Protecting Location-privacy with Encryption

Protocol in OMSN

Rui Huang

School of Electrical Engineering

University of Ottawa

Ottawa, ON, Canada, K1N 6N5

Email: rhuan025@uottawa.ca

Bidi Ying

School of Electrical Engineering and Computer Science

University of Ottawa

Ottawa, ON, Canada, K1N 6N5

Email: yingbidi@mail.zjgsu.edu.cn

Amiya Nayak

School of Electrical Engineering and Computer Science

University of Ottawa

Ottawa, ON, Canada, K1N 6N5

Email: nayak@uottawa.ca

*Abstract*—Users face location-privacy risks when accessing Location-Based Services (LBSs) in an Opportunistic Mobile Social Networks (OMSNs). In order to protect the original requester’s identity and location, we propose a location-privacy obfuscation protocol Multi-Hop Location-Privacy Protection (MHLPP) protocol that utilizes social ties between users. To increase chances of completing obfuscation operations, users detect and make contacts with one-hop or multi-hop neighbor friends in social networks. Encrypted obfuscation queries avoid users learning important information except the original requester who generates queries and trusted users, especially the original requester’s identity and location. Simulation results show that our protocol can reach a higher query success ratio comparing to the existing protocol.

Keywords—obfuscation; encryption; location-privacy; opportunistic mobile social networks

# Introduction

Location-privacy is becoming a major concern in OMSNs which is a kind of Delay Tolerant Networks (DTNs) [1] featuring lack of continuous connectivity. More specifically, in OMSNs, it is not necessary for senders to have an end-to-end routing path to their destinations. Nodes make contacts when they encounter each other. LBSs are common applications in OMSNs and they are widely used in "military and government industries, emergency services and the commercial sector" [2], especially after the proliferation of localization technologies, like GPS. LBS users often send their location to LBS providers. Many people access to LBSs with their portable devices, so their locations are also bound to the devices. In this case, LBS users face a continuous risk which their location may be leaked from LBS applications. This makes people unwilling to use LBSs. Thus, protect location privacy has been a critical issue in LBSs.

Early location privacy protection methods such as obfuscation algorithms generate anonymized areas for the original LBS requesters so that the requesters are mixed in a group of other LBS users, like [3]. Users send an anonymized area instead of an exact coordinator to LBS providers when launching a LBS request. After that, social ties are imported into obfuscation algorithms to improve their performances in security. [4] is a typical one among these algorithms. It uses social ties to determine trustable friends who could be chosen as agencies to forward obfuscation queries. [5] and [6] are algorithms which aims to improve its delivery performance. However, compared to [4], the query success ratios of [5] and [6] increase only 5% and 11% respectively, which are not obvious.

These papers assume that attackers can access the LBS servers, which enables them to learn LBS users’ identities and locations. If a user sends a query to a LBS servers with his real identity and location (e.g., a query asking for a path from his current location to a certain place), attackers can locate the user easily. Therefore, hiding the original requester’s real identity and location is the key of this problem.

Inspired by [5] and [7] that uses social network for messages forwarding in mobile ad hoc networks, we propose a distributed location-privacy algorithm MHLPP to guarantee location-privacy and reach a higher query success ratio. The introduction of social networks enables us to hide the original requester’s information behind his friends. When a user wants to send a query, he starts to look for friends based on information in their social networks. He sends his query to the first encountered friend who are responsible to take the query away from the original requester after receiving the query. This friend can also give the query to his friends when they encounter. When the distance between the user carrying this query and the location where the query leaves the original requester is larger than a specific value, the user sends the query to the LBS server. He also replaces the original requester’s information with its own identity and location, which enables the LBS server to receive the query without any information about the original requester. After receiving the query, the LBS server replies to the last friend(the user sending the query to the LBS), and he transmits it to the original requester.

MHLPP contains an obfuscation phase and a free phase. The process before the last friend sends the query out is called an obfuscation phase, then it is a free phase. That is similar as [5], but there are several major differences between them. Instead of finding *k* friends to finish the obfuscation phase, MHLPP takes the query to a place a certain distance away from the original requester. Moreover, [5] only selects friends among its neighbors while MHLPP selects friends among one-hop and multi-hop neighbors. This improvement enables MHLPP to gain a higher query success ratio when there are a fewer friends in the network. In order to provide a secure communication among requesters and friends, especially multi-hop neighbor friends, encryption algorithms are used in MHLPP. Simulation results show that both one-hop and multi-hop connection between friends are acceptable in MHLPP while preventing un-trusted intermediate users from knowing queries' information.

