information

Article

Location Privacy-Preserving Query Scheme Based on the Moore Curve and Multi-User Cache

Zhenpeng Liu [1,2], Qiannan Liu [1], Jianhang Wei [2,3,*], Dewei Miao [1] and Jingyi Wang [4]

[1] School of Cyberspace Security and Computer, Hebei University, Baoding 071002, China
[2] Information Technology Center, Hebei University, Baoding 071002, China
[3] Network and Experiment Management Center, Xinjiang University of Science & Technology, Korla 841000, China
[4] School of Electronic Information Engineering, Hebei University, Baoding 071002, China
* Correspondence: wei@hbu.edu.cn

Abstract: With the rapid development of the Internet of Things, location-based services have emerged in many social and business fields. In obtaining the service, the user needs to transmit the query data to an untrusted location service provider for query and then obtain the required content. Most existing schemes tend to protect the user's location privacy information while ignoring the user's query privacy. This paper proposes a secure and effective query privacy protection scheme. The multi-user cache is used to store historical query results, reduce the number of communications between users and untrusted servers, and introduce trust computing for malicious users in neighbor caches, thereby reducing the possibility of privacy leakage. When the cache cannot meet the demand, the user's location coordinates are converted using the Moore curve, processed using encryption technology, and sent to the location service provider to prevent malicious entities from accessing the transformed data. Finally, we simulate and evaluate the scheme on real datasets, and the experimental results demonstrate the safety and effectiveness of the scheme.

Keywords: location-based services; privacy protection; multi-user cache; trust computing; Moore curve

Citation: Liu, Z.; Liu, Q.; Wei, J.; Miao, D.; Wang, J. Location Privacy-Preserving Query Scheme Based on the Moore Curve and Multi-User Cache. *Information* **2022**, *13*, 417. https://doi.org/10.3390/info13090417

Academic Editors: Sokratis Katsikas and Sherali Zeadally

Received: 27 June 2022
Accepted: 26 August 2022
Published: 6 September 2022

1. Introduction

As a typical application of intelligent terminals, location-based services [1,2] (LBS) have received much popularity and are becoming one of the fastest-growing services. The user sends their interests or locations to the location service provider (LSP). Then LSP returns points of interest near the user's current location according to the user's interests [3]. However, the LSPs may deduce the user's personal information based on the collected user query record information and even disclose the information to a third party to obtain commercial interests [4]. This not only results in the leakage of user location query information but also may lead to the leakage of more sensitive information, so it is essential to protect user privacy.

Given the privacy problem of location query in LBS, experts have proposed many methods, such as dummy location [5], spatial transformation [6], spatial cloaking [7], and encryption technology [8]. Although these methods can protect users' privacy, there are still some shortcomings. Most existing schemes pay attention to the user's location privacy while ignoring the user's query privacy. In location-based services, the user's privacy is tied to both, losing one will also implicate the other. Therefore, two aspects must be considered to protect users' privacy better.

The cache-based method [9] means that each node in the system can be regarded as a simple server, and its role is to provide queries for other nodes. However, most of the existing cache methods rely on the assumption that the nodes in the cache are credible, which is unlikely to be the case under realistic conditions. For example, Cui et al. [10] proposed a cache-based LBSs user privacy scheme. This method avoids the central attack

problem caused by the TTP server through cooperation between users, thereby reducing the possibility of privacy leakage. However, when there is untrusted behavior between users, it is necessary to ensure that malicious behavior between users can be detected and effectively suppressed to achieve privacy protection for users.

Based on the above considerations, this paper proposes a location privacy-preserving query scheme based on the Moore curve and multi-user cache. Unlike the previous methods, this paper not only protects the user's location privacy and query privacy but also uses trust computing to suppress malicious users and achieve trusted multi-user collaboration. For queries that neighbor users cannot solve, the data is transformed using the Moore curve, encrypted, and sent to the LSP for processing. The main contributions are summarized below.

(1) Given the untrustworthy problem of the location servers, we introduce cache technology to effectively utilize historical query data, thereby reducing the number of times users send to LSPs and reducing the risk of user location privacy leakage.
(2) Aiming at the problem of distrust among neighbor users, the authentication and trust calculation strategies are adopted to eliminate the mobile user equipment whose average weighted trust value is lower than the threshold, so as to protect the security of neighbor users.
(3) We conduct sufficient experiments on the scheme on real and uniform datasets. The results show that the proposed location query privacy protection method not only protects the user's query privacy and location privacy but also improves the query efficiency.

The organization of this article is as follows. We introduce the related works in Section 2. In Section 3, we introduce the definition of the scheme. Subsequently, the system model is given in Section 4. In Section 5, we introduce the system structure of the scheme. In Section 6, we conduct a safety analysis of the scheme. Next, we perform the simulation analysis of the scheme in Section 7. Finally, we conclude the article in Section 8.

