Development of Cybersecurity Lab Exercises for Mobile Health

Hongmei Chi
Dept. of Comp.& Info Sciences
Florida A&M University
Tallahassee, Fl, USA
hongmei.chi@famu.edu

Meysam Ghaffari
Dept. of Computer Science
Florida State University
Tallahassee, Fl, USA
ghaffari@cs.fsu.edu

Ashok Srinivasan
Dept. of Computer Science
University of West Florida
Pensacola, Fl, USA
asrinivasan@uwf.edu

Jinwei Liu
Dept. of Comp.& Info Sciences
Florida A&M University
Tallahassee, Fl, USA
jinwei.liu@famu.edu

***Abstract*—There is an emerging class of public health applications where non-health data from mobile apps, such as social media data, are used in subsequent models that identify threats to public health. On one hand, these models require accurate data, which would have an immense impact on public health. On the other hand, results from these models could compromise the privacy of an individual's health status even without directly using health data. In addition, privacy could also be affected if systems hosting these models are compromised through security breaches. Students ought to be trained in evaluating the effectiveness of different protocols in ensuring privacy while providing useful data to the models.**

There is a lacuna in current cybersecurity education in training students in the context of both the above types of mobile health applications. The objective of this paper is to describe novel educational material to augment current cybersecurity courses for undergraduate and graduate students. We develop material to teach about fundamental concepts and issues related to security and privacy in mobile health applications and describe a cloud-based hands-on lab that lets students explore the consequences of different solution strategies. Hands-on lab exercises will provide students with insight into the development of practical solutions based on sound theoretical foundations.

Keywords—hands-on lab, mHealth app, privacy and security, health data, practical solutions, K-anonymity

I. Introduction

Mobile devices, such as smartphones and tablets, are a convenient means of managing health care and an effective means of promoting healthy behaviors because they are popular, and most users carry them with themselves everywhere [1]. For example, in 2017, 95% of US adults had a cell phone and 77% of those phones were smartphones. While a survey in 2017 showed only 31% of users were using healthcare apps, recent surveys show 62% of cell phone holders have used at least one healthcare app [2]. This is a substantial growth in the number of users of such services.

Mobile health applications are proved to be useful for the early detection of physical and mental health illnesses. This has led to the majority of US smartphone users installing at least one health application on their phones. There is usually an exchange of data between such applications and servers on the cloud that store and process much of the information. The security of health data and systems hosting them is clearly important. Cybersecurity education ought to train the workforce on techniques for protecting such information and systems.

While the above deals with individual health, there is also an emerging class of public health applications that uses non-health data to predict consequences to public health. For example, a recent study on using mobile devices for modeling disease outbreaks after an earthquake used cell phone location data to determine human movement patterns, which proved to be more accurate than government data [3]. Google Flu Trends and Google Dengue Trends used search query results to try to predict disease outbreaks. Google Health Trends continues making new data available to researchers. There is also much current research that uses social media data to identify health status for the purpose of public health [6]. Much of the social media data and search query data are generated through mobile devices. Models could use non-health information, gleaned from these sources, to identify the likelihood of diseases for an individual. Due to these sophisticated scientific models, security vulnerabilities of otherwise innocuous data could be used to infer health information, thereby compromising the privacy of individuals. These vulnerabilities could arise either in the transmission of the information or in the cloud-based systems running the models.

Privacy is an important issue in these mobile healthcare applications, because users are concerned about their health information being revealed. With increasing concerns about users' privacy, it became mandatory that organizations follow strict security and privacy protections [2]. In addition, protocols can be designed so that privacy is not compromised. Consequently, much research has been performed on preserving privacy while providing a user with mobile health features [4]. Techniques such as anonymization, using a trusted anonymization server or other methods such as P2P anonymization, address conventional privacy concerns [5]. However, we also need to address issues that arise from the use of public health models that use non-health data. On one hand, such models may infer health information from data that appears innocuous. On the other hand, it is useful, from a social perspective, for such models to have adequately accurate data.

Currently, there is a lack of education material that trains students in security and privacy solutions arising from mobile

979-8-6425-4896-7/20/$26.00 ©2020 CISSE

health applications. Given the rapid developments in this field, students with training in this field would have good employment opportunities. We aim to address this lacuna in cybersecurity education by developing tutorial material on security and privacy issues in mobile health contexts, along with a set of exercises deployed on a cloud-based lab, that will help students explore different security and privacy vulnerability scenarios. This could have a transformative effect on cybersecurity education.