The rest of this paper is organized as follows: section II presents related protocols in privacy-protection. The process of MHLPP is described in III section, then in the IV section, we show our experiment results. Lastly, we conclude the paper.

# Related work

Users face risks when they access to a semi-trusted LBS provider, because anyone who has access to data in LBSs is able to threaten LBS users’ location-privacy. Considering that LBSs rely on location-aware computing, it is unavoidable to leak users' location from LBSs. Therefore, balancing “these two competing aims of location privacy and location awareness” [8] is always a challenge.

Some early solutions, like [9] and [3], are generating a specific area based on *k*-anonymity [10] for each user who needs to send queries. For example, [9] gives a rectangle as an anonymized area, in which all nodes form a group to hide the original requester. But it requires at least *k* connected agents to complete its obfuscation process. [3] uses a central anonymity server as a mix router. As a result, it is necessary for each node to have a continuous connection with the server. That is hard to achieve in a spare DTN. With a similar problem as in [3], [11] employs a matchmaker which is used to match users and advertisements, then users can achieve anonymization of their identities and locations from the matchmaker. However, the matchmaker is a high-risk in the network, because it collects so much private information. In the work in [12], exact locations and requests from clients are replaced by a location anonymization engine before they arrive at LBS providers. Since the anonymization engine learns all exact locations and requests, it becomes a better target for malicious attack.

There are protocols with more servers. The servers are settled in the network and each one of them takes charge of a certain area. [13] uses roadside units (RSUs) as mix servers in a vehicular DTN, and the destination is encrypted during forwarding, so eavesdropping queries cannot help attackers to locate users. But deploying the RSUs is not always feasible. In [14], sensor nodes which are scattered throughout the network provide anonymized locations for users. Since the sensor nodes’ coverage should possess a non-overlapping characteristic, it is difficult to deploy them in real-world. Besides, the mix servers and sensor nodes might be more prominent targets than LBS providers.

However, the system could also be a distributed one, like protocols [4], [5] and [6]. The obfuscation processes are performed by each separated node independently without any help from a third-part device. The utilization of label as [15] makes it easier for nodes to mix themselves into a group, which is a significant difference from these protocols and the previous ones. Algorithms above use groups instead of an area to protect users. [5] imports 2 concepts: the obfuscation phase and the free phase. In the obfuscation phase, queries must be transmitted between friends for *k* times. When there are only a few friends in the network, it is hard for a node to find an available next hop in obfuscation phase. [5] and [6] attempt to improve [4]’s performance in the obfuscation phase, which might be a safety tradeoff, because some ineligible users in [4] are chosen as friends based on the additional standards imported by [5] and [6].

# Appointment Card Protocol

# Performance simulations

We use the map of Helsinki in our simulator to evaluate MHLPP. It is also compared to a previous protocol Hybrid and Social-aware Location-Privacy in Opportunistic mobile social networks (HSLPO) [5]. The parameters of the simulator are shown in TABLE II. All pedestrians and cars are users in MHLPP. These users are moving on the map along streets continuously. There is a LBSP fixed at a random location on the map. For each user, we give him random social values between 0% and 100% to all the other users. Each value has the same probabilities, so that there is an expectation rate between social value and the number of its friends. For example, if given a privacy threshold of 85%, then there might be 15% (100%-85%) users are a certain user’s friends.

As shown in Table II, there are 126 users in the map. For each of them, say user *i*, we give him 126 random *SV* values, which denotes the relationship strength between him and other users. So that there are 1262 *SV* in our simulator. The *SV*s are between 0 and 100. As a result, if the *Tmin* is equal to 85, the average number of friends of each user should be 18.9 ()