2. Related Works

2.1. Query Privacy

Point-of-interest queries often involve searching more extensive data sets, and most solutions rely on spatial partitioning mechanisms [11] to reduce the search space. Zhu et al. [12] proposed an efficient and secure polygon spatial query framework for location-based services. LSP outsources encrypted LBS data to cloud servers, and registered users can query any polygon range to obtain accurate results. Zen et al. [13] proposed a practical privacy protection general LBS query scheme, which implements data privacy and query privacy. Privacy protection is achieved based on the CircleTest building block and cryptographic reverse indexing technology, but the client-side needs to decrypt the n-dimensional vector matrix. Biswa et al. [14] proposed a lightweight framework for finding the nearest neighbor, using the distance between the query point and the distant candidate neighbor to optimize the search space. Wang et al. [15] index multi-criteria POIs with MRS-tree, based on which they proposed an efficient solution to query processing and authentication. Lian et al. [16] proposed an efficient spatial query protocol and another new secure spatial query protocol. The protocol uses full homomorphic encryption to privatize query processing of outsourced data, transform data using the Moore curve, and use random arrangements to protect location privacy. Kim et al. [17] proposed an encryption scheme based on the Hilbert curve transformation to balance data security and query efficiency. They used the Hilbert curve to transform the data, hiding the original location. The data is clustered locally and then encrypted using AES, and the encrypted information is stored on the server. After the user requests a query, the LSP sends the retrieved encrypted data to the user, who decrypts it with a key. However, the scheme requires multiple communications. Wang et al. [18] treat edge nodes as an anonymous central server based on the potential for improved computational efficiency and data accuracy due to the potential for real-time data

preprocessing of edge nodes. Finally, they proposed a query scheme based on the Hilbert curve to divide two-dimensional spatial data and use differential privacy technology.

2.2. Cache Technology

Zhang et al. [19] are based on a unified mesh, employing order-preserved symmetric encryption (OPSE) and k-anonymity techniques combined with cache techniques to protect users' location privacy. Later, Zhang et al. [20] proposed a scheme to enhance user privacy through caching and spatial K-anonymity in continuous LBS, using multi-level caching to reduce the risk of user information exposure to untrusted LSPs. The method uses a Markov model for prediction and builds k-anonymity based on the location of the forecast, cache contribution rate, and data freshness. Jung et al. [21] proposed a P2P architecture that utilizes personal data storage (PDS) to protect user location privacy in LBS and introduced a collaborative caching technique that shares additional query results among users to mitigate performance degradation.

Compared to the above technologies, our solutions reduce communication overhead and increase efficiency while ensuring accurate information.

3. Preliminaries

3.1. Space-Filling Curve

From a mathematical point of view, a space-filling curve is a mapping function that converts N-dimensional spatial data into a one-dimensional continuous space. Elements that are adjacent in N-dimensional space are still adjacent to each other in one-dimensional space. The most commonly used are Z-curves and Hilbert curves [22]. Hilbert curve is widely used in the nearest neighbor query due to its superior clustering and location reservation characteristics. The Moore curve is a variant of the Hilbert curve with the feature of end-point-connected. Figure 1 compares the Moore curve of the third order with the Hilbert curve. The Moore curve passes through square cells of $2^n \times 2^n$, each represented by a Moore index, ranging from $[0, 2^{2n} - 1]$. The points of interest in its two-dimensional region can form circular structures with cyclic connectivity.

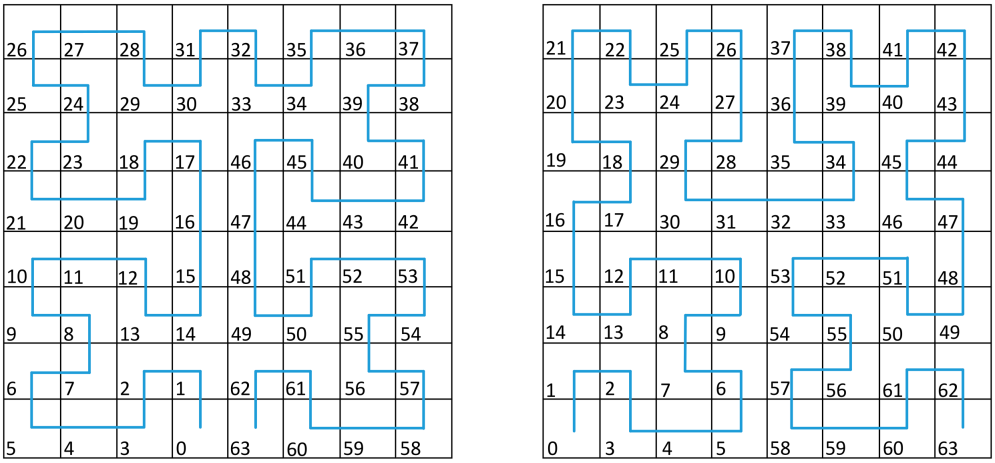


Figure 1. Moore curve and Hilbert curve.

3.2. Trust Calculation

The specific definition of a trust calculation consists of the following sections. Edge device set:

$$ED = \{d_1, d_2, \ldots, d_m\}$$

Edge node set:

$$EN = \{en_1, en_2, \ldots, en_m\}$$

1. Trust attribute set

The edge node formulates the trust attribute set of edge devices $d_j(d_j \in ED)$, represented by $TA = \{T_1, T_2, \ldots, T_l, \ldots, T_s\}$. In our scheme, transmission accuracy and speed are used as attributes to determine the trust value of the edge device.

2. Trust assessment set

After the $d_j(d_j \in ED)$ interacts with the d_j, the property values of the d_j are evaluated. The trust assessment set can be expressed as Equation (1). $e_{T_l}^{d_i d_j}$ denoted the result of the evaluation of the d_j of T_l against the d_i. Edge device evaluation $e_{T_l}^{d_i d_j}$ is expressed as Equation (2).

$$E_{d_i,d_j} = \left\{ e_{T_1}^{d_i,d_j}, e_{T_2}^{d_i,d_j}, \ldots, e_{T_l}^{d_i,d_j}, \ldots, e_{T_s}^{d_i,d_j} \right\} \tag{1}$$

$$\begin{cases} e_{T_l}^{d_i,d_j} = 0, & d_j \textit{ is insecure} \\ e_{T_l}^{d_i,d_j} \in (0, 0.5] & d_j \textit{ is secure but inefficient} \\ e_{T_l}^{d_i,d_j} \in (0.5, 1] & d_j \textit{ is secure and efficient} \end{cases} \tag{2}$$

3. Attribute weight set

When edge nodes perform trust management on d_j, the importance of each property is different. Therefore, our scheme introduces a set of attribute weights to meet the different trust requirements of the edge node. The set of attribute weights can be expressed as $\omega = \{\omega_1, \omega_2, \ldots, \omega_l, \ldots, \omega_s\}$, where $0 \leq \omega_l \leq 1$ and $\sum_{l=1}^{s} \omega_l = 1$.

