Peer to Peer Location Sharing

Dmitry Namiot
Lomonosov Moscow State University
Faculty of Computational Math and Cybernetics
Moscow, Russia
dnamiot@gmail.com

Manfred Sneps-Sneppe
Ventspils University College
Ventspils International Radio Astronomy Centre
Ventspils, Latvia
manfreds.sneps@gmail.com

Abstract— **This paper describes several new models for sharing location information without disclosing identity data to some third party server. All our services proposed in this research share the common background idea. It could be described as a safe location sharing. It combines server-side (centralized) location information with the locally based distributed identity information. In this distributed data store, identity info is always saved locally. The proposed approach eliminates one of the biggest concerns for location based systems adoption – privacy. This article describes our approach as well as several generic implementations.**

Keywords-location; privacy; lbs; mobile; HTML5; geo coding; social networks

I. INTRODUCTION

It is a well-known fact that the question "Where are you?" is one of the most often asked during the communications. More than 600 billion text messages per year in the US ask "Where are you?" [1]. A huge amount of services use location exchange as a key feature. Location plays a basic role in context-aware applications. The classical definition [2] describes context as location, identities of nearby people and objects, and changes to those objects. As per N. Hristova [3], context-related information can consist of user profiles and preferences, their current location, the type of connection that to the mobile network, the type of wireless device being used, the objects that are currently in the user's proximity, and/or information about their behavioral history. Actually, most of the authors define context awareness as complementary element to location awareness, whereas location may serve as a determinant for resident processes. By this reason, all the context-aware applications are linked to location exchange.

Location, as one of the most widely adopted sensor readings of a modern smart phone, is probably the first attribute (candidate) to share for mobile users. The typical applications are well known and include for example geo-tagged context, friend-finder, recommendation systems, turn-by-turn navigation, etc.

In location-based service (LBS) scenarios we can describe the following actors [4]:

- Intended recipient. For example, the service company, friends, parents, etc. This usually involves the use of a service provider that offers to forward your location to the intended recipient. Actually, it is a main goal for our research. How can we deal with intended recioients without telling too much data to service providers? The final goal is to create a safe location sharing system without explicit centre for all circulated data.
- Service provider. For example, Google providing you with the Latitude application or Yelp provides a restaurant recommendation system for near-by places. In contrast to the intended recipient, users usually do not have a primary goal of letting the service provider know their location – it is a by-product of getting a restaurant review or staying in touch with friends.
- Infrastructure provider. The typical example is a mobile operator. While self-positioning systems such as GPS can work without an infrastructure provider, mobile phone users are often implicitly located in order to provide communication services (for example, route phone calls or emergency communications).
- Unintended recipients. For example, we can mention accidental recipient, illegal recipient and law enforcement.

In the most cases, by describing various LBS, we assume that for a given system, the infrastructure provider needs to be trusted. In other words, the need for sharing location data with infrastructure providers is always a non-discussable topic. In the same time, Palen [5] argues that privacy is not simply a problem of setting rules and enforcing them. It is rather an ongoing and organic process of negotiating boundaries of disclosure, identity, and time. Authors suggest genres of disclosure for managing interpersonal privacy, which are "socially-constructed patterns of privacy management," as a sort of design pattern approach to support the development of privacy-sensitive applications. Examples might include creating and managing accounts at shopping Web sites, taking appropriate photographs at social events, exchanging contact information with a new acquaintance, etc. [6].

This paper summarizes our efforts in safe location sharing. The rest of the paper is organized as follows. Section II contains an analysis of existing projects devoted to privacy in location sharing applications. In Section III, we consider our Geo Messages service and related applications. In Section IV, we describe our WATN application..

II. LOCATION SHARING AND PRIVACY

In the most cases, location sharing is implemented as the ability for the mobile user (participants) write down (save)

 ISBN: 978-1-61208-262-2

own location info in the some special place (e.g., special mobile application).

But, it means of course that user must be registered in this service or deploy some specially downloaded application. What is more important here – everyone who needs this information must use the same service too [1]. This chicken-egg problem is a typical for many LBS applications. The service is useless until many users register there, but there are no reasons to register due to lack of useful information from userless service.