TABLE II SIMULATION PARAMETERS

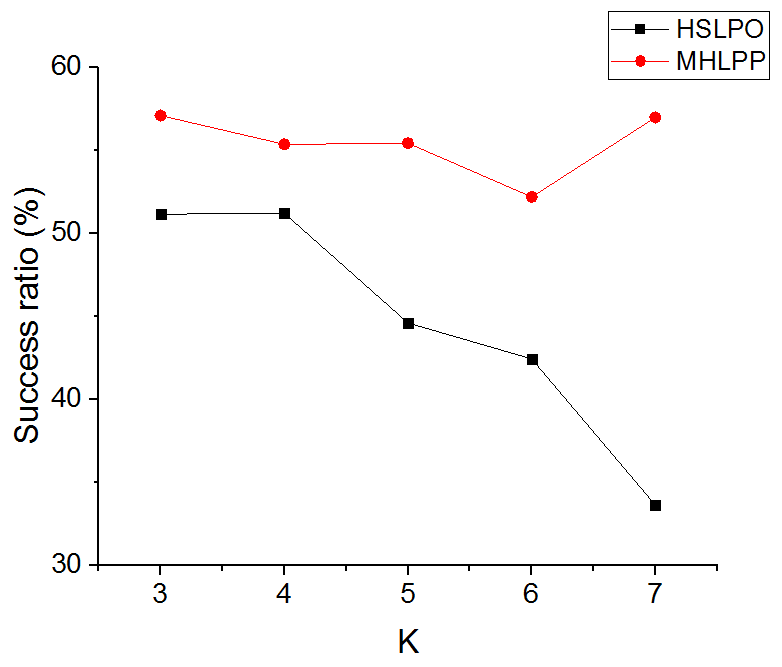
|  |  |
| --- | --- |
| Parameter | Value |
| Simulation Time | 10 minutes |
| Map Size (W x H) | 4500 m x 3400 m |
| Total number of users | 126 |
| Pedestrians/ Cars | 84/42 |
| Communication Area Radius | 10m – 90 m |
| Pedestrians Speed | 1.8-5.4 Km/h |
| Cars Speed | 10-50 Km/h |

Users are settled at random locations at the beginning of each experiment. We choose another random point for each user, so that he can move back and forth along streets between that point and the point his starts at. Their speed depends on its type (pedestrians or cars) and set randomly. All Queries a 10-minute time out. The queries which is expired before it reaches the LBSP (the destination) are failed in our success ratio statistics.

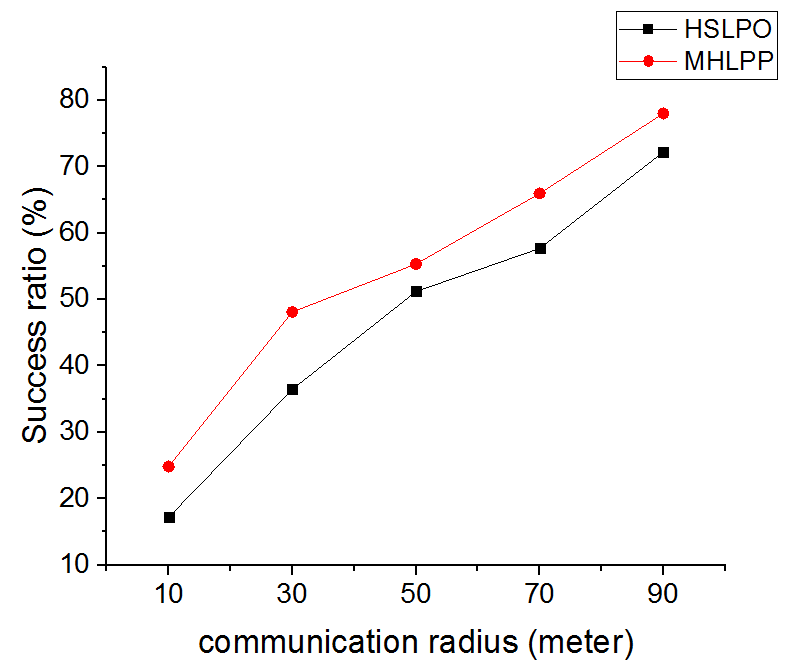
Fig. 3-5 compare performances between HSLPO [5] and MHLPP from different values of *k*, communication radius and privacy threshold (*Tmin* in MHLPP). The *k* is the privacy-level requirement in HSLPO. HSLPO and MHLPP have different conditions in which a query can switch to the free phase. To make them comparable, we import a parameter obfuscation distance. If a query leaves *N*0 at location *La* and switches to the free phase on *Nf* at location *Lb*, then obfuscation distance is the straight-line distance between *La* and *Lb* . We test the obfuscation distances of HSLPO with different parameters, then we set the inner radius of MHLPP to those values. The query success ratio is the ratio of delivered queries’ number to the total. Intermediate-hop number is the number of intermediate users between *N*0 and the destination (LBSP). We statistic the number of users surrounding the last friend in a specific range, which is *k* times the communication radius. We calculate the entropy from reciprocals of surrounding users’ number.

