JointCache: Collaborative Path Confusion through Lightweight P2P Communication

Kai Dong*, Tao Gu†, Xianping Tao*, Jian Lu*
*State Key Laboratory for Novel Software Technology, Nanjing University
Email: kaidong@smail.nju.edu.cn, txp@nju.edu.cn, lj@nju.edu.cn.
†School of Computer Science and Information Technology, RMIT University
Email: tao.gu@rmit.edu.au

***Abstract*—The emerging location based services have brought forth a privacy risk where users expose their location information. Most of the existing solutions preserve location privacy relying on a trusted anonymizer, which may not exist in real situation. Several peer-to-peer approaches are proposed, but these techniques require large-scale data storage and communication, and they cannot provide up-to-date context information to users. To address this issue, we propose a novel method *JointCache* to preserve location privacy by confusing nearby users' paths. Using this technique, users do not need to buffer context information of a large area and they only need to transfer little information to each other for cooperation. We use simulation to compare *JointCache* with the traditional path confusion, the results show that our technique solves the same place same time problem, and achieves higher anonymity even in a low user density situation.**

***Keywords*-LBS, Location Privacy, K-Anonymity, Path Confusion, JointCache.**

I. Introduction

Advanced mobile devices, such as smart phones, have integrated the Global Positioning System (GPS) receivers, giving rise to a wide range of location based services (LBSs). In LBSs, users are allowed to access location based information and entertainment services, provided that they report their locations to the location based service providers (LSPs). However, the collection and sharing of location data pose significant privacy risks since the service providers may disclose users' location information to others.

Pseudonyms or anonymous techniques are often used to protect users' location privacy. However, if a user's queries are sufficiently frequent, adversaries are able to track the individual. For example, by mounting an identification attack [1], multiple location based requests can be linked to the same subject, thus de-anonymizing the user.

To address this problem, much work have been proposed and most of them require a trusted third party to serve as an anonymizer. However, such a trusted third party may not exist in real world. By breaking this assumption, a recent approach leveraging on ad-hoc communication [2] has been proposed. This approach requires the mobile devices to buffer location based context information (i.e., the service results to this location), and be capable of ad-hoc peer-to-peer (P2P) communication. Every time two users approach to each other, i.e., within the Wi-Fi range (about 100 meters theoretically), they exchange the information they buffered. Instead of querying LBSs, a user may obtain the requested information directly from nearby user if available. In this way, a user's path can be made hidden partially, and his location privacy is thus protected. Since different users meet each other in different places at different times, the buffered context information may be spread to a large number of users, and each user has to keep a large amount of context information.

The ad-hoc P2P approach has a significant drawback that the amount of data storage and communication overhead can be huge, and it is hard to maintain up-to-date information. In this paper, we propose a novel P2P solution named *JointCache* to protect accurate, continuous and real-time location information, and it does not require users to buffer their context information. The intuition of *JointCache* is to make use of Wi-Fi signals to exchange information of users' moving patterns among nearby users, and let users cooperate with each other to confuse their paths. Using this technique, nearby users generate a "mix zone" [3] like area satisfying that they occupy this area at the same time. When users enter this area, they change their pseudonyms and query LBSs for the whole area. In this way, the LSP is not able to distinguish these users.

Designing *JointCache* is a nontrivial task. A naive way to generate a "mix zone" like area for a set of users is to simply draw a circle on the map which covers a large area containing all the users. But this area will be too large for accessing a LBS. A solution is that a user can divide the area into many small ones to obtain the context information of the whole area from the LSP. However, in this case, extra resources are consumed in generating, processing and replying to the queries. So, it is difficult to keep the "mix zone" like areas small.

II. Related Work

Gruteser et al. [1] proposed spatial and temporal cloaking, where multiple users are managed together as an anonymity

978-1-4673-5077-8/13/$31.00 ©2013 IEEE

Authorized licensed use limited to: RMIT University Library. Downloaded on January 10,2021 at 16:33:30 UTC from IEEE Xplore. Restrictions apply.

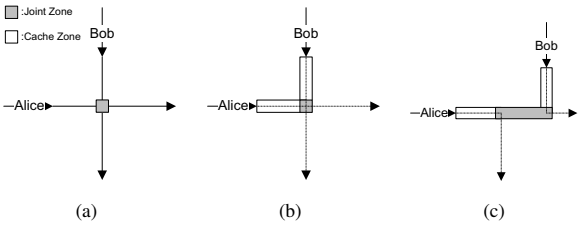


Figure 1: *JointCache* can confuse two users' paths in three conditions: a) they locate in the same place at the same time, b) in the same place at different times, c) in different places at different times.