4. Threshold

The threshold set by the edge node is represented by Equation (3), where τ is the threshold adjustment parameter.

$$\rho = \frac{\tau \sum_{j=1}^{m} E_{d_j}}{m} \tag{3}$$

4. System Models

In a traditional location service architecture, if a user wants to compute or cache data, he needs to forward the data to an anonymizer or a central server for processing. Our system makes full use of the advantages of local servers and neighbor users, allowing users to have their local cache, and able to use the cache of neighbors to query information, significantly reducing the number of data sent to LBS and increasing security. We introduce a trust computing strategy to ensure efficiency and security issues between neighbor users. Figure 2 shows the system model, which contains primarily two significant entities; the users include authenticated query user AU and neighbors user Ni. Each user carries a mobile device with functions such as information processing, data storage, and positioning. The user stores the historical query information in the cache and communicates with the surrounding neighbors through wireless communication protocols such as IEEE 802.11, Bluetooth, and radio frequency identification. LSP stands for Location Service Provider and is mainly responsible for computing data, providing user query services, and computing content that meets user needs using homomorphic encryption and returning it to users.

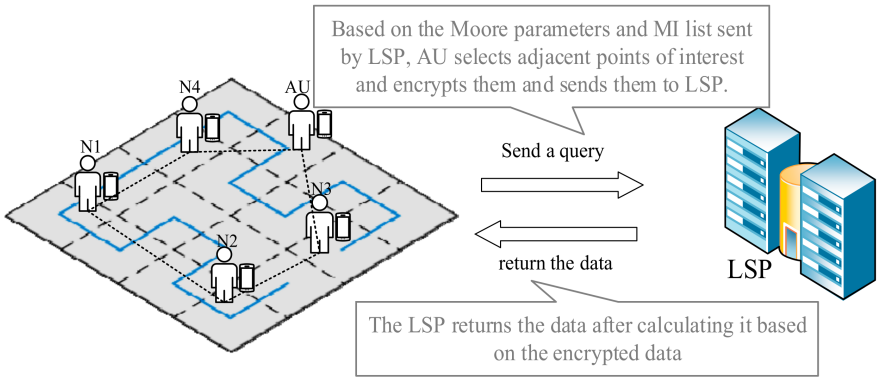


Figure 2. Query system framework.

5. Location Privacy-Preserving Query Scheme

This section uses a trust computing strategy [23] to suppress malicious behavior between neighboring users and improves on a query strategy based on Moore transformation and encryption.

5.1. Multi-User Cache Policy Based on Trust Calculations

The caching technology can reduce frequent communication between users and servers. For example, when a user requests the same data at different times, they can query from the local cache and a neighbor user without having to send the request to the LSP again.

When a user sends a broadcast to a neighbor node, malicious behavior can occur between neighboring users. We introduce a caching strategy with trust calculations to prevent malicious behavior in neighbor members. Trust computation is done by edge devices and edge nodes [24]. After the edge devices interact with each other, the trust value of each other is evaluated, and the edge node calculates the average weighted trust value of each user edge device. The device is expelled from the group for cases where the average weighted trust value falls below the threshold. The user who sends the query broadcasts a message, and after the neighbor receives the broadcast message, he searches for the data stored in the cache and sends it to the user if there is a matching point of interest. The solution not only protects users' privacy but also improves communication efficiency.

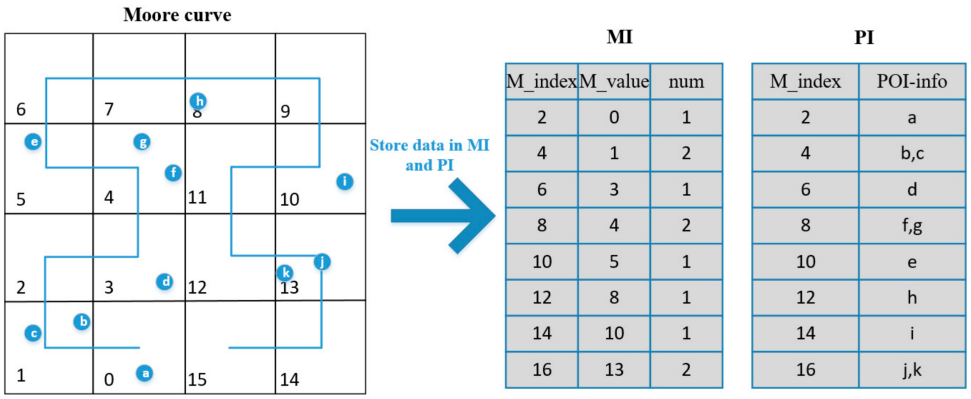
5.2. Query Strategies Based on Moore Transformation and Encryption

For queries that cannot be solved by caching, the scheme uses the methods in [25] to calculate. It uses the strategy of sending to the location service provider to obtain the information. While location service providers have the robust computing power to support large-scale data computation, large-scale data can cause an unavoidable time delay. Therefore, we convert the two-dimensional data of the point of interest into one-dimensional data according to the Moore curve, thereby reducing overhead. The data is stored in the POI-table and Index-table. The user generates the query content in the form of vectors, encrypts it, and sends it to the LSP for calculation. The LSP returns the results to the user after calculation. The scheme ensures the privacy of both parties.