## Query success ratio

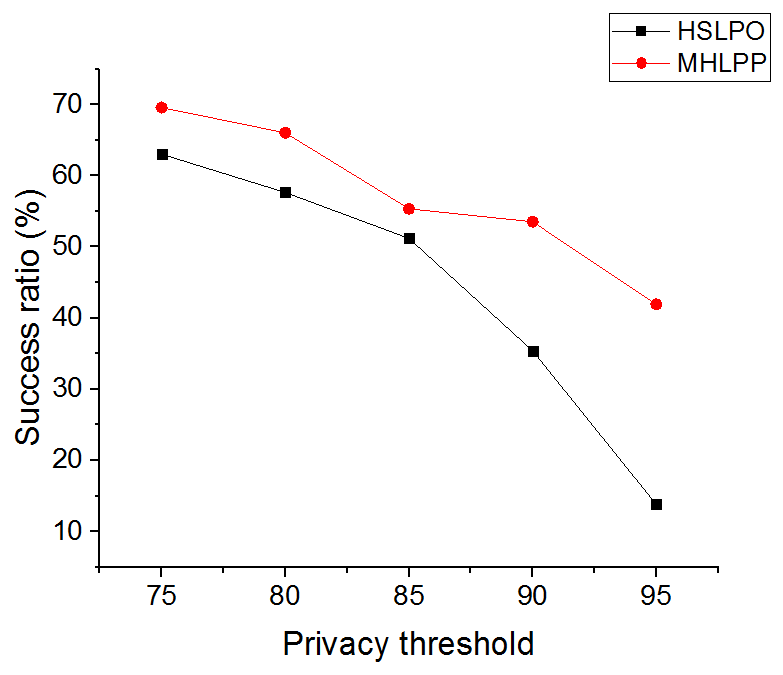
The query success ratio is the percentage of delivered queries among a number of attempts. Based on the time out value in table II, a query is delivered successfully, if it arrives the LBSP (the destination) before it expires; otherwise it fails. We use the query success ratio to evaluate the delivery performance of MHLPP.



(a)



(b)



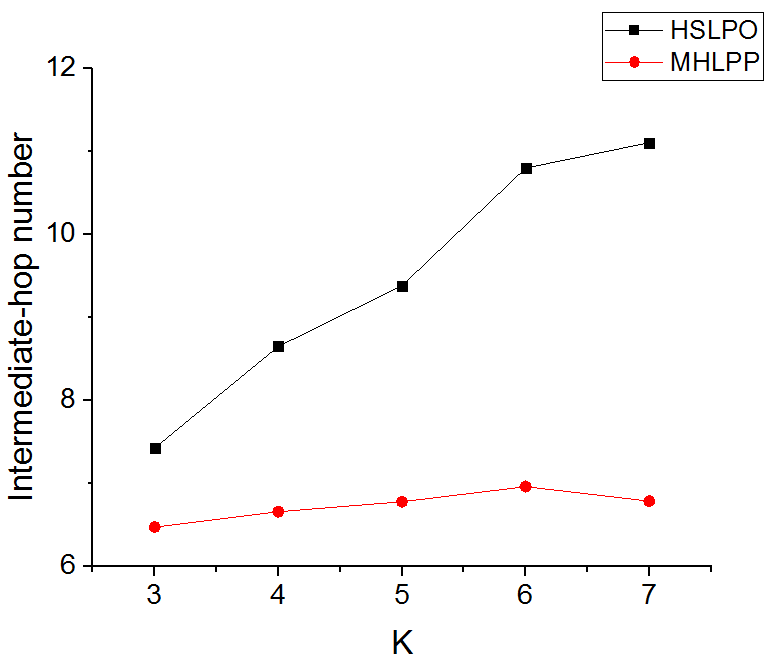
(c)

Fig. 3. The query success ratio comparison between HSLPO and MHLPSP. (a) Query success ratio with various K. (b) Query success ratio with various communication radiuses. (c) Query success ratio with various privacy thresholds.