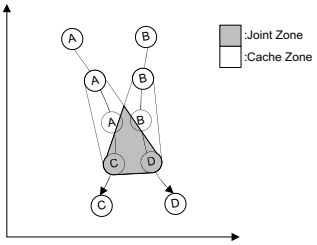


Figure 2: User A and user B each generates a cache zone and access LBS with the whole cache zone. If two cache zones have an overlap area (which we name it the joint zone) and each user leaves his cache zone from this area, they become indistinguishable from each other.

set to achieve k-anonymity. To ensure privacy protection, they sacrifice the granularity of location information.

In Mix Zone [3] and Path Confusion [4], multiple users achieve anonymity if they change their pseudonyms when they occupy the same place at the same time. They have the same place same time problem, i.e., if user density is low, users have few chances to occupy the same place.

Shokri et al. [2] introduce a P2P MobiCrowd which requires large-scale data storage and communication, and cannot provide real-time context information to the users.

Query obfuscation [5] is also used to enhance location privacy. The problem of this approach is that fake queries consume resources, and they may be still distinguished from real user move patterns [6].

III. *JointCache*

The intuition behind our idea is that we want to dynamically generate a "mix zone" among nearby users to confuse their paths. If two nearby users collaborate with each other to query a single area which covers both users' current locations, they become indistinguishable after they leave this area (as shown in Fig. 1).

In the situation that two vehicles located in the same place at the same time, *JointCache* is an extension of Path Confusion, as shown in Fig. 1a. Alice and Bob are located in the joint zone (the grey area) at the same time, after they leave this zone, they change their pseudonyms, (e.g., Carl heading right and David heading bottom), hence they become indistinguishable since no one can tell Alice is Carl or David.

In contrast to Path Confusion, *JointCache* does not require users to appear in the same place with a short delay (a.k.a. the "Same Place Same Time" problem), as shown in Fig. 1b and Fig. 1c. Alice and Bob discover each other through Wi-Fi. They then establish a P2P connection, and exchange their current locations and directions. Based on both users' locations and directions, they each generate a cache zone which covers his future path in a period of time, while generating in cooperation a joint zone which connects two

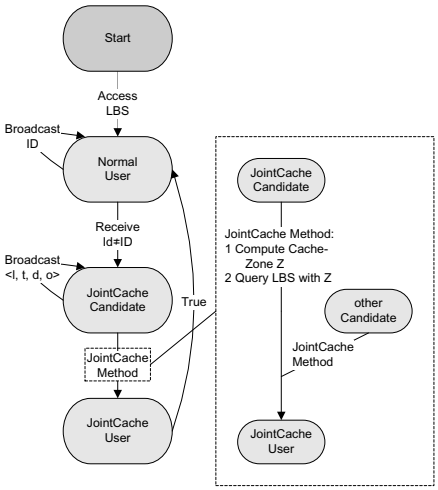


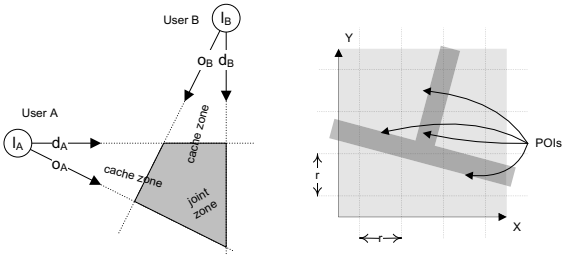
Figure 3: Finite state machine diagrams of the *JointCache* Protocol.

users' paths. Then each user queries LBSs for the whole cache zone and the joint zone, and keeps the service results in his local cache. When moving in the cache zone or the joint zone, the user does not query LBSs but uses the local cached information. Thus, it seems to the LSP that Alice and Bob locate in the joint zone at the same time, even if they never really meet each other (as illustrated in Fig. 2). After both Alice and Bob leave the joint zone, they change their pseudonyms, thus they become indistinguishable from each other.

In the previous example, Alice and Bob discover and connect each other directly, thus no ad-hoc routing protocol is required. Instead, to compute the cache zone and the joint zone, we develop the *JointCache* protocol for nearby users to confuse their traces via Wi-Fi based lightweight broadcasting.

