JES
Journal of Engineering Sciences

Vol 10, Issue 11, Nov /2019
ISSN NO:0377-9254

PRIVACY PROTECTION IN AD-HOC MOBILE CLOUD ENVIRONMENT

[#1] **SUNKARI SUBASH**, *M.Tech Student,*

[#2]**K.DEEPTHI**, *Assistant Professor,*

Dept of CSE,

SCIENT INSTITUTE OF TECHNOLOGY, IBRAHIMPATNAM , RANGAREDDY TELANGANA.

Abstract—Mobile cloud computing is an emerging cloud computing paradigm that integrates cloud computing and mobile computing to enable many useful mobile applications. However, the large-scale deployment of mobile cloud computing is hindered by the concerns on possible privacy leakage. In this paper, we investigate the privacy issues in the ad hoc mobile cloud computing, and propose a framework that can protect the location privacy when allocating tasks to mobile devices. Our mechanism is based on differential privacy and geo cast, and allows mobile devices to contribute their resources to the ad hoc mobile cloud without leaking their location information. We develop analytical models and task allocation strategies that balance privacy, utility, and system overhead in an ad hoc mobile cloud. We also conduct extensive experiments based on real-world datasets, and the results show that our framework can protect location privacy for mobile devices while providing effective services with low system overhead.

Index Terms—Mobile cloud computing, location privacy, task allocation, reputation.

1 INTRODUCTION

NOWADAYS, mobile devices such as smartphones and tablets have gained tremendous popularity. These devices are often equipped with a variety of sensors such as camera, microphone, GPS, accelerometer, gyroscope, and compass. The data (e.g., position, speed, temperature, and heart rate) generated by these sensors enable many useful mobile applications, including location-based services [1], [2], mobile sensing [3], and mobile crowdsourcing [4], [5]. Although improved largely over the past several years, mobile devices are still resource-constrained mainly due to the limited battery lifetime. On the other hand, cloud computing has widely been regarded as the next-generation computing paradigm which provides "unlimited" cloud resources to end-users in an on-demand fashion. The rich cloud resources in cloud computing can be exploited to increase, enhance, and optimize capabilities of mobile devices, leading to the concept of mobile cloud computing (MCC). According to [6], MCC integrates cloud computing technologies with mobile devices to make the mobile devices more capable in terms of computational power, memory, storage, energy, and context awareness. There are generally two types of mobile clouds in MCC: infrastructure-based and ad hoc [6]. The infrastructure-based mobile cloud consists of stationary computing resources and provides services to the mobile users via the Internet. Alternatively, in the ad hoc mobile cloud, a collection of mobile devices (hereafter referred to as "mobile servers") performs as cloud resources and provides access to local or Internet-based cloud services to other mobile users (hereafter referred to as "mobile clients"). In this paper, we focus on the second case, namely, the ad hoc mobile cloud. The main benefit of utilizing ad hoc mobile cloud resources is their distributed and context-awareness features. As explained in [7]–[10], incentivized by the mobile cloud computing platform (CCP), individual mobile users contribute their mobile devices as mobile servers in the ad hoc mobile cloud, and these mobile servers can be used to perform location-dependent tasks such as epidemic monitoring, traffic monitoring, image/video capturing, and price checking for mobile clients. Despite many promising applications, ad hoc mobile clouds pose several challenges. First, mobile cloud resources in an ad hoc mobile cloud are dynamic and diverse. As a result, some mobile servers may drop the task they are performing and leave the cloud. Some mobile servers may be "spammers" that only want to collect rewards and submit arbitrary answers