The spatial data is transformed using the Moore curve, and LSP constructs Moore Index (MI) and POI Information (PI) based on Moore curve parameters. As shown in Figure 3, LSP starts from the first grid with a Moore value of 0 and arranges POI in ascending order of the Moore index values. Moore values for empty cells are not stored in the list. M_index, M_value, and P_num are stored in MI. M_index and P_info are stored in PI, where M_index is numbered according to the common difference d to form a uniformly distributed sequence. The M_index value of the ith POI is calculated according

to Equation (4), where d is an integer greater than or equal to 1, and i is the ascending order of the index value M_value in a given Moore curve.

$$\mathrm{M\_index}(i) = d \times i \tag{4}$$



MI

M_index	M_value	num
2	0	1
4	1	2
6	3	1
8	4	2
10	5	1
12	8	1
14	10	1
16	13	2

PI

M_index	POI-info
2	a
4	b,c
6	d
8	f,g
10	e
12	h
14	i
16	j,k

Figure 3. MI and PI list.

The query user generates a corresponding vector δ of what it needs, represented as $\varepsilon_{pk}(\delta)$, and then sends the encrypted δ to the LSP. The vector δ is defined as Formulas (5) and (6), where j represents the jth record in the MI selected by the user.

$$\delta = |\delta_i|_{1 \le i \le r} \tag{5}$$

$$\delta_i = \begin{cases} 1, & i = j \\ 0, & otherwise \end{cases} \tag{6}$$

LSP uses homomorphic encryption [26] to calculate the encrypted content and sends the encrypted result of the calculation to the user, and the user obtains the content after decryption.

5.3. Details of the Query Scheme

Figure 4 represents the specific components of the query scheme. In the first stage, the user queries the data from the local and neighbor cache. RSA authentication [27] is performed before interaction and then evaluated by edge nodes to calculate the average weighted trust value. If the first stage can meet the demand, the query ends. The second stage is that when the cache cannot satisfy the query, the query content is encrypted and sent to the server for calculation. The server cannot know the point of interest information of the user's needs, ensuring the user's privacy.

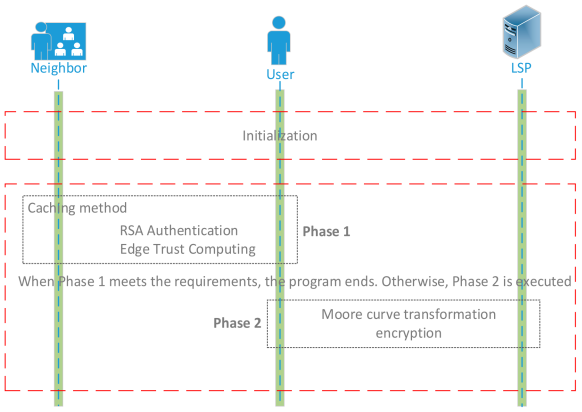


Figure 4. Query scheme.

During initialization, the server constructs a Moore curve based on the Moore curve parameters and generates MI and PI based on the constructed Moore index values. The specific build process is described in Section 5.2. The Moore curve parameter is sent to the user. Next, the user generates the public and private keys, and the public key pk is sent to LSP.

In the first phase, the caching technique is used to improve the problem of server untrusting. After the edge device is validated into the group, each user has its edge device, and the users are divided into different groups according to the geographical distribution. Edge node establishes trust attribute set and attributes weight set. d_i interact with, d_j. Edge node calculates weighted trust value and average weighted trust value. The average weighted trust value can be evaluated as

$$E'_{d_i,d_j} = \sum_{l=1}^{s} \omega_l e_{T_l}^{d_i,d_j} \tag{7}$$

$$E_{d_j} = \frac{\sum_{1\le i\le m, i\ne j} E'_{d_i,d_j}}{m-1} \tag{8}$$

In the second stage, the querying user requests the server for problems the user cache cannot solve. The server sends an encrypted MI to the querying user. After decryption, the user locates his position according to the Moore curve parameters and selects the point of interest by comparing the Moore distance between the two adjacent locations until the number of POI greater than or equal to k is retrieved. The result is represented as a vector, encrypted, and sent to the server. The k nearest neighboring POIs obtained here are rough results.

Due to the different distribution of points of interest in different cases, if we select points of interest by increasing the radius, this can reduce the coverage area. However, the efficiency is not high, and the communication cost is high. For this issue, caching technology reduces the number of requests sent to the LSP. The information is sent to the LSP for processing when the cache does not meet the demand. When the LSP receives the data, it performs the calculation in an encrypted case and returns the result to the user. After the user decrypts it, it calculates the distance between the real locations and sorts it. We note the sorted dataset as $P' = \{p'_1, p'_2, \ldots, p'_s\}(s > k)$. The first radius is $p'_j(j = \lceil k/2 \rceil)$, where p'_j represents the jth POI in the dataset p'_j. When the number of POI in the circle is insufficient, the radius is increased to $p'_j(j = \lceil 3 * k/4 \rceil)$. If it is not satisfied, the radius is increased to $p'_j(j = \lceil k \rceil)$. The query content is encrypted and sent, excluding previously sent content, and the user obtains accurate results by filtering.

5.4. Algorithm Implementation

This section describes the algorithm for the scheme. Algorithm 1 mainly explains that in the process of cache interaction, with the assistance of edge nodes, the malicious behavior problem between neighbor users in the cache is constrained, and the edge devices below the threshold are removed. Algorithms 2 and 3 describe transforming the user's location information by the Moore curve, constructing two lists, and performing query operations.