Figure 3 illustrates the *JointCache* Protocol. Each user of a LBS (who accesses the LBS) randomly chooses a pseudo

Authorized licensed use limited to: RMIT University Library. Downloaded on January 10,2021 at 16:33:30 UTC from IEEE Xplore. Restrictions apply.



(a) (b)

Figure 4: Details: a) Generating joint zone and cache zone. b) Deciding POIs of an area.

identity and continuously broadcasts this pseudo-ID along his path. If a user receives a message with a different pseudo-ID, he becomes a *JointCache* candidate since he discovers a nearby user's existence and knows the current location of this user is within its Wi-Fi range, i.e., 100 meters. As a *JointCache* candidate, he then broadcasts a message with his current location, direction and the time stamp, and monitors the Wi-Fi channel for receiving a corresponding message from a nearby user. If both users receive this information from each other, they become the *JointCache* users, and each user can separately compute a cache zone which is used for future access to the LBS.

IV. Details and Analysis

A. JointCache *with two users*

Now we describe in detail how two users, for example, Alice and Bob, collaboratively generate a joint zone and two cache zones.

A method $\mathcal{F}_Z$ is called by both Alice and Bob to generate two cache zones: $\mathcal{F}_Z : \mathcal{L}, \mathcal{D}, \mathcal{O} \rightarrow \mathcal{Z}$. It takes as input both users' current locations $l \in \mathcal{L}$, both users' directions $d \in \mathcal{D}$ and both users' orientations $o \in \mathcal{O}$: Here, the direction and the orientation have different meanings. For example, a vehicle moves from the north to the south then makes a turn to the east. Its direction changes from the south to the east, however, its orientation never changes which may be still the southeast. It outputs two areas $z_i \in \mathcal{Z}$.

As shown in Fig. 4a, if two users Alice and Bob approach to each other, the extensions of d_A, o_A and o_B make an area, which is Alice's cache zone; and the extensions of d_B, o_A and o_B make Bob's cache zone. The two cache zones have an overlap which is the joint zone of Alice and Bob. In this example, the paths of Alice and Bob are confused in their joint zone. In the case that two users move away from each other, the extensions cannot make an area, this means that they have just left their cache zones or they are not close enough to generate the cache zones.

Figure 6: The City of Leimen

B. Queries in Zone

With the knowledge of a local map, $\mathcal{F}_Z$ can be improved. We describe the points of interest (POIs). A continuous area z_i is composed of infinite location points, and only a definite number of these location points can be the inputs of a certain LBS. For any z_i, we use Google Maps API [7] and find a set of location points $\{l_i\}$, satisfying that $a)$ l_i is a point of a road in Google Map, $b)$ the discrete location points $\mathcal{L}_i$ are sufficiently dense to ensure accurate QoS, and $c)$ l_i are as sparse as possible to have a minimum i. These location points are POIs in *JointCache*. Figure 4b illustrates how to decide the POIs for a given area, there are 4 POIs in the grey square area. In the case that a user generates and enters a cache zone, he queries LBSs with only these POIs.

C. JointCache *with Multiple Users*

We now demonstrate that the *JointCache* protocol is secure in the multi-user situation. Suppose two users u_1, u_2 compute a joint zone z_1 in which their paths are confused. Shortly afterwards, u_1 discovers u_3 and computes a joint zone z_2. If z_1 equals z_2, u_1, u_2, u_3 become indistinguishable since they confuse their paths in z_1, thus the probability of distinguishing u_1 is now 1/3; Otherwise, the path of u_1 is confused two times, the first with u_2, and the second with u_3, thus the probability of distinguishing u_1 is still 1/3.

V. Simulation

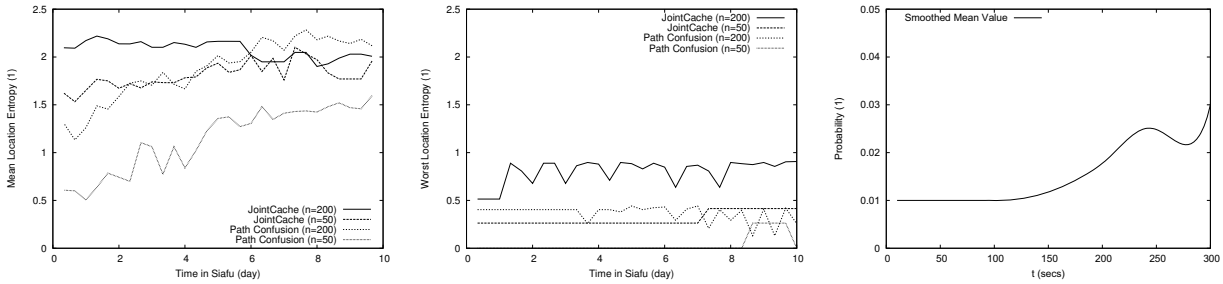
We use an open source context simulator Siafu [1] to evaluate *JointCache* under realistic conditions. Our simulation is based on an existing simulation scenario of Leimen [2] which is a small city in Germany, and the size is about $1.3km \times 1.2km$. In our simulation, people in Leimen lead a simple life. They wake up in the morning, drive to office, then drive back home, or drive for parties after work. The same pattern repeats day and day. We treat them as LBS users.