without looking at the specific task. Moreover, some mobile servers may not be powerful enough to provide sensing data at the required accuracy. Therefore, how to allocate tasks to ensure the quality of the service provided by these dynamic mobile servers is challenging. Second, as pointed out by [7], security and privacy of mobile devices as service providers is a critical concern in the ad hoc mobile cloud. In order to allocate tasks and provide effective services, mobile servers in an ad hoc mobile cloud need to share their location data with the CCP, which could reveal a lot of personal information such as a user's identity, health status, personal activities, and political views [11]. Hence, it is mandatory to provide privacy guarantee in order to engage more mobile devices in the cloud. Finally, there is an inherent conflict between quality of service (i.e., utility) and privacy in task allocation. If an ad hoc mobile cloud ensures privacy of mobile servers, it is difficult to guarantee the utility of their MCC service. Finding a solution that ensures privacy while guaranteeing utility for task allocation is a major challenge in such systems. Several solutions to privacy issues in mobile applications have been proposed. For example, aggregation is a common approach to hiding individual sensitive information when only statistics of users are required [12]. However, this approach only calculates statistics and thus cannot be used to select mobile servers in an ad hoc mobile cloud. Another approach is used in location-based services, where accurate locations are obfuscated in location-based queries, and the service provider returns results based on the obfuscated query [13], [14]. In our scenario, however, the private information is no longer part of a location-based query, but the result of a location-based query regarding the task. Some papers [15], [16] consider queries on private locations in an outsourced database, but they only protect private data from an intermediate service provider while assuming a trust relationship between the data owner and the querying entity. This is not true in our scenario because mobile servers and the CCP may not share an inherent trust relationship. A recent work by To and Ghinita [17] has been proposed to protect location privacy of crowdsourcing workers in spatial crowdsourcing. However, their solution does not consider worker reputation, and thus cannot provide any quality control over the final result. Therefore, it cannot be easily applied to the mobile cloud computing scenario where service quality is very important. In this paper, we propose a framework that provides solutions to the above challenges, where both location privacy and service quality are considered. In our framework, the CCP only has access to sanitized location data of mobile servers according to differential privacy (DP). Since every mobile server is subscribed to a cellular service provider (CSP) with which it already has a trust relationship, the CSP can integrate mobile server location and reputation information, and provides the data to the CCP in noisy form according to DP. To generate the noisy mobile server data, we adapt the Private Spatial Decomposition (PSD) approach proposed in [17], [18], and construct a new structure called Reputation based PSD (R-PSD). Since fake points need to be created in the DP model, geocast is used to disseminate tasks to mobile servers to prevent the CCP from identifying these points. To summarize, our main contributions are as follows: 1) We identify the specific challenges for task allocation in ad hoc mobile clouds, and propose a framework that can achieve differential privacy for mobile server location data while providing high service quality. 2) We introduce a new structure called R-PSD that partitions the space based on both reputation and location information, and develop an efficient search strategy that finds appropriate R-PSD partitions to ensure high quality of service. 3) We use a geocast mechanism when disseminating tasks to mobile servers to overcome the restrictions imposed by DP, and the overhead during this process is incorporated into the design of the search strategy. 4) We conduct extensive experiments based on real-world datasets to show the effectiveness of the proposed framework. The remainder of this paper is organized as follows. We present background on several techniques we use in Section 2. In Section 3, we describe the system model for the proposed framework. Section 4 and Section 5 describe the detailed solutions, i.e., R-PSD generation and task allocation based on R-PSD. Thereafter, we discuss the experimental results and evaluate the system overhead in Section 7. Section 8 reviews the related work and Section 9 concludes the paper.

II.BACKGROUND

In this section, we introduce background on differential privacy (DP) and Private Spatial Decomposition (PSD). 2.1 Differential Privacy The privacy guarantee provided in our framework is

