SCIENCE CHINA
Information Sciences

• LETTER •

March 2018, Vol. 61 039101:1–039101:3
doi: 10.1007/s11432-017-9090-6

Secure and efficient k-nearest neighbor query for location-based services in outsourced environments

Haiqin WU[1], Liangmin WANG[1*] & Tao JIANG[2]

[1]*Department of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang* 212013*, China;*
[2]*School of Cyber Engineering, Xidian University, Xi'an* 710071*, China*

Received 16 January 2017/Revised 15 March 2017/Accepted 19 April 2017/Published online 30 June 2017

Citation Wu H Q, Wang L M, Jiang T. Secure and efficient k-nearest neighbor query for location-based services in outsourced environments. Sci China Inf Sci, 2018, 61(3): 039101, doi: 10.1007/s11432-017-9090-6

Dear editor,

Recently, due to the proliferation of location-aware mobile devices, k nearest neighbor query (kNN) has become increasing popular in location-based services, especially in outsourced environments where data owners (DO) outsource their private points-of-interests (POIs) to the location service provider (LSP) and allow authorized clients to query k POIs nearest to his location. However, the location privacy of both POIs and mobile clients has been two crucial security concerns as LSP is usually semi-honest in the outsourced paradigm.

Existing solutions for this issue mostly focused on encryption schemes and various kNN query technologies over encrypted data. In [1], asymmetric scalar-product-preserving encryption (ASPE) was proposed to compute the kNN on encrypted data. However, this scheme is vulnerable to the chosen-plaintext attack (CPA) which also exists in [2, 3]. Although Elmehdwi et al. [4] solved this problem with a Paillier crypotosystem-based SkNN protocol, large computation cost is incurred by homomorphic encryption. Recently, some researchers proposed partition-based solutions, such as those based on secure Vornoi diagram (SVD) [5], Delaunay triangulations (TkNN) [6] and Hilbert curve transformation (HCT) [7]. Unfortunately, these solutions only return a relevant encrypted partition with far more than k POIs, which induces expensive computation overhead for further filtering at the client.

To address the security and efficiency issues, in this letter, we propose a secure and Hilbert curve-based kNN query framework (HkNN) in combination with the secure distance comparison protocol (SDCP). Specifically, we introduce two entities for LSP, termed storage and proxy servers, to collaborate and provide stronger security. Moreover, two indices based on HCT are designed to facilitate the query on encrypted data. Extensive theoretical and experimental analysis show that HkNN achieves superior query performances to two state-of-the-art approaches: TkNN [6] and HCT [7], in terms of security, preprocessing cost, kNN query processing time, communication and computation cost at the client.

System overview. To perform secure and efficient kNN query, DO first transforms n 2-dimensional (2-D) POIs into 1-D Hilbert value using Hilbert transformation function $h(\cdot)$ [8]. Next, we group POIs according to their transformed values and design two indices to manage them efficiently, which are then encrypted with AES and mutable order preserving encoding (mOPE)[1)], respectively. When an authorized client issues a kNN query to the LSP with his transformed location, the proxy server can quickly find the group (record) containing the query point, and obtain

* Corresponding author (email: wanglm@ujs.edu.cn)
The authors declare that they have no conflict of interest.

1) mOPE can achieve the ideal security called indistinguishability under ordered chosen-plaintext attack (IND-OCPA).

the encrypted candidate POIs from the storage server. Finally, SDCP [6] is followed securely between the proxy server and the client, which determines which POI is closer to the query client on the encrypted data. Specifically, given two encrypted data points $D_i(x_i, y_i)$, $D_j(x_j, y_j)$ and a query point $Q(x_q, y_q)$, D_i is closer to Q if and only if the following inequality is satisfied:

$$y_q - S_{i,j} \cdot x_q > -1 \cdot S_{i,j} \cdot \frac{x_i + x_j}{2} + \frac{y_i + y_j}{2}, \quad (1)$$

where $S_{i,j}$ denotes the slope of the midperpendicular of the segment that connects D_i and D_j.

Indices design. For efficient retrieval, we divide original POIs into groups based on their Hilbert values and then create data information index (DII) and Hilbert aggregation index (HAI) to store corresponding partition information. Specifically, a unified threshold τ is preset to confine the number of POIs stored in each DII record, such that the LSP cannot deduce the density of POIs by analyzing the same Hilbert value.

DII contains four parts $\langle$ID, data info, R, $S\rangle$, where ID is the record identifier and data info stores τ POIs' locations with same or similar Hilbert value in the ascending order. R and S are the set of right-hand side of inequality (1) and the midperpendicular slope between two arbitrary data points in each record.

