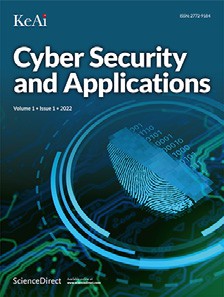
[Cyber Security and Applications 1 (2023) 100007](https://doi.org/10.1016/j.csa.2022.100007)

Contents lists available at [ScienceDirect](http://www.ScienceDirect.com/)

Cyber Security and Applications

journal homepage: <http://www.keaipublishing.com/en/journals/cyber-security-and-applications/>

PPT-LBS: Privacy-preserving top-k query scheme for outsourced data of location-based services

Yousheng Zhou[a](#_bookmark0), Xia Li [a](#_bookmark0),[∗](#_bookmark2), Ming Wang[b](#_bookmark1), Yuanni Liu[a](#_bookmark0)

a *School of Cyber Security and Information Law, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*

b *College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*

a r t i c l e i n f o a b s t r a c t

*Keywords:*

Privacy protection Location-based services Top-k query Outsourcing computing

Location-based service (LBS) is enjoying a great popularity with the fast growth of mobile Internet. As the volume of data increases dramatically, an increasing number of location service providers (LSPs) are moving LBS data to cloud platforms for benefit of affordability and stability. However, while cloud server provides convenience and stability, it also leads to data security and user privacy leakage. Aiming at the problems of insuﬃcient privacy protection and ineﬃcient query in the existing LBS data outsourcing schemes, this paper presents a novel privacy-preserving top-k query for outsourcing situations. Firstly, to ensure data security of LSP and privacy of the user, the enhanced asymmetric scalar-product preserving encryption and public key searchable encryption have been adopted to encrypt outsourced data and LBS query, which can effectively lower the computational cost and realize the privacy protection search. Secondly, an eﬃcient and secure index structure is constructed by using a coded quadtree and the bloom filter, so that the cloud server can quickly locate the user’s query region to improve retrieval eﬃciency. Finally, the formal security analysis is given under the random oracle model, and the performance is evaluated by experiments which demonstrates that our scheme is preferable to existing schemes.

# Introduction

Location-based service (LBS) has been used in many fields, such as military, medical treatment, emergency rescue, etc., due to the rapid popularity of mobile devices [[1]](#_bookmark28). However, as the upsurge of LBS dataset, LBS data’s high storage and computation costs produce a heavy burden on location service providers (LSPs). The rapid development of cloud computing provides a new operation mode for LBS, that is, LSP up- loads a large amount of LBS data onto the cloud to process user’s queries with its powerful computing power, which effectively reduces LSP cost. However, cloud computing brings data security and user privacy prob- lems while facilitating data computing and storage. In a traditional com- puting mode, the user’s data is usually on the LSP-controlled or trusted platform for processing. Still, after outsourcing data to a cloud server (CS), the data’s physical control capability is handed over to the cloud. In the outsourcing environment, the CS is usually supposed to be “semi- honest”, it will perform the user’s query request honestly. Meanwhile, it also attempts to derive useful information from the user’s query and the stored data [[2–5]](#_bookmark29). Hence, LBS data secure storage and computation in an untrusted cloud environment have become a critical issue that needs to be resolved urgently.

To achieve the LBS system’s privacy protection in an outsourcing

environment, researchers present a series of location privacy protection

∗ Corresponding author.

*E-mail address:* [s190802004@stu.cqupt.edu.cn](mailto:s190802004@stu.cqupt.edu.cn) (X. Li).

methods. Zeng et al. [[6]](#_bookmark31) presented a new search scheme based on en- hanced asymmetric scalar-product preserving encryption algorithm and encrypted inverted index technology to support generic LBS query over encrypted data for cloud environment. In this scheme, the user can spec- ify the geographical scope and search keywords. After searching, the CS returns the point of interest (POI) records which matches the given area and keywords. Yang et al. [[7]](#_bookmark32) presented a verifiable privacy pro- tection scheme for kNN query under road network environment based on Voronoi diagram, 2-Hop tag index, and some cryptographic prim- itive, which could simultaneously preserve the privacy of spatial data and kNN query, and verify the reliability of query results. Xie and Wang

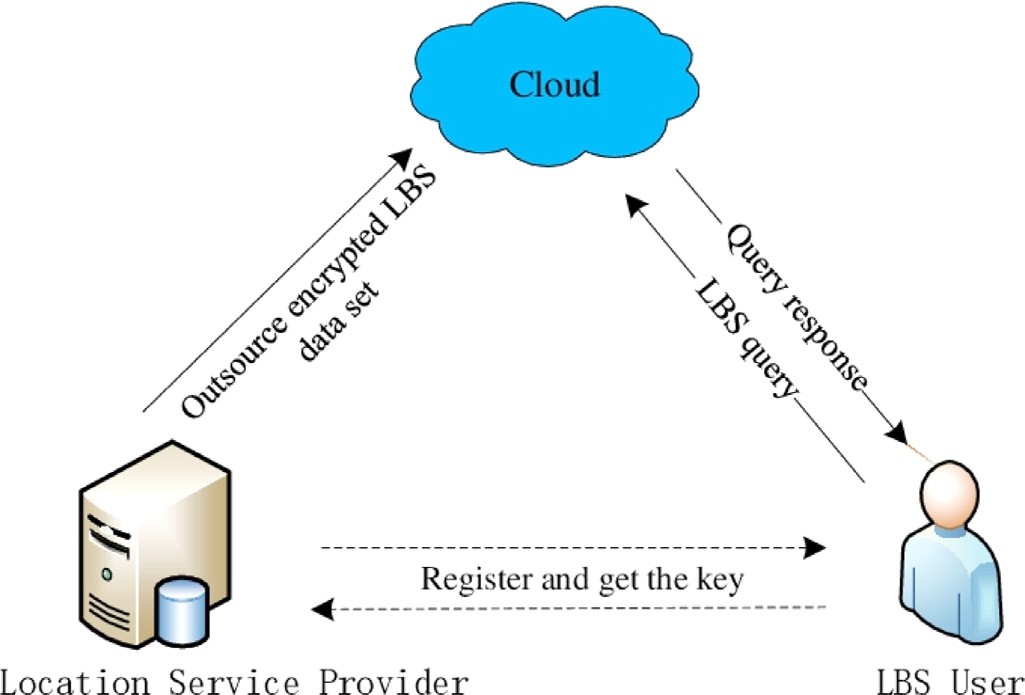
[[8]](#_bookmark33) put forward an LBS privacy protection scheme based on attribute en- cryption of ciphertext policy (CP-ABE), which calculate and compare lo- cation distance for the authorized user without revealing its privacy. Zhu et al. [[9,10]](#_bookmark34), aiming at the privacy and eﬃciency problems in the LBS range query service process, designed an LBS circle region and polygon region range query scheme with eﬃcient privacy protection by using an improved homomorphic encryption algorithm, which can provide query service while ensuring the user’s query privacy and the data confiden- tiality of LSP. However, the existing research on privacy protection for LBS data outsourcing mainly focuses on the geographic scope or interest keywords query and rarely supports top-k query. Besides, the retrieval algorithm of LBS data fails to protect the user’s search mode.

<https://doi.org/10.1016/j.csa.2022.100007>

Received 19 April 2022; Received in revised form 10 June 2022; Accepted 18 August 2022

Available online 23 August 2022

2772-9184/© 2022 The Authors. Published by Elsevier B.V. on behalf of KeAi Communications Co., Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)



**Fig. 1.** The system model.

Our contributions are outlined as follow:

1. Firstly, we present a top-k query scheme for LBS outsourced data. To protect the LBS data and user’s query against the attacker, the en- hanced asymmetric scalar-product preserving encryption and public key searchable encryption are utilized to construct our secure query scheme while supporting accurate top-k search over the encrypted LBS data.

to study useful information from the user’s query and stored LBS data. As with other related studies [[2,3,6,7,9]](#_bookmark29), LSP and user are assumed to be honest, and LSP and user do not conspire with CS to obtain other user’s privacy. Therefore, to protect the LBS data and the user’s query, our scheme achieves the following security objectives:

1. **Confidentiality:** The CS cannot understand any content of LBS data stored by LSP. The outsourced LBS data is encrypted to prevent the CS from obtaining any valid information from the data set.
2. **Privacy:** The user’s query should be confidential to the CS since it contains private information, such as area and personal interest. To ensure that the user’s query privacy is not disclosed, this paper encrypts the user’s query request and submits it to CS in the form of a query trap- door, thus preventing CS from obtaining any useful information about the user’s query.

# Preliminaries

Some basics used in our scheme are introduced in this section, includ- ing bilinear pairing map, hard problem assumptions, and bloom filter.

* 1. *Bilinear pairing map*

𝔾 and 𝔾*𝑇* are two *𝑞*-order multiplication cyclic groups, with the gen- erator *𝑔* of 𝔾. *𝑒* ∶ 𝔾 × 𝔾 → 𝔾*𝑇* is a bilinear map, and it has the follow

properties:

* + 1. **Bilinearity:** for any *𝑥, 𝑦* ∈ ℤ∗, there is *𝑒*(*𝑔𝑥, 𝑔𝑦*) = *𝑒*(*𝑔, 𝑔*)*𝑥𝑦*.

1. Secondly, we construct a secure index structure based on coded

quadtree and bloom filter, which enables the CS to fast locate the user’s

# Non-degenerate:

*𝑞*

*𝑒*(*𝑔, 𝑔*) ≠ 1.

query region with the user’s query request, thereby improving the re- trieval eﬃciency.

1. Lastly, we conduct formal security proof and experiments based performance analysis, which exhibit that our presented scheme is secure and eﬃcient.

**Organization.** The remainder of our thesis is organized as follows. [Section 2](#_bookmark5) introduces the system model and design goals. In [Section 3](#_bookmark3), we give some preliminaries. Then, the detailed construction of the proposed scheme is illustrated in [Section 4](#_bookmark6). The security analysis and performance evaluation are demonstrated in [Section 5](#_bookmark13) and [Section 6](#_bookmark15), respectively. Finally, the paper is concluded in [Section 7](#_bookmark20).

# Problem formulations

(PPT) algorithm to evaluate *𝑒*(*𝑔, 𝑔*). **(3) Computability:** there exists a probabilistic polynomial-time

* 1. *Hard problem assumptions*

that there is a *𝑞*-order group 𝔾 and a generator *𝑔*. Provided with **Decisional Bilinear Diffie-Hellman (DBDH) Assumption:** Assume

*𝑔𝑎, 𝑔𝑏, 𝑔𝑐* ∈ 𝔾*, 𝑍* ∈ 𝔾*𝑇* , where *𝑎, 𝑏, 𝑐* ∈ ℤ∗, the DBDH problem is to de-

*𝑞*

cide *𝑍* =? *𝑒*(*𝑔, 𝑔*)*𝑎𝑏𝑐* .

**m Decisional Linear (mDLIN) Problem:** The mDLIN problem is a

order is *𝑞* and the generator is *𝑔*. Provided with *𝑔𝑎, 𝑔𝑏, 𝑔𝑟𝑎, 𝑔𝑡*∕*𝑏, 𝑔𝑐* , where variant of the DLIN problem. Assume that there is a group 𝔾, where the

*𝑎, 𝑏, 𝑟, 𝑡, 𝑐* are randomly chosen from ℤ∗. mDLIN problem is to determine

*𝑞*

* 1. *System model*

Our scheme aims to provide the user with secure and eﬃcient query services while ensuring LBS data security and user query privacy. There are three entities: the location service provider (LSP), the cloud server(CS), and the LBS user, as shown in [Fig. 1](#_bookmark4).