ϙdifferential privacy [19], [20]. DP provides protection of datasets against adversaries with arbitrary background information. By sanitizing the data, DP prevents an adversary from knowing whether a certain individual record is present or not in the database. Formally speaking, we have the following formal definition. Definition 1. A randomized algorithm F satisfies ϙ-DP if for any two datasets D1 and D2 which differ in only one element, and ∀O ⊆ range(F), the following inequality holds: ln Pr[F(D1) ∈ O] Pr[F(D2) ∈ O] ≤ ϙ. (1) In the definition, the parameter ϙ bounds the ratio of probability distributions of two datasets differing on at most one element. It specifies the amount of privacy protection, and a smaller value of ϙ indicates better protection. We call this parameter the privacy budget. In order to achieve ϙ-DP in a dataset, the raw data is sanitized by adding random noise to the released query set QS. The amount of noise is determined by the sensitivity of QS, which is defined as follows: Definition 2. Given any two datasets D1 and D2 which differ in one element, the sensitivity of the released query set QS is σ(QS) = max D1,D2 kQSD1 − QSD2 k1 . (2) Given the sensitivity, a sufficient condition to achieve ϙ-DP is to add to each query result randomly distributed Laplace noise with mean λ = σ(QS)/ϙ [21]. The results from a database usually involve several stages of analyses Mi . The privacy level of the composition of several stages can be computed by the following results [22]: Theorem 1 (Sequential composition). If Mi are a set of analyses, each providing ϙi-DP, then their sequential composition satisfies (P i ϙi)-DP. Theorem 2 (Parallel composition). If Mi are a set of analyses, each providing ϙi-DP, then their parallel composition satisfies maxi (ϙi)-DP. These theorems enable us to calculate privacy level of an aggregated result based on the privacy level of each individual result. 2.2 Private Spatial Decomposition (PSD) The Private Spatial Decomposition (PSD) approach is first introduced in [18] to construct a spatial dataset that achieves DP. A PSD is a spatial index where each index node is associated with a spatial region, and the value for each node is the noisy count of data points (mobile servers in our scenario) in that region. The data structure for spatial index can be grids, k-d trees, or quadtrees [23]. Choice of data structure and its parameters (fan-out and height) can heavily influence the accuracy of PSD. In spacebased partitioning PSD such as grids and quad trees, the splitting

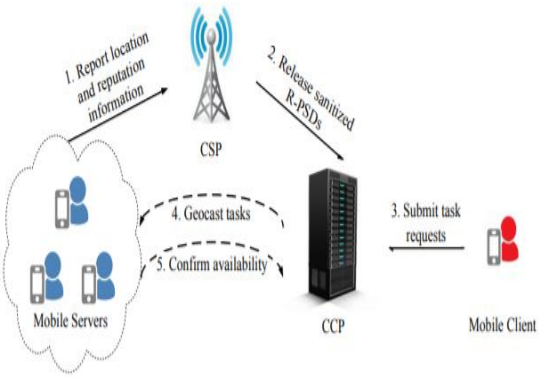

Fig. 1: Privacy-preserving framework for task allocation in MCC.

positions of space is independent of MS locations. Thus privacy budget is only consumed when calculating the noisy count of mobile servers. Typically, index nodes at the same level cover non-overlapping extents, resulting in a low sensitivity of 2 (i.e., the location change of a single MS affects at most 2 cells in a level). The privacy budget ϙ is distributed across levels according to geometric allocation strategy in [18], where leaf nodes are allocated more budget than higher level nodes. Space-based PSD are easy to construct, but they can become unbalanced when mobile servers are not uniformly distributed in space. On the other hand, object-based structures such as k-d trees [18] split space based on the locations of mobile servers. Since location data are used both for calculating splitting positions and computing noisy counts, the privacy budget should be split between the two processes as well. Object-based structures are expected to be more balanced than space-based PSD; however, they are not very robust in the sense that their accuracy may decrease abruptly with a slight change of the PSD parameters or input dataset distributions. The work in [24] proposes an adaptive grid (AG) approach with two-level grids. The first-level grid is uniformly divided, and the granularity of the second-level grid depends on the noisy counts obtained in the first-level. AG is a hybrid approach that inherits the simplicity and robustness of space-based approach, but still utilizes some data-dependent information when choosing the granularity for the second-level grid. In this paper, we adapt their approach to construct our PSD.

III. PROPOSED SYSTEM

We propose a framework that provides solutions to the above challenges, where both location privacy and service quality are considered. In our framework, the CCP only has access to sanitized location data of mobile servers according to differential privacy (DP). Since every mobile server is subscribed to a cellular service provider (CSP) with which it already has a trust relationship, the CSP can integrate mobile server location and reputation information, and provides the data to the CCP in noisy form according to DP. To generate the noisy mobile server data, we adapt the Private Spatial Decomposition (PSD) approach proposed in construct a new structure called Reputation- based PSD (R-PSD). Since fake points need to be created in the DP model, geocast is used to disseminate tasks to mobile servers to prevent the CCP from identifying these points.