The simulation runs continuously, and we record the time, the coordinates, the moving direction, and the destination information for all the users (i.e., cars) for every 10 seconds, and mark each record with the user's name. The trace is recorded, simulating a real-time stream of location updates from users. For any time and any place, if there exist two

[1] http://siafusimulator.sourceforge.net

[2] http://siafusimulator.sourceforge.net/?what=simulations&simtitle=leimen

Authorized licensed use limited to: RMIT University Library. Downloaded on January 10,2021 at 16:33:30 UTC from IEEE Xplore. Restrictions apply.



(a) Mean Location Entropy (b) Worst Location Entropy (c) Probability that a user can never generate a cache zone during the drive

Figure 5: Privacy Performance of *JointCache*. *JointCache* achieves higher anonymity, mainly because that we have solved the same place and same time problem raise from Path Confusion.

users and their distance is below 100 meters, they obey the *JointCache* protocol to confuse their paths.

In location sensitive applications, location privacy is typically measured by location entropy [8]. It is defined as the number of bits E, $E = -\sum P(x,y) \cdot \log_2 P(x,y)$. For each user in Leimen, we calculate his location entropy and the results are shown in Fig. 5.

A. Results and Analysis

For each user, we calculate his E_L and E_P. Figure 5 shows the results for a total number of users of 200 and 50, respectively. The mean location entropy is shown in Fig. 5a. In the case that there are 200 users, both *JointCache* and Path Confusion provide certain degree of anonymity, the location entropy is around 2, this implies that the probability of tracking a LBS user is about 1/4. In the case that there are 50 users, *JointCache* still provides enough protection, the location entropy is around 1.5. This is much higher than that in Path Confusion, which is around 1. Figures 5b shows location entropy in the worst case. The location entropy in *JointCache* is higher than that in Path Confusion. Note that when there are only 50 users, Path Confusion cannot provide any protection in the worst case. Figure 5c shows the probability for a user who never has a chance to confuse his path, and it ranges from 0.01 to 0.03. This probability is quite low, implying that most people are protected by *JointCache*. We have checked all the cases that a user does not get protected, and find out that these users never encounter another user is mainly because they drive for a very short period of time (e.g., less than 5 minutes). This means in the case that people drive a very short distance, they may not be protected.

VI. Conclusion

In this paper, we propose a collaborative path confusion mechanism which do not require large-scale context information storage and communication. We evaluate our mechanism and the results show that users' location privacy can be preserved even in low user density areas.

Acknowledgment

This work was supported by Natural Science Foundation of China (NSFC) under grant 61021062, 61073031, National 973 Program of China under grant 2009CB320702, National 863 Program of China under grant 2012AA011205.

References

[1] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st international conference on Mobile systems, applications and services*. ACM New York, NY, USA, 2003, pp. 31–42.

[2] R. Shokri, P. Papadimitratos, G. Theodorakopoulos, and J. Hubaux, "Collaborative location privacy," in *Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on*. IEEE, 2011, pp. 500–509.

[3] A. Beresford and F. Stajano, "Location privacy in pervasive computing," *Pervasive Computing*, pp. 46–55, 2003.

[4] B. Hoh and M. Gruteser, "Protecting location privacy through path confusion," in *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on*. IEEE, 2005, pp. 194–205.

[5] J. Krumm, "Realistic driving trips for location privacy," *Pervasive Computing*, pp. 25–41, 2009.

[6] S. Peddinti, N. Saxena, and A. Birmingham, "On the limitations of query obfuscation techniques for location privacy," in *International conference on Ubiquitous computing*, 2011.

[7] "Google maps api," https://developers.google.com/maps/.

[8] C. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 3–55, 2001.

Authorized licensed use limited to: RMIT University Library. Downloaded on January 10,2021 at 16:33:30 UTC from IEEE Xplore. Restrictions apply.