Arunkumar, S., Srivatsa, M. & Rajarajan, M. (2014). Location Security - Where to Enforce?. Paper presented at the IEEE MILCOM, 06-10-2014 - 08-10-2014, Baltimore, USA.

Location Security - Where to Enforce ?

Saritha Arunkumar[‡], Mudhakar Srivatsa[†], Muttukrishnan Rajarajan[*]
IBM Hursley Labs, UK[‡] IBM Research, USA[†] City University, London[*]
saritha.arun@uk.ibm.com, msrivats@us.ibm.com, R.Muttukrishnan@city.ac.uk

***Abstract*—Enforcing security in location based services is very crucial in the current mobile world. Past literature has examined both location and identity obfuscation techniques in order to optimally tradeoff security/privacy with utility – this primarily addresses the 'how to enforce location security problem'; however, it does not address the 'where to enforce location security problem'. This paper examines the 'where' problem and in particular, examines tradeoffs between enforcing location security at a device vs. enforcing location security at an edge location server. This paper also sketches an implementation of location security solutions at both the device and the edge location server and presents detailed experiments using real mobility and user profile data sets collected from various data sources (taxicabs, Smartphones). Our results show that while device-based solutions do not require trust in the edge location server, they either suffer from high false positive rate (about 25% probability of not meeting the desired security requirement) or low utility (about 600 meters higher error in obfuscated location data).**

I. Introduction

Mobile devices have become an important tool in modern communication. Mobile and other handheld devices such as ipads and tablets have over taken laptops and desktops and hence there has been an increasing research interest in this area in recent years. This includes improving the quality of communication and the overall end-to-end data security in day-to-day transactions. Mobile devices can request for location based services from service providers. This leads to various attacks made in order to tamper the end user security. Hence end user security is another major issue along with the data security flowing from one end to the other. On requesting a service from a mobile service provider, the location and identity of the individual making the request is unknowingly accessed by the Service Provider leading to taking advantages and misusing it. Hence preserving the security of the individual including location and identity is a very difficult open research problem today.

One of the approaches is to ensure the location security and the other is through identity security. Both these techniques are equally important as either of them can compromise the security in the mobile space. There is a vast body of work in the area of location security [8] [3] [6] [2] [4] [12] [1] and [7]. But there is no thorough evaluation on where to enforce location security and what are the tradeoffs involved. There are two approaches for information flow control in mobile environments.

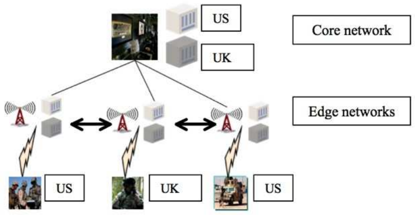

Fig. 1: A tactical network scenario enabling efficient computations over dynamic networks

It is very important to understand the placement of the location security solution and where to enforce security is key to this paper. Introducing mobile micro-cloud in this paper will help in understanding the placement of the solution. Mobile micro-cloud [11] envisions that applications (or computing tasks) will be deployed in a mobile micro-cloud, a logical network composed of two components, the core (e.g., the command and control center) with access to large quantities of static (and possibly stale) information and the edge (e.g., the forward operating base) with access to smaller quantities of more real-time and dynamic data. The edge and core are separated by dynamic and performance constrained networks with a many-to-one relationship between the core and the edge. It is also possible for edge nodes to communicate with each other. Further, the (edge and core) nodes can belong to different coalition partners, raising the question of security and operational policies for handling of data and computation. Figure 1 illustrates a typical architecture of the mobile micro-cloud in the army coalition context. The benefits of embedding storage and computation into such a micro-cloud tactical network are two fold: (i) Effective provisioning for diverse information requirements the micro- cloud supports users with different latency requirements and access rights and (ii) Effective information exchange in a constrained environment Complete shuffling of information is impractical in a tactical network and the micro-cloud reduces congestion by providing computation at the edge.