II. Design

Communication from a device to its destination can be secured using an encryption protocol. However, this is not adequate when private information, such as location and health condition, is sent because identity and health status could be inferred indirectly. Ideally, even the server running the model or any node in the system should not be able to infer the identity of the user and the user's private information at the same time. Several protocols have been developed to attain this objective, including by one of the authors [5].

The simplest method is *Pseudonym* where, instead of the ID, users use a pseudonym that could be changed periodically. One limitation of this method is its communication overhead when the pseudonym is changed. Moreover, it is possible to link different pseudonyms to each other based on their behavior patterns. Furthermore, it does not preserve privacy in sparsely populated locations. For example, consider an extreme situation in which a public health model indicates the risk of acquiring a certain disease, and the person using that model notices that there is only one other person in that location. The user could recognize that person as a likely carrier of that disease.

Sending fake queries is another approach, which hides the real queries amongst fake ones. A significant limitation of this method is that it results in substantial overhead in the server due to the large number of fake queries sent along with the real ones. Furthermore, techniques such as map matching [5] can cull the majority of fake queries.

The Perturbation method tries to achieve privacy by adding noise to the queries, such that the exact location of the user cannot be detected. An advantage of this method lies in its ease of deployment, without the need for much additional information. One limitation of this method is that it is hard to determine the amount of noise that is required to hide the identity of the user. In addition, any reasonable noise may be inadequate to address the problem mentioned above with sparse zones. Besides, the loss in quality of the queries due to noise could lead to a loss in the quality of the answers. This problem is magnified by the fact that the optimum noise is not known. Furthermore, it has downstream consequences, such as affecting the results of public-health models that rely on the data collected.

Spatial and temporal cloaking are specific types of perturbation, where the query location and time are respectively perturbed. They are explained further in Section III. Identifying the optimum perturbation value is not easy. A small value would decrease privacy while a high value could decrease the quality of service.

A trusted central server approach has been proposed by Gruteser and Goldwin [11] in order to find the optimum spatial and temporal cloaking for the queries and is shown in Figure 1. In this approach, users' queries are sent to a central trusted server. This server applies temporal and spatial cloaking to ensure K-anonymity; that is, metadata related to the user in a query is indistinguishable from user information for K-1 other users, which hinders re-identification of the original user. This trusted server then sends the K-anonymized queries to the main server. So, the final queries are anonymized, and an intruder cannot re-identify the user. Moreover, since the trusted server knows the different queries from different users, it could add optimum spatial and temporal cloaking to the queries, thus minimizing the quality loss.

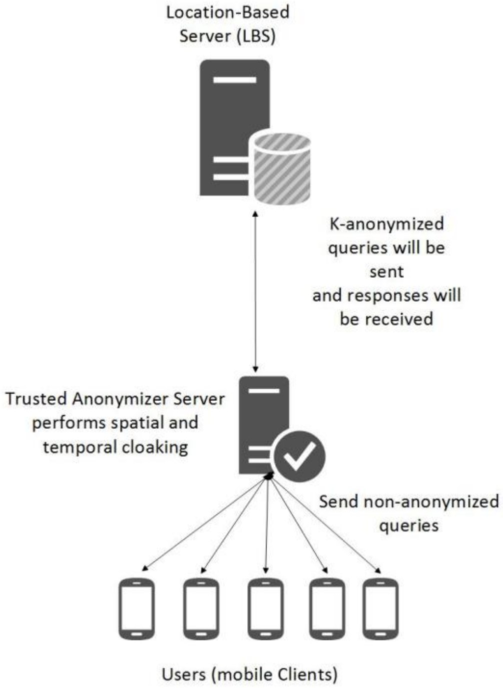

Fig. 1. Architecture of Central Trusted Server

On the other hand, the trusted server could become a security bottleneck, since all the queries are sent to it; in case of an unauthorized access, users' information could be revealed. It can also become a performance bottleneck.

A P2P anonymization technique can be used to reduce the risk of such a central bottleneck. Here, nodes cooperate with each other to perform anonymization, and users' sensitive information will not be revealed [5].

We implement a variety of anonymization techniques on user queries, including peer-to-peer anonymization techniques, such that an intruder cannot acquire identifiable information about specific users. We also need to implement a secure connection between users such that the nodes cannot access the data before anonymization. In the proposed methods, the spatial and temporal cloaking will be applied to the users' queries dynamically to preserve the users' privacy while minimizing the quality of service loss. Students will also study the tradeoff among privacy, quality of service loss, and impact on public-health models that use such data.