Algorithm 1 Trust Computing

Input: Edge Devices Set ED, Trust Attribute Set TA, Trust Weight Set ω
Output: $\{E_{d_1}, E_{d_2}, \ldots, E_{d_m}\}$
1. **for** j = 1 to m do
2. **for** i = 1 to m & i ! = j do
3. d_i computes trust evaluation set E_{d_i,d_j} of d_j after an interaction
4. en_k computes weighted trust value E'_{d_i,d_j} with weight set ω
5. **end**
6. en_k computes average weighted trust value E_{d_j} of d_j
7. **if** $E_{d_j} < \rho$
8. en_k kicks d_j out of group
9. **end**
10. **end**
11. **return** $\{E_{d_1}, E_{d_2}, \ldots, E_{d_m}\}$

Algorithm 2 Moore Curve List Construction

Input: Spatial Data Point, $P = (p_1, \ldots, p_s)$, Encryption Key, K
Output: MI, PI
1. **for** all p_i in P do
2. normalize p_i
3. convert the coordinate p_i and add the converted value to Z
4. **end for**
5. Sort the set of filled Moore cells, Z, in ascending order
6. **for** all z_i in Z **do**
7. M_value = z_i
8. P_num = count(poi)
9. P_info = p_i
10. **end for**
11. Encrypt M_value, P_num, P_info using K
12. **return** MI, PI

Algorithm 3 Query processing

1. **if** the Local cache concludes POI
2. precise POI
3. **end if**
4. get the MI published by LSP, determine the number corresponding POI type
5. query in a multi-user cache
6. **if** the neighbor user concludes POI
7. cache information locally
8. **else**
9. LSP sends MI to the user
10. The user locates the current Moore converted position
11. **end else**
12. **end if**
13. **while** not satisfied k **do**
14. query by list and store
15. **end while**
16. finally, form a query record request vector $\delta = \{\delta_1, \delta_2, \ldots, \delta_r\}$, sent to the LBS
17. LBS calculates the result, sends it to the user, and the user decrypts the result

6. Security Analysis

6.1. Privacy between Neighbors

When the user wants to obtain a location query service, he first queries locally, and then initiates a query to the neighbors. Because there may be a situation of mistrust between neighbor users, neighbors will get some sensitive information, such as the user's

location. The scheme performs two-step verification on neighbor users to protect users' data from being leaked. RSA verification is performed first to ensure the legitimacy of the user. Afterward, the edge node calculates the average weighted trust evaluation value and removes users below the threshold, ensuring the security between neighbor users. In sum, our scheme protects the user's location information and query information.

6.2. Privacy for LSP

In this article, when a user makes a query request, the search results are first found in the local cache and the user neighbor cache, and if there are results in the cache that meet the user's needs, the user can directly obtain the result data. In this process, the user does not send any queries to the untrusted LSP, so the LSP does not get any query record information from the user.

When a user sends a query to the LSP, the user transforms the real location, finds the location from the MI, retrieves the target according to the PI, and then sends it to the LSP. The user's geographic coordinates are not transmitted during the interaction, and the user sends $\varepsilon_{pk}(\delta)$ to the LSP. The LSP calculates without decryption and returns the result. During this query, no plaintext data is transmitted, and sensitive information is not leaked to the server or the attacker. Therefore, the user's location privacy is effectively protected.

6.3. The Link Attacks

The user sends $\varepsilon_{pk}(\delta)$ to the LSP, and when the user queries, a different random number is used to encrypt the vector δ, and the user will scramble these random numbers. This way, the user's sensitive information is not leaked to the server or adversary. In sum, our scheme is robust to this attack.

7. Experimental Evaluation and Results

This paper utilizes the published large-scale LBSN Foursquare and Uniform datasets for simulation in Python. The solution uses edge nodes for trust computing and implements location query privacy protection through Moore curve transformation and encryption. In this section, we first test whether the trust calculation suppresses the behavior of malicious nodes and then evaluate the scheme's performance in terms of Moore curve construction time, query accuracy, and communication overhead.

7.1. The Effect of Trust Calculations

This section tests the use of the edge node to perform trust calculations to ensure its security. The experiment set three malicious proportions: 30%, 20%, and 10%.

As shown in Figure 5, the results show that with the increase in simulation time, the proportion of malicious behavior continues to decrease. Therefore, the scheme can suppress the occurrence of malicious behavior through the trust value of malicious users.

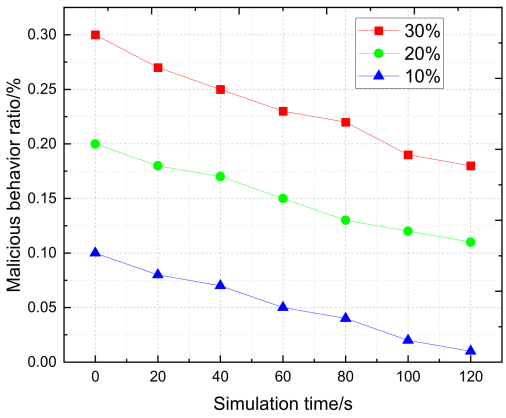


Figure 5. Comparison of the proportion of malicious behavior.

7.2. Moore Curve Construction Time

This article uses the Moore curve to divide areas, thereby reducing search time. Figure 6 shows the calculation time required to construct the different orders of the Moore curve. As you can see from the figure, as the order of the Moore curve increases, the time needed to build the curve also increases, but in the overall scheme, the time is relatively short, so it is efficient to divide the region with the Moore curve.