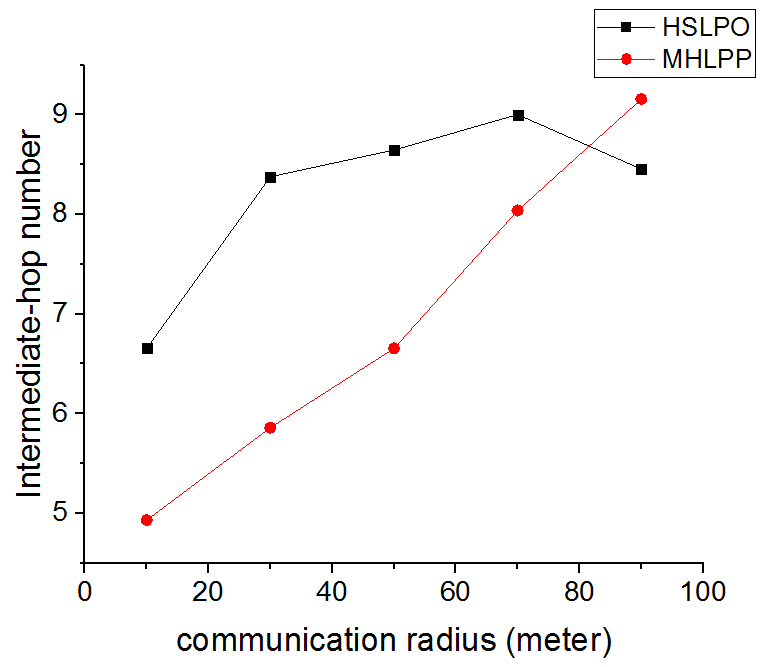
As shown in figure 3a, the success ratio in MHLPP is always higher than that in HSLPO. As the value of *k* increases, HSLPO success ratio drops sharply while MHLPP is more stable. This is because the larger *k* is, the harder it is for HSLPO to find enough friends in a limited time. The lack of friends has less impact on MHLPP. We observe that the success ratio of MHLPP rises when *k* = 7. That is because it depends on the inner radius which is equal to the obfuscation distance of HSLPO. The obfuscation distance decreases when *k* = 7, because most of queries which complete their obfuscation have a short obfuscation distance. In figure 3b, both HSLPO and MSLPSP values increase and have the same trend when given a larger communication radius . The reason is that the communication radius effects the free phase more than the obfuscation phase for both. As shown in figure 3c, higher privacy threshold leads to lower success ratio in two algorithms. Its impact on HSLPO is more intense than that on MHLPP, which is the most important characteristic of MHLPP. MHLPP has a better performance than HSLPO especially when there are less friends in the network. MHLPP can transmit messages with the help from strangers in its obfuscation phase while HSLPO cannot.

## Intermediate-hop number

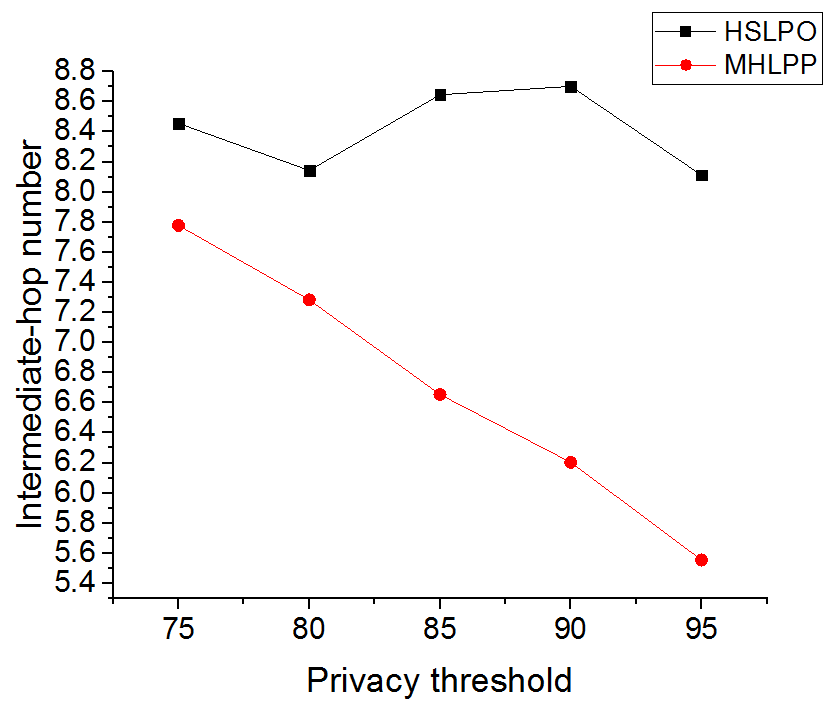
We count the number of intermediate-hop of queries which are delivered successfully. Every user who takes part in the delivery process is counted in the intermediate-hop number. We introduce this criterion to measure the routing path length of the algorithm. MHLPP is more sensitive to the probability that a user encounters a friend than HSLPO does. The reason is that MHLPP aims to reach a certain distance other than a certain hop count. In another word, MHLPP continues sending queries to other friends until queries enter their obfuscation area. In this process, MHLPP takes every chance to send queries. If it is hard for MHLPP to find friends, it can also take the queries to obfuscation areas with less friends. Therefore, the probability that users encounter friends has less effect on the performance of MHLPP. The experiment result is shown in figure 4.