III. Case Study

In this section, we outline two real examples of hands-on labs for our students.

Hands-on Lab 1: Goals: At the end of this lab, the students will,

- Become familiar with the concept of anonymity
- Learn how location cloaking can help preserve users' privacy
- Study about concepts of K-anonymity
- Implement and work with a trusted central anonymizer

Tools: Ubuntu, Wireshark, and Amazon AWS.

Scenario: This exercise simulates a realistic scenario during a disease outbreak (such as Zika virus), where a public health model relies on data from users to identify high-risk locations. It provides students training on applying techniques that preserve anonymity and exploring the tradeoff between preserving anonymity and privacy on one hand versus enabling the public-health model functionality on the other hand. In this system, users will send their information, such as hometown and current location, to the server using an application. In response, the application tells them if there is a zone with a high risk of a disease nearby. The server analyzes the information of users in a zone and assigns a risk factor to the zone based on information of users located there, such as their home location, places they have visited, and their health information. The server then announces this risk factor to all nearby users, so that they may avoid that location if they consider the risk is too high.

The challenge here is the resolution of the announced zones. A higher resolution results in a high quality of service. Higher quality of service means that the application has smaller tagged zones; if a small zone is marked as risky, one could avoid it easily since it is small. However, this might violate other users' privacy or anonymity. For example, if a high-risk zone contains just a few users, others observing that location can guess that people there are probably sick (see Figure 2). There is a clear trade-off between the quality of the service and the privacy of the users.

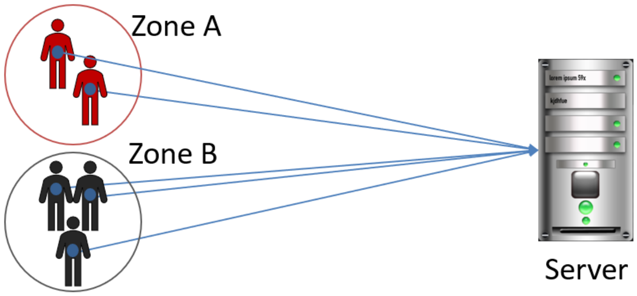

Fig. 2. Defining the risk in each zone without applying anonymity could cause privacy concerns for the users

In addition, an intruder with access to the server can acquire the user data. An intruder may be someone who owns or is using the server. Alternatively, this may be an online intruder, for example in the case of having a man-in-the-middle attack, which can happen anywhere in the data flow between the user and the server. In particular, since all the data is stored on the server, access to the server provides access to the entire user information.

The following are common methods to preserve privacy.

Fake queries: Send fake queries along with the real one such that detecting the real query becomes hard.

Pseudonyms: Instead of using a username, use a pseudonym as an ID, and change it over time to make it hard to detect the real identity of the user.

Spatial cloaking: This method considers a zone around a user's query and changes the location of the query randomly to somewhere in that zone. This method prevents detecting the user based on the query location. For example, if the query location is (X,Y), then after spatial cloaking it will change to $(X+\alpha 1, Y+\alpha 2)$, where $\alpha 1$ and $\alpha 2$ are two random variables in the range of (-R, R). R is the radius of the cloaking zone around the query.

Temporal cloaking: Temporal cloaking considers a specific time interval and changes the query time randomly within that time interval so that the exact query time is not detectable. For example, the query could be delayed for t seconds, where t is between 0 and 100 seconds.

Spatial and temporal cloaking can help to conceal a user's identity by making the time and location of a few queries from different users similar to each other. For example, if there are two users sending a query to the server, and their location is within a radius R, the intruder cannot identify the exact source of each query when spatial cloaking has changed their location within radius R randomly. However, it is not possible for the user to know the optimum radius R to perform spatial cloaking since the user lacks information on surrounding users and the numbers of their queries. To solve this problem, a central trusted anonymizer has been proposed [11]. In this approach, a trusted server receives the queries of the users, and performs spatial and temporal cloaking on them such that at least K of them are within the radius R and their query time is within a time span of t. By this means, an intruder cannot identify the source of each query from its time and location. Making K queries indistinguishable in their user metadata is called K-anonymity, which means the intruder has at least K possible source for each query (Figure 3).