This paper focuses on Device Vs Edge based implementation and the tradeoffs in them. This paper quantifies the tradeoffs and proves that edge based solution is the better solution for enforcing security. Our results show that while device-based solutions do not require trust in the edge location server, they either suffer from high false positive rate (about 25% probability of not meeting the desired security requirement) or low utility (about 600 meters higher error in obfuscated location data).

Solution at the Core

The core is the centralised network and hence has a lot of bandwidth and can maintain huge repository of information. It also has a lot of computational power allowing it to process complex solutions. It is important to note that it takes longer time for transactions to work between the device and the core. This is a major drawback to the location based solution as decisions needs to be made rapidly else will lead to delays in the decisions to be taken and hence weakens the system. Solution at the core will retain same false positive and false negative and will have a very high latency.

Solution on the Device

The delays caused due the solution being placed at the core of the network gave rise to the new wave of solutions that were placed on the device. It is important to notice that the device doesn't have a lot of flexibility, bandwidth, computation power. Besides any of these, the device does not have visibility of the other devices in the network. Hence any kind of computations performed by the device will not be leading to accurate results. It could very well lead to misleading answers to the user's request.

This leads us to the new methodology that we introduce in this paper called the solution at the edge of the network.

Solution at the Edge

The edge of the network is closer to the device and is an intermediate channel between the device and the core of the network. The edge has visibility of all the other users in the network and the edge can perform computations faster and provide with results spontaneously to the device. The advantage of having the solution at the edge is that edge will have information about other people and hence solution will have lower false positive and lower false negative. The only catch with this solution is that trust with the edge is needed. The edge will have the raw obfuscated data or slightly obfuscated location data . Latency with this solution is higher than device based solution and is lower that the solution at the core. This helps the device user make decisions on the location based service requests that one has. Hence this solution is the best solution compared to the 3 solutions explained in this paper.

In this paper we compare optimal choice on the device (based on historical data) with optimal choice on the edge and examines the tradeoffs between enforcing location security at a device vs. enforcing location security at an edge location server. To the best of our knowledge this the first attempt to quantify the effectiveness gap between the optimal solution at the device versus that at the edge/core.

The organization of the paper is as follows: Section I provides a brief introduction to location and identity security solution placement. Section II describes location security and k-anonymity. Section III shows the evaluation of the location information flow control with datasets. Section IV describes an implementation of the device and edge based solution and shows the comparison between the two solutions. Finally, Section V concludes the paper.

II. Location Security

Location Security mainly deals with the location of the requester. In mobile environments, users requests for information related to location ever so often during requesting location based services. For example, when the user requests nearby restaurant information from the location based server, the location based server needs to know the location of the user and hence the location information is normally requested. However, in most of the cases, the user doesnt want to disclose the location information to arbitrary location based service providers. This can be achieved by a number of different mechanisms. One of the well known methods is k-anonymity. In this method, users location information is updated with pseudo-ids and then the generalized location information is sent to the location based service provider. Due to some groups being created that fail to provide overall anonymity, another mechanism called s-proximity has been implemented [3] . This mechanism creates a larger number of anonymous user profiles to ensure that the location based service provider cannot identify the location of the requestor. Another location Security mechanism that is described is Casper [6] . Casper is a combination of location anonymizer and Security aware query processor. Few other mechanisms like the Encrypted data store [8], key agreement [5], privacy tools [9]; In-device spatial cloaking assisted by cloud [10] is also part of the location security.

k-anonymity is a well known metric that is used for quantifying location security in mobile environments. Identity obfuscation is equally important when mobile device users are requesting for location based services. The location and identity are both being captured by the service providers in order to provide the user with the accurate results to the query. Hence it is very important to obfuscate both the identity and location so that the security is enforced but accurately providing the same set of results to the query.

This paper describes the implications of location security solution placement at the core, edge and device. It also presents experimentation results based on taxi cab, cellular and Watson datasets indicating the performance of the solution thus proving that location security is best placed at the edge of the network using a trusted edge server.