(a)



(b)



(c)

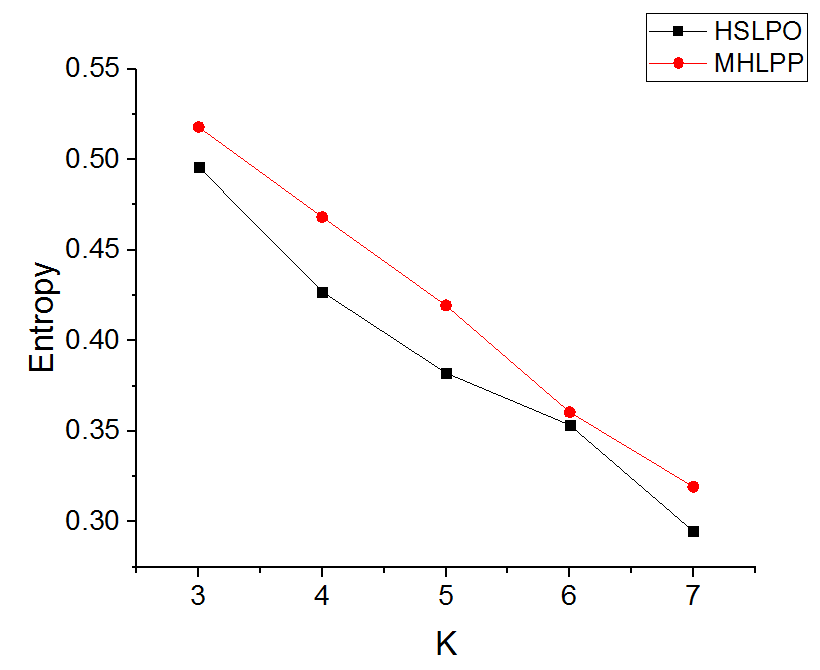
Fig. 4. The intermediate-hop number comparison between HSLPO and MHLPSP. (a) number of hops with various K. (b) number of hops with various communication radiuses. (c) number of hops with various privacy thresholds

In figure 4a, the Intermediate-hop number of HSLPO is affected by parameter *k* obviously. Especially, the first *k* hops forwarders must be friends, which makes it hard for HSLPO to have queries to be transmitted successfully. Both protocols have similar intermediate-hop numbers in their free phases, but HSLPO has exact *k* hops in its obfuscation phase, while MHLPP can have less hops. In figure 4b, the value of MHLPP grows with communication radius obviously, while HSLPO does not change a lot. Because only successful delivery queries can be counted in the statistics. Given a large communication radius, MHLPP has a much higher success ratio to delivery queries as it can connect more friends. In figure 4c, the value of MHLPP drops when given a higher privacy threshold. The reason is the same as figure 4b. For HSLPO, no matter how hard it is to find a friend, it attempts to find *k* friends. However, if it is too hard to find a friend, MHLPP’s friend can carry the query moving and complete the obfuscation process. When given a higher threshold (less friends), MHLPP chooses to carry queries other than to find friends. That results in a drop in intermediate-hop number.

