J Inf Process Syst, Vol.16, No.1, pp.96~112, February 2020
https://doi.org/10.3745/JIPS.04.0163

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Privacy Protection Model for Location-Based Services

Lihao Ni[*, **], Yanshen Liu[**], and Yi Liu[**]

Abstract
Solving the disclosure problem of sensitive information with the k-nearest neighbor query, location dummy technique, or interfering data in location-based services (LBSs) is a new research topic. Although they reduced security threats, previous studies will be ineffective in the case of sparse users or K-successive privacy, and additional calculations will deteriorate the performance of LBS application systems. Therefore, a model is proposed herein, which is based on geohash-encoding technology instead of latitude and longitude, memcached server cluster, encryption and decryption, and authentication. Simulation results based on PHP and MySQL show that the model offers approximately 10× speedup over the conventional approach. Two problems are solved using the model: sensitive information in LBS application is not disclosed, and the relationship between an individual and a track is not leaked.

Keywords
Geohash-Encoding, Location-Based Services, Memcached Server Cluster, Point of Interest, Privacy Protection Model

1. Introduction

Recently, mobile phone service providers have been offering location-based services (LBSs) that track the exact geolocations of users through location positioning devices, e.g., the Global Positioning System (GPS). A point of interest (POI) is a focused geographic entity, such as a landmark, school, historical building, or business. POIs are the bases for most data used for supporting LBSs. LBS- and POI-based applications are developed rapidly, parallel with the upgrade of wireless communication technologies and the popularity of smart phones [1-4].

LBSs are often used to search POIs. The exact latitude and longitude information and query keywords of users should first be sent to an LBS server. In this process, the user's private information may be disclosed because a premeditated adversary or malicious LBS provider can easily intercept and wiretap important or private information on the Internet. Subsequently, those who access this information can easily speculate users' privacy, such as their lifestyle, exchequer, physical health, education, marital status, disease, and religion. In fact, privacy invasion has occurred frequently worldwide recently, including the interception or wiretap of users' geolocation information and query keywords in LBS applications [5,6].

Manuscript received June 8, 2018; first revision September 28, 2018; second revision February 8, 2019; third revision April 16, 2019; accepted April 22, 2019.
Corresponding Author: Lihao Ni (157372883@qq.com)
* National Engineering Research Center for E-Learning (NERCEL), Central China Normal University, Wuhan, China (157372883@qq.com)
** Hubei Research Center for Educational Informationization, Central China Normal University, Wuhan, China ({yanshenliu, liuyi136}@126.com)

LBS and privacy principles are always in contradiction. Academic researchers have been investigating the balance between providing quality LBSs and user location privacy protection, which enables users' location and query keywords information to be controlled when being exposed at different sites when being exposed at different sites of networks when acquiring LBS data.

2. Related Studies

Recently, privacy protection in LBSs has garnered the attention of many scholars. The current studies focus primarily on the following aspects.

Fuzzy location [7], a widespread location protection method, was adopted to deliberately reduce location precision such that attackers could only retrieve coarse-grained location information. However, by adopting this method, the attack model must consider the following: as the attackers know the map well, they may eliminate the unreachable zones in the fuzzy region and use graph matching to increase the accuracy of the known location. Different types of protection methods exist based on the privacy target, and most of them use the k-anonymous pattern to protect users' identity [8]. Generally, these methods involve obtaining a set containing k users who are indistinguishable from each other. Therefore, the attackers cannot identify an individual user from the user set. These methods generally achieve anonymity based on TTP components.

In [9], the authors classified the existing methods into TTP-dependent and non-TTP-dependent methods. However, they did not consider different attack models. The classification methods proposed by Ghinita et al. [10] posed the same problem as well. Toch et al. [11] and He and Chan [12] considered different attack models of location privacy.

The current P2P-based model exhibits two shortcomings: first, the collaborative users selected are uniformly distributed around the targeted user; second, the anonymous region may not be sufficiently large to satisfy the minimum requirements of each moving user. Consequently, the probability of the attackers obtaining the targeted user's real location is high [13]. Niu et al. [13] proposed a deviation-based attack and alleviated it using the R-cloak method. Chow et al. [14] offered the sharing of peer information and historical location information to reduce communication overhead; however, while eliminating the bottleneck of the system, such a structure increased the overhead at the mobile terminal, and it was difficult to identify the integrity of the surrounding users: if a vicious collaborative anonymous user exists, then the privacy of other users may be compromised. Hence, Chen et al. [15] suggested a game theory-based location privacy protection method via user collaboration, computed the anchor points, and solved the dishonest collaboration through a secure sum computation method. Utilizing digital signature techniques, Yang and Yuan [16] adopted a receiver's private key to encrypt the location information to verify honest users and avoid location privacy being compromised. Yang et al. [17] introduced the direct Internet reciprocity mechanism to LBS privacy protection and directly benefitted from providing services to other nodes via a node in the network. Such a mechanism is dependent on the auction model proposed by Levin et al. [18] in 2008, rendering them the sole group to introduce an incentive mechanism in LBS privacy protection. Based on fuzzy logic of soft computing, Li et al. [18] proposed a credit incentive-based mechanism, in which a user may obtain and accumulate credit, and the user is allowed to help others only when his/her credit reaches a threshold.