IV.ALGORITHM

Greedy Algorithm with PSD Input: Task t, dmax, ARk, k Output: Geocast region Ω

1: Initialize Ω = ∅, ARk = 0;

2: Let U denote the square of length 2 × dmax centered at the task location;

3: Let AR(·) denote the overall acceptance rate ARk of a region;

4: Q ← {the level-2 cell that covers task t};

5: repeat

6: if Q = ∅ then

7: return Ω

8: else

9: c ∗ ← argmaxc∈Q AR (GR ∪ c);

10: Q ← Q \ {c ∗};

11: Ω ← Ω ∪ {c ∗};

12: ARk ← AR(Ω);

13: S ← ({neighbors of c ∗} \ Ω) ∩ U;

14: Q ← Q ∪ S;

15: end if

16: until ARk ≥ ARk

17: return Ω;

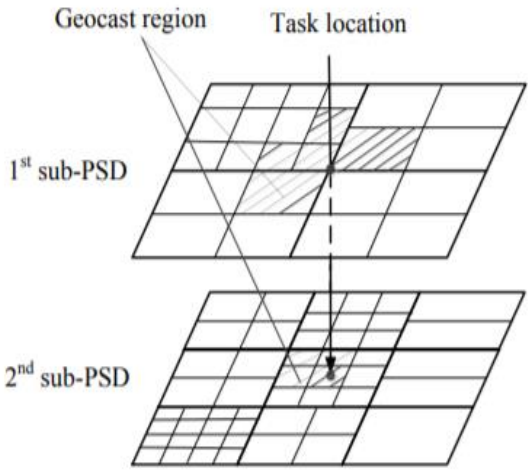

Fig. 4: Illustration of a geocast region with R-PSD.

geocast region in this case is a combination of cells in all subPSDs for the R-PSD. An example of a geocast region is illustrated in Fig. 4. The input to the algorithm is task t, R-PSD with l sub-PSDs, and parameters dmax, ARk, ρ, and k. The variable wi represents the noisy count of servers included in the geocast region GR that belongs to the i-th sub-PSD, i = 1, 2, . . . , l. In addition to the constraint ARk ≥ ARk considered in Algorithm 1, we add a new constraint ρ({wi} l 1) < ρ, which guarantees the service quality of the chosen mobile servers. The geocast region GR is first initialized to an empty set and then expanded iteratively. In each iteration, a new cell that both best improves ARk and ensures ρ({wi} l 1) < ρ is selected and added to GR. The geocast region stops expanding when no new cells within distance of dmax can be added or until ARk exceeds ARk. The algorithm is a greedy approach that always chooses the cell with the highest acceptance rate while guaranteeing service quality at each iteration.

VI. PERFORMANCE EVALUATIONS

In this section, we evaluate the performance of our proposed framework using real-world datasets. 7.1 Experimental Setup We use two real-world datasets: Gowalla [36] and CrowdFlower [37]. The Gowalla dataset is