One of the biggest concerns for all location-based services is user's privacy. Despite the increased availability of these location-sharing applications, we have not seen yet wide adoption for the most of them. It has been suggested that the reason for this lack of adoption may be users' privacy concerns regarding the sharing and use of their location information.

For example, the widely cited review of social networks practices [7] concluded, that location information is preferably shared on a need to know basis, not broadcast.

Participants were biased against sharing their location constantly, without explicit consent each time their location is requested. This suggests that people are cautious about sharing their location and need to be reassured that their private information is only being disclosed when necessary and is not readily available to everybody.

The key point for any existing service is some third party server that keeps identities and locations. We can vary the approaches for sharing (identity, locations) pairs but we could not remove the main part in privacy related concerns – the third part server itself.

As it is mentioned in [8], peer opinion and technical achievements contribute most to whether or not participants thought they would continue to use a mobile location technology. In this connection, Hong [6] suggests the following end-user and application developer requirements divided into four high-level groups:

- A decentralized architecture, where as much personal information about an end-user is captured, stored, and processed on local devices owned by that end-user.
- A range of mechanisms for control and feedback by end users over the access, flow, and retention of personal information, to support the development of pessimistic, optimistic, and mixed-initiative applications.
- A level of plausible deniability built in.
- Special exceptions for emergencies.

Actually, we will present below some mix (mashup) of decentralized and server-based architectures.

One possible solution is using peer-to-peer location sharing. The easiest way to apparently "solve" location privacy problems is to authorize (or do not authorize) manually or automatically the disclosure of location information to others. But, we should see in the same time the other privacy related problem that is not eliminated. Your location will be disclosed to (saved on) some third party server. For example, you can share location info in Google Latitude on "per friend" mode, but there is still some third party server (Google) that keeps your location and your identity. In other words, we still should have some trusted source. This source keeps all information.

Typically we have now two models for location sharing in services. At the first hand, it is some form of passive location monitoring and future access to the accumulated data trough some API; it is Google Latitude, for example. The first possible problem with this approach is privacy. There is some third party tool that constantly monitors my location and what is more important – saves it on the some external server. The second problem is the shorted life for handset's batteries.

Another model for location sharing is check-in. It could be an active (e.g., Foursquare), when user directly writes down his current location or passive (e.g., Twitter), when location info could be added to the current message. A check-in is a simple way to keep tabs on where you've been, broadcast to your friends where you are, and discover more about other people in your community. But, here we can see not only privacy related issues, when all my friends (followers) can see my location. We have here also a noise related issue too - my location info could be actually interested for the physical friends only. For the majority of followers my location info (e.g., Foursquare's statuses in my Twitter's time line) is just a noise [9].

There is another interesting solution associated with check-ins. At the first hand, technically check-ins could be customized [10]. But, after the customization, we can make the next step also and create a new form of check-ins that is disconnected from the social stream. We can create a new type of check-in records and separate them from rest of stream. It means that we will provide a separate database that just contains a list of accounts from the social network being concentrated (at this moment!) nearby some place. It is a temporal database, check-in records could be changed constantly and it does not contain the social stream itself – just IDs (e.g., nick names) for accounts confirmed check-ins. Figure 1 illustrates the new check-in form.

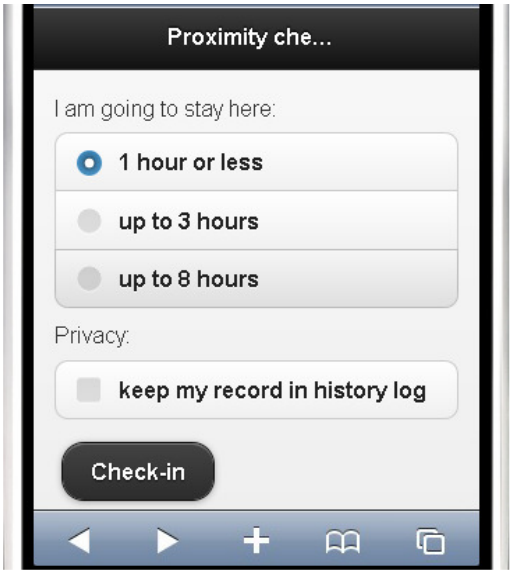

Figure 1. External check-in

 ISBN: 978-1-61208-262-2

It is one of the features for the upcoming new SpotEx [11] version. This temporal database lists people who are currently nearby some selected point (Wi-Fi access point in Spotex). And more important, that this database is visible only to those of people who are nearby themselves too. It is some form of temporal geo-limited location sharing [12].