* + - **Location Service Provider (LSP):** An LSP owns a large number of

LBS resources. It outsources massive LBS data to the CS to benefit from

cheap storage and reliable computation services. To guarantee LBS data confidentiality, each LBS data will be encrypted at first and then up- loaded to the cloud. Besides, LSP also provides registration service for the LBS user. Once the user passes registration, LSP sends an authenti- cation certificate and the key to the user via a secure communication channel.

* + - **Cloud Server (CS):** CS has abundant storage and computing re-

sources, it is responsible for storing ciphertext data sets from LSP and

providing LBS query services for users.

*𝑔𝑐* =? *𝑔𝑟*+*𝑡* .

* 1. *Bloom filter*

the current set, the core is a bit array and *𝑡* independent hash function The bloom filter can easily determine whether an element belongs to

*𝐻𝑖* ∶ {0*,* 1}∗ → {1*,* 2*,* … *,* … *, 𝑚*}*, 𝑖* ∈ [1*, 𝑡*], the initial state of the values in the array is 0. Given a set *𝑋* = {*𝑥*1 *, 𝑥*2 ⋯ *𝑥𝑛* }, select *𝑡* hash functions to

map each element to the bloom filter and set the bit position of each generated hash value to 1.

When verifying whether an certain element *𝑥𝑖 , 𝑖* ∈ [1*, 𝑛*] exists in the

set *𝑋*, use *𝑡* hash function to map *𝑥𝑖* to the bloom filter. If one of the corresponding positions is 0, then *𝑥𝑖* must not belong to *𝑋*.If each of these corresponding positions is 1, then maybe *𝑥𝑖* is a member of *𝑋*, as

shown in [Fig. 2](#_bookmark7).

The error rate *𝑝* of the bloom filter is defined as:

−*𝑡𝑛 𝑡*

* **LBS User:** A LBS user first registers with LSP to obtain the key.

*𝑝* = 1 − (*𝑒 𝑚* )

(1)

In order to protect privacy, the query requests of users are submitted to the CS in the way of trapdoor.

* 1. *Design goals*

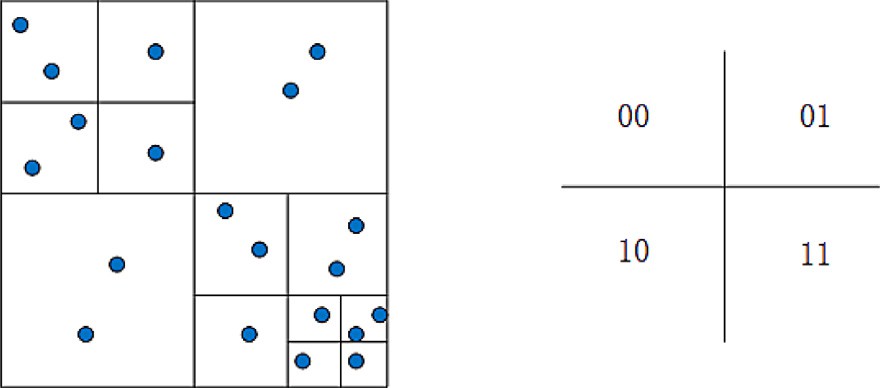
In this paper, it is assumed that CS is “honest but curious”. In other words, it will execute the user’s query request honestly, but it also tries

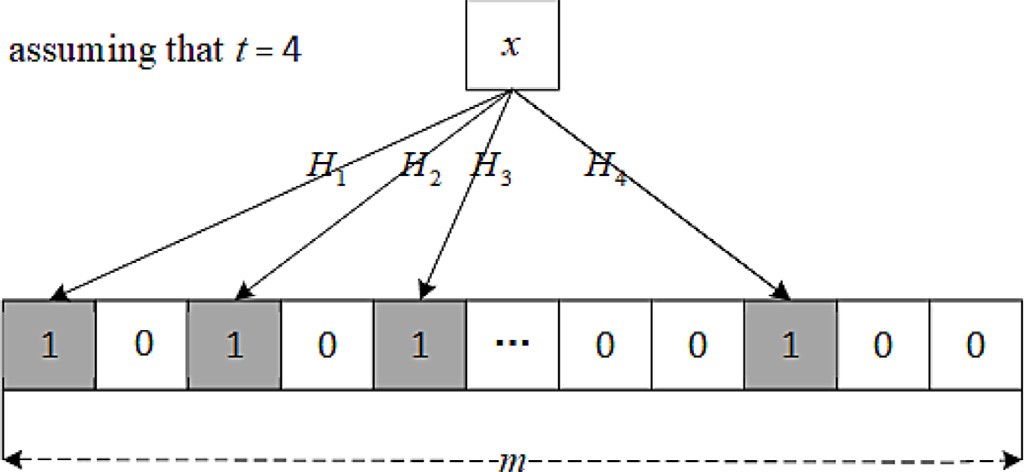
where *𝑚* is the length of bit array, *𝑛* denotes the size of *𝑋*,and *𝑡* denotes

the amount of hash functions [[3]](#_bookmark30).

# Construction of PPT-LBS

The concrete construction of our proposed scheme is described in this section. Our scheme is improved based on Zhao et al. [[11]](#_bookmark21), which



**Fig. 3.** Map devision.

**Table 1**

**Fig. 2.** Schematic diagram of bloom filter.

(1) The user randomly selects *𝑥𝑢*

*𝑥𝑢*, the public key is *𝑃 𝐾𝑢* = *𝑔𝑥𝑢* ;

*𝑞*

∈ ℤ∗ and sets its private key *𝑆𝐾𝑢* =

Symbols definition.

Notations Definition

***M***1 *,* ***M***2 invertible matrix

*𝑙* matrix dimension, set *𝑙* = 80 in this scheme

*𝑏* ∈ {0*,* 1}*𝑙* a bit string of length *𝑙*

((*𝑥, 𝑦*)*, 𝑤*) one POI data

((*𝑥**𝑢, 𝑦𝑢* )*, 𝑅*) the user’s coordinate is *𝑥𝑢 , 𝑦𝑢* and the query radius is *𝑅*

*( )*

*𝑤𝑞* user’s query keyword

*𝑘* the user want to query the number of POIs closest to his/her location

*𝑄* the intersection region

ℜ∗ a set of all expanded code sequences

enables users can perform keyword query of POI and top-k query. For example, Alice wants to query the three restaurants closest to him/her within 500 m nearby. At the same time, under the premise of ensuring that her location and query keywords are not leaked, the CS returns the POI records that meet Alice’s conditions.

The proposed scheme is made up of eight phases: system initializa- tion, key generation, user registration, LBS data encryption, index con- struction, LBS query generation, LBS data retrieval, respectively. Sym- bols used are defined as in [Table 1](#_bookmark8).

* 1. *System setup*

System setup is used to generate some public parameters, which are performed by LSP. With the following steps:

* + 1. LSP selects two multiplication cyclic groups 𝔾 and 𝔾*𝑇* with the large prime order *𝑞*. Let *𝑔* be a generator of 𝔾 and a bilinear pairing map
    2. The user calculates registration information *𝑀𝐼𝐷𝑢* = *𝐻*0(*𝐼𝐷𝑢*),

and send it to the LSP for registration request;

sent by the user, LSP computes an authentication certificate *𝐶𝑢* = (3) When the LSP receives the registration request message

*𝑒*((*𝑀𝐼𝐷𝑢*)*𝑆𝐾𝐿 , 𝑔*) for the user and send the *𝑘𝑒𝑦* to the user via secure

communication channel;

(4) LSP sends the *𝐶𝑢* to the CS, which stores it in the user list for

subsequent user authentication.

* 1. *LBS Data encryption*

The POI data set stored in the LSP is represented as ***DB*** =

{*𝑑*1*, 𝑑*2*,* … *, 𝑑𝑛*}. To be simple, use *𝑑𝑖* = {(*𝑥, 𝑦*)*, 𝑤*} to represent one of POI record, where (*𝑥, 𝑦*) means the coordinates of the POI, *𝑤* shows the key-

word of the POI(e.g., restaurants, hotels, bars, etc.). In order to ensure data confidentiality, LSP outsources data set ***DB*** to the CS in the form of ciphertext.

# Coordinate Encryption

For POI coordinate (*𝑥, 𝑦*), LSP generates an *𝑙*-dimensional vector ***p****̂*, where the first three dimensions of data is (*𝑥, 𝑦,* −0*.*5(*𝑥*2 + *𝑦*2)). For *𝑗* ∈ [4*, 𝑙*], set ***p****̂*[*𝑗*] = *𝛼* . Then, ***p****̂* is split according to the bit string *𝑏* into two

*𝑗*−3

*𝑙* dimensional vector ***p****̂𝑎* and ***p****̂𝑏*:

* + - * if *𝑏*[*𝑖*] = 1*, 𝑖* ∈ [1*, 𝑙*], set ***p****̂𝑎*[*𝑖*] + ***p****̂𝑏*[*𝑖*] = ***p****̂*[*𝑖*];
      * if *𝑏*[*𝑖*] = 0*, 𝑖* ∈ [1*, 𝑙*], set ***p****̂𝑎*[*𝑖*] = ***p****̂𝑏*[*𝑖*] = ***p****̂*[*𝑖*]. Calculate:***C****𝑎* = ***M***1*𝑇* ***p****̂𝑎,* ***C****𝑏* = ***M***2*𝑇* ***p****̂𝑏*.

Output coordinat ciphertext: *𝐶𝑐𝑜𝑟𝑑* = {***C****𝑎,* ***C****𝑏*}.

# Keyword Encryption

LSP randomly selects *𝑟* ∈ ℤ∗, and derives the keywords ciphertext

*𝐶𝑤* = {*𝐶𝑤*1 *, 𝐶𝑤*2 }, where

*𝑞*

*𝑒* ∶ 𝔾 × 𝔾 → 𝔾*𝑇* ;

(2) LSP selects a one-way hash function *𝐻*

∶ {0*,* 1}∗ → *𝐺* and a hash

*𝐶𝑤* = *𝐻*0(*𝑤*)*𝑆𝐾*

*𝐿* ⋅ *𝑔𝑟,* (2)

family with *𝑡* hash functions *𝐻* ∶ {0 1}∗

1

0

{1 2 … …

} ∈ [1 ],

*𝑖 ,*

→ *, ,*

*, , 𝑚 , 𝑖*

*, 𝑡*

where *𝑚* denotes the size of the bloom filter, whose main function is

to map a bit string into the bloom filter vector.

LSP open public parameters *𝑝𝑎𝑟𝑎𝑚* = {𝔾*,* 𝔾*𝑇 , 𝑔, 𝐻*0*, 𝐻𝑖*}*, 𝑖* ∈ [1*, 𝑡*].