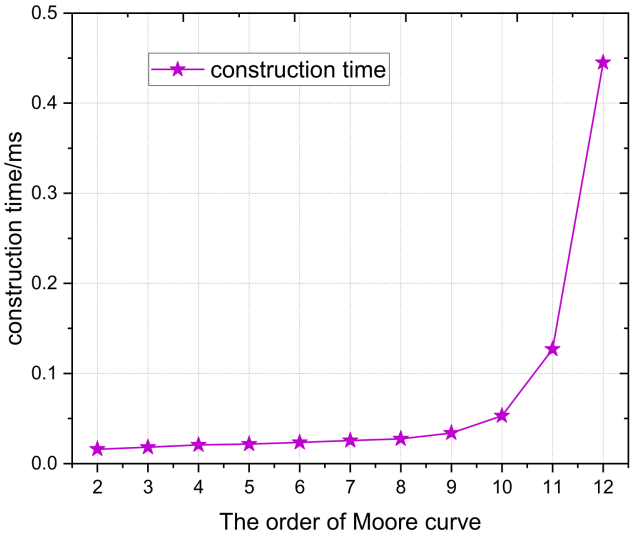


Figure 6. Time cost for constructing Moore curve.

7.3. Accuracy

The query accuracy rate occupies a vital position in the query process, ensuring the quality of the user's service, setting GR′ as the result set returned and *GR* as the actual result set based on geographical coordinates. We define the accuracy as Equation (9).

$$ACC = \frac{|\mathrm{GR}' \cap GR|}{|GR|} \tag{9}$$

To evaluate the effect of the scheme, we compared LPCQP [28], EPCQP [29], and SEQP [25] with our scheme. The scheme [28] constructs the POI table using the Moore curve, removes unnecessary POI information from the requesting user by splitting and aggregating the POI table, and protects the user's privacy through encryption. The scheme [29] builds on [28] by secretly looping and shifting the encrypted POI message to hide the user's location. The scheme [25] uses the Moore curve to transform data and uses a secure optimization method, oblivious transfer, to protect the privacy of query records. Figures 7 and 8 show the comparison of the accuracy of each scheme under the real data set and the uniform data set, respectively. The accuracy rate of our scheme and SEQP is higher than that of other schemes. The reason is that the scheme of this article implements an exact query by constructing a circular region that covers the actual result. LPCQP searches by selecting the Mth sub-table. The number of sub-tables directly affects the accuracy of the query. After dividing the POI table into M sub-tables, some POIs will be lost due to the aggregation of the factor table. EPCQP is an improvement on the original LPCQP solution, which is higher than the initial accuracy rate but lower than the accuracy rate of our solution.

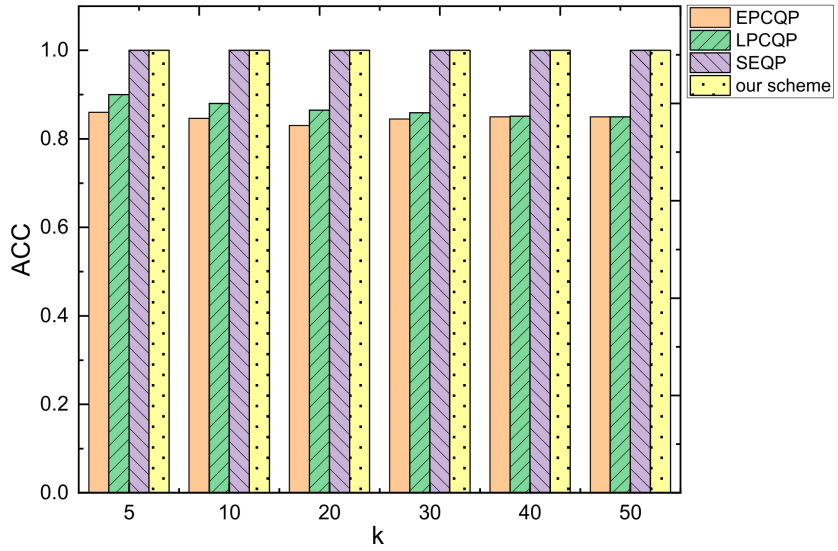


Figure 7. Query accuracy in a real dataset.

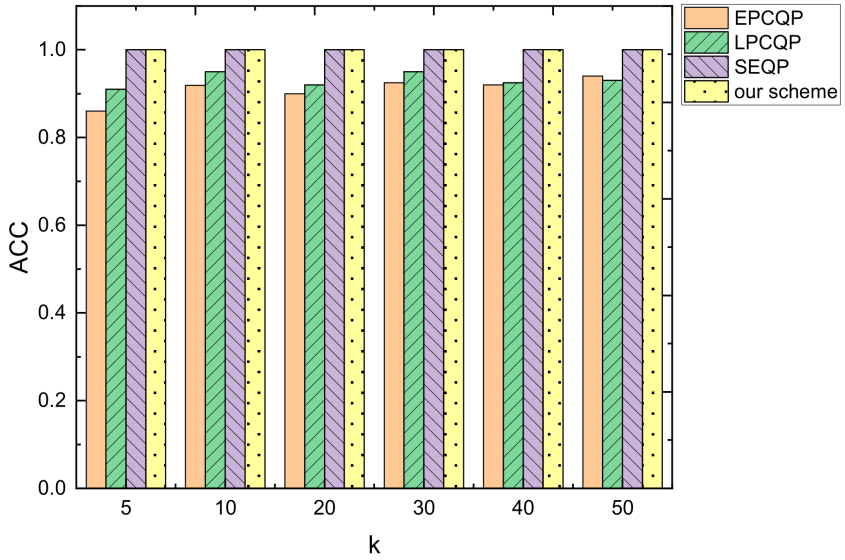


Figure 8. Query accuracy under uniform dataset.