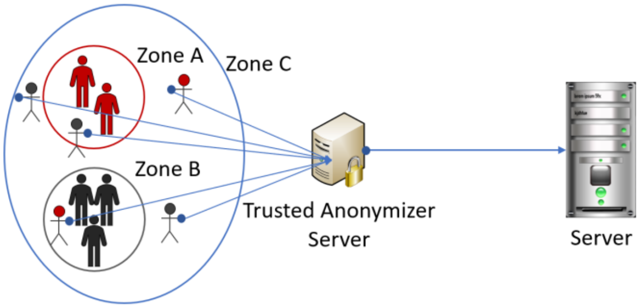

Fig. 3. The anonymizer performs spatial cloaking to the user queries and send the new locations to the server. In this case the privacy of the user will be preserved.

If K is sufficiently high, then an observer cannot relate the risk of the disease to any of the users in that zone. Using a trusted anonymizer, the resolution of the zones could be changed dynamically such that each zone has at least K different users. So, the intruder cannot correlate the risk to any specific person and the privacy of the users will be preserved. On the other hand, the risk zone by the public health model would become larger, which decreases the quality of the service. For example, instead of announcing a house as a high-risk zone, the server may announce a part of a street as a high-risk zone.

If there are few users in a specific zone, the trusted anonymizer combines this zone with one of the adjacent zones to have at least K users in the newly created zone (Zone C in Figure 3). Then it changes the location of the users' queries in this new zone to make them indistinguishable (see Figure 3). After performing spatial cloaking, the location of queries is changed so that they do not reveal the exact location of the original sender of the queries. The anonymizer then sends the queries to the server.

While the server cannot realize the exact location or identity of the original sender of a query, it can process the queries and send the responses back to the anonymizer. Finally, the anonymizer sends the responses to the users since it knows the owner of each query. Using this method, there are no direct interactions between the user and the server. In this scenario, we assume that the trusted anonymizer server is safe and reliable and will not reveal those user identities or sensitive information.

As an example, assume we have zone A with 2 infected patients and zone B with 3 healthy persons, with K defined as 5. The anonymizer will anonymize the queries and sends the queries as zone C (which includes both zones A, B), such that in the new zone we have two infected and three healthy people. Using this approach, the privacy of the users will be preserved while the server is still able to send a response with an acceptable resolution.

In our proposed plan, students will connect to the cloud where the central trusted anonymizer, several user nodes, and the server are implemented. They will use open source Wireshark software to check the packets and understand more about the anonymizer and its benefits.

By checking the packet contents before and after anonymization, they will understand this process better. In some cases, there might be zones with a single person or a few people from one country. In these cases, having the risk factor of these zones, conclusions can be drawn about that single person or the country of people in a zone. This can be a violation of user privacy. Students can try finding such scenarios to understand the importance of privacy-preserving techniques that we are teaching in these labs. In addition, students will have more sense about privacy persistence in mHealth app scenarios. In the meantime, we reveal other use scenarios that could result in the absence of privacy persistence.

Hands-on Lab 2: the second hands-on lab focuses on distributed anonymization.

Goals: At the end of this lab, students will

- Learn about connecting and using the cloud
- Become familiar with a distributed system
- Implement and work with the distributed anonymizer
- Learn about symmetric and asymmetric encryption methods
- Learn to use Wireshark for sniffing
- Learn how distributed anonymizer can help preserve users' privacy while avoiding a security bottleneck

Tools: Ubuntu, Wireshark, and Amazon AWS.

Scenario: During a disease outbreak, tracing human movements can help identify high-risk zones, which can yield effective public health interventions. We consider a mobile app that uses a user's locations and time to inform them of spatial zones nearby that are at high risk of disease spread, using a public health model. While users can benefit from such a service, they would not want their individual disease status or disease risk to be revealed.

The central trusted server described in the previous hands-on lab is not the ideal solution for the following reason. If an intruder has access to the trusted server, the intruder can analyze the user's private information. To solve this problem, the distributed anonymization technique has been introduced. This hands-on lab exercise provides students training in applying techniques that preserve anonymity using a distributed architecture. In addition, it enables students to explore the tradeoff between preserving anonymity and privacy on the one hand versus enabling the public health application's functionality on the other hand. Students will also learn about symmetric and asymmetric encryptions and their features. We provide further details below.

As discussed above, the central trusted server has security and computational bottlenecks. So, here we discuss the distributed anonymizer architecture that performs K-anonymity. In this architecture, each device (typically a cell phone) used by a user also serves as an anonymizer. The distributed architecture where the nodes collaborate with each other and communicate with each other is called a peer to peer architecture. So, the system discussed here is a peer to peer anonymizer system (see Figure 4).