III. Security Metrics

In this section we present an empirical evaluation of the proposed location information flow control solution. Figure 2 shows a summary of the datasets used for evaluation. Three of the datasets `Shanghai`, `San Francisco` and `Stockholm` are taxicab traces obtained from the respective cities. The fourth (`Cellular`) is a user location trace and URL accesses obtained from a cellular network. The fifth (`Watson`) is an enterprise dataset obtained from WiFi location traces and URL accesses.

In the `Shanghai` and `San Francisco` datasets, there are explicit markers that indicate when the taxicab is occupied; in the `Stockholm` dataset collection of location trace is turned off when the taxicab is occupied (i.e., we only have trajectory information when the taxicab is not occupied). We use these datasets to quantify tradeoffs between the extent of obfuscation and anonymity.

In addition to these datasets, we use coarse grained mobility data from 16K mobile users obtained from CDRs (Call Detail

Records) and from about 1.2K enterprise users obtained from WiFi and web data accesses. While a taxicab's trajectory may be viewed as a mixture of several user trajectories (i.e., multiple passenger trajectories), this dataset captures movement information at the granularity of each user. However, location information is captured is at the level of cellular Basestation association, which depending upon urban/rural areas can range from a few 100 meters to about 5,000 meters. From a population of about 11.6M users, we selected about 16K users that had more than 400 CDRs per day (i.e., >400 location samples and data accesses per day). While we use the taxicab dataset to analyze fine grained trajectories (each corresponding to one trip), we use the cellular and enterprise dataset to analyze mobility across multiple trips undertaken by a single user.

Figures 3, 4, 5, 6 show the average anonymity as the extent of obfuscation is varied for times 7am-10am, 10am-4pm, 4pm-7pm and 7pm-7am respectively. As the extend of obfuscation is increased so does the extent of anonymity; further anonymity is generally higher during busy hours in the morning and the evening because several mobile users are active within a small spatial extent. The key challenge in practice is that these measures of anonymities are averages over the respective dataset. Hence, given a user location at a point in date and time, the challenge is to identify the amount of obfuscation required to can achieve a desired level of anonymity.

Figure 7 shows the number of users on the y-axis and similarity on x-axis. A point (x, y) in the figure indicates that there are at least y users whose profiles have a similarity of at least x with a randomly selected user. Similarity between user profiles is computed using a cosine distance on the set of URLs (web pages) accessed by a user with that of another user.

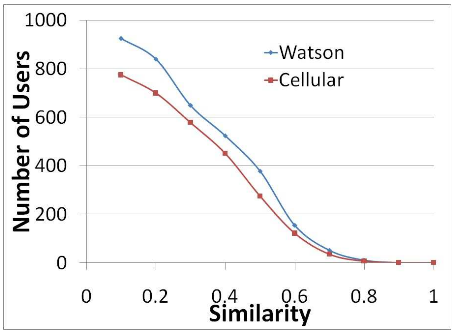

Fig. 7: Similarity of user profiles (based on data accesses)

Figures 8, 9, 10 show the complexity of a device-based model and false positive and false negative rates in enforcing the desired level of anonymity. A choice of obfuscation k is said to result in a false positive if it results in cloaking $< k$ users; and in a false negative if it results in cloaking $\geq k$ users. A false negative is an indicator of over obfuscation which would in turn affect the utility of the obfuscated data; while a false positive is in direct violation of the k-anonymity security requirement. In order to determine the level of obfuscation we analyze historical data using decision tree based machine learning algorithms – parameterized by location (typically encoded as latitude/longitude boxes) and timestamps (typically time of day and week). We tradeoff model complexity (i.e., number of nodes in the decision tree) with accuracy (i.e., being able to predict the desired level of obfuscation). We observe that increasing model complexity beyond a desired level increases the error primarily due to over fitting. We observe that in most cases the false positive and false negative rate of an optimal device-based algorithm (with large model complexity) varies between 0.12 and 0.25 for our datasets. This captures the extent of sub-optimality in a device-based solutions in comparison with an edge-based solution.