7.4. Communication Overhead

In [17], the user roughly searches for k nearest points based on the number of POI in the grid, drawing circles with the distance of the kth point of interest as the radius. When the distribution of points of interest is dense, if a circle is drawn with the distance of the kth point of interest as the radius, the result is a large amount of redundant data. In [17], the scheme reduces redundant data by progressively increasing the radius to draw circles to acquire points of interest. However, it is not efficient and has a high number of communications. Experiments show that when k >= 30, the circle by k/2 contains POI that does not meet the needs, and we need to draw a larger circle to meet the conditions. The initial circle drawing operation incurs unnecessary communication overhead. Figures 9 and 10 respectively show the number of interest points returned by each scheme under different requirements under the real data set and the uniform data set. In our scheme, the information is first queried using caching technology, and if no results are found, it is sent to the LSP for processing. Redundant data is reduced on the one hand, and efficiency is increased on the other.

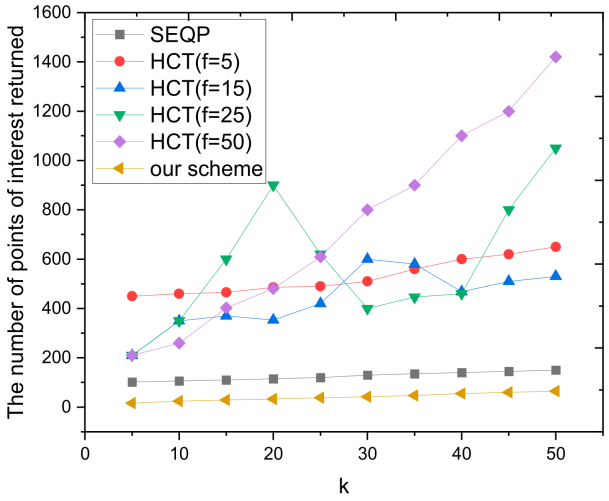


Figure 9. The number of points of interest returned under different requirements in real data.

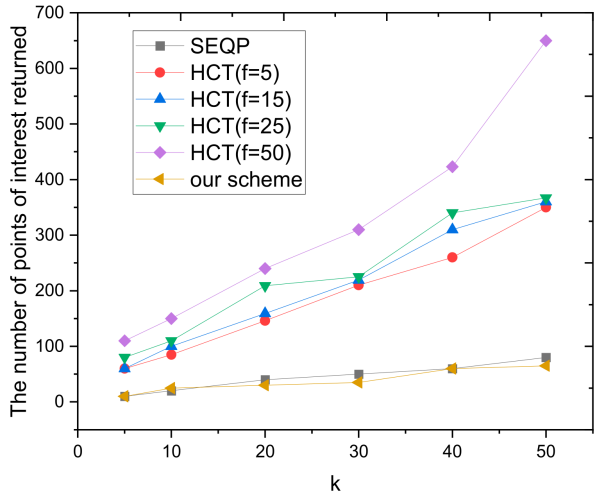


Figure 10. The number of points of interest returned under different requirements in the uniform dataset.

8. Conclusions

In this paper, we propose a location privacy-preserving query scheme based on the Moore curve and multi-user caching. Edge nodes and edge devices perform trust computation to ensure security between members. The user can quickly get results from the neighbor cache for duplicate query requests, which improves efficiency. If no results are found from the neighbor cache, they are obtained from the LSP using the Moore curve encryption policy to ensure that the user gets the service. Simulation results show that our scheme not only suppresses the occurrence of malicious behavior but also improves query efficiency. This research will expand to privacy protection for continuous location queries in future work.