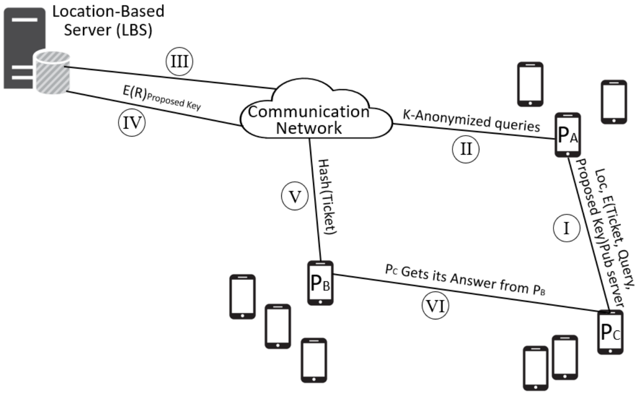
Location-Based Server (LBS)
III
E(R)Proposed Key
IV
Communication Network
K-Anonymized queries
II
PA
Loc, E(Ticket, Query, Proposed Key)Pub server
I
Hash(Ticket)
V
PB
Pc Gets its Answer from PB
VI
Pc

Fig. 4. Distributed anonymizer

In this architecture, the user sends the user location to another node which is responsible for the anonymization of that zone. That node performs anonymization on queries received from multiple users and sends them to the server. The server will send the response to a different node in the system (Broker node). The user will request the broker and get the response from it. In order to limit the access of each node in the system, the user will encrypt the query with the server public key. However, the user's location data will be sent as plain text beside the encrypted query. The anonymizer will get the query and location but cannot analyze the query since it is encrypted. The anonymizer performs the location cloaking on the users' queries to make them K-anonymized before sending the group of anonymized queries to the server. The server gets the queries, decrypts them, and processes them. However, it cannot identify the owner of any query since they are anonymized. The server then sends the response of each query to a broker. The broker is different for each query and it will be based on the hash of a random number that is suggested by the user. So, the server cannot predict the broker that each user will select. Finally, the user will ask the broker for the desired response. The response is also encrypted by the user's suggested key, so the broker cannot analyze the response either.

Using this architecture, the server or the intruder cannot get the user's private information, since the information is encrypted using symmetric and asymmetric technique and each node can only access the required data and not more. The sequence of data communication is as follows.

- User → anonymizer: (Query, Query ID (Random), Proposed encryption key, Random number for selecting the broker)_Encrypted with server public key, location
- Anonymizer → server: (Query, Query ID (Random), Proposed encryption key, Random number for selecting the broker)_Encrypted with server public key, (location)_Anonymized
- Server → broker: (Response)_Proposed encryption key, Query ID
- User → broker: Query ID
- Broker → user: (Response)_Proposed encryption key

In our proposed plan, students connect to the cloud where the peer to peer architecture, several user nodes, and the server are implemented. They use Wireshark software to check the packets and understand more about the anonymizer and its benefits.

By checking the packet contents before and after anonymization, they will understand this process better. Moreover, they will learn about the symmetric and asymmetric encryption and their differences by analyzing the queries that are sent to the server and the responses.

Description of each step

In the first step, the students install a virtual machine image or connect to the cloud. Over there, they have access to the simulated user server connections.

There are multiple user threads running on the cloud that send queries to the distributed system. Each query will be assigned to one anonymizer based on the zones that are assigned to each anonymizer in the system. The anonymizer will perform location cloaking and send the queries to the server. The anonymizer cannot process the contents of the query since it is encrypted using the server public key.

In the next step, students will analyze the user queries sent to the anonymizer and the cloaked queries sent to the server. By analyzing and comparing the queries before and after anonymization, they will learn about the effect of anonymization by spatial cloaking and find out their differences. They should be able to identify the application and benefits of cloaking and symmetric and asymmetric encryptions.

In the next step, they will try to analyze the packets in anonymizer, server, and broker to see if they can re-identify the users from them.

Finally, we will have students evaluate the tradeoff between privacy and service. We will consider users in an airport, which includes persons from a region suffering from a disease outbreak. The app will use a public health model to identify zones within the airport that are at high risk of disease spread. Students will evaluate the tradeoff between cloaking parameters that ensure the privacy of users and their effectiveness in enabling the public health application to identify precise zones that are at high risk of disease spread.