used to simulate the spatial distribution of mobile servers in our experiments, which contains a total of 6, 442, 890 check-ins on a location-based social networking website from Feb. 2009 to Oct. 2010. We use the check-in history of Gowalla users as the task allocation history of mobile servers. We consider Gowalla users as the mobile servers. We assume all check-ins of a Gowalla user, except the latest one, are tasks that have been completed by him/her, and the latest check-in location is treated as his/her current location. Due to data sparsity, there are no data points in some area of the dataset. Hence we overlay the dataset with a set of uniformly distributed mobile servers. The resulting mobile server distribution is shown in Fig. 5a, where each cross in the figure represents the current location of a mobile server. We extract the reputation scores of mobile servers based on a study carried out in [38], which asks participants to report traffic events in Dublin. They created and assigned approximately 4000 tasks and calculated reputation scores of participants based on the ground truth of the tasks and historical performance in CrowdFlower. We randomly assign the reputation scores to Gowalla users so that we would get a dataset which contains both task performance history and reputation scores of servers. The reputation distribution is given in Fig. 5b. As we presented in Section 5, the service quality of the task can be captured by function ρ(·). In our experiment, we use the results in [39] to estimate the service quality of the task when k mobile servers with potentially different reputation levels perform the same task. In their paper, the error rate of completing a task, denoted as ER, is the metric to quantify the service quality. Suppose we use the majority voting to aggregate the results of mobile servers for a task. It is proved in [39] that the error rate ER, the required number of servers k, and the collective quality Q satisfy the following inequality: kQ2 4 ≤ ln 1 ER . (9) The collective quality Q is calculated in the following way. Define X as a random variable to describe the event that a mobile server submits a correct answer. We have Pr(X = True) = pr and Pr(X = False) = 1 − pr, where pr is the reputation score of a mobile server in our scenario. If the reputation scores of mobile servers are independent and identically distributed, we have Q = E (2pr − 1)2 , (10) where the expectation is take with respect to the distribution of reputation scores. Therefore, given an error rate requirement ER for a task and the number of required mobile servers k, we can deduce a corresponding requirement on the reputation score distribution in the geocast region. In our experiments, we suppose that mobile servers are divided into two groups whose reputation scores fall into [0, 0.5] and (0.5, 1], respectively. The number of servers in each group is w1 and w2, respectively. For a given geocast region, the collective quality depends on the ratio of the number of mobile servers in each group, i.e., w1/w2. If the reputation score in each reputation level follows a uniform distribution, the collective quality Q can be calculated from (10) as

$$\begin{aligned} Q &= \mathbf{E}\left[(2p_r-1)^2\right] \\ &= \int_0^{0.5} \frac{w_1}{w_1+w_2}(2x-1)^2\frac{1}{0.5}\,\mathrm{d}x \\ &+ \int_{0.5}^{1} \frac{w_2}{w_1+w_2}(2x-1)^2\frac{1}{0.5}\,\mathrm{d}x \\ &= \frac{w_1}{(w_1+w_2)\times 3}. \end{aligned}$$

Given a requirement on ER and the number of mobile servers k, we can deduce a lower bound for Q and further calculate a requirement on w1 and w2. When constructing the geocast region, the CCP needs to ensure that the region can satisfy this requirement. We randomly generate 1, 000 tasks which are uniformly distributed in an area, and use our algorithms to calculate GR regions for each task. We also implement a baseline algorithm that is privacy-oblivious. The baseline algorithm has access to exact locations of all servers and always adds the nearest server to a set until the acceptance rate of the set surpasses the acceptance threshold ARk.

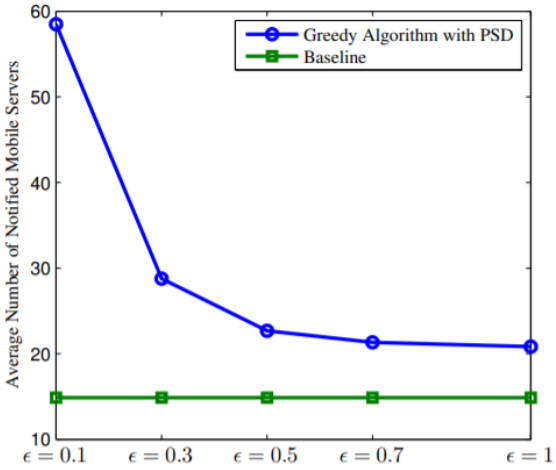

Fig. 1: Effect of privacy budget ϱ when ARk = 0.9.

Method. Fig.1 presents the system overhead for our private algorithm (the greedy algorithm based on PSD) and the baseline algorithm when varying privacy budget ϱ. As ϱ increases, which means mobile servers are less sensitive to their privacy breach, the PSD provides more accurate data for geocast, and the geocast overhead decreases as well. Additionally, we can observe that compared with the baseline, our private algorithm does not significantly increase the system overhead, especially when the privacy budget ϱ is larger than 0.3. This shows the ability for our algorithm to choose nearby mobile servers for a task.