Author Contributions: Conceptualization, Z.L. and Q.L.; methodology, Z.L.; software, J.W. (Jianhang Wei); validation, D.M. and J.W. (Jingyi Wang); formal analysis, Q.L.; investigation, Z.L.; resources, D.M.; data curation, J.W. (Jianhang Wei); writing—original draft preparation, Q.L.; writing—review and editing, Q.L.; visualization, Z.L.; supervision, Z.L.; project administration, Z.L.; funding acquisition, Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Natural Science Foundation of Hebei Province, China under Grant No. F2019201427 and Fund for Integration of Cloud Computing and Big Data, Innovation of Science and Education of China under Grant No. 2017A20004.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yin, C.; Xi, J.; Sun, R.; Wang, J. Location Privacy Protection Based on Differential Privacy Strategy for Big Data in Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3628–3636. [CrossRef]
2. Li, Y.; Qin, Y.; Wang, H. K-Nearest Neighbor Privacy Protection Query for Distributed Storage in Location-based Service. *Wirel. Pers. Commun.* **2021**, *121*, 1509–1532. [CrossRef]
3. Yang, G.; He, Y.; Xiao, K.; Tang, Q.; Xin, Y.; Zhu, H. Privacy-Preserving Query Scheme (PPQS) for Location-Based Services in Outsourced Cloud. *Secur. Commun. Netw.* **2022**, *2022*, 9360899. [CrossRef]
4. Zhang, Q.; Zhang, X.; Wang, M.; Li, X. DPLQ: Location-based service privacy protection scheme based on differential privacy. *IET Inf. Secur.* **2021**, *15*, 442–456. [CrossRef]
5. Parmar, D.; Rao, U.P. Dummy generation-based privacy preservation for location-based services. In Proceedings of the 21st International Conference on Distributed Computing and Networking, Kolkata, India, 4–7 January 2020.
6. Liu, Z.; Wu, L.; Meng, W.; Wang, H.; Wang, W. Accurate Range Query with Privacy Preservation for Outsourced Location-Based Service in IoT. *IEEE Internet Things J.* **2021**, *8*, 14322–14337. [CrossRef]
7. Liu, Y.; Tian, J.; Du, Y.; Li, S. A Random Sensitive Area Based Privacy Preservation Algorithm for Location-Based Service. *Wirel. Pers. Commun.* **2021**, *119*, 1179–1192. [CrossRef]
8. Li, L.; Lu, R.; Huang, C. EPLQ: Efficient privacy-preserving location-based query over outsourced encrypted data. *IEEE Internet Things J.* **2015**, *3*, 206–218. [CrossRef]
9. Niu, B.; Li, Q.; Zhu, X.; Cao, G.; Li, H. Enhancing privacy through caching in location-based services. In Proceedings of the 2015 IEEE Conference on Computer Communication, Hong Kong, China, 24 August 2015.
10. Cui, Y.; Gao, F.; Li, W.; Shi, Y.; Zhang, H.; Wen, Q.; Panaousis, E. Cache-Based Privacy-Preserving Solution for Location and Content Protection in Location-Based Services. *Sensors* **2020**, *20*, 4651. [CrossRef]
11. Zhang, C.; Almpanidis, G.; Hasibi, F.; Fan, G. Gridvoronoi: An efficient spatial index for nearest neighbor query processing. *IEEE Access* **2019**, *7*, 120997–121014. [CrossRef]
12. Zhu, H.; Liu, F.; Li, H. Efficient and Privacy-Preserving Polygons Spatial Query Framework for Location-Based Services. *IEEE Internet Things J.* **2017**, *4*, 536–545. [CrossRef]
13. Zeng, M.; Zhang, K.; Chen, J.; Qian, H. P3gq: A practical privacy-preserving generic location-based services query scheme. *Pervasive Mob. Comput.* **2018**, *51*, 56–72. [CrossRef]
14. Biswas, P.; Dandapat, S.K.; Sairam, A.S. Ripple: An approach to locate k nearest neighbours for location-based. *Inf. Syst.* **2022**, *105*, 101933. [CrossRef]
15. Wang, Y.; Hassan, A.; Duan, X.; Zhang, X. An efficient multiple-user location-based query authentication approach for social networking. *J. Inf. Secur. Appl.* **2019**, *47*, 284–294. [CrossRef]
16. Lian, H.; Qiu, W.; Yan, D.; Huang, Z.; Tang, P. Privacy-Preserving Location-Based Query Over Encrypted Data in Outsourced Environment. In Proceedings of the 2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC), Hangzhou, China, 23–25 June 2019.
17. Kim, H.I.; Hong, S.; Chang, J.W. Hilbert curve-based cryptographic transformation scheme for spatial query processing on outsourced private data. *Data Knowl. Eng.* **2016**, *104*, 32–44. [CrossRef]
18. Miao, Q.; Jing, W.; Song, H. Differential privacy-based location privacy enhancing in edge computing. *Concurr. Comput. Pract. Exp.* **2019**, *31*, e4735. [CrossRef]
19. Zhang, S.; Choo, K.K.R.; Liu, Q.; Wang, G. Enhancing privacy through uniform grid and caching in location-based services. *Future Gener. Comput. Syst.* **2018**, *86*, 881–892. [CrossRef]
20. Zhang, S.; Li, X.; Tan, Z.; Peng, T.; Wang, G. A caching and spatial K-anonymity driven privacy enhancement scheme in continuous location-based services. *Future Gener. Comput. Syst.* **2019**, *94*, 40–50. [CrossRef]
21. Jung, K.; Park, S. Collaborative caching techniques for privacy-preserving location-based services in peer-to-peer environments. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–17 December 2017.
22. Wang, J.; Wang, F.; Li, H. Differential Privacy Location Protection Scheme Based on Hilbert Curve. *Secur. Commun. Netw.* **2021**, *2021*, 5574415. [CrossRef]
23. Zhang, L.; Zou, Y.; Wang, W.; Jin, Z.; Su, Y.; Chen, H. Resource allocation and trust computing for blockchain-enabled edge computing system. *Comput. Secur.* **2021**, *105*, 102249. [CrossRef]
24. Zhang, P.; Wang, Y.; Kumar, N.; Jiang, C.; Shi, G. A Security- and Privacy-Preserving Approach Based on Data Disturbance for Collaborative Edge Computing in Social IoT Systems. *IEEE Trans. Comput. Soc. Syst.* **2022**, *9*, 97–108. [CrossRef]
25. Lian, H.; Qiu, W.; Yan, D.; Guo, J.; Li, Z.; Tang, P. Privacy-preserving spatial query protocol based on the Moore curve for location-based service. *Comput. Secur.* **2020**, *96*, 101845. [CrossRef]
26. Cheon, J.H.; Choe, H.; Lee, D.; Son, Y. Faster Linear Transformations in HElib, Revisited. *IEEE Access* **2019**, *7*, 50595–50604. [CrossRef]

27. Imam, R.; Areeb, Q.M.; Alturki, A.; Anwer, F. Systematic and Critical Review of RSA Based Public Key Cryptographic Schemes: Past and Present Status. *IEEE Access* **2021**, *9*, 155949–155976. [CrossRef]
28. Utsunomiya, Y.; Toyoda, K.; Sasase, I. LPCQP: Lightweight private circular query protocol with divided POI-table and somewhat homomorphic encryption for privacy-preserving k-NN search. *J. Inf. Processing* **2016**, *24*, 109–122. [CrossRef]
29. Lian, H.; Qiu, W.; Yan, D.; Huang, Z.; Guo, J. Efficient Privacy-Preserving Protocol for k-NN Search over Encrypted Data in Location-Based Service. *Complexity* **2017**, *2017*, 1490283. [CrossRef]