HAI contains three parts $\langle$ID, SGI, EGI$\rangle$, where ID is the identifier of HAI, and SGI, EGI denote the start and end grid cell index (i.e., Hilbert values) corresponding to the DII record respectively.

Our HkNN query protocol. Without loss of generality, we assume that both DII and HAI include m records, and each DII record contains τ POIs information except the last one (may be less than τ). For each record $z \in [1, m]$, its corresponding DII is represented by $\langle P_z, R_z, S_z\rangle$, where P_z, R_z and S_z denote the set of POIs, the right-hand sides and bisector slopes of all pairs of POIs in record z, respectively. With the well-designed indices, given a query $Q = (x_q, y_q)$, our HkNN query framework between four entities is shown in Figure 1. The detailed operations at each entity and interactions between them are illustrated as follows.

First, after indices construction, DO sends the following messages to the storage and proxy server, respectively:

$$\mathrm{Msg}_1 = \{E_{k_1}(P_z), \varepsilon_{k_2}(R_z), E_{k_1}(S_z)\},$$
$$\mathrm{Msg}_2 = \{\varepsilon_{k_2}(\mathrm{SGI}_z), \varepsilon_{k_2}(\mathrm{EGI}_z)\}.$$

To preserve the data confidentiality, each POI $p \in P_z$ and all bisector slopes in S_z are encrypted with AES (denoted as $E_{k_1}(\cdot)$), while the right-hand side R_z, start grid index SGI_z and end grid index EGI_z are encoded with mOPE (denoted as $\varepsilon_{k_2}(\cdot)$) to support secure and efficient comparison.

Next, DO sends the relevant key message $\mathrm{Msg}_3 = \{k_1, k_2, \mathrm{HTP}\}$ to the authorized client securely, where HTP denote the Hilbert transformation parameter used in function $h(\cdot)$. Note that no additional information about keys is revealed to adversaries in this process. For the authorized client, after transforming his location, he submits the encoded Hilbert value $\mathrm{Msg}_4 = \varepsilon_{k_2}(h(x_q, y_q))$ to the proxy server.

Subsequently, the proxy server can retrieve the candidate HAI record(s) in accord with the query point quickly by comparing $\varepsilon_{k_2}(h(x_q, y_q))$ with both $\varepsilon_{k_2}(\mathrm{SGI}_z)$ and $\varepsilon_{k_2}(\mathrm{EGI}_z)$. Note that, if τ is less than k (i.e., one record is not sufficient to answer kNN query), then the proxy server needs to expand the HAI record to the neighbor ID until the total number of POIs is greater than k. Here the expansion is bidirectional, which means the neighbor IDs above and below the current record are both considered. Finally, the candidate ID set $\mathbb{C}$ is sent to the storage server ($\mathrm{Msg}_5 = \mathbb{C}$).

Upon receiving Msg_5, the storage server returns the corresponding DII record to the proxy server, denoted by $\mathrm{Msg}_6 = \{E_{k_1}(P_z), \varepsilon_{k_2}(R_z), E_{k_1}(S_z)\}, z \in \mathbb{C}$. Then, SDCP will be performed with client on the candidate POIs to find kNN securely. The interactions between them are as follows.

• The proxy server sends encrypted slopes in set $\mathbb{C}$ to the client, which is $\mathrm{Msg}_7 = E_{k_1}(S_z), z \in \mathbb{C}$;

• The client decrypts the message with k_1 to get S_z and then computes the left-hand set L_z using inequality (1) for each pair of POIs in record z. Then L_z is encoded with k_2 (i.e., $\varepsilon_{k_2}(L_z)$);

• The client submits $\mathrm{Msg}_8 = \varepsilon_{k_2}(L_z), z \in \mathbb{C}$ to the proxy server, followed by the comparison between $\varepsilon_{k_2}(L_z)$ and $\varepsilon_{k_2}(R_z)$ at the proxy server. It is sufficient to find k POIs nearest to the query point in this step;

• The encrypted query result $\mathrm{Msg}_9 = E_{k_1}(\mathbb{R})$ is returned to the client who will decrypt it with k_1 to get the result $\mathbb{R}$.

For better understanding, we illustrate our HkNN with a simple example in Appendix A. In particular, with respect to data updates, it is not necessary to recreate the entire HAI and DII. Instead, $O((\tau-1)\cdot 4)$ update time is incurred at most for small-range motions of POIs. While for adding or deleting POIs, our scheme can flexibly adapt τ so that indices can be updated incrementally.