Figure 11, 12, 13 show the false positive rate (i.e., the odds of not meeting the desired level of anonymity) and location error. Location error is only computed when the choice of obfuscation meets the desired level of anonymity. If the choice of obfuscation meets the desired level of anonymity and nothing more than location error is zero. Otherwise, location error is computed as the difference between the extent of obfuscation chosen and the optimal obfuscation needed to achieve the desired level of anonymity.

Figure 14, 15, 16 shows the false positive rate (i.e., the probability of not meeting the desired level of anonymity) and location error with and without consideration to user similarity respectively. For this experiment the desired level of anonymity k = 16 and the desired level of user similarity is 0.0 (first case that ignores user profiles), 0.7 (in the second case) and 0.9 (in the third case). For instance when user similarity threshold is 0.7, amongst the set of users that are within the extent of obfuscation only those users whose profiles are at least 70% similar to the given user are considered for quantification of anonymity. This figure shows the additional cost (higher false positive rate and higher location error) that is incurred when enforcing location security based on profile cloning. We observe that when the similarity threshold is low the device-based solution pays a high penalty in terms of location error, while when the threshold is high the device-based solution pays a higher penalty in terms of false positive rate (i.e., the inability to meet the security requirement).

Figure 17, 18, 19 show the false positive rate (i.e., the odds of not meeting the desired level of anonymity) and location error while requiring a user similarity threshold of 0.7. Profiles for entities are drawn at random from the `Watson` dataset with the goal of showcasing tradeoffs between location security and identity/profile based obfuscation. Similar to prior experiments, location error is only computed when the choice of obfuscation meets the desired level of anonymity. If the choice of obfuscation meets the desired level of anonymity and nothing more than location error is zero. Otherwise, location error is computed as the difference between the extent of obfuscation chosen and the optimal obfuscation needed to achieve the desired level of anonymity.

IV. Android based implementation

This work has been implemented as an android based system. An application has been implemented in the android device in order to showcase the difference in the 2 methodologies. The solution at the device and the solution at the edge

Characteristic	Shanghai	San Francisco	Stockholm	Cellular	Watson
Sampling rate	2/min	12/min	1/min	>400/day	all web accesses
Number of entities	∼ 10,000	∼ 500	∼ 2000	∼ 16,000	∼1200
Source type	GPS	GPS	GPS	Cellular Basestation association	WiFi
Privacy	None	None	No sampling when taxi hired	Coarse grained samples	None
Timeline	1 month	1 month	1 month	1 month	1 month
Total number of trips	1,335,360	26,767	570,690	55,200	-
Total number of web accesses	-	-	-	12.4M	5.6M