V. CONCLUSION

In this paper, we have investigated the privacy issues in the ad hoc mobile cloud computing, and have proposed a framework that protects the location privacy of mobile servers when allocating mobile cloud computing tasks. Considering the dynamic and diverse nature of mobile servers, we have designed a new data structure R-PSD and developed an efficient search strategy that finds appropriate R-PSD partitions to ensure high service quality. We have conducted extensive experiments based on real-world datasets to demonstrate the effectiveness of our proposed framework.

REFERENCES

1. J. Schiller and A. Voisard, Location-based services. Elsevier, 2004.
2. M. Spreitzer and M. Theimer, "Providing location information in a ubiquitous computing environment," Mobile Computing, pp. 397–423, 1996.
3. T. Choudhury, S. Consolvo, B. Harrison, J. Hightower, A. LaMarca, L. LeGrand, A. Rahimi, A. Rea, G. Bordello, B. Hemingway et al., "The mobile sensing platform: An embedded activity recognition system," Pervasive Computing, IEEE, vol. 7, no. 2, pp. 32–41, 2008.
4. S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "Bikenet: A mobile sensing system for cyclist experience mapping," ACM Transactions on Sensor Networks (TOSN), vol. 6, no. 1, p. 6, 2009.
5. F. Alt, A. S. Shirazi, A. Schmidt, U. Kramer, and Z. Nawaz, "Locationbased crowdsourcing: extending crowdsourcing to the real world," in Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries. ACM, 2010, pp. 13–22.
6. A. R. Khan, M. Othman, S. A. Madani, and S. U. Khan, "A survey of mobile cloud computing application models," Communications Surveys & Tutorials, IEEE, vol. 16, no. 1, pp. 393–413, 2014.
7. S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya, "Cloudbased augmentation for mobile devices: motivation, taxonomies, and open challenges," Communications Surveys & Tutorials, IEEE, vol. 16, no. 1, pp. 337–368, 2014.
8. N. Fernando, S. W. Loke, and W. Rahayu, "Dynamic mobile cloud computing: Ad hoc and opportunistic job sharing," in Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on. IEEE, 2011, pp. 281–286.
9. G. Huerta-Canepa and D. Lee, "A virtual cloud computing provider for mobile devices," in Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond. ACM, 2010, p. 6.
10. E. E. Marinelli, "Hyrax: cloud computing on mobile devices using mapreduce," Master's thesis, Carnegie Mellon University, 2009.
11. I. Krontiris, F. C. Freiling, and T. Dimitriou, "Location privacy in urban sensing networks: research challenges and directions," Wireless Communications, IEEE, vol. 17, no. 5, pp. 30–35, 2010.

12. J. W. Brown, O. Ohrimenko, and R. Tamassia, "Haze: Privacy-preserving real-time traffic statistics," in Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, 2013, pp. 530–533.
13. M. Duckham and L. Kulik, "A formal model of obfuscation and negotiation for location privacy," in Pervasive computing. Springer, 2005, pp. 152–170.
14. R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.- Y. Le Boudec, "Protecting location privacy: optimal strategy against localization attacks," in Proceedings of the 2012 ACM conference on Computer and communications security. ACM, 2012, pp. 617–627.
15. M. L. Yiu, G. Ghinita, C. S. Jensen, and P. Kalnis, "Enabling search services on outsourced private spatial data," The VLDB Journal, vol. 19, no. 3, pp. 363–384, 2010.
16. B. Yao, F. Li, and X. Xiao, "Secure nearest neighbor revisited," in Data Engineering (ICDE), 2013 IEEE 29th International Conference on. IEEE, 2013, pp. 733–744.
17. H. To, G. Ghinita, and C. Shahabi, "A framework for protecting worker location privacy in spatial crowdsourcing," Proceedings of the VLDB Endowment, vol. 7, no. 10, 2014.
18. G. Cormode, C. Procopiuc, D. Srivastava, E. Shen, and T. Yu, "Differentially private spatial decompositions," in Data Engineering (ICDE), 2012 IEEE 28th International Conference on. IEEE, 2012, pp. 20–31.
19. C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in Advances in Cryptology-EUROCRYPT 2006. Springer, 2006, pp. 486–503.
20. C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in Theory of Cryptography. Springer, 2006, pp. 265–284.