Security analysis. Our proposed HkNN scheme is robust to resist against the ciphertext-only attack and the estimation attack. For the former, the attacker learns nothing about the POIs dis-

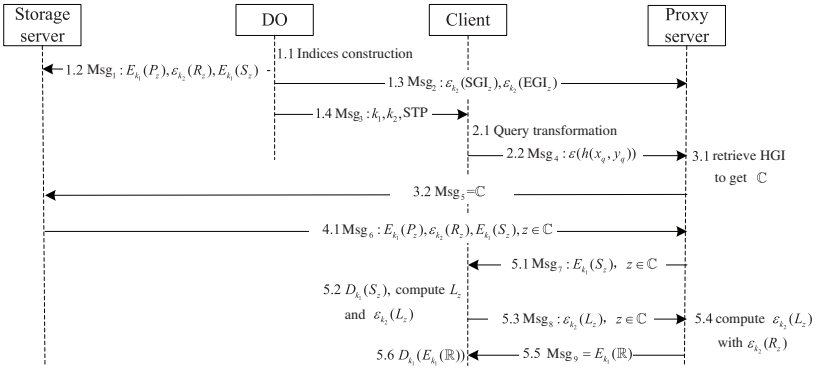

Figure 1 The framework of HkNN query protocol.

tribution except for the existence of densely distributed area. While for the latter, it is computationally intractable to estimate the locations of other original POIs and query points. Therefore, HkNN achieves our security goal that both data confidentiality of POIs and query privacy of authorized clients should be protected. Due to space limitations, we provide the detailed analysis and proofs in Appendix B.

Experiments. We perform our HkNN scheme on three kinds of datasets: a real-world dataset from North East USA[2)] which contains 123593 POIs and two synthetic datasets with uniform and Gaussian distribution ($\mu = 0.5$ and $\sigma = 0.1$), respectively. We mainly compare our method with two existing partition-based solutions TkNN [6] and HCT [7]. The results show that HkNN achieves better query performance than TkNN with lower preprocessing cost, less kNN query time and client's communication cost on three different datasets. In particular, 44.5% preprocessing cost and 45% query time are reduced mostly on the uniform dataset. In addition, our scheme further depresses the client's computation cost dramatically (about 50% time is saved compared with HCT). Detailed results can be found in Appendix C.

Conclusion. In this letter, to realize location privacy preservation and query efficiency simultaneously, we propose a secure and efficient kNN query framework HkNN that supports incremental data updates. Particularly, HkNN provides strong security leveraging two-entity model and two secure encryption methods, the query is facilitated significantly by retrieving our well-designed indices. Additionally, we further reduce the client's cost with SDCP. Theoretical analysis and experiments demonstrated our superiority to others.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61272074, U1405255), and Industrial Science and Technology Foundation of Zhenjiang City (Grant No. GY2013030).

Supporting information Appendixes A–C. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

2) http://www.chorochronos.datastories.org/.

References

1 Wong W K, Cheung D W, Kao B, et al. Secure kNN computation on encrypted databases. In: Proceedings of the ACM SIGMOD International Conference on Management of data, New York, 2009. 139–152
2 Hu H, Xu J, Ren C, et al. Processing private queries over untrusted data cloud through privacy homomorphism. In: Proceedings of IEEE 27th International Conference on Data Engineering, Hannover, 2011. 601–612
3 Padashetty N, Nadagoudar R. Confidential and secure query services in the cloud with rasp. Int J Comput Sci Mobile Comput, 2015, 4: 592–600
4 Elmehdwi Y, Samanthula B K, Jiang W. Secure k-nearest neighbor query over encrypted data in outsourced environments. In: Proceedings of IEEE 30th International Conference on Data Engineering, Chicago, 2014. 664–675
5 Yao B, Li F F, Xiao X K. Secure nearest neighbor revisited. In: Proceedings of IEEE 29th International Conference on Data Engineering (ICDE), Brisbane, 2013. 733–744
6 Choi S, Ghinita G, Lim H S, et al. Secure knn query processing in untrusted cloud environments. IEEE Trans Knowl Data Eng, 2014, 26: 2818–2831
7 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
8 Xu H. An approximate nearest neighbor query algorithm based on Hilbert curve. In: Proceedings of IEEE International Conference on Internet Computing & Information Services (ICICIS). Washington: IEEE Computer Society, 2011. 514–517