to generate query trapdoor for QU is shown in Figure 6A. From Figure 6A, TA only spent 6.52 s when processing 60,000 requests. This shows that TA has a high processing efficiency in trapdoors generation.

Figure 6B shows the overhead of the CS dealing with query requests. We can see that the time of CS performing query requests is linearly related to the number of query requests. It indicates that the CS has reasonable performance in responding to query requests.

7 | CONCLUSION

To prevent the privacy leakage of MCS applications applied in metro passenger flow query, in this article, we proposed a PDAQ scheme based on MCS, which protected the identity and location of the sensing users by adopting Paillier cryptosystem and pseudonym mechanism, and used the secure kNN algorithm to protect the query privacy. Furthermore, our PDAQ scheme achieved the sensing user accountability mechanism without revealing the identity information to the untrusted entities. Finally, we gave the security analysis, the performance evaluation showed the time cost of each phase, and the experimental result demonstrated the PDAQ scheme in practice.

In our future research, we will focus on the effective incentive mechanism to attract more sensing users and build the passenger flow prediction model with large-scale dataset.

ACKNOWLEDGMENTS

This work was supported in part by the National Key Research and Development Program of China under grant numbers 2019YFA0706200 and 2019YFB1703600, the National Natural Science Foundation of China under grant numbers 61702195, 61751202, U1813203, U1801262, and 61751205.

DATA AVAILABILITY STATEMENT

Data are available on request from the authors.

ORCID

Yuanyuan Zhang https://orcid.org/0000-0002-2302-836X

REFERENCES

1. Zhang C, Zhu L, Ni J, Huang C, Shen X. Verifiable and privacy-preserving traffic flow statistics for advanced traffic management systems. *IEEE Trans Veh Technol*. 2020;69(9):10336-10347.
2. Xiong J, Ma R, Chen L, et al. A personalized privacy protection framework for mobile crowdsensing in IIoT. *IEEE Trans Ind Inform*. 2019;16(6):4231-4241.
3. Zhao B, Tang S, Liu X, Zhang X, Chen WN. iTAM: bilateral privacy-preserving task assignment for mobile crowdsensing. *IEEE Trans Mob Comput*. 2020;20(12):3351-3366.
4. Ma M, Preum SM, Ahmed MY, Tärneberg W, Hendawi A, Stankovic JA. Data sets, modeling, and decision making in smart cities: a survey. *ACM Trans Cyber-Phys Syst*. 2019;4(2):1-28.
5. Zhang Y, Chen CP. Secure heterogeneous data deduplication via fog-assisted mobile crowdsensing in 5G-enabled IIoT. *IEEE Trans Ind Inform*. 2021;18(4):2849-2857.