Lets us describe some existing approaches in LBS development that targets the privacy.

One of the most popular methods for location privacy is obfuscation [13]. Obfuscating location information lowers its precision, e.g., showing only street or city level location instead of the actual coordinates, so that the visible (within our system) location does not correspond to the real one. For example, in Google Latitude we can allow some of the users get our own location info on the city level only. Sometimes even the random noise could be added to the real location data [14]. But, once again – it is just a visible location. The central point (points) for such a system can have all the information.

Some papers prefer the term spatial cloaking and describe it as the most commonly used privacy-enhancing technique in LBS. In spatial cloaking algorithm, mobile and stationary users can entertain location-based services without revealing their exact location information. The main idea is that before requesting any location-based service, the mobile user will form a group from his peers via single-hop communication and/or multi-hop routing. Then, the spatial cloaked area is computed as the region that covers the entire group of peers. [15].

Another popular approach in the area of location privacy is "k-anonymity" [16]. As per this approach, the actual location is substituted by a region containing at least k − 1 other users, thus ensuring that a particular request can only be attributed to "1 out of k" people. Of course, this approach has the disadvantage that if the region contains too few people, it has to be enlarged until it contains the right number of people. But, in general k-anonymity protects identity information in a location-oriented context [17]. In the same time, the group-composing algorithm is complex and the member peers are dynamic. The big question here is the level of implementation. Who is responsible for this anonymity? In other words, what kind of data do we have inside of our system – anonymous location info right from the moment data being put into our system or it is just a view and our data saved internally in the raw formats?

Of course, the deployment of location privacy methods depends on the tasks our system is going to target. For example, obfuscating location information in case of emergency help system could not be a good idea. But, from other side many geo-context aware applications (e.g., geo search) can use approximate location info.

Also we need to highlight the role of identity in LBS. It looks like combining identity with location info is just an attempt for delivering more targeted advertising rather then the need of the services themselves. It is obvious, for example, that local search for some points of interests (e.g., café) should work for the anonymous users too.

More traditionally, peer-to-peer LBS refer to the way sharing information is traversed over the network [18]. For example, the peer-to-peer k-anonymity algorithm has several steps: select a central peer who will act as an agent for the group, next, the central peer will discover other k-1 different peers via single-hop or multi-hop to compose the group, and finally find a cloaked region covering all locations that every peer may arrive.

In our article, we are using "peer-to-peer" term at the first hand for highlighting the target party for the location-sharing request. It is "another peer" directly, rather than the central server (data store).

In terms of patterns for LBS, this approach targets at the first hand such tasks as 'Friend finder' and the similar. In other words, it is anything that could be linked to location monitoring.

III. GEO MESSAGES

Our idea of the signed geo messages service (geo mail, geo SMS) based on the adding user's location info to the standard messages like SMS or email. Just as a signature. So with this service for telling somebody 'where I am' it would be just enough to send him/her a message. And your partner does not need to use any additional service in order to get information about your location. All the needed information will be simply delivered to him as a part of the incoming message.

It is obviously peer-to-peer sharing and does not require any social network. And it does not require one central point for sharing location with by the way. Our location signature has got a form of the map with the marker at the shared location. And what is important here – the map itself has no information about the sender and recipient. That information exists only in the message itself. The map (marker) has no information about the creator for example. That is all about privacy [9].

Signed geo message service offers a mobile web mashup that lets users add a signature with geo information to the standard messages (SMS, email). As any other signature, our signature is just a text. And this text simply contains a dynamically generated link that leads either directly to the mobile map or to some landing page where mobile map is a part of it. And that mobile map (visible area) shows the current position of the sender. It is where the name of service is coming from – signed geo messages or geo signatures. The Figure 2 illustrates this approach. This figure illustrates Geo Mail client. Geo Messages approach has been implemented as a series of web mashups. All of them are based on the same principles. For example, Geo Mail mashup detects user's location (via W3C Geo location, supported in HTML5 browsers) and lets user choose the format of the signature (static or dynamic map, just a pair of latitude, longitude). By the similar principles we can share location info via SMS (it could be based on SMS URI scheme), Facebook Messages, Twitter's direct messages, etc.