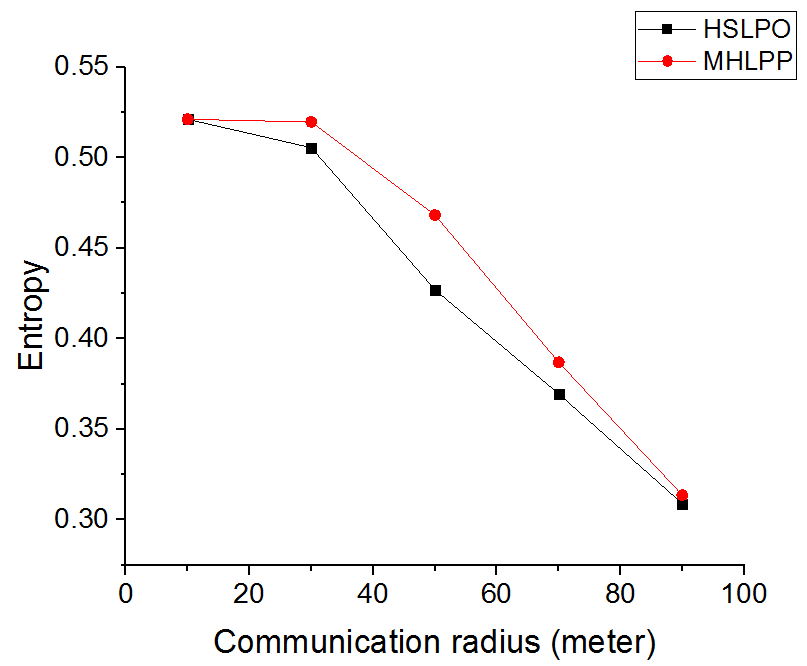
## Security

Since the principles with that two protocols protect original requesters are different, we evaluate the probability that attackers locate the original requester if the distance between him and the last friend is shorter than a value *r*, which is equal to *k* times the communication radius. Since the last friend reveal himself to the LBSP, attackers might locate him accurately. We assume that attackers know the privacy parameter *k*. In the worst case, the distance between the original requester and the last friend is shorter than *r* when attackers start to locate the original requester. That gives attackers a chance to locate the original requester. We count all users who is inside a *r* radius of the last friend and the original requester is one of them. For example, if there are *m* users in the area, the probability should be . We change it to entropy, which is shown in figure 5.

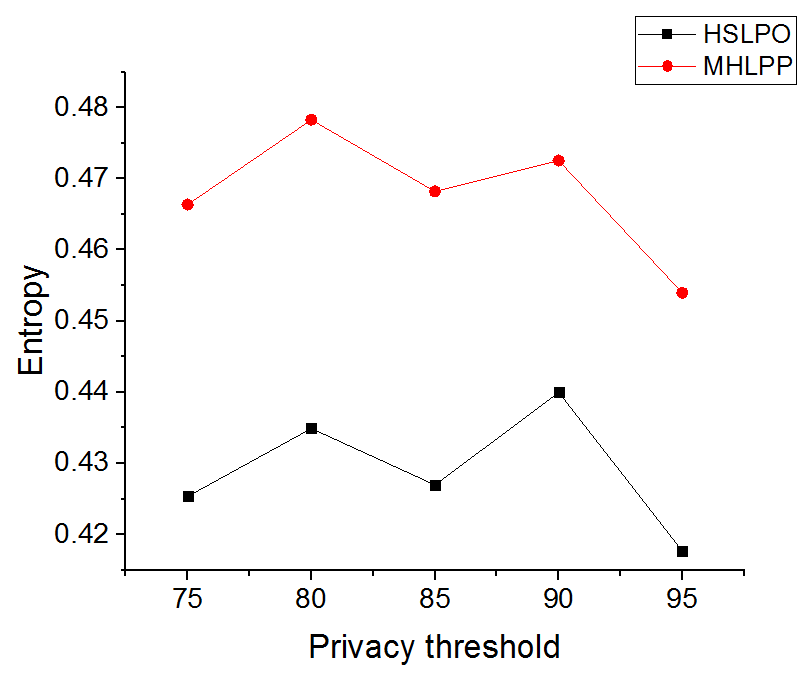
From figure 5a, we learn that MHLPP has about 0.04 more entropy than that in HSLPO. When the original requester is in the circle centered at the last friend, HSLPO is a little more secure than MHLPP. That is because HSLPO always switches to the free phase when the last friend encounters the previous friend, so the previous friend must in the circle. MHLPP does not have this limit. From figure 5b, the trend looks like figure 5a. The curve of MHLPP is also a little higher than that of HSLPO. While two curves almost meet when the communication radius are small enough or large enough. Given a small radius, the entropy of both protocols are small. When the radius is large enough, we can ignore the effect of the previous friend mentioned in figure 5a as there are so many users in the circle. From figure 5c, since the circle neither expands or shrinks, two curves keep the similar changes. The values of HSLPO are always lower than correlated values of MHLPP. However, as we observed in the experiment, the last friend always hundreds of meters away from the requester.