3. Problem Setting

3.1 Motivations

Herein, a new location privacy protection model for LBSs is presented. In the scheme, the location information is expressed by geohash geographical information encoding [19] rather than the latitude and longitude of users' positions, returning k-nearest POIs from the LBS per query, and attaining k-anonymity. Furthermore, a lightweight cache module is added to the scheme, which implements the backward function on the cache of data storage and access, resulting in a more efficient model-based LBS system. Furthermore, all required parameters and data between the server and client are encrypted, which can enhance the safety of the LBS system. Additionally, LBS providers will only need to add a geohash encoding field in the original POI record, whose mechanism is useful in solving the multi-index ranking of relational databases.

3.2 System Architecture

The system architecture is shown in Fig. 1, which comprises a lightweight trusted agent module, pre-LBS system module, and LBS system. Although the LBS client does not belong to the system architecture, it is the starting and ending point of information.

The order of execution is 1, 2, 3, 4, 5, and 6, as shown in Fig. 1.

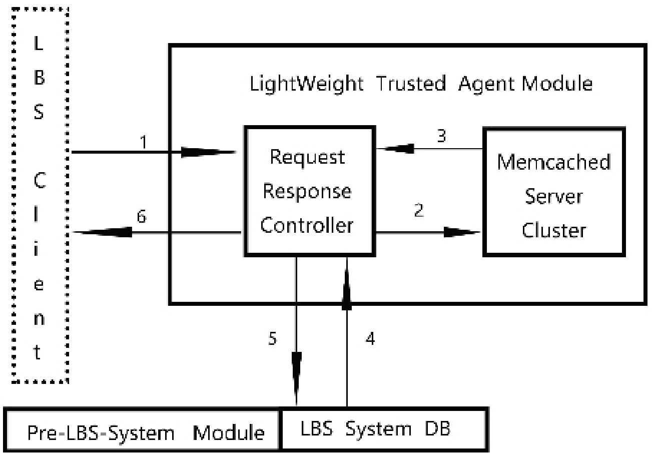

Fig. 1. System architecture.

3.3 Function Definition

As shown in Fig. 1, the system model comprises four parts: the client, lightweight trusted agent module, pre-LBS-system module, and LBS system.

Definition 1. An intelligent terminal refers to electronic devices with positioning function, such as smart mobile phones, tablet computers, navigators, and desktop computers.

Definition 2. A lightweight trusted agent module is connected to an intelligent terminal and LBS system, with a distributed structure and a large capacity cache.

Definition 3. A pre-LBS-system module is a software “module” used to translate the latitude and longitude of POIs of the original LBS systems into a geohash sequence.

Definition 4. An LBS system is a retrieval system for obtaining POIs.

4. Preliminaries

4.1 Geohash

Geohash encoding is a type of address coding technology that can convert the location information of every point on the planet from longitude and latitude encoding to hash-coding. At a given accuracy, the two-dimensional latitude and longitude information is translated into the corresponding one-dimensional string code, which was originally used in genhash.org services [20]. Currently, geohash encoding is widely used in addressing spatial data, such as Google App Engine, MongoDB, HBase, Lucene, and Solr [21-23].

To introduce the geohash encoding algorithm, a point (28.026380, 120.642445) was used as a reference, which is located in Wenzhou City, Zhejiang Province, China. The latitude coding is shown in Table 1. First, the latitude range (-90, 90) was divided into two intervals, i.e., (-90, 0) and (0, 90). If the target latitude value of 28.026380 is within the former interval, then the target zone code is set as 0; otherwise, it will be set as 1. In this example, the latitude value (28.026380) belongs to (0, 90); therefore, the target zone code was set as 1. Subsequently, by dividing the interval (0, 90) into (0, 45) and (45, 90), it is clear that the latitude value (28.026380) belongs to (0, 45). Consequently, the next target zone code was set as 0. By analogy, the calculation was performed until the accuracy satisfied the requirements. After the calculation was completed, the latitude was encoded as 1010 0111 1101 1100 0001.

Table 1. Relationship between geohash bit(s) and distance

Bit of geohash sequence	Bit of latitude	Bit of longitude	Distance in latitude	Distance in longitude	Spherical distance (km)
1	2	3	±22.5	±22.5	±2500
2	5	5	±2.8125	±5.625	±630
3	7	8	±0.703	±0.703	±78
4	10	10	±0.0879	±0.176	±20
5	12	13	±0.022	±0.02197	±2.4
6	15	15	±0.00275	±0.00549	±0.6
7	17	18	±0.000688	±0.000687	±0.076
8	20	20	±0.000086	±0.000172	±0.019
9	22	23	±0.000021	±0.0000215	±0.002

The same algorithm could be used to encode the longitude. The longitude 120.642445 can be encoded into a string of 1101 0101 1100 1110 0101. Next, the longitude and latitude are merged from the right to the left; the odd digit is the latitude, while the even number is the longitude. The combined coding string can be obtained, i.e., 11100 11000 11011 11111 00011 11110 00001 00011. From the right to the left, every 5 bits are stipulated as a unit; if the number of bits is less than 5, then a zero-padded must occur on the left. According to the coding rules of base32, the set of latitude and longitude (28.026380, 120.642445) can be encoded into “wsvz3y13”. This coding algorithm can be easily implemented by programming languages.