The interesting moment that should be mentioned here, is the technical ability to implement such approach on the level of SIM-card. In other words, it could be provided as a standard feature by telecom operators. As it is described in [9], it could be implemented via Smart Cards Web Server Servlet that requests local info. This servlet can perform

 ISBN: 978-1-61208-262-2

proactive command for getting Location Information (MCC, MNC, LAC and Cell Identity). Such approach let us provide web mashups for smart cards. Obtained location info could be processed by the external web service. External service will convert Cell Id data into location data (latitude, longitude) and use obtained information for signing outgoing requests.

Figure 2. Geo Mail client

Because we can use Geo Messages as web mashups, there is no need to download (install) some applications in order to share location info. Of course, there is a common problem with mobile users that are not aware about Geo Email at all. One viral trick suggests modifications for Geo Mail, so the modified version sends a link to Geo Mail application instead of the question 'Where are you?'. For example, a signature for Geo Mail can include a link to Geo Mail itself. It lets the recipients answer to the question 'Where are you?' just by the opening Geo Mail link.

The next step is Geo Mail integration with Address Book on mobile. This feature at thi moment could be implemented via mobile application only. Currently, it exists as Android app. We are looking a way for creting portable version for it.

IV. WATN

Geo Messages approach works and really eliminates the problems with identity revealing. It provides a good solution for 'Where Are You Now?' question, but in the same time is not very convenient for the constant monitoring with several participants. It is simply not very convenient to jump from one message to another. WATN ('Where Are They Now') [1] application solves this problem. It provides a new peer-to-peer service that solves the privacy issues and lets you deal with several location-related feeds (location peers) simultaneously.

In Geo Messages approach the standard header for messages (e.g., 'From' header field for email, phone number for SMS, etc.) has been used for the identification. There is no own identification in Geo Messages. For multiple location-related feeds, we need to identify (distinguish) them by some way.

WATN provides own identification scheme, but in the same time it separates locations and identity. Actually, it is the basic idea behind WATN. In other words, rather than use one server that keeps all our data in some centralized system (like Google's Lattitude), we will switch to some mashup of distributed and centralized architectures.

We can separate location info and identity data just in three steps:

a) assign to any participant some unique ID (just an ID, without any connection to the personality)

b) save location data on the central server with links to the above-mentioned IDs

c) keep the legend (descriptions for IDs, who is behind that ID) locally

In this case, any participant may request location data for other participants from third party server (as per sharing rules, of course), get data with meaningless IDs and map them against locally saved database with names. With such replacement we can show location data in the "natural" form (replace meaningless IDs with mnemonic names). And in the same time, our server (third party server for our users) is not aware about IDs decrypting.

Obviously, that in this model each client keeps own legend info. And because our clients are not aware about each other and there are no third party servers that know all registered clients. It means, obviously, that in this model the same ID may have different legends. Technically, each client can assign own name (nick) for the same ID. Our social graph saves information (links between participants) using the above-mentioned meaningless IDs only. And the human readable interpretation for that graph can vary from client to client.

The next basic moment is the implementation for our distributed system. The local sub-part has been based on HTML5 standards. There is so called local storage specification [19]. As per W3C documents, HTML5 web storage is local data storage, web pages can store data within the user's browser.

The concept is similar to cookies, but it's designed for larger quantities of information. Cookies are limited in size, and mobile browser can send them back to the web server (use them as a part of HTTP requets). Web Storage is more secure and faster and our data is not included with every server request, but used only when asked for. It is also possible to store large amounts of data, without affecting the website's performance. The data is stored in key/value pairs, and web pages can only access data stored by them. In other words, Web Storage follows to standard same-origin policies.

Storage is defined by the HTML5 standards as this:

interface Storage {

 ISBN: 978-1-61208-262-2

```
    readonly attribute unsigned long length;
    DOMString key(unsigned long index);
    getter DOMString getItem(DOMString key);
    setter creator void setItem(DOMString key, DOMString value