IV. Student Feedback

During Spring 2019, hands-on Lab 1 was given to an introduction cybersecurity class and a pre-survey and post-survey were conducted before and after this lab was done. During the survey, students were asked a series of questions regarding security and privacy in Wireshark, mobile health applications, K-anonymity, and other preserving privacy methods.

Twenty-six students responded to the questionnaire of pre-survey and post-survey. When students were asked, “Are you familiar with hands-on labs related to mobile health app preserving privacy?” Only one student (3.8%) stated “Yes”, and the rest students answered “No”. When students were asked, “Have you heard of K-anonymity and other preserving privacy methods?” 76.92% stated, “No” and 23.08% “Yes”. Students who stated, “Yes” described K-anonymity incorrectly. Those preliminary results show that our future cybersecurity professionals lack of training in preserving privacy related to mobile health, especially when measuring human mobility and disease connectivity [8,9].

After the lab exercises, students were asked the following questions. "Do you like to do more hands-on labs related to mobile health app preserving privacy?” 88% stated, “Yes” and 12% “No”. "Did this hands-on lab inspire you to learn more about preserving privacy related to an outbreak disease?" 84% stated, “Yes”, 4% “No”, and 12% "Maybe". This indicates the need for such labs and student recognition of their value.

V. Conclusions

In this paper, we describe multiple novel hands-on labs based on K-anonymity and other privacy-preserving methods [10]. Feedbacks from our students are positive and promising. In the future, we will prepare more hands-on labs to cover trending topics and security tools based on applying anonymity frameworks [11]. In addition, we will update the materials to make them more student-friendly and efficient based on the students’ feedback.

Acknowledgement

This work has been supported by CyberFlorida for Capacity Building Award Program. The opinions expressed in this paper do not necessarily reflect those of the funding agencies.

References

[1] C. Free, G. Phillips, L. Galli, L. Watson, L. Felix, P. Edwards, V. Patel, and A. Haines. "The effectiveness of mobile-health technology-based health behaviour change or disease management interventions for health care consumers: a systematic review." PLoS Medicine 10.1 (2013): e1001362.

[2] S. Fox and M. Duggan. Mobile health 2010. Washington, DC: Pew Internet & American Life Project, 2010.

[3] L. Bengtsson, X. Lu, A. Thorson, R. Garfield, and J. Schreeb. "Improved response to disasters and outbreaks by tracking population movements with mobile phone network data: a post-earthquake geospatial study in Haiti." PLoS Medicine 8.8 (2011): e1001083.

[4] R. Lu, X. Lin, and X. Shen. "SPOC: A secure and privacy-preserving opportunistic computing framework for mobile-healthcare emergency." IEEE Transactions on Parallel and Distributed Systems 24.3 (2013): 614-624.

[5] M. Ghaffari, N. Ghadiri, M. Manshaei, and M. Lahijani. "P4QS: A Peer-to-Peer Privacy Preserving Query Service for Location-Based Mobile Applications." IEEE Transactions on Vehicular Technology 66.10 (2017): 9458-9469.

[6] L. Li, K. Qian, Q. Chen, R. Hasan, G. Shao. "Developing Hands-on Labware for Emerging Database Security." Proceedings of the 17th Annual Conference on Information Technology Education. ACM, 2016.

[7] T. Peng, Q. Liu, G. Wang, Y. Xiang, and S. Chen. Multidimensional privacy preservation in location-based services. Future Generation Computer Systems, 93, 312-326, 2019.

[8] M. Ghaffari and J. Wang. Integrating Travel and Epidemic Models through Computational Analysis. In Proceedings of the Practice and Experience on Advanced Research Computing (p. 91). ACM, July, 2018.

[9] C. Panigutti, M. Tizzoni, P. Bajardi, Z. Smoreda, and V. Colizza. (2017). Assessing the use of mobile phone data to describe recurrent mobility patterns in spatial epidemic models. Royal Society Open Science, 4(5), 160950.

[10] S. Lai, A. Farnham, N. Ruktanonchai, and A. Tatem. (2019). Measuring mobility, disease connectivity and individual risk: a review of using mobile phone data and mHealth for travel medicine. Journal of Travel Medicine, 26(3), taz019.

[11] M. Gruteser and D. Grunwald. "Anonymous usage of location-based services through spatial and temporal cloaking." *Proceedings of the 1st international conference on Mobile systems, applications and services*. ACM, 2003