(a)



(b)



(c)

Fig. 5. Locating probability entropy comparison between HSLPO and MHLPSP. (a)Locating probability entropy with various K. (b)Locating probability entropy with various communication radiuses. (c)Locating probability entropy with various privacy thresholds.

# Conclusion

In this paper, we proposed a distributed location-privacy preserving protocol named MSLPP based on social-relationship and encryption. Simulation results show that it has a better performance on delivery success ratio and provides an acceptable obfuscation.

##### References

1. K. Fall, "A Delay-Tolerant Network Architecture for Challenged," in Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications , pp 27-34, Karlsruhe, Germany, August 25 - 29, 2003
2. J. Schiller, A. Voisard, "Location-Based Services," Morgan Kaufmann , 2004
3. M. Gruteser, 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, pp 31-42, 2003
4. S. Zakhary, M. Radenkovic, "Utilizing social links for location privacy in opportunistic delay-tolerant networks," in ICC, 29 November 2012
5. S. Zakhary, M. Radenkovic, A. Benslimane, "The quest for location-privacy in opportunistic mobile social networks," in IWCMC, 2013 9th International
6. S. Zakhary, M. Radenkovic, A. Benslimane, "Efficient Location Privacy-Aware Forwarding in Opportunistic Mobile Networks," in IEEE Transactions on Vehicular Technology, Volume: 63, Issue: 2, Feb. 2014
7. E. Daly, M. Haahr, “Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs,” in Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing, 2007
8. M. Duckham, “Moving forward: Location privacy and location awareness,” in Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS, pp 1-3, 2010
9. T. Hashem, L. Kulik, “safeguarding location privacy in wireless ad-hoc networks,” in UbiComp, pp 372-390, 2007
10. L. Sweeney, "k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY," in Int. J. Unc. Fuzz. Knowl. Based Syst., 10, 557 (2002).
11. M. Mano, Y. Ishikawa, “Anonymizing User Location and Profile Information for Privacy-aware Mobile Services,” in Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks, pp 68-75, 2010
12. T. Wang, L. Liu, “Privacy-aware mobile services over road networks,” in Proceedings of the VLDB Endowment, pp1042-1053, Vol 2, August 2009
13. R. Lu, X. Lin, X. Shen, “SPRING: A Social-based Privacy-preserving Packet Forwarding Protocol for Vehicular Delay Tolerant Networks,” in INFOCOM, 2010 Proceedings IEEE
14. J. Zhang, C. Chow, “REAL: A Reciprocal Protocol for Location Privacy in Wireless Sensor Networks,” in IEEE Transactions on Dependable and Secure Computing, Vol 12, 2015
15. P. Hui, J. Crowcroft, “How Small Labels create Big Improvements,” in PERCOMW ’07. IEEE, pp 65–70, 2007
16. Diffie, W., Hellman, M., "New directions in cryptography," in IEEE Transactions on Information Theory, 1976
17. P. Jacquet, P. Muhlethaler, T. Clausen, “Optimized link state routing protocol for ad hoc networks,” in Multi Topic Conference, 2001
18. R. Xiang, J. Neville, M. Rogati, “Modeling Relationship Strength in Online Social Networks,” in Proceedings of the 19th international conference on World wide web, pp 981-990, April 26 - 30, 2010
19. T. Spyropoulos, K. Psounis, C. S. Raghavendra, "Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks," in Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, pp 252-259, 2005