Fig. 2: Summary of datasets

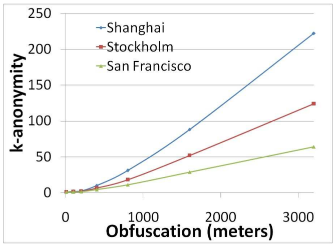

Fig. 3: 7-10am

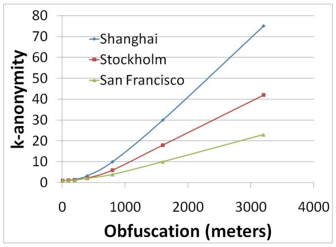

Fig. 4: 10am-4pm

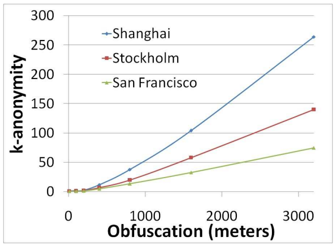

Fig. 5: 4-7pm

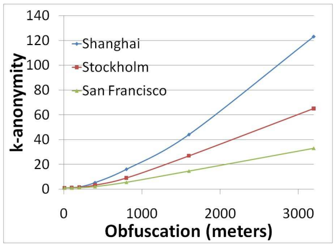

Fig. 6: 7pm-7am

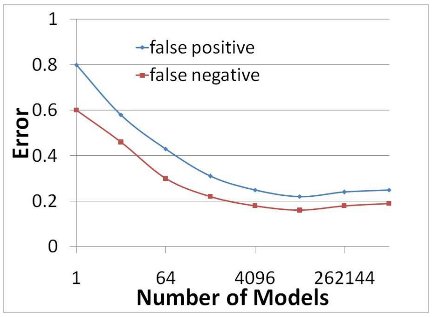

Fig. 8: Shanghai

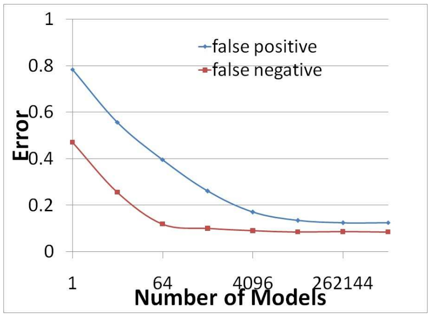

Fig. 9: Stockholm

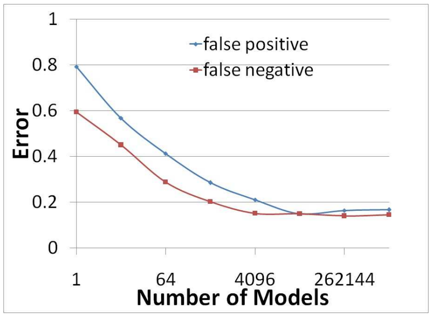

Fig. 10: San Francisco

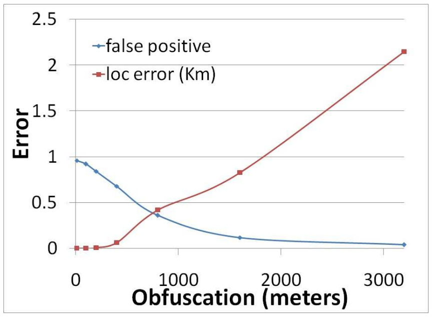

Fig. 11: Shanghai

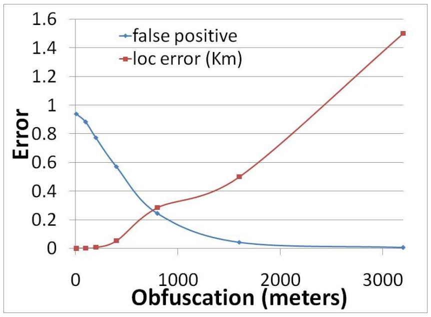

Fig. 12: Stockholm

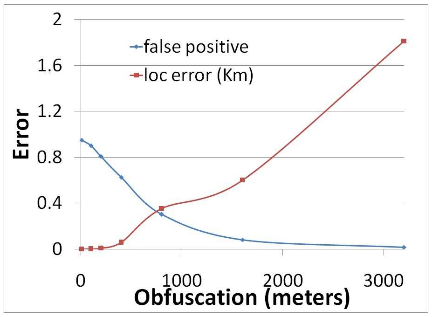

Fig. 13: San Francisco

have been implemented using an example of the London Boris bikes. Boris bikes are the easiest way to hire a cycle, ride it where you like and return it to any docking station. In this implementation, we have shown the means of how the system solution works when the solution is at the edge and when its at the device. In order to perform the implementation, we have made use of an application in an android device and then have implemented an edge server on a windows server. This server behaves as an edge which has the visibility to all the devices in the network and perform computations accordingly. The device based solution shows an android application with the map of London in it indicating the boris bikes available for hire. Request from the mobile device is shown on the map by indicating the current location of the device. By performing obfuscation on the device, it can be noticed that the obfuscation is not accurate enough as the device does not have visibility to other devices in the network. When the user then makes a request for the bikes, the responses received are not accurate due to the drawback of inaccurate obfuscation. In the case of solution at the edge, the edge has visibility to all the devices. When the user makes a request asking for the nearest bike hire from the current location, the edge takes care of obfuscating the current location of the device in comparison with the other devices in the network who would have made similar requests. The request is then sent from the obfuscated location and this results in accurate responses for the user requesting the locations of the bikes nearby from his location. Figures show the different stages in the demonstration of the location based request with the anonymised location and the results of the query. The solution has been implemented using the Eclipse development kit and has been tested with real use case scenarios. Figure 20 shows the device based solution