6. Xiong J, Zhao M, Bhuiyan MZA, Chen L, Tian Y. An AI-enabled three-party game framework for guaranteed data privacy in mobile edge crowdsensing of IoT. *IEEE Trans Ind Inform*. 2021;17(2):922-933.
7. Weppner J, Lukowicz P. Bluetooth based collaborative crowd density estimation with mobile phones. Proceedings of the 2013 IEEE International Conference on Pervasive Computing And Communications (PerCom); 2013:193-200.
8. Thiagarajan A, Ravindranath L, Lacurts K, et al. VTrack:accurate, energy-aware road traffic delay estimation using mobile phones. Proceedings of the ACM Conference on Embedded Networked Sensor Systems; 2009:85-98.
9. Koukoumidis E, Peh LS, Martonosi MR. Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory. Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services; 2011:127-140.
10. Xiong J, Bi R, Tian Y, Liu X, Ma J. Security and privacy in mobile crowdsensing: models, progresses, and trends. *Chin J Comput*. 2021;44(9):1949-1966.
11. Xia Z, Wang X, Sun X, Wang Q. A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. *IEEE Trans Parallel Distrib Syst*. 2016;27(2):340-352.
12. Xiong J, Ren J, Chen L, et al. Enhancing privacy and availability for data clustering in intelligent electrical service of IoT. *IEEE Internet Things J*. 2018;6(2):1530-1540.
13. Hoh B, Gruteser M, Herring R, et al. Virtual trip lines for distributed privacy-preserving traffic monitoring. Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services; 2008:15-28.
14. Hoh B, Gruteser M, Xiong H, Alrabady A. Preserving privacy in gps traces via uncertainty-aware path cloaking. Proceedings of the 14th ACM Conference on Computer and Communications Security; 2007:161-171; ACM, New York, NY.
15. Gruteser M, Grunwald D. Anonymous usage of location-based services through spatial and temporal cloaking. Proceedings of the 1st International Conference on Mobile Systems, Applications and Services; 2003:31-42.
16. Gedik B, Liu L. Location privacy in mobile systems: a personalized anonymization model. Proceedings of the 2005 IEEE 25th International Conference on Distributed Computing Systems(ICDCS); 2005:620-629.
17. Krumm J. Inference attacks on location tracks. Proceedings of the International Conference on Pervasive Computing; 2007:127-143.
18. Popa RA, Blumberg AJ, Balakrishnan H, Li FH. Privacy and accountability for location-based aggregate statistics. Proceedings of the 18th ACM Conference on Computer and Communications Security; 2011:653-666.
19. Chen J, Ma H, Wei DS, Zhao D. Participant-density-aware privacy-preserving aggregate statistics for mobile crowd-sensing. Proceedings of the 2015 IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS); 2015:140-147; IEEE.
20. Yang Z, Wang R, Wu D, Wang H, Song H, Ma X. Local trajectory privacy protection in 5G enabled industrial intelligent logistics. *IEEE Trans Ind Inform*. 2021;18(4):2868-2876.
21. Zhang J, Yang F, Ma Z, Wang Z, Liu X, Ma J. A decentralized location privacy-preserving spatial crowdsourcing for internet of vehicles. *IEEE Trans Intell Transp Syst*. 2020;22(4):2299-2313.
22. Wu D, Yang Z, Yang B, Wang R, Zhang P. From centralized management to edge collaboration: a privacy-preserving task assignment framework for mobile crowdsensing. *IEEE Internet Things J*. 2020;8(6):4579-4589.
23. Chow CY, Mokbel MF, Aref WG. Casper: query processing for location services without compromising privacy. *ACM Trans Database Syst (TODS)*. 2009;34(4):24.
24. Ghinita G, Kalnis P, Khoshgozaran A, Shahabi C, Tan KL. Private queries in location based services: anonymizers are not necessary. Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data; 2008:121-132; ACM, New York, NY.
25. Niu B, Li Q, Zhu X, Cao G, Li H. Achieving k-anonymity in privacy-aware location-based services. Proceedings of the IEEE INFOCOM 2014-IEEE Conference on Computer Communications; 2014:754-762; IEEE.
26. Long S, Long W, Li Z, Li K, Xia Y, Tang Z. A game-based approach for cost-aware task assignment with QoS constraint in collaborative edge and cloud environments. *IEEE Trans Parallel Distrib Syst*. 2020;32(7):1629-1640.
27. Guo B, Yu Z, Zhou X, Zhang D. From participatory sensing to mobile crowd sensing. Proceedings of the 2014 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops); 2014:593-598; IEEE.
28. Zhao B, Tang S, Liu X, Zhang X. PACE: privacy-preserving and quality-aware incentive mechanism for mobile crowdsensing. *IEEE Trans Mob Comput*. 2021;20(5):1924-1939.

How to cite this article: Zhang Y, Ying Z, Zhao B, Chen CLP. A privacy preserving data aggregation and query for metro passenger flow via mobile crowdsensing. *Concurrency Computat Pract Exper*. 2022;e6965. doi: 10.1002/cpe.6965