The decoding process is as follows: first, wsvz3y13 was converted into 11100 11000 11011 11111 00011 11110 00001 00011 in accordance with base32; subsequently, the latitude and longitude strings were separated; finally, the latitude and longitude ranges were subdivided according to the binary code sequence. Because geohash represents a certain interval, a longer coding length results in a higher precision. However, it is impossible to decode a completely consistent address.

4.2. Memcached

Memcached is an open-source, high-performance distributed memory object caching system based on text line protocol and libevent event handling mechanism, with a C/S and B/S hybrid architecture; it can be deployed into a distributed cluster system. All the data are organized in a large hash table in memory and accessed by a key/value interface. This mechanism reduces the frequency of access to the database significantly and improves the response speed of back-end services [23].

5. Model

The operation flow diagram of the model is shown in Fig. 2.

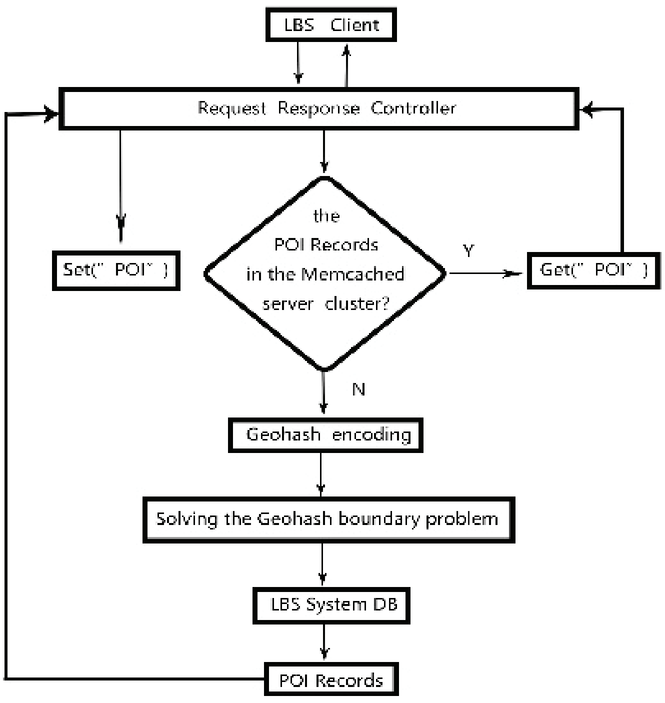
LBS Client
Request Response Controller
Set(" POI")
the POI Records in the Memcached server cluster?
Y
Get(" POI")
N
Geohash encoding
Solving the Geohash boundary problem
LBS System DB
POI Records

Fig. 2. Operation flow diagram of the model.

The LBS client sends a POI query to the request response controller. The controller first queries whether POI records exist in the memcached server cluster. If they exist, then they are sent directly to the LBS Client. Otherwise, the longitude and latitude of the POI query are first converted to geohash encoding, with the problem of boundary catastrophe solved, and then sent to the third party's LBS system DB. POI record sets are retrieved from the third-party LBS system and then returned to the request response controller in a secure form. Subsequently, the controller pushes the POI record sets to the LBS client and writes them to the memcached server cluster.

5.1 Client End

In the traditional LBS model, an LBS user can directly send the query keywords, such as "find industrial and Commercial Bank of China 1 kilometers around" to the LBS providers. The query request is formatted by the application (APP) of users' intelligent equipment into four parameters and subsequently submitted directly to the LBS server program for processing. As an example, based on the author's current position, the four parameters are as follows:

Lat = 30.527883,
Lon = 114.36384,
Range = 1000,
Keyword = Industrial and Commercial Bank of China.

After receiving the user's query request, the LBS application system will query POI records in the LBS database, standardize the results, and subsequently respond to the user-side APP.

As shown in Fig. 1, in this geohash encoding model, the geohash boundary is solved and the standardizing query results processed at the user side.

5.1.1 Geohash encoding

The process of latitude manual coding in Section 4.1 can be performed repeatedly. The latitude x = 28.026380 always belongs to a certain range [a, b]. The range decreases with the iterations, and its value will eventually converge to 28.026380. Therefore, for any given ε, an N always exists such that $\delta = \left|\frac{x-a}{2^N}\right| \leq \varepsilon$ is true, where x can be any latitude. When x = 28.026380, the coding string 1010 0111 1101 1100 0001 can be obtained after 20 iterations (N = 20). On the contrary, we can retrace latitude x belonging to the interval (28.02629471,28.02646638) from the string 1010 0111 1101 1100 0001; however, one cannot ascertain whether the x is equal to 28.026380, which means that an error may exist although this algorithm is reversible.

It is impossible to execute an infinite iteration, and it is unnecessary to set N as infinity in the application. In the N-step iteration, the interval length to which x belongs is $\frac{b-a}{2^N}$. Using a latitude as an example: the length of the latitudinal interval is $\frac{180}{2^N}$ and its error is equal to $\frac{90}{2^N}$ or $-\frac{90}{2^N}$ .When N = 20, the value of this error is $\frac{90}{2^N} = \frac{90}{2^{20}}$ = 0.000086; furthermore, $-\frac{90}{2^N} = -\frac{90}{2^{20}}$ = -0.000086 and N = 20, x = 28.026380 is set, where x belongs to (28.02629471, 28.02656638). The calculation (28.02646638-28.02629471)/2 valid~~a~~tes the formula.