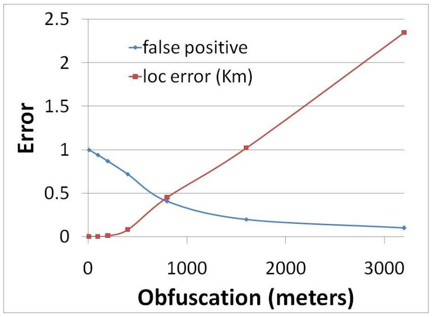

Fig. 14: Cellular: Sim Thr 0.0

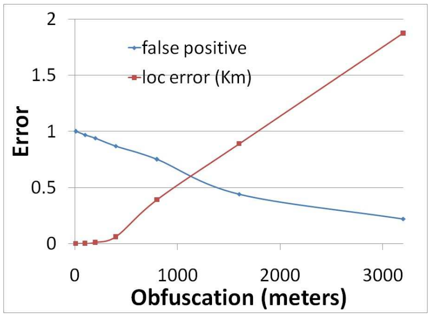

Fig. 15: Cellular: Sim Thr 0.7

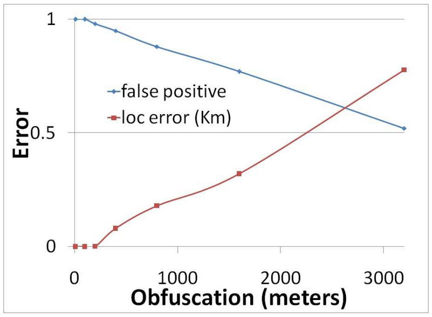

Fig. 16: Cellular: Sim Thr 0.9

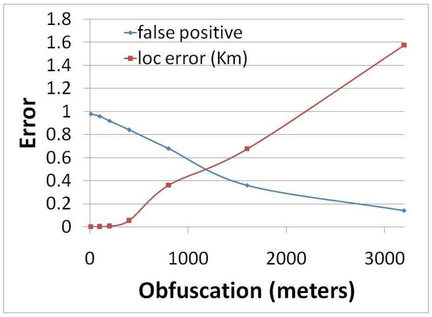

Fig. 17: Shanghai: Sim Thr 0.7

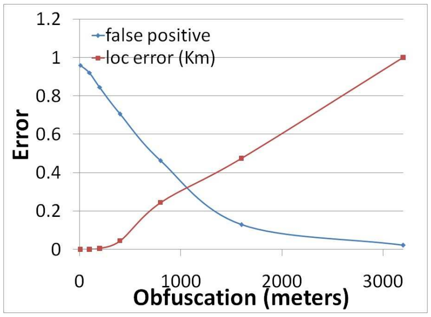

Fig. 18: Stockholm: Sim Thr 0.7

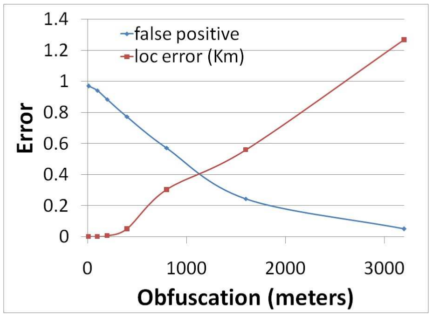

Fig. 19: San Francisco: Sim Thr 0.7

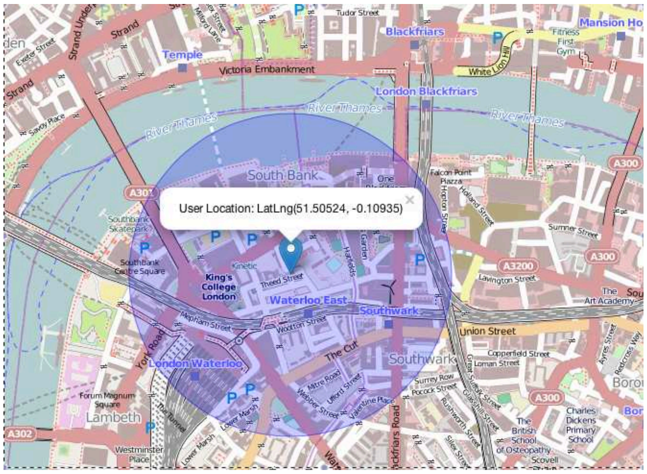

Fig. 20: Deviced based solution view of the London Thames region

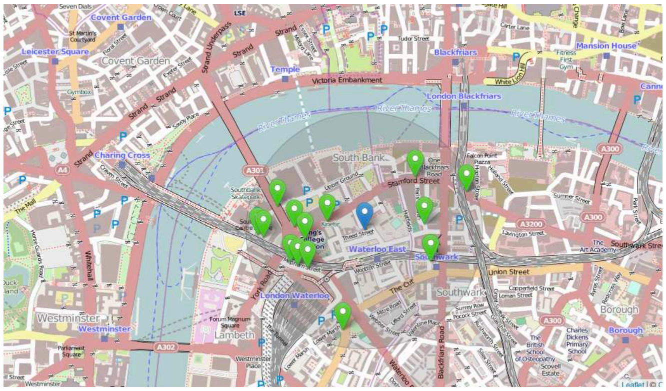

Fig. 21: Search results for the device based solution

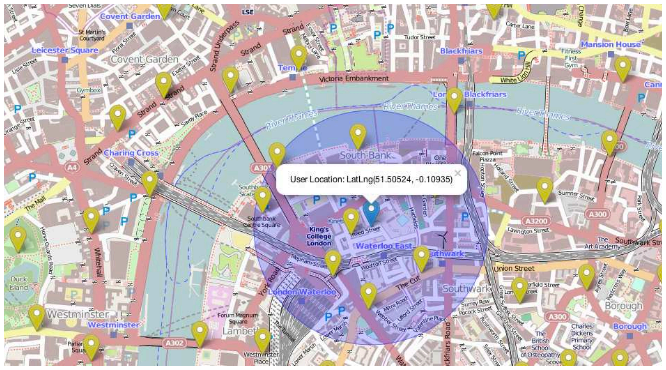

Fig. 22: Devices that are visible to the edge server

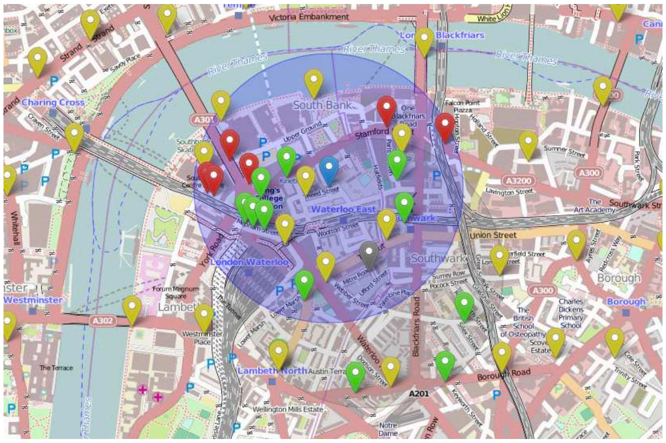

Fig. 23: Query results from true and obfuscated location

where user clicks on a particular point and then checks are done to see if the chosen location has enough obfuscation. Device level obfuscation cannot be performed as the device has no visibility to the other devices. Hence checks are done at the edge server to ensure that the obfuscation is good enough to make a query. Figure 21 shows the search results for boris bike using the device based solution. Figure 22 shows the view that the edge server would have of all the devices. Since the server can see all the devices, when a device makes a request for the bikes, the server can obfuscate the location based on the other devices in the area. On searching for the bikes based on the new obfuscated location, the results are displayed in figure 23. The comparison of results based on the search from the true location and the obfuscated location is shown using the 2 circles. This proves that the edge server functions close enough to the query made directly to the Boris bikes provider without any obfuscation.