* 1. *Key generation*
     1. LSP selects two *𝑙* × *𝑙* dimensions invertible matrix ***M***1*,* ***M***2, a bit string *𝑏* of length *𝑙* and *𝑙* − 3 random number {*𝛼𝑗* }*𝑗*∈[1*,𝑙*−3]. Here the matrix dimension *𝑙* should be big enough to resist brute force attacks, matrix dimensions are set *𝑙* = 80;
     2. *𝑘𝑒𝑦* = {***M***1*,* ***M***2*, 𝑏,* {*𝛼𝑗* }*𝑗*∈[1*,𝑙*−3]} is stored as a secure key in LSP,

which is used for user to generate trapdoor and decrypt data ;

* + 1. LSP randomly chooses *𝑥𝐿* ∈ ℤ∗ and sets its private key *𝑆𝐾𝐿* = *𝑥𝐿*, the public key is *𝑃 𝐾𝐿* = *𝑔𝑥𝐿* .

*𝑞*

* 1. *User registration*

When a new user join the LBS system, LSP registers the user at this phase.

*𝐶𝑤*2 = *𝑃 𝐾𝑢 .* (3)

So far, a piece of POI ciphertext data *𝐶𝑑* is represented as: *𝐶𝑑* =

*𝑟*

{*𝐶𝑐𝑜𝑟𝑑 , 𝐶𝑤*}.

sented as: ***EDB*** = {*𝐶𝑑* }*, 𝑖* ∈ [1*, 𝑛*]. LSP encrypts all POI data in ***DB***, and the ciphertext data set is repre-

*𝑖*

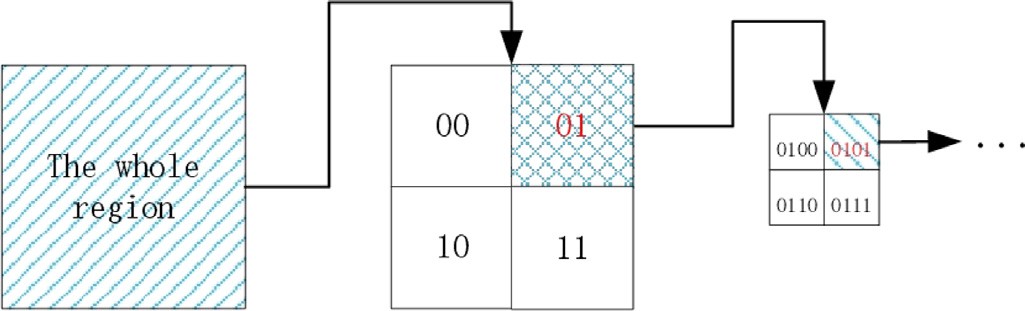
The LSP uploads the ciphertext data set ***EDB*** to the CS.

* 1. *Index construction*

# Binary coding of geographic data

LSP recursively divides the map into four regions until the number of POI stored in each subregion does not exceed a specified threshold. At the same time, LSP selects 00,01,10,11 to denote four regions, each of which can be represented by a bit string, as shown in [Fig. 3](#_bookmark7).

For example, in [Fig. 4](#_bookmark9), the entire region is divided into four subre- gions, encode as 00,01,10,11. The upper right subregion with the bit string “01” is further divided into four subregions encoded as 0100, 0101, 0110, 0111, and the bit string “0101” region is divided again, and so on so that each region is a unique bit string encode.



**Fig. 4.** Hierarchical coding of a region.

# Coded quadtree index construction

The LSP uses a quadtree to store bit string and POI ciphertext data of the above-divided regions. In this coded quadtree index structure, the root node denotes the entire region, and each non-leaf node has 4 child nodes denoting four subregions. Non-leaf nodes store bit strings of each region, and leaf nodes store the ciphertext of POI within the

region. For a given POI data *𝑑*, LSP first encrypts it to *𝐶𝑑* , then find its

subregion encode, and finally adds it to the corresponding leaf node of

the subregion of the coded quadtree.

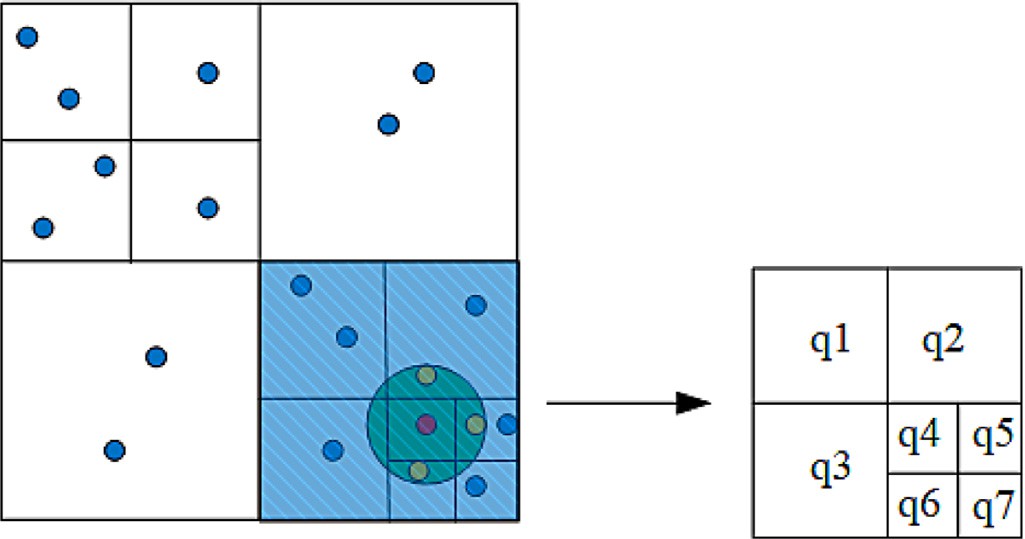
Take the example of inserting data in the region of the bit string as “001001”. It starts from the root node to search the data record pertain- ing to the subregions 00, 010, 01001, respectively. Afterwards, the data record is interjected into subregion 001001, as shown in [Fig. 5](#_bookmark10).

* 1. *LBS Query generation*

An LBS user’s query request can be expressed as:

*{((𝑥𝑢, 𝑦𝑢 ), 𝑅), 𝑤𝑞 , 𝑘}*, where (*(𝑥𝑢, 𝑦𝑢 ), 𝑅*) is the user’s current coor-

**Fig. 6.** Example of user’s query.

* + 1. **Index trapdoor *V*** *𝐵𝐹* **generation**

according to the current position *𝑥𝑢 , 𝑦𝑢* and the query radius *𝑅*, as First of all, the user computes the circular region as the query range

*(* [*)*](#_bookmark9)

the user’s query range and the subregions divided is expressed as *𝑄* = shown in the shaded green part of [Fig. 6](#_bookmark9). The intersecting region of

{*𝑞*1 *, 𝑞*2 *,* … *, 𝑞𝑘* }, as shown in the blue shaded part of [Fig. 6](#_bookmark9).

Once *𝑄* is determined, i.e.,*𝑄* = {*𝑞*1 *, 𝑞*2 *,* … *, 𝑞*7 }, it can be expanded into a series of bit strings by listing all prefix substrings of *𝑞𝑖 , 𝑖* ∈ [1*,* 7]

for every two neighbouring bits.

quence is represented as ℜ*𝑖*, ℜ*𝑖* = {11*,* 1111*,* 111101}. User combine the For instance, suppose a bit string is 111101, the extended code se- extended code sequence of all query subregions *𝑞𝑖* (*𝑖* ∈ [1*, 𝑘*]) to get a set

dinates and the query radius, *𝑤𝑞* is the user’s query keyword, *𝑘* is user

*𝑖*=*𝑘*

of code sequence, ℜ∗ = ℜ*𝑖*. For every bit string in ℜ∗, the user uses

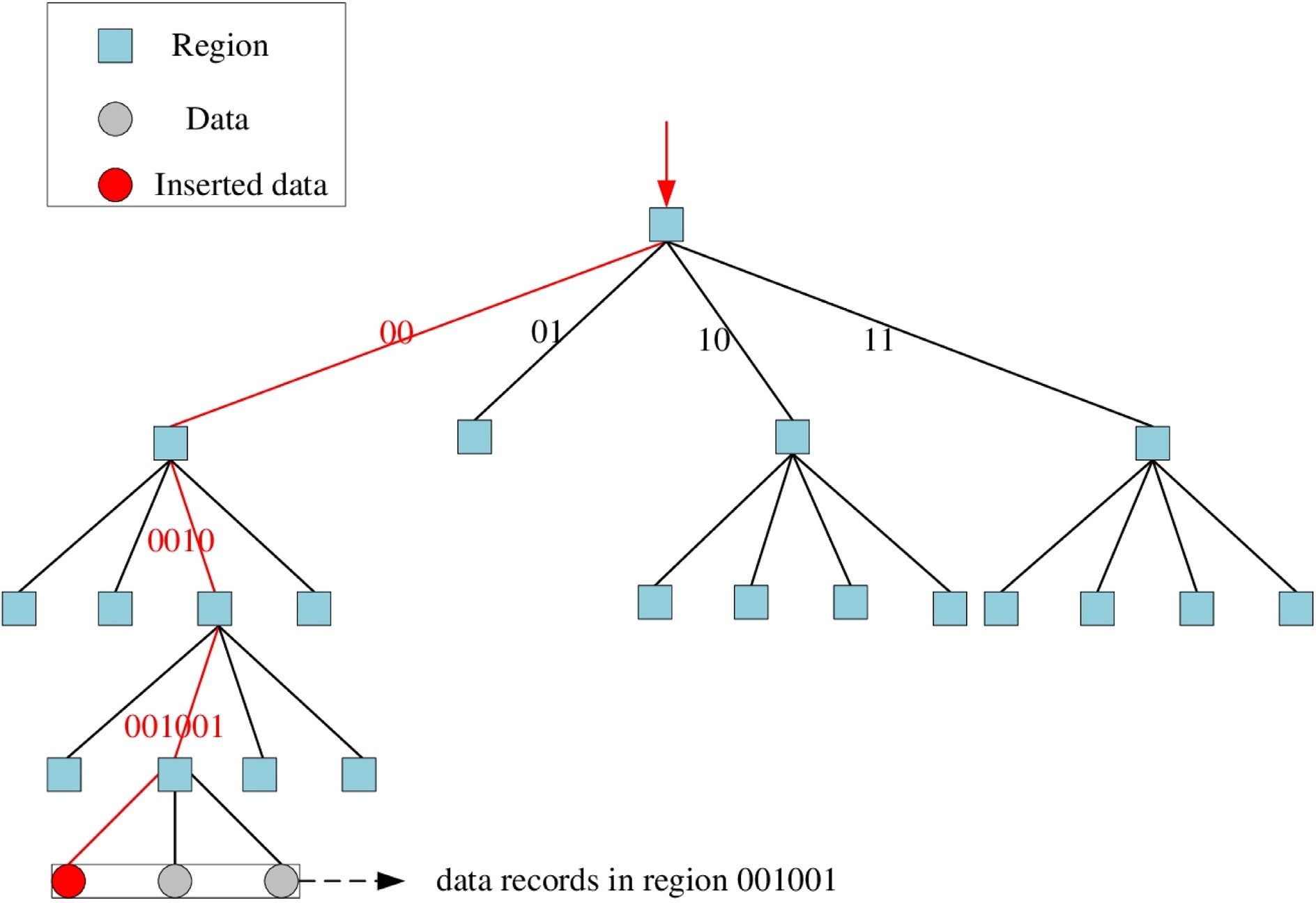
*⋃*

want to query the number of POIs closest to his/her location. In the cloud-based LBS system, in order to prevent the private query being disclosed, the user will send the query in the form of trapdoor.

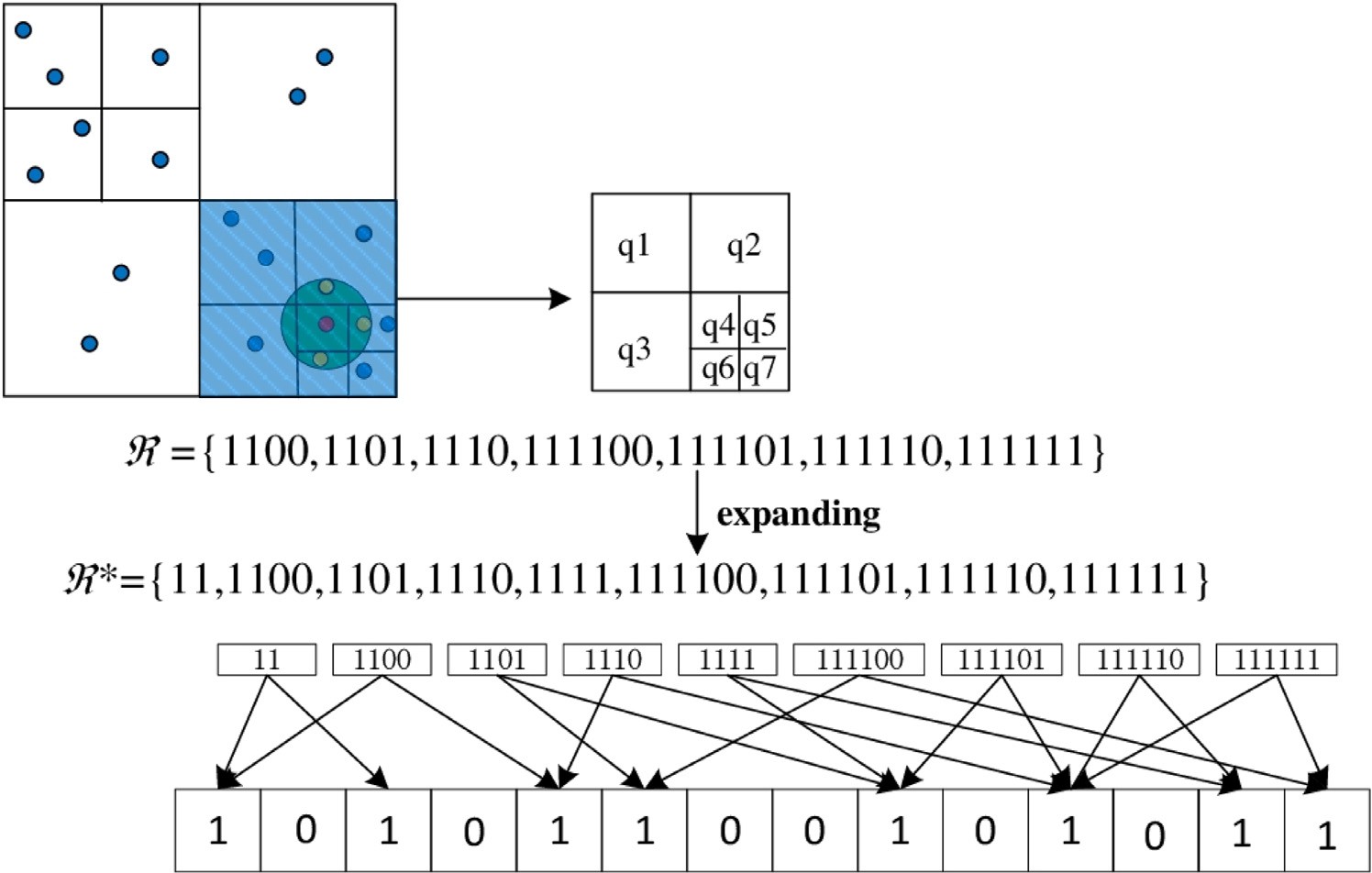
*𝑖*=1

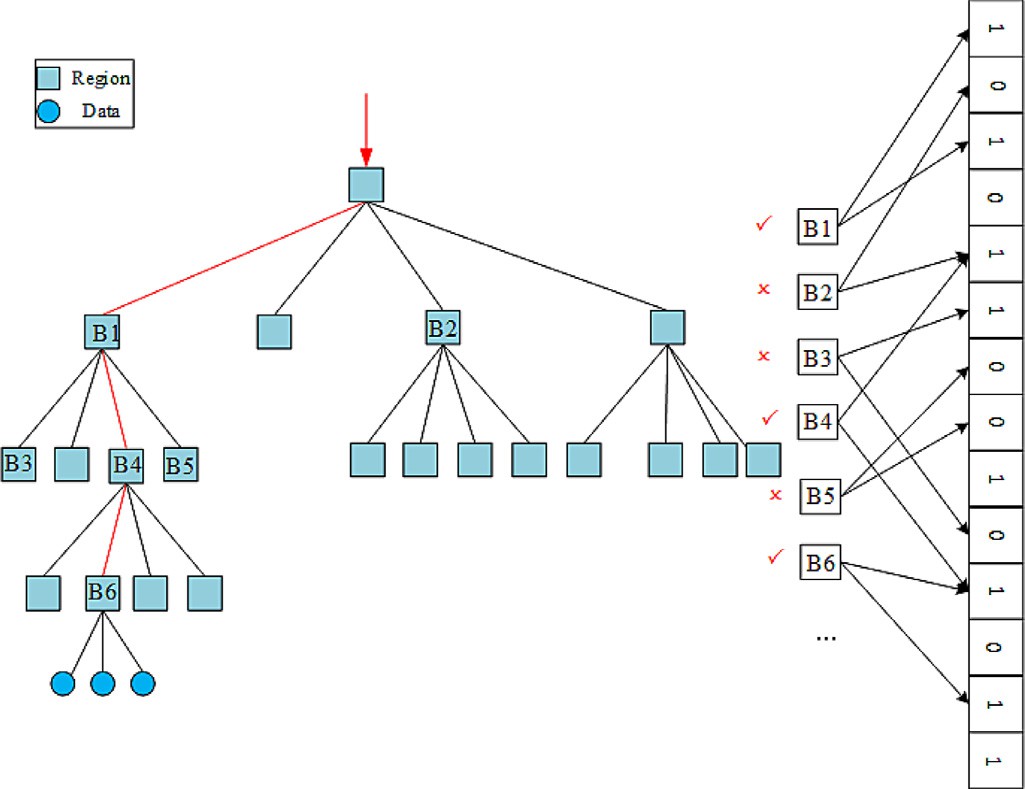
*𝑡* hash functions *𝐻𝑖* ∶ {0*,* 1} → {1*,* 2 …… *𝑚*}, *𝑖* ∈ [1*, 𝑡*] to hash each bit string sequence in ℜ∗ to get an index trapdoor ***V*** *𝐵𝐹* .

∗



**Fig. 5.** Example of data insertion.



Take an example in [Fig. 7](#_bookmark11). Assume that the current coordinate of

to the query radius *𝑅*, which is represented as the green circular region, the user is (26, 81). The user first determines the query range according

which is represented as *𝑄* = {*𝑞*1*,* …… *, 𝑞*5}. The bit string of the seven and its intersection with the subregion divided is the blue shaded part, subregions is ℜ={1100*,* 1101*,* 1110*,* 111100*,* 111101*,* 111110*,* 111111}.

The user then expands these bit strings into collections ℜ∗ =

{11*,* 1100*,* 1101*,* 1110*,* 1111*,* 111100*,* 111101*,* 111110*,* 111111}. Finally,

the user uses *𝑡* hash functions to hash all the bit strings in ℜ∗ to obtain index trapdoor ***V*** *𝐵𝐹* =10101100101011. **(2) Generation of coordinate**

# trapdoor

User randomly selects a positive integer *𝜆*, computes the first 3 di- mensions of data (*𝜆𝑥𝑢 , 𝜆𝑦𝑢 , 𝜆*). For *𝑗* ∈ [4*, 𝑙*-1], set ***q****̂* [*𝑗*]=*𝜛𝑗* (*𝜛𝑗* is a ran-

dom number). For the last dimension, set ***q***[ ] = − *∑𝑙*−1 *𝛼𝑖*−3 *𝜛𝑖* . Then, ***q*** is

**Fig. 7.** Example of range derivation.

*̂ 𝑙*

*𝑖*=4 *̂*

*𝛼𝑖*−3

split according to the bit string *𝑏* into two *𝑙* dimensional data ***q****̂𝑎* and ***q****̂𝑏* :

* if *𝑏*[*𝑖*] = 0*, 𝑖* ∈ [1*, 𝑙*], set ***q****̂𝑎* [*𝑖*] + ***q****̂𝑏* [*𝑖*] = ***q****̂* [*𝑖*];
* if *𝑏*[*𝑖*] = 1*, 𝑖* ∈ [1*, 𝑙*], set ***q****̂𝑎* [*𝑖*] = ***q****̂𝑏* [*𝑖*] = ***q****̂* [*𝑖*]. Calculate:***T****𝑎* = ***M***1 −1 ***q****̂𝑎 ,* ***T****𝑏* = ***M***2 −1 ***q****̂𝑏* . Output coordinat trapdoor: *𝑇𝐿* = {***T****𝑎,* ***T****𝑏*}.

# (3) Generation of keyword trapdoor

Since the keyword searched by the user may expose sensitive infor- mation such as interests, hobbies, behavior habits, etc., the user needs

**Fig. 8.** Example of coordinate matching.

# Coordinate matching

to encrypt the search keyword before sending the query request. User

first chooses their query keyword *𝑤𝑞* , and then randomly choose *𝑟*′ ∈ ℤ∗

After receiving the ***V***

*𝐵𝐹*

in the query trapdoor from the user, CS uti-

to encrypt

*𝑤𝑞* into *𝑇𝑤* = {*𝑇𝑤*1 *, 𝑇𝑤*2 *, 𝑇𝑤*3 }, where

*𝑞* lizes it to search the coded quadtree. In the coded quadtree, each non-

leaf node is denoted by a bit string. CS examines if the mapping posi-

*𝑇* = *𝑒*(*𝐻* (*𝑤* )*𝑆𝐾𝑢 , 𝑃 𝐾 𝑟*′ )*,* (4)

tions of *𝐵𝑖* of the node are all “1” in ***V*** *𝐵𝐹* . If it is, CS subsequently search

*𝑤*1

*𝑇𝑤*2

*𝑇*

0 *𝑞 𝐿*

= *𝑔𝑟*′ *,* (5)

= *𝑃 𝐾 𝑟*′ *.* (6)

on the search until all data in ℜ∗ is hit, as shown in [Fig. 8](#_bookmark12). CS stores down the encoding quadtree; otherwise, CS changes the branch to carry

these data in the temporary resource list (TRL) and then goes to step

# ; (2) Keyword matching

Verify whether data *𝑇𝑤*1 ⋅ *𝑒*(*𝐶𝑤*2 *, 𝑇𝑤*2 ) = *𝑒*(*𝐶𝑤*1 *, 𝑇𝑤*3 ) in the TRL is es-

*𝑤*3

*𝑢*

The user sends the query trapdoor ***TRAPDOOR*** =

tablished. If it is established, it means that the POI record meets the user’s query keyword, and go to step (3); otherwise, select the next POI

{*𝑀 𝐼 𝐷𝑢,* ***V*** *𝐵𝐹 , 𝑇𝐿, 𝑇𝑤, 𝑘*} to the CS.

* 1. *LBS Data retrieval*

user’s identity and determines whether the *𝐶𝑢* = *𝑒*(*𝑃 𝐾𝐿, 𝑀 𝐼 𝐷𝑢* ) is estab- Once the CS receives ***TRAPDOOR*** from the user, it first verifies the

lished. If not, CS rejects the query; otherwise, CS searches user query. Specific steps are as follows:

record for matching;

# top *𝑘*POI closest

Suppose {*𝐶*1*𝑎, 𝐶*1*𝑏* } and {*𝐶*2*𝑎, 𝐶*2*𝑏*} are the ciphertext data of the data

*𝑃 𝑂𝐼* 1 and *𝑃 𝑂𝐼* 2 respectively, {*𝑇𝑎, 𝑇𝑏*} is the trapdoor of the user query. Determine whether (***C***1*𝑎* − ***C***2*𝑎* )***T****𝑎* + (***C***1*𝑏* − ***C***2*𝑏* )***T****𝑏 >* 0 is true. If it is true, the *𝑃 𝑂𝐼* 1 is closer to the user than the *𝑃 𝑂𝐼* 2 .

the first *𝑘* encrypted POI to the user. Finally, CS performs distance comparison and sorting, and returns

* 1. *User decryption*

1. **Trapdoor query** U*𝑇* **:** A sends the keyword *𝑤𝑖* to C for trapdoor

query and C retrieves *𝑤 , ℎ , 𝛼 , 𝜏*

*⟨*

from the list *𝐿*

. If *𝜏* =0, it randomly

*𝑖 𝑖*

′

*𝑖 𝑖 ⟩*

*𝐻 𝑖*

User get *𝑘* specified POI ciphertext records from the CS, expressed as

{***C****𝑎𝑖 ,* ***C****𝑏𝑖* }*, 𝑖* ∈ [1*, 𝑘*]. For one of the POI ciphertext records, the user first

calculates:

***p****̂* = *𝜋* (***M****𝑇* )−1 × ***C*** *,* (7)

*𝑎 𝑑* 1 *𝑎*

outputs a bit *𝜇* ∈ {0,1} as its guess of *𝜇*. Otherwise, it randomly se- lects *𝑟𝑖* ∈ ℤ∗ and derives the trapdoor *𝑇𝑤* = {*𝑇*1 *, 𝑇*2 *, 𝑇*3 } = {*𝑒*(*𝑔𝛼𝑖 𝑎 , 𝑔𝑏𝑟𝑖* ) =

*𝑒*(*𝐻* (*𝑤𝑖* )*𝑎, 𝑔𝑏𝑟𝑖* )*, 𝑔𝑟𝑖 , 𝑔𝑎𝑟𝑖* }. C then returns *𝑇𝑤* to A.

*𝑞*

# Challenge:

After polynomial queries, A sends two keywords *𝑤*∗*, 𝑤*∗ to C, which

0 1

***p****̂* = *𝜋* (***M****𝑇* )−1

× ***C****𝑏,* (8)

have not been queried to U*𝑇* nor U*𝐶* . Then, C performs the following

operations:

2  C performs a Hash query on *𝑤*∗*, 𝑤*∗ respectively: *𝐻* (*𝑤*∗) =

*𝑏 𝑑*

0 1 0

where *𝜋𝑑* = (*𝐼𝑑 ,* 0) is a 2 × 80 matrix, *𝐼𝑑* is a 2 ×2 identity matrix.

and ***p****̂* , if *𝑏* = 1*, 𝑡* ∈ [1*,* 2], set ***p****̂* [*𝑡*] = ***p****̂* [*𝑡*]+***p****̂* [*𝑡*]; else set

0

1

1

0

1

0

0

0

0

1

1

1

1

0

1

*ℎ*∗ and *𝐻* (*𝑤*∗) = *ℎ*∗. *𝑤*∗*, 𝑤*∗ correspond to tuples *⟨𝑤*\**, ℎ*\**, 𝛼*\**, 𝜏*\**⟩* and

*𝑎 𝑏 𝑡*

For ***p****̂*

[ ] = ] [

***p****̂𝑜 𝑡*

***p****̂𝑎* [*𝑡* =***p****̂𝑏 𝑡*]. The real coordinates of POI is

*𝑜 𝑎*

(

*𝑥* = ***p****̂𝑜* [1]*, 𝑦*

*𝑏*

= [2])

***p****̂𝑜*

.

*⟨𝑤*\**, ℎ*\**, 𝛼*\**, 𝜏*\**⟩*, respectively. If *𝜏*\**, 𝜏*\* are both 1, C aborts the query and

randomly outputs *𝜇* ∈ {0,1} as its guess of *𝜇*;

′

 If at least one of *𝜏*\* and *𝜏*\* is 0, set *𝜇̂* be the bit such that

# Security analysis

0 1

\* *𝑏*

*𝜏*∗ = 0. C computes the ciphertext *𝐶* ∗ = (*𝐶* ∗*, 𝐶* ∗) = (*𝑊* ⋅ (*𝑔𝛼𝜇̂* )

⋅ *𝑔𝑘𝑖 , 𝑔𝑎𝑘𝑖* ).

*𝜇̂* 1 2

* 1. *Data confidentiality*

If *𝑊* = *𝑔*

*𝑟*+*𝑡*

, then *𝐶* \* = *𝑔𝑟*+*𝑡*

⋅ *𝑔*

*𝑏𝛼*\*

*𝜇̂ 𝑔 𝑖 𝑔*

1

⋅ *𝑘* =

(*𝑟*+*𝑘𝑖* )+(*𝑡*+*𝑏𝛼*\* ) \*

2

*𝜇̂ , 𝐶* = *𝑔*

*𝑎𝑘*

*𝑖* , where

For each LBS data, LSP outsources it to the CS in the form of ci- phertext. Specifically, for the coordinates of POI record, we use the enhanced ASPE algorithm [[12]](#_bookmark22) to encrypt it. In the known ciphertext model, if CS wants to get the real value of POI coordinates, it must re-

store two matrix ***M*** *,* ***M*** from the ciphertext and correctly guess the

*𝑟* + *𝑘𝑖* is random in A perspective. If *𝑊* is a random element of 𝔾,

so is *𝐶* \*. In addition, since *𝑘𝑖* is a random *𝐶* \* is also random in A

1 2

perspective.

# More trapdoor queries:

A sends *𝑤̃* to C for more trapdoor queries, where *𝑤̃* ≠ *𝑤*∗*, 𝑤̃* ≠ *𝑤*∗

1 2 and C respond to the query as before. 0 1

bit string *𝑏*. For CS, the equations used to determine the transformation

matrices are:***C****𝑎* = ***M****𝑇* × ***p****̂𝑎* and ***C****𝑏* = ***M****𝑇* × ***p****̂𝑏* , where ***M***1 and ***M***2 are two

1 2

*𝑙*2

# Guess:

A outputs its guess *𝜇̂*′ ∈ {0*,* 1}. If *𝜇̂*′ = *𝜇̂*, then C outputs *𝜇*′ = 0; oth-

unknown *𝑙* × *𝑙* dimensional matrixs, there are 2 unknowns in ***M***1 and ***M***2 . The vector ***p****̂𝑎 ,* ***p****̂𝑏* are split from the *𝑙*-dimensional bit string *𝑏*, which has 2*𝑙* unknowns. Since given only 2*𝑙* equations, which are less than the

amount of unknowns, the transformation matrix cannot be solved by the adversary without enough information. Also, Wong et al. [[12]](#_bookmark22) secu-

rity analysis shows that when *𝑙*=80, its security is equivalent to that of

the 1024-bit key RSA encryption algorithm. Thus, under the premise of

key security, the CS cannot restore the real location of the POI through

erwise, C outputs *𝜇*′ = 1.

Refer to [[13]](#_bookmark23), we use *𝑡𝑒𝑟* to denote two cases in which Challenger C

aborts during the game, as follows:

 When C simulates U*𝑇* and U*𝐶* , *𝜏𝑖* = 0. Since that each *𝜏𝑖* is picked

is Pr *𝑡𝑒𝑟*1 = (1 − *𝜃*)*𝑞𝑇* +*𝑞𝐶* , where *𝑞𝑇* and *𝑞𝑐* represent the adversary A randomly and independently, the probability that C aborts the game invokes at most *𝑞𝑇 , 𝑞𝐶* queries to U*𝑇* and U*𝐶* , respectively.

*| |*

 In the challenge keywords chosen by adversary A, *𝜏*∗=*𝜏*∗=1, the

ciphertext data. 0 1

2

**Lemma 1.** *Assuming that the problem of mDLIN is hard, then for any prob-*

probability that C aborts the game is Pr *𝑡𝑒𝑟*2 = 1 − (1 − *𝜃*) . Hence, the

probability of C not aborts in the game is Pr *|𝑡𝑒𝑟|* = ((1 − *𝜃*)*𝑞𝑇* +*𝑞𝐶* )(1 −

*| |*

*𝑞* +*𝑞* +2

*ability polynomial time adversary* A *, the probability advantage* Adv*𝐶* (*𝜆*) *of*

*√*  *𝑞* +*𝑞*

*| || |*

A (1 − *𝜃*)2). When *𝜃*=1-

*| |*

*distinguishing the keyword ciphertext is negligible.*

*| |*

*𝑇 𝐶*

, the probability *𝑃 𝑟 𝑡𝑒𝑟* takes the maxi-

**Proof 1.** Assuming that exists an adversary A that can correctly distin- guish the keyword ciphertext with a non-negligible probability advan-

mum value:

*𝑃 𝑟|𝑡𝑒𝑟|*=*(*  *𝑞𝑇* + *𝑞𝐶 )*

*𝑇 𝐶*

*𝑞𝑇* +*𝑞𝐶*

⋅ 2 *,* (9)

2

tage *𝜖𝐶* . We demonstrate that the challenger C can solve the mDLIN

*| | 𝑞𝑇* + *𝑞𝐶* +2 *𝑞𝑇* + *𝑞𝐶* +2

problem with a non-negligible advantage *𝜖𝐶* . Given an instance of

the mDLIN problem with parameters (*𝑔, 𝑔𝑎, 𝑔𝑏, 𝑔𝑟𝑎, 𝑔𝑡*∕*𝑏, 𝑊* ) ∈ 𝔾, where

which nearly equal to 2

*𝑞𝑇* +*𝑞𝐶* )

( *𝑒*

, and thus non-negligible. Therefore, the

*𝑎, 𝑏, 𝑟, 𝑡* ∈ ℤ∗, the goal of C is to determine *𝑊* = *𝑔𝑟*+*𝑡* or a random ele- ment of 𝔾. C set *𝜇* ∈ {0*,* 1}, if *𝑊* = *𝑔𝑟*+*𝑡*, *𝜇*=0; if *𝑊* is random, *𝜇*=1.

*𝑞*

# Initialization:

The Challenger C sets (*𝑃 𝐾 , 𝑃 𝐾* ) = (*𝑔𝑎, 𝑔𝑏*) and sents it to the adver- sary A along with the public parameter *𝑝𝑎𝑟𝑎𝑚* = (𝔾*,* 𝔾*𝑇 , 𝑒, 𝑞, 𝑔*).

*𝑢*

*𝐿*

# Query:

A can ask C for the following query:

* 1. **Hash query** U*𝐻* **:** A sends the keyword *𝑤𝑖* to C for hash query. C

will do the following:

success probability that of C guessing the bit *𝜇* (i.e., solving the mDLIN

problem) is:

Pr *𝜇* = *𝜇* = + *𝜖* ⋅ Pr *𝑡𝑒𝑟 .* (10)

′

1

*| |* 2 *𝐶 | |*

If *𝜖𝐶* is non-negligibl*|*e, s*|*o is Pr[*𝜇* = *𝜇*] − 1∕2 .

*| |*

′

* 1. *User’s query privacy*

Each LBS data will be encrypted and submitted to CS as a query trap-

 C maintains an initially empty list *𝐿*

*ℎ𝑖* to A.

*𝑤 , ℎ , 𝛼 , 𝜏*

for A’s query,

door when a user query. Specifically, for POI coordinates, as discussed

*𝐻 ⟨ 𝑖 𝑖 𝑖*

and if *𝑤𝑖* already appears in *𝐿𝐻* , C sends a response *𝐻* (*𝑤𝑖*

*𝑖⟩* ) =

in the encryption phase, the vector ***q****̂* will also be split into two random

 Otherwise, C produces a random number *𝜏𝑖* ∈ {0,1} in probability

such that *𝑃 𝑟*[*𝜏𝑖* = 0] = *𝜃*.

If *𝜏𝑖* =0, randomly chooses *𝛼𝑖* ∈ ℤ∗ and computes *ℎ𝑖* = *𝑔𝑡*∕*𝑏* ⋅ *𝑔𝛼𝑖* ; Oth- erwise, sets *ℎ𝑖* = *𝑔𝛼𝑖* .

*𝑞*

vector ***q****̂𝑎* and ***q****̂𝑏* according to the bit string *𝑏*. Since the CS cannot obtain

the invertible matrix ***M***1, ***M***2 and the bit string *𝑏*, after a series of op-

erations such as splitting and matrix multiplication, CS can not get the

user’s actual coordinates and query radius with the user’s query trap-

 C stores the tuple *𝑤 , ℎ , 𝛼 , 𝜏*

*⟨*

in *𝐿*

and returns *𝐻* (*𝑤* ) = *ℎ* to

door.

*𝑖*

A.

*𝑖 𝑖*

*𝑖 ⟩ 𝐻*

*𝑖 𝑖*

**Lemma 2.** *Allowing the problem of DBDH is hard, then for any PPT* A *, the*

* 1. **Ciphertext query** U*𝐶* **:** A sends the keyword *𝑤𝑖* to C for ci-

*probability advantage* Adv*𝑇* (*𝜆*) *who breaches the keywords trapdoor privacy*

phertext query and C retrieves

*𝑤 , ℎ , 𝛼 , 𝜏*

from the list *𝐿*

. If *𝜏* =0, A

*𝑖 𝑖*

*⟨*

′

*𝑖 𝑖 ⟩*

*𝐻 𝑖*

*of our scheme is negligible.*

it randomly outputs a bit *𝜇* ∈ {0,1} as its guess of *𝜇*. Otherwise, it randomly selects *𝑘𝑖* ∈ ℤ∗ and computes the ciphertext *𝐶𝑤* = (*𝐶*1 *, 𝐶*2 ) =

*𝑞*

((*𝑔𝛼𝑖* )*𝑏* ⋅ *𝑔𝑘𝑖* =*𝑔𝛼𝑖 𝑏*+*𝑘𝑖 ,* (*𝑔𝑎*)*𝑘𝑖* ). C then returns *𝐶𝑤* to A.

**Proof 2.** Assuming that there is a probabilistic polynomial time adver- sary A can breach the trapdoor privacy of our proposed scheme with a

non-negligible advantage *𝜖𝑇* , there is a challenger C whose probability

of the DBDH problem with parameters (*𝑔, 𝑔𝑎, 𝑔𝑏, 𝑔𝑐*) ∈ 𝔾*, 𝑊* ∈ 𝔾*𝑇* , where of solving the DBDH problem cannot be negligibled. Given an instance

*𝑎, 𝑏, 𝑐* ∈ ℤ∗, the goal of C is to determine *𝑊* = *𝑒*(*𝑔, 𝑔*)*𝑎𝑏𝑐* or a random el-

*𝑞*

ement in 𝔾*𝑇* . C sets *𝜇* ∈ {0*,* 1}, if *𝑊* = *𝑒*(*𝑔, 𝑔*)*𝑎𝑏𝑐*, *𝜇*=0; if *𝑊* is a random element from 𝔾*𝑇* , *𝜇*=1.

# Initialization:

The C sets (*𝑃 𝐾𝑢, 𝑃 𝐾𝐿*) = (*𝑔𝑎, 𝑔𝑏*) and sents it to the adversary along with the public parameter *𝑝𝑎𝑟𝑎𝑚* = (𝔾*,* 𝔾*𝑇 , 𝑒, 𝑞, 𝑔*).

# Query:

A can ask C for the following query:

**Table 2**

Symbol definition.

|  |  |
| --- | --- |
| Symbol | Definition |
| *𝑁* | The amount of LBS data |
| *𝑛* | The data dimension, this paper assumes that *𝑛*=2 |
| *𝑇𝑒* | The running time of one exponentiation operation over the group |
| *𝑇𝑚* | The running time of one multiplication operation over the group |
| *𝑇𝑝* | The running time of a bilinear pairing operation |
| *𝑇𝑜𝑚* | The running time of a normal multiplication operation |

mum value:

* 1. **Hash query** U*𝐻*

**:** A sends the keyword *𝑤𝑖* to C for hash query. C

*| | (*  *𝑞𝑇* + *𝑞𝐶 )*

*𝑞𝑇* +*𝑞𝐶*

2 2

will do the following:

*𝑃 𝑟 𝑡𝑒𝑟* =

*𝑞* + *𝑞* +2

⋅ *𝑞*

+ *𝑞* +2 *,* (11)

 C maintains an initially empty list *𝐿 𝑤 , ℎ , 𝛼 , 𝜏* for A’s query,

*𝐻 ⟨ 𝑖 𝑖 𝑖 𝑖⟩*

*| | 𝑇 𝐶*

*𝑇 𝐶*

*| |*

2

 Otherwise, C randomly chooses *𝜏𝑖* ∈ {0,1} in probability such that

and if *𝑤𝑖* appears in *𝐿𝐻* , C sends a response *𝐻* (*𝑤𝑖*) = *ℎ𝑖* to A.

*𝑃 𝑟*[*𝜏𝑖* = 0] = *𝜃*.

If *𝜏𝑖*=0, randomly chooses *𝛼𝑖* ∈ ℤ∗ and sets *ℎ𝑖* = *𝑔𝑧*+*𝛼𝑖* ; Otherwise, set

*ℎ𝑖* = *𝑔𝛼𝑖* .

 C stores the tuple *𝐿 𝑤 , ℎ , 𝛼 , 𝜏* to *𝐿* and returns *𝐻* (*𝑤* ) = *ℎ*

*𝑞*

*𝑞𝑇* +*𝑞𝐶* )

success probability that of C guessing the bit *𝜇* (i.e., solving the DBDH

which nearly equal to (

*𝑒* , and thus non-negligible. Therefore, the

problem) is:

Pr *|𝜇* = *𝜇|* = + *𝜖𝑇* ⋅ Pr *|𝑡𝑒𝑟|.* (12)

′

*| |*

*|*

*|*

2

*|*

*|*

′

1

to A.

*𝐻 ⟨ 𝑖 𝑖 𝑖 𝑖⟩ 𝐻 𝑖 𝑖*

* 1. **Ciphertext query** U*𝐶* **:** A sends the keyword *𝑤𝑖* to C for ciphertext

If *𝜖𝑇* is non-negligible, so is Pr[*𝜇* = *𝜇*] − 1∕2 .

query and C retrieves *𝑤 , ℎ , 𝛼 , 𝜏*

from the list *𝐿*

. If *𝜏* =0, it aborts the

* 1. *Index security*

*⟨ 𝑖*

*𝑖 𝑖*

*𝑖⟩* ′

*𝐻 𝑖*

query and randomly outputs a bit *𝜇* ∈ {0,1} as its guess of *𝜇*. Otherwise,

it randomly picks *𝑘𝑖* ∈ ℤ∗ and computes the ciphertext *𝐶𝑤* = (*𝐶*1*, 𝐶*2) = ((*𝑔𝛼𝑖* )*𝑏* ⋅ *𝑔𝑘𝑖* =*𝑔𝛼𝑖 𝑏*+*𝑘𝑖 ,* (*𝑔𝑎*)*𝑘𝑖* ). C then returns *𝐶𝑤* to A.

*𝑞*

* 1. **Trapdoor query** U*𝑇* **:** A sends the keyword *𝑤𝑖* to C for trapdoor

**Lemma 3.** *Even if an adversary* A *obtains the index, the probability of the*

A *successfully deriving the keyword or coordinates is negligible.*

query and C retrieves

*⟨ 𝑖 𝑖 𝑖* ′ *𝑖⟩*

from *𝐿𝐻* . If *𝜏𝑖*=0, it aborts the query

}

as its guess of *𝜇*. Otherwise,

**Proof 3.** In our scheme, the bit string of the subregion is stored in each non-leaf node of the coded quadtree, and the LBS data ciphertext

it randomly picks *𝑟𝑖* ∈ ℤ∗ and derives the trapdoor *𝑇𝑤* = {*𝑇*1*, 𝑇*2*, 𝑇*3} =

*𝑞*

and randomly outputs a bit *𝜇* ∈ {0,1

*𝑤 , ℎ , 𝛼 , 𝜏*

{*𝑒*(*𝑔𝛼𝑖 𝑎, 𝑔𝑏𝑟𝑖* ) = *𝑒*(*𝐻* (*𝑤𝑖*)*𝑎, 𝑔𝑏𝑟𝑖* )*, 𝑔𝑟𝑖 , 𝑔𝑎𝑟𝑖* }. C then returns *𝑇𝑤* to A.

# Challenge:

After polynomial queries, A sends two keywords *𝑤*∗*, 𝑤*∗ to C, which

is stored in the leaf node. Even if the A obtains the coded quadtree,

query, it is assumed that the query range is {(*𝑥𝑙 , 𝑦𝑙* )*,* (*𝑥𝑟, 𝑦𝑟*)*, 𝑛*}, where the data ciphertext cannot be decrypted without the key. When a user’s

(*𝑥𝑙 , 𝑦𝑙* ) and (*𝑥𝑟, 𝑦𝑟*) represent the top left and bottom right vertices of

have not been queried to U

0 1

nor U . Then, C

query region *𝑄* respectively, and *𝑛* denotes the amount of subregions in

operations:

*𝑇 𝐶*

performs the following

query region *𝑄*. Since the user’s position coordinates can be in any of the

 C performs a Hash query on *𝑤*∗*, 𝑤*∗ respectively: *𝐻* (*𝑤*∗) =

divided subregions, the probability of correctly guessing the user’s loca-

0 1 0

*⟨ ⟩*

*ℎ*∗ and *𝐻* (*𝑤*∗) = *ℎ*∗. *𝑤*∗*, 𝑤*∗ correspond to tuples *𝑤*\**, ℎ*\**, 𝛼*\**, 𝜏*\* and

0

1

1

0

1

0

0

0

0

*⟨𝑤*\**, ℎ*\**, 𝛼*\**, 𝜏*\**⟩*, respectively. If *𝜏*\**, 𝜏*\* are both 1, C aborts the query and

1

1

1

1

′

0

1

not obtain any valuable information.

tion in a certain subregion is 1/n. In addition, in view of the one-wayness of the hash function, even if the index ***V****𝐵𝐹* is leaked, the adversary can

randomly outputs a bit *𝜇* ∈ {0,1}as its guess of *𝜇*;

 If at least one of *𝜏*\*

and *𝜏*\*

is 0, let

*𝜇̂*

be the bit sat-

isfying *𝜏*∗ = 0. C

0 1 *𝑇* \* = {*𝑇* ∗*, 𝑇* ∗*, 𝑇* ∗}, where

# Performance evaluation

*𝜇̂*

calculates the trapdoor

1 2 3

∗ *𝑎 𝑟 𝑏 𝛼*∗ ∗ *𝑟* ∗ *𝑎𝑟*

*𝑎𝑏𝑐* \*

*𝑇*1 = *𝑊* ⋅ *𝑒*(*𝑔 , 𝑔 𝑖* ) *𝜇̂ , 𝑇*2 = *𝑔 𝑖 , 𝑇*2 = *𝑔 𝑖* . If *𝑊* = *𝑒*(*𝑔, 𝑔*) , then *𝑇*1 =

*𝑒*(*𝑔, 𝑔*)*𝑎𝑏*(*𝑐*+*𝑟𝑖 𝛼*∗ ). Since *𝑟 𝛼*∗ is a random value, so *𝑇* \* = *𝑒*(*𝑔, 𝑔*)*𝑎𝑏*(*𝑐*+*𝑟𝑖 𝛼*∗ ) =

In this section, we analyze the performance of PPT-LBS from the perspectives of data encryption, trapdoor generation and data retrieval,

*𝜇̂*

*𝑖 𝜇̂*

*𝜇̂*

1

*𝑒*(*ℎ𝜇̂*′ *, 𝑔𝑎𝑏*), C outputs 0; if *𝑊* is a random member of 𝔾*𝑇* , then *𝑇* \* is also

respectively. The experiments are run in Windows 10 operating system

a random element in 𝔾*𝑇* , C outputs 1. In addition, since *𝑟*

value, *𝑇* \**, 𝑇* \* are also random for A.

1

*𝑖* is a random

with Intel (R) Core (TM) i5-10200H CPU @2.40GHz and 16GB mem- ory. The cryptographic operations have been realized by using the Java

2 3 Pairing-Based Cryptography (JPBC) library.

# More trapdoor queries:

A sends *𝑤̃* to C for more trapdoor queries, where *𝑤̃* ≠ *𝑤*∗*, 𝑤̃* ≠ *𝑤*∗

# Comparison of computational cost

and C respond to the query as before.

# Guess:

0 1 For convenience, [Table 2](#_bookmark14) lists the symbol description definitions used

in the comparison. Since the time cost of matrix and vector multiplica-

A outputs its guess *𝜇̂*′ ∈ {0*,* 1}, if *𝜇̂*′ = *𝜇̂*, then C outputs *𝜇*′ = 0, oth-

erwise C outputs *𝜇*′ = 1.

Refer to [[13]](#_bookmark23), we use *𝑡𝑒𝑟* to denote two cases in which Challenger C

aborts during the game, as follows:

 When C simulates U*𝑇* and U*𝐶* , *𝜏𝑖* = 0. Since that each *𝜏𝑖*

the game is Pr *𝑡𝑒𝑟*1 = (1 − *𝜃*)*𝑞𝑇* +*𝑞𝐶* , where *𝑞𝑇* and *𝑞𝑐* represent is picked randomly and separately, the probability that C aborts the adversary A invokes at most *𝑞𝑇 , 𝑞𝐶* queries to U*𝑇* and U*𝐶* ,

*| |*

respectively.

 In the challenge keywords chosen by adversary A, *𝜏*∗=*𝜏*∗=1, the

tion operation, hashing operation, symmetric encryption and decryption operation is relatively low, it is ignored in the comparison.

[Table 3](#_bookmark16) shows the comparison results of PPT-LBS and the existing schemes [[9,14,15]](#_bookmark34) with regard to computational cost.

# Comparison of characteristics

In [Table 4](#_bookmark17), characteristics of our scheme and the existing schemes [[9,14,15]](#_bookmark34) are compared.

# Comparison of experiments

[Figure 9](#_bookmark18) indicates the comparison results of our scheme and the ex- isting similar scheme [[9,14,15]](#_bookmark34) in the data encryption phase. Among

0 1 them, our scheme and Ou et al.’s [[14]](#_bookmark24) scheme use matrix operation and

2

probability that C aborts the game is Pr *𝑡𝑒𝑟*2 = 1 − (1 − *𝜃*) . Hence, the

*| |*

probability of C not aborts in the game is Pr *|𝑡𝑒𝑟|* = ((1 − *𝜃*)*𝑞𝑇* +*𝑞𝐶* )(1 −

public key searchable encryption to process LBS coordinates and key-

(1 −

*𝜃*

. When *𝜃*= -

*𝑇 𝐶* , the probability *𝑃 𝑟 𝑡𝑒𝑟*

takes the maxi-

and Zhu et al.’s [[9]](#_bookmark34) scheme, which uses a homomorphic encryption

)2)

1 *√ 𝑞* +*𝑞*

*| || |*

word. The computational cost is lower than Lin et al.’s [[15]](#_bookmark25) scheme

*𝑞𝑇* +*𝑞𝐶* +2

*| |*

**Table 3**

Comparison of computational cost in each phase.

**Table 4**

Data encryption Trapdoor generation Data retrieval Ou et al. [[14]](#_bookmark24) *𝑁 𝑇* + *𝑇* 2*𝑇 𝑁* 2*𝑇* + *𝑇*

*( 𝑝 𝑚 ) 𝑒 ( 𝑝 𝑚 )*

*( 𝑒 𝑚 ) 𝑒* 2 *( 𝑚 𝑝 )*

Zhu et al. [[9]](#_bookmark34) *𝑁* 6*𝑇* + 2*𝑇* 4*𝑁𝑇 𝑙𝑜𝑔 𝑁* ⋅ 5*𝑇* + 2*𝑇*

Lin et al. [[15]](#_bookmark25) *𝑁𝑛* 3*𝑇* + *𝑇* +*𝑇* 2*𝑛*(*𝑇* + *𝑇* ) *𝑙𝑜𝑔 𝑁* ⋅ (*𝑛𝑇* + (2*𝑛* + 1)*𝑇* )

*( 𝑒 𝑜𝑚 𝑚 ) 𝑒 𝑜𝑚* 2 *𝑝 𝑚*

Ours 3*𝑁𝑇𝑒* 2*𝑇𝑒* + *𝑇𝑝 𝑙𝑜𝑔*4 *𝑁* ⋅ *(*2*𝑇𝑝* + *𝑇𝑚 )*

Comparison of characteristics.

Query privacy Location privacy Geographic range query Keyword query Top-k query

Ou et al. [[14]](#_bookmark24) Zhu et al. [[9]](#_bookmark34) Lin et al. [[15]](#_bookmark25) Ours

*√ √ √*

*√ √ √*

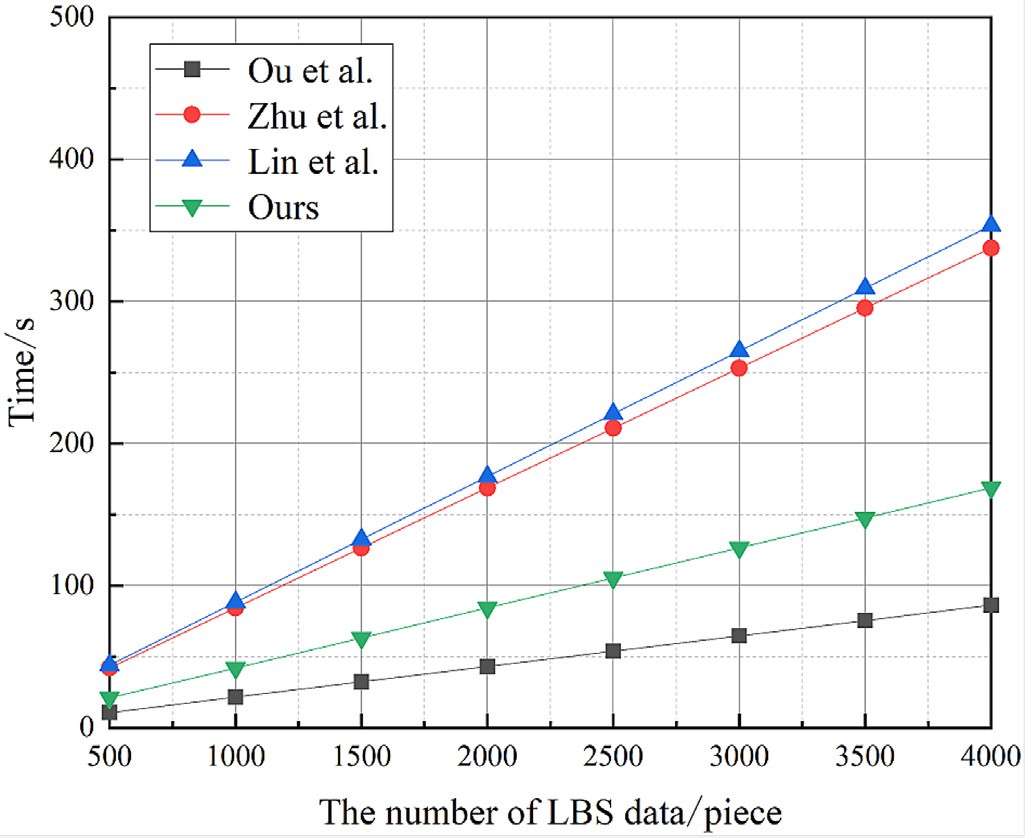
×*√* ×

× ×

×

*√ √*



**Fig. 9.** The experiment comparison of LSP data encryption.



data to *𝑛* dimension on the basis of Zhu et al.’s scheme, so it has the mechanism over the composite order group. Lin et al.’s scheme extends

than ours, the key matrix of Ou et al.’s scheme is only 3 ×3 dimen- highest cost. Although Ou et al.’s scheme’s calculation cost is lower

sions, which is easy to be cracked by force and has certain security risks.

[Figure 10](#_bookmark18) shows the comparison results of ours scheme and the ex- isting similar scheme [[9,14,15]](#_bookmark34) in the trapdoor generation phase. Sim- ilar to the encryption process, our scheme has a lower cost than the Zhu et al.’s [[9]](#_bookmark34) scheme and Lin et al.’s [[15]](#_bookmark25) scheme using a homo- morphic encryption mechanism. Besides, in Zhu et al.’s scheme and Lin et al.’s scheme, users do not support constructing a query trap- door based on a keyword but can only return all the data in the query area through the CS to decrypt and filter the required data. In the Ou et al.’s [[14]](#_bookmark24) scheme, when a user query the same keyword, the same parameters will be generated in the trapdoor, which is not random and easy to disclose the user query mode, resulting in the disclosure of user privacy.

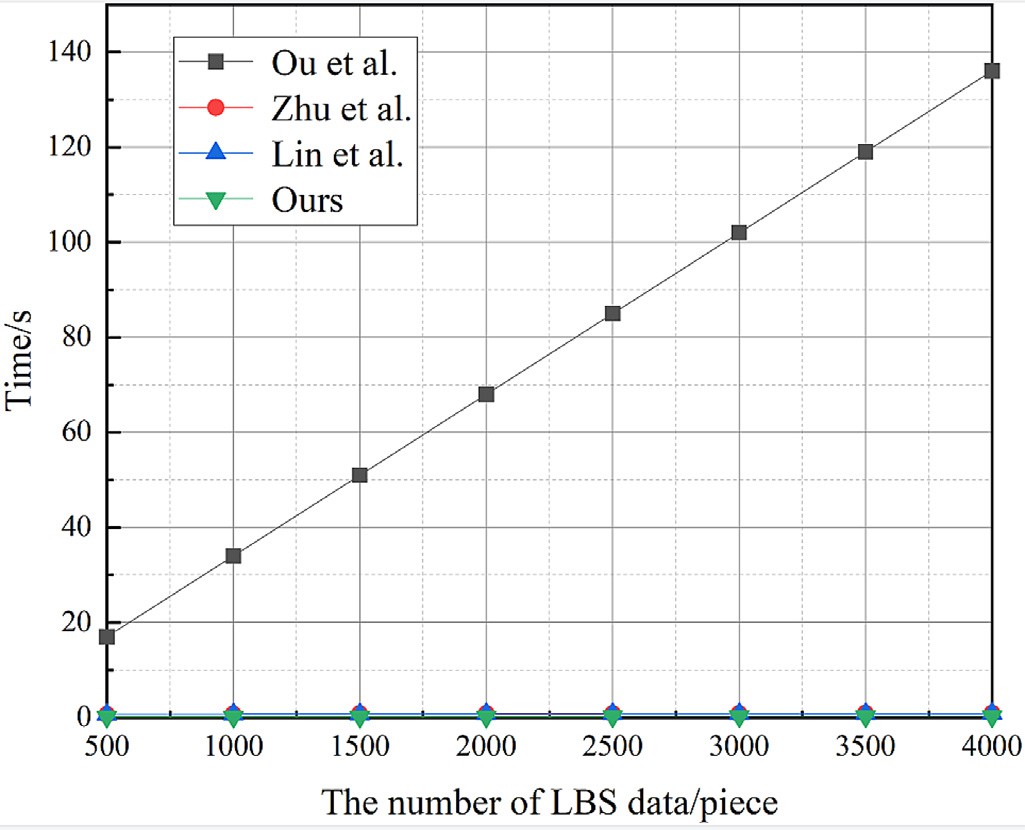
[Figure 11](#_bookmark19) shows the comparison results of ours scheme and the ex- isting similar scheme [[9,14,15]](#_bookmark34) in the data retrieval phase. Zhu et al.’s

[[9]](#_bookmark34) scheme and Lin et al.’s [[15]](#_bookmark25) scheme use dichotomy to retrieve data, our scheme uses coded quadtree to retrieve data, the time complexity are

both *𝑂*(*𝑙𝑜𝑔𝑁* ). Still, neither Zhu et al.’s scheme nor Lin et al.’s scheme

supports distance top-k query, while Ou et al.’s [[14]](#_bookmark24) scheme need to

**Fig. 10.** The experiment comparison of user’s trapdoor generation.



**Fig. 11.** The experiment comparison of the data retrieval.

traverse all of the data and time complexity is *𝑂*(*𝑁* ), which cannot be applied to massive data.

Therefore, comprehensively, our scheme is suitable for LBS query in outsourcing environment.

# Conclusion

A privacy-preserving top-k query scheme for outsourcing situation is constructed based on enhanced asymmetric scalar-product preserv- ing encryption and public key searchable encryption in this paper. In addition, we use the coded quadtree and bloom filter to construct an index structure, which enables the CS fast locate the user’s query area in the massive encrypted data. Security analysis demonstrates that our scheme not only ensures the data security of LSP but also protects the privacy of users’ query, and has preferable to similar scheme in performance.

# Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

# Acknowledgments

[Our work was jointly supported by the National Natural Science Foundation of China (No.](https://doi.org/10.13039/501100001809) [61872051](#_bookmark26)[, No.](https://doi.org/10.13039/501100001809) [61702067](#_bookmark26)[), the Chongqing](https://doi.org/10.13039/501100001809) [Natural Science Foundation of China](https://doi.org/10.13039/501100001809) (No. [cstc2020jcyj-msxmX0343](#_bookmark26)), and the [Venture & Innovation Support Program for Chongqing Overseas](https://doi.org/10.13039/501100013160) Returnees (No. [CX2018122](#_bookmark27)).

# Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:[10.1016/j.ejps.2020.105216](https://doi.org/10.1016/j.ejps.2020.105216).

# References

1. [X. Zhang, X. Gui, Z. Wu, Privacy preservation for location-based services: a survey,](http://refhub.elsevier.com/S2772-9184(22)00007-8/sbref0001)

[J. Softw. 26 (9) (2015) 2373–2395.](http://refhub.elsevier.com/S2772-9184(22)00007-8/sbref0001)

1. [X. Zhu, E. Ayday, R. Vitenberg, A privacy-preserving framework for outsourcing location-based services to the cloud, IEEE Trans. Dependable Secure Comput. (2019).](http://refhub.elsevier.com/S2772-9184(22)00007-8/sbref0002)
2. [Z. Liu, L. Wu, J. Ke, W. Qu, W. Wang, H. Wang, Accountable outsourc- ing location-based services with privacy preservation, IEEE Access 7 (2019) 117258–117273.](http://refhub.elsevier.com/S2772-9184(22)00007-8/sbref0003)
3. [R. Li, A.X. Liu, A.L. Wang, B. Bruhadeshwar, Fast and scalable range query processing with strong privacy protection for cloud computing, IEEE/ACM Trans. Netw. 24 (4) (2015) 2305–2318.](http://refhub.elsevier.com/S2772-9184(22)00007-8/sbref0004)
4. [R. Li, A.X. Liu, Adaptively secure conjunctive query processing over encrypted data for cloud computing, in: 2017 IEEE 33rd International Conference on Data Engineer- ing (ICDE), IEEE, 2017, pp. 697–708.](http://refhub.elsevier.com/S2772-9184(22)00007-8/sbref0005)
5. [M. Zeng, K. Zhang, J. Chen, H. Qian, P3GQ: a practical privacy-preserving generic location-based services query scheme, Pervasive Mob. Comput. 51 (2018) 56–72.](http://refhub.elsevier.com/S2772-9184(22)00007-8/sbref0006)
6. [S. Yang, S. Tang, X. Zhang, Privacy-preserving k nearest neighbor query with au- thentication on road networks, J. Parallel Distrib. Comput. 134 (2019) 25–36.](http://refhub.elsevier.com/S2772-9184(22)00007-8/sbref0007)
7. [Q. Xie, L. Wang, Privacy-preserving location-based service scheme for mobile sens- ing data, Sensors 16 (12) (2016) 1993.](http://refhub.elsevier.com/S2772-9184(22)00007-8/sbref0008)
8. [H. Zhu, R. Lu, C. Huang, L. Chen, H. Li, An eﬃcient privacy-preserving loca- tion-based services query scheme in outsourced cloud, IEEE Trans. Veh. Technol. 65 (9) (2015) 7729–7739.](http://refhub.elsevier.com/S2772-9184(22)00007-8/sbref0009)
9. [H. Zhu, F. Liu, H. Li, Eﬃcient and privacy-preserving polygons spatial query frame- work for location-based services, IEEE Internet Things J. 4 (2) (2016) 536–545.](http://refhub.elsevier.com/S2772-9184(22)00007-8/sbref0010)
10. [L. Zhao, Q. Liu, H. Huang, X. Jia, Eﬃcient privacy-preserving query processing on outsourced geographic databases, in: 2018 IEEE Global Communications Conference (GLOBECOM), IEEE, 2018, pp. 1–6.](http://refhub.elsevier.com/S2772-9184(22)00007-8/sbref0011)
11. [W.K. Wong, D.W.-l. Cheung, B. Kao, N. Mamoulis, Secure kNN computation on en- crypted databases, in: Proceedings of the 2009 ACM SIGMOD International Confer- ence on Management of data, 2009, pp. 139–152.](http://refhub.elsevier.com/S2772-9184(22)00007-8/sbref0012)
12. [Q. Huang, H. Li, An eﬃcient public-key searchable encryption scheme secure against inside keyword guessing attacks, Inf. Sci. 403 (2017) 1–14.](http://refhub.elsevier.com/S2772-9184(22)00007-8/sbref0013)
13. [L. Ou, H. Yin, Z. Qin, S. Xiao, G. Yang, Y. Hu, An eﬃcient and privacy-preserving multiuser cloud-based LBS query scheme, Secur. Commun. Netw. 2018 (2018).](http://refhub.elsevier.com/S2772-9184(22)00007-8/sbref0014)
14. [J. Lin, J. Niu, H. Li, M. Atiquzzaman, A secure and eﬃcient location-based service scheme for smart transportation, Future Gener. Comput. Syst. 92 (2019) 694–704.](http://refhub.elsevier.com/S2772-9184(22)00007-8/sbref0015)

**Zhou Yousheng**, received the PhD degree from the Beijing University of Posts and Telecommunications, in 2011. He is currently an associate professor with the Chongqing Univer- sity of Posts and Telecommunications. He has published more than 20 academic papers in peer-reviewed international jour- nals. His research interests include mobile security and cloud security. Email: [zhouys@cqupt.edu.cn](mailto:zhouys@cqupt.edu.cn)

**Li Xia**, is currently a master student of School of Cyber Secu- rity and Information Law, Chongqing University of Posts and Telecommunications. Her research interest include blockchain security and the IoT security. Email: [lixiacr7@163.com](mailto:lixiacr7@163.com)

**Wang Ming**, received the master’ s degree from the Chongqing University of Posts and Telecommunications. His research in- terests include mobile security and the IoT security. Email: [s180231886@stu.cqupt.edu.cn](mailto:s180231886@stu.cqupt.edu.cn)

**Liu Yuanni**, is an associate professor at the Institute of Fu- ture Network Technologies, ChongQing University of Posts and Telecommunications, China. She received her PhD from the Department of Network Technology Institute, Beijing Uni- versity of Posts and Telecommunications, China, in 2011. Her research interests include mobile crowd sensing, IoT security, and data virtualization. Email: [liuyn@cqupt.edu.cn](mailto:liuyn@cqupt.edu.cn)