For real LBS-based applications, Table 1 can be used as a reference for the errors between the lengths of geohash-based encoding and geohash-based latitude and longitude, as well as the maximum error based on spherical distance.

Table 1 shows that the accuracy is sufficiently high when the geohash sequence is 9-bits long in an LBS application.

According to the previous geohash manual encoding method, an algorithm to generate a geohash sequence and the required parameters of geohash sequence length as well as latitude and longitude can be designed easily.

Algorithm 1. Geohash encoding

```
1. Input: latitude, longitude, depth of Geohash sequence: depth.
2. Output: result of Geohash encoding: geohash.
3. Initialization parameters: BASE3←"0123456789bcdefghjkmnpqrstuvwxyz," bits←array (16, 8, 4,
   2, 1), lat←array (-90.0, 90.0), lon←array (-180.0, 180.0). The count constant bit, ch, and i are 0.
   The flag bit of the latitude and longitude parameter is even=1, the temporary variable of interval
   halving is mid, and the geohash is defined to store temporarily generated geohash encoding.
4. According to the geohash encoding rules: first, the longitude is divided into two sections;
   subsequently, the latitude and longitude are divided sequentially. In each division, the target value
   of the latitude or longitude belongs to the former (represented by a value of 0) or the latter
   (represented by a value of 1). Merging 0s and 1s, one base32 code is generated according to every
   five bits.
    a) Partition interval.
    While i<deep
    {
        if it is even, it is true
        {
            mid ←The median of the original longitude range
            if longitude >mid
            {
                ch ← the result of the "or" operation between the bits array and the
                original ch variable. The bits array starts from subscript 0, and 1 is added
                to the subscript after each longitude or latitude is processed. When the
                value is 5, the value of subscript is set to 0.
                lon ← (mid, lon[1])
            }
            if longitude < mid
            {
                lon← (lon[0], mid)
            }
        }
        if it is even, it is false
        {
            mid ←The median of the original latitude
            if latitude>mid
            {
                ch← the result of the "or" operation between the bits array and the
                original ch variable. The bits array starts from subscript 0, and 1 is added
                to the subscript after each longitude or latitude is processed. When the
                value is 5, the value of the subscript is set to 0.
```

```
                        lat ← (mid, lat [1])
                }
                if latitude< mid
                {
                        lat ← (lat [0], mid)
                }
        }
    b) Converted to 1-bit base32 code for each 5-bit binary number.
        ! is even
        {
                Determine whether the bits array subscript variable bit is 4; if it is less than 4, then
                add 1 to the bit; otherwise, it means that 5 bits of latitude and longitude have been
                processed, and a base32 code will be obtained. Simultaneously, the variable I plus 1
                and the bit, and ch are reset.
        }
    }
    Return the geohash sequence.
```

5.1.2 Solving the geohash boundary problem

Geohash is Peano space filling curve, and the mutation of the Peano curve is highly abrupt [21]. Therefore, the geohash boundary problem must be solved. The steps performed were as follows:

1. Draw a circle with the center of the users' current location expressed as (Lat, Lon), and radius r = Range.
2. Search a region that is the circumscribed rectangle area of the circle.
3. Determine the retrieval accuracy based on Table 2, which ensures the number of geohash sequence bits required.
4. Calculate the geohash sequence with the specified accuracy for the current users' position. The key point of this calculation is to determine whether the lower-left and upper-right points of the rectangle are covered in the area indicated by the geohash sequence. If both of them are covered, then the geohash is the only query parameter; otherwise, further calculations are required to determine the 8-geohash sequence, which is close to the acquired geohash sequence, and the 9-geohash sequence's intersection is the query parameter.

Table 2. Time of extracting various POI records in different schemas

POI records	5-bit geohash sequence (sec)		6-bit geohash sequence (sec)	
	Based on coordinates of latitude and longitude	Based on geohash sequence	Based on latitude and longitude coordinates	Based on geohash sequence
1,000	169.196	0.188	153.216	0.186
10,000	1,714.140	1.823	1,425.206	1.608
100,000	8,312.128	18.178	14,263.218	15.733

5.1.3 Standardize query results

As shown in Fig. 1, after submitting a POI query, the user will receive POI data returned from the lightweight trusted proxy module; meanwhile, its location information will be expressed as latitudes and longitudes, and the users' LBS APPs will filter, sort, and format these data.

5.2 Lightweight Trusted Agent Module

The lightweight trusted agent module includes a request response controller and a memcached server cluster, as shown in Fig. 1.

5.2.1 Data structure and operation flow

The query request's data structure sent to the request response controller from an LBS client is (userid, geohash, query, timestamp), and the userid uniquely identifies the user device. According to the userid, the request response controller can send the search results to the LBS client. In one query, the geohash parameter may include many different geohash sequences; the query parameter includes the classification information and key word of the query, and the timestamp parameter is the query's time stamp.

The memcached value is expressed as a key value by setting key={geohash} and value={POI, CreatedTimestamp, accessFrequency}. Here, POI refers to the record set returned from the LBS provider's system, including the latitude and longitude information of the POI, administrative divisions, user comments, and other expandable information. CreatedTimestamp refers to the datetime that the recordset is written into the memcached server, while accessFrequency displays the number of times the recordset has been visited. Both createdTimestamp and accessFrequency are indicators for selecting an algorithm for the memcached server's data elimination mechanism.

The operational procedure of the model is as follows:

1. The request response control module will verify whether the request data are in the memcached server after receiving a query request from an LBS client. If the request data are present, the corresponding POIs are returned from the LBS client's request data to the LBS requester, and the LBS provider's database is not operated. The processing path is 1326. Otherwise, the request response control module will query the LBS provider's database and return the POIs obtained by the query operation to the LBS requester. Meanwhile, a copy to the memcached server will be set. The processing path is 1324563.
2. Once the LBS server database has begun updating, the request response controller is notified to update the data in the memcached server, which enables the data in database and memcached server to remain consistent.
3. If the memcached is out of memory, then the corresponding data in memcached will be deleted according to the expiration policy and the least-recently-used strategy. In the latter, memcached uses a lazy-expiration strategy, and whether the key/value is overdue will not be monitored. Whether the key/value is expired will be verified when the key/value is obtained by viewing the time record, which can reduce the load of the memcached server.

5.2.2 Memcached distributed architecture

The memcached servers are originally independent and disabled for communication with each other, and they do not have a distributed architecture. In this architecture, the request response control module serves as the memcached's client. The control and management of the memcached server are unified by the request response control module, which realizes the distributed function. The writing and reading of data in a memcached cluster distributed structure are shown in Figs. 3 and 4, respectively.

Two steps are required to access data in a memcached cluster: first, select a memcached server and then directly access the data.

Consistent hashing involves two typical algorithms: the remainder and hash algorithms. The remainder algorithm is simpler and more efficient than the hash algorithm. Therefore, in this study, the remainder algorithm was adopted.

The process of the remainder algorithm is as follows: first, the integer hash value of a key is calculated; subsequently, the number of servers is divided by the integer hash value; finally, the access server will be identified according to the value of the remainder, i.e., the remainder is the number of target servers.

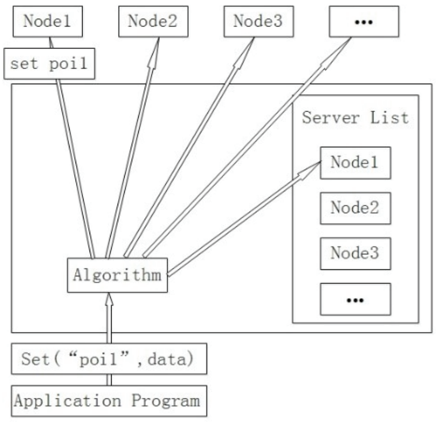

Fig. 3. Writing data in memcached cluster distributed structure.

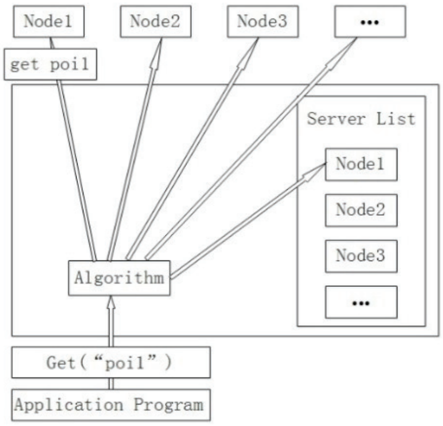

Fig. 4. Reading data in memcached cluster distributed structure.

5.2.3 Security controls

1. Two-way authentication between a request response controller and an LBS server.
2. The data, which are returned from the LBS provider database in a query origin by a query response controller, are encrypted and stored in memcached. The latter contains only the application ID of the LBS client, including the LBS provider information, POI information, and time stamp. This mechanism ensures the security of the application, i.e., the data are encrypted during information storage and transmission, and they do not contain any LBS client identity information in memcached.
3. The statement of querying LBS server databases, formatted with a tuple containing the userid, geohash, query, and timestamp, does not contain LBS users' identity information. It is impossible to leak LBS users' privacy at the LBS server end.

4. All the operations and data between the memcached cache and LBS server database are isolated.
5. In a distributed architecture, the memcached server cannot recover data from disasters.

5.2.4 The realization mechanism of k-anonymous privacy protection

When an LBS user initiates a POI query, a lightweight trusted agent module will transform the user's latitude and longitude information into geohash coding. Within a specific time period, multiple LBS users in a specific region may have the same Geohash coding despite their different actual latitude and longitude information, thereby achieving the goals of fuzzy position and k-anonymity.

An LBS user's latitude and longitude information will be identified when he/she initiates a POI query through an APP. Once the query information enters the lightweight trusted agent module, fuzzy location and k-anonymity will be achieved. A two-way transformation occurs between the latitude/longitude and geohash coding within the lightweight trusted agent module. Although the final POI information obtained from the LBS provider contains latitude and longitude information, it is impossible to trace back to the latitude and longitude information of the real initiator and the exact APP through which the POI query is initiated.

5.3 Pre-LBS System Module

In this system, the sequence of the geo-geohash coding is set to a 45-bit string; the corresponding base32 encoding is 45/5 = 9 bytes, i.e., the length of the geohash encoding is 9 bits. To adapt to the searching of the position accuracy, a module is built into the LBS service system, and the field "geohash, varchar (9)" is added to the original POI information table to store the 9-bit-geo-geohash coding that corresponds to the latitude and longitude data of the POI records.

In the LBS provider system, the POI retrieval operation from the agent module is executed. By matching the location parameters and the geohash, the database's multidimensional external sort can be avoided. The latitude and longitude data of the POI records and the geohash are both included in the returned query results.

5.4 Improvement of LBS Provider System

In this study, an LBS request response control module was added into the LBS service system, between the LBS client end and the service. The authentication between the module and the LBS server is important.

It is assumed that the URL to access the LBS request response control module is "https://contror/index.php." One of the LBS providers is named LBSA, who provides a POI query interface for the LBS request response control module, while the parameter is set by the URL of "https://contror/index.php":

URL: https://lbs-servera/api.php,

Application-Token: password for this service,

Encoding-AES-key: Message encryption and decryption key.

After the LBS request response control module receives information from LBS service provider LBSA, the following procedures will be executed.

5.4.1 LBS request response control module posts data to LBS service

When the data from LBS service provider LBSA arrive at the LBS request response control module, a POST request shall be sent to the URL provided by LBSA, using the arguments of S, T, RN, and RS, which refer to the signature, application time stamp, application of random number, and application of random string, respectively. Combined with the application-Token, T parameters, and RN parameters, S as an encrypted digital signature is filled by the POST request from LBSA.

5.4.2 LBSA verifies the parameters from the POST request

LBSA verifies the POST request by the program, and the core code in the PHP version is listed as follows:

```
$tArray = array ($application-Token, $T, $RN);
sort ($tArray, SORT_STRING);
$tString = implode ($tArray);
$tString = sha1($tString);
if ($tString == $S) {
   …
}
```

In particular, the application-Token parameter is set as a constant in the validation page:

if $tString == $S, then LBSA must return the value of RS. After receiving the RS, the request response controller will verify the authenticity of LBSA's identity. Two aspects are illustrated in this process:

1. The application-token submitted to the LBS request response control module is the password of LBSA.
2. The URL requested from the LBS request response module is registered in the LBS request response module by LBSA.
3. Obtaining data by visiting the URL of https://lbs-servera/api.php.

The core code in the PHP version for the LBS request response module to obtain API data from LBSA is written as follows:

```
function get_post ($u, $p)
{
   $o = array ( );
   return file_get_contents ($u, false, stream_context_create($o));
}
```

Defines the query string $q, and calls the get_post() function to obtain the value.

```
$q=array ('POI_BX'=>$lon,'POI_BY'=>$lat,'POI_TYPE'=>$type, T=>$T, RN=>$RN, RS=>$RS, S=>$S);
$data =get_post ($url, $q);
```

When the page of https://lbs-servera/api.php has received the request from the LBS request response module and the query connection parameters are verified, the POI queries will be executed; subsequently, the result will be returned to the LBS request response module.

The API combined with JSON (instead of Services Web) is a lightweight and platform-independent solution that contributes to the access cost reduction of the LBS provider.

6. Performance Analysis

An LBS information system based on the lightweight location privacy protect model was implemented.

In this section, a performance evaluation scheme is designed for the new model, which is composed of two parts: one is a contrast experiment of the query processing efficiency in LBS application systems based on the latitude and longitude of the users' location and the geohash encoding; the other is a contrast experiment of the response times and the agent module, based on memcached, redis, or tair.

6.1 First Experimental Environment Setting

A program was implemented to observe and record the times of extracting 1,000, 10,000, and 100,000 POIs records from 80,000 real POIs by a stochastic function in the precise sequence of 5-bit and 6-bit geohash in LBS application systems based on the latitude and longitude of the users' location and the geohash sequence. The results are shown in Table 2.

6.2 First Experimental Result Analysis

Certain generalizations can be derived from the data in Table 2.

1. The data accuracy of geohash encoding is closely related to the POI query efficiency. A higher-accuracy geohash results in more bits in a geohash sequence, which represents a smaller region and fewer POI records in an LBS database; furthermore, the times of query and results returned from LBS system normalization are reduced considerably.
2. The total time to return the POI records from an LBS system based on an agent module is 1/1000 that of a system without an agent module; furthermore, regardless of the number of POI records, this feature is stable.
3. The reason contributing to the good performance of our LBS service system based on geohash encoding is analyzed. Primarily, two factors exist: first, from the design of the system's database and the perspective of technology realization, geohash encoding uses the string form; the partition technique provides a good solution to the storage management of mass POI records in an RDBMS, and the database's multidimensional external sort is avoided; it is detrimental when a client frequently queries the rows in an RDBMS with large tables. The other factor is that the LBS has received a query parameter, including the latitude and longitude information of the user's position and radius, and the POI records must be determined by a specific function.

6.3 Second Experimental Environment Setting

Another program was conducted to observe and record the times of extracting 1,000, 10,000, and 100,000 POIs records from 80,000 real POIs by a stochastic function in the precise sequence of 5-bit and 6-bit geohash in LBS application systems based on memcached, redis, or tair. The results are shown in Table 3.

Table 3. Query time of agent module based on memcached, redis, or tair

POI records	5-bit geohash sequence (sec)			6-bit geohash sequence (sec)		
	memcached	redis	tair	memcached	redis	tair
1,000	0.188	0.165	0.202	0.186	0.163	0.200
10,000	1.823	2.419	3.470	1.608	2.144	3.055
100,000	18.178	Time out	33.374	15.733	Time out	28.948

6.4 Second Experimental Result Analysis

Certain generalizations can be derived from the data in Table 3.

1. As shown in the Table 3, the geohash sequence length imposes a certain effect on the query efficiency. The longer the geohash sequence, the higher is the query speed.
2. The retrieval performance of the LBS system based on redis decreases significantly in a large sample, and a timeout occurs while querying for more than 100,000 POIs. Under the same conditions, only 18.178 seconds are required by the memchached-based LBS system. Compared with redis and tair, the memcached-based LBS system yields better performance. The differences between memcached, redis, and tair can be attributed to the distributed architectures and the multithreaded LBS systems.

6.5 Performance and Privacy Protection Capability Comparison

As shown in Tables 2 and 3, our memcached-based models are superior to redis or tair for POI records in 5-bit and 6-bit geohash sequences.

Data anonymity has garnered significant attention because it enables many types of data to be processed and real data to be published, which can satisfy the requirements of many practical applications. Data anonymity is a complex process, which requires many factors to be considered, such as raw data, anonymous data, background knowledge, anonymity technology, and attack. The results are shown in Table 4.

The solutions for geohash primarily include the following: (1) sphinx geohash index, (2) MongoDB geohash index, and (3) redis geohash index and memcached geohash index. Performance comparison from the following aspects: (1) support sorting by distance, (2) support paging, (3) support multi-condition screening, and (4) occupy resources. The results are shown in Table 5.

Table 4. Performance evaluation of privacy protection technology

Solution	Privacy protection	Computing overhead	Data defect	Data dependence	Communicati on overhead
Privacy-preserving technology based on data distortion	Medium	Low	High	High	Low
Privacy-preserving technology based on data encryption	High	High	Low	Low	High
Data anonymization	High	Medium	Medium	Low	Low
Our solution	High	Low	Low	Low	Low

Table 5. Performance evaluation of solutions based on geohash

Solution	Support sorting by distance	Support paging	Support multicondition screening	Occupy resources
Sphinx+geohash	Supporting	Supporting	No supporting	More resources
mongodb+geohash	Supporting	Supporting	Supporting	More resources
redis+geohash	No supporting	Supporting	No supporting	Average level
memcached+geohash	supporting	Supporting	Supporting	Less resources

Privacy protection capability involves two basic indicators: query privacy (QP) and location privacy (LP). The former means protecting the corresponding relationship between query requests and users, while the latter refers to protecting accurate user location information. This geohash-based privacy protection model is better than other privacy protection methods in terms of query efficiency and accuracy (Table 6).

Table 6. Comparison of geohash with other query privacy methods

Method	Efficiency	Accuracy	Privacy
Mixed-zone	Low	Medium	QP
Hidden-zone	Medium	Low	QP, LP
Disturb	Low	Low	QP, LP
Geohash	High	High	QP, LP

7. Conclusions

In this study, a new system architecture was designed to guard privacy in LBS applications. Trusted agents were added to LBS clients and servers. The added geographical coordinates were expressed by geohash encoding, and this system architecture was efficient and lightweight. Using a geohash-based coding method, the two-dimensional information of latitude and longitude was reduced into a one-dimensional string. Every geohash encoding string was a different region identifier, which was important for protecting users' location privacy. Additionally, the geohash encoding string could avoid the sorting of multidimension indexes in a large set of data when querying POIs.

A query controller and memcached server cluster were designed for improving efficiency and security; therefore, they were critical to the model. Because of the cache mechanism, it was extremely difficult for the LBS server to identify the real inquirer from the LBS application client, and its effect was highly significant when few LBS clients existed or consecutive queries were generated from users. This agent module was a barrier between the LBS client and provider, and all the data were encrypted during storage and communication.

Peano space filling curves were adopted for the current geohash algorithms. These curves often mutate and yield similar coding with a large distance. Therefore, while querying the POI in adjacent regions, POIs with similar geohash coding would be identified first for the calculation of practical distance, which would result in additional calculations. Geohash is spatial indexing method that applies well to point data but not to panel data.

Acknowledgement

This research is supported by Hubei Research Center for Educational Informationization (Central China Normal University).

References

[1] T. Xu, Y. Ma, and Q. Wang, "Cross-urban point-of-interest recommendation for non-natives," *International Journal of Web Services Research*, vol. 15, no. 3, pp. 82-102, 2018.

[2] A. Arampatzis and G. Kalamatianos, "Suggesting points-of-interest via content-based, collaborative, and hybrid fusion methods in mobile devices," *ACM Transactions on Information Systems*, vol. 36, no. 3, pp. 1-28, 2017.

[3] F. Abbas and H. Oh, "A step towards user privacy while using location-based services," *Journal of Information Processing System*, vol. 10, no. 4, pp. 618-627, 2017.

[4] N. S. Kumar and M. Thangamani, "Multi-ontology based points of interests (MO-POIS) and parallel fuzzy clustering (PFC) algorithm for travel sequence recommendation with mobile communication on big social media," *Wireless Personal Communications*, vol. 103, pp. 991-1010, 2018.

[5] J. Bao, C. Y. Chow, M. F. Mokbel, and W. S. Ku, "Efficient evaluation of k-range nearest neighbor queries in road networks," in *Proceedings of 2010 11th International Conference on Mobile Data Management*, Kansas City, MO, 2010, pp. 115-124.

[6] T. Novack, R. Peters, and A. Zipf, "Graph-based matching of points-of-interest from collaborative geo-datasets," *ISPRS International Journal of Geo-Information*, vol. 7, article no. 117, 2018.

[7] Z. Liu, D. Luo, J. Li, X. Chen, and C. Jia, "N-Mobishare: new privacy-preserving location-sharing system for mobile online social networks," *International Journal of Computer Mathematics*, vol. 93, no. 2, pp. 384-400, 2016.

[8] C. Zhou, C. Ma, S. T. Yang, and Z. P. Li, "Low-communication inquiry method for protecting privacy-based LBS neighbor points of interest," *Journal of Sichuan University (Engineering Science Edition)*, vol. 47, no. 3, pp. 114-122, 2015.

[9] A. Papageorgiou, M. Strigkos, E. Politou, E. Alepis, A. Solanas, and C. Patsakis, "Security and privacy analysis of mobile health applications: the alarming state of practice," *IEEE Access*, vol. 6, pp. 9390-9403, 2018.

[10] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K. L. Tan, "Private queries in location based services: anonymizers are not necessary," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, Vancouver, Canada, 2008, pp. 121-132.

[11] E. Toch, C. Bettini, E. Shmueli, L. Radaelli, A. Lanzi, D. Riboni, and B. Lepri, "The privacy implications of cyber security systems: a technological survey," *ACM Computing Surveys*, vol. 51, no. 2, pp. 1-27, 2018.

[12] S. He and S. H. G. Chan, "Tilejunction: mitigating signal noise for fingerprint-based indoor localization," *IEEE Transactions on Mobile Computing*, vol. 15, no. 6, pp. 1554-1568, 2015.

[13] B. Niu, X. Zhu, Q. Li, J. Chen, and H. Li, "A novel attack to spatial cloaking schemes in location-based services," *Future Generation Computer Systems*, vol. 49, pp. 125-132, 2015.

[14] C. Y. Chow, M. F. Mokbel, and X. Liu, "Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments," *GeoInformatica*, vol. 15, no. 2, pp. 351-380, 2011.

[15] Y. F. Chen, X. J. Liu, and B. Li, "Collaborative position privacy protection method based on game theory," *Computer Science*, vol. 40, no. 10, pp. 92-97, 2013.

[16] Y. Yang and J. Yuan, "Research on incredible users cooperate constructing anonymous region in LBS," *Computer Engineering and Applications*, vol. 50, no. 14, pp. 82-87, 2014.

[17] D. Yang, X. Fang, and G. Xue, "Truthful incentive mechanisms for k-anonymity location privacy," in Proceedings of *2013 IEEE INFOCOM*, Turin, Italy, 2013, pp. 2994-3002.

[18] D. Levin, K. LaCurts, N. Spring, and B. Bhattacharjee, "Bittorrent is an auction: analyzing and improving bittorrent's incentives," in *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication*, Seattle, WA, 2008, pp. 243-254.

[19] M. B. Brahim, W. Drira, F. Filali, and N. Hamdi, "Spatial data extension for Cassandra NoSQL database," *Journal of Big Data*, vol. 3, article no. 11, 2016.

[20] S. B. Kylasa, G. Kollias, and A. Grama, "Social ties and checkin sites: connections and latent structures in location-based social networks," *Social Network Analysis and Mining*, vol. 6, article no. 95, 2016.

[21] S. Nishimura, S. Das, D. Agrawal, and A. El Abbadi, "*MD*-HBase: design and implementation of an elastic data infrastructure for cloud-scale location services," *Distributed and Parallel Databases*, vol. 31, no. 2, pp. 289-319, 2013.

[22] X. Ye, J. Xu, L. Lu, X. Li, X. Fang, and J. Kong, "Equipment-free nucleic acid extraction and amplification on a simple paper disc for point-of-care diagnosis of rotavirus A," *Analytica Chimica Acta*, vol. 1018, pp. 78-85, 2018.

[23] A. Tabarcea, N. Gali, and P. Franti, "Framework for location-aware search engine," *Journal of Location Based Services*, vol. 11, no. 1, pp. 50-74, 2017.

Lihao Ni **https://orcid.org/0000-0002-8049-6151**

He received his B.S. degree in Mechanical Engineering from Jiamusi University in 2006. He received his M.S. degree in Software Engineering from Huazhong University of Science & Technology in 2012. He is currently a senior engineer of the Wenzhou University of science and technology. His research interests are location based services and big education data.

Yanshen Liu **https://orcid.org/0000-0002-4110-5891**

He is currently the professor and director of Hubei Research Center for Educational Informationization. He is the visiting scholar of Columbia University and Kent State University. His research interests are multimedia technology and distance education.

Yi Liu **https://orcid.org/0000-0003-0748-8275**

She received her bachelor's degree and master's degree in Computer Science from Central China Normal University respectively in 1996 and 2001, respectively. She received her Doctor of Technical Science in Management Science and Engineering in Huazhong University of Science & Technology in 2007. She is currently a senior engineer of the Central China Normal University (CCNU). Her research interests are modern educational theory and experimental techniques.