V. Summary

This paper builds upon the vast literature in location anonymity by investigating a large unexplored facet of this problem – where to enforce location security and what are the tradeoffs in doing so? We have explored both device and edge based enforcement of location security and quantified the gap between optimal device-based enforcement with that of the edge-based enforcement. In particular, we have identified machine learning algorithms that determine the extent of location obfuscation that is needed to achieve a desired level of anonymity. We have shown that even with good models a device based solution (that is unaware of the instantaneous locations of other entities or their profiles) is largely suboptimal in determining extent of location obfuscation. Our experiments on various mobility datasets show that device-based solutions either suffer from high false positive rate (about 25% chance of not meeting the desired security requirement) or low utility (about 600 meters higher error in obfuscated location data).

Acknowledgments

Research was partly sponsored by US Army Research laboratory and the UK Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the U.S. Government, the UK Ministry of Defense, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] D. R. L. P. C. Bettini and S. Jajodia. Preserving anonymity of recurrent location-based queries. In *16th International Symposium on Temporal Representation and Reasoning, IEEE Computer Society*, 2009.

[2] H.-Y. Chien. New efficient user authentication scheme with user anonymity facilitating e-commerce applications. In *9th IEEE International Conference on E-Commerce Technology*, 2007.

[3] M. Chowdhury S. Hasan, Sheikh I. Ahamed. A privacy enhancing approach for identity inference protection in location-based services. In *33rd Annual IEEE International Computer Software and Applications Conference*, 2009.

[4] J. V. H Shin, V Atluri. A profile anonymization model for privacy in a personalized location based service environment. In *9th International Conference on Mobile Data Management. MDM'08,*, pages 73–80, 2008.

[5] G. K. M E Skarkala and S. Gritzalis. Milc: A secure and privacy-preserving mobile instant locator with chatting. In *Springer Science plus Business Media, LLC*, 2010.

[6] C. Y. C. M. F. Mokbel and W. G. Aref. The new casper: Query processing for location services without compromising privacy. In *In Proc. of VLDB*, 2006.

[7] D. P.Deivanai, Mrs .J. Jesu Vedha Nayahi. A hybrid data anonymization integrated with suppression for preserving privacy in mining multi party data. In *IEEE-International Conference on Recent Trends in Information Technology, ICRTIT*, 2011.

[8] K. Puttaswamy and B. Zhao. Preserving privacy in location-based mobile social applications. In *In HotMobile*, pages 1–6, 2010.

[9] J. F. R Shokri and J.-P. Hubaux. A unified framework for location privacy. In *In HotPETS*, 2010.

[10] W. X. S. W Song. In-device spatial cloaking for mobile user privacy assisted by the cloud. In *Eleventh International Conference on Mobile Data Management (MDM)*, pages 381–386, 2010.

[11] S. W. G.-H. T. R. G. T. H. K. L. H. T. K. Warr and M. Zafer. Mobile micro-cloud: Application classification, mapping, and deployment. In *Proc. of Annual Fall Meeting of ITA (AMITA) 2013, Oct. 2013.*, 2013.

[12] R. W. Z Liang. Efficient k-anonymization for privacy preservation. In *Computer Supported Cooperative Work in Design, 12th International Conference*, pages 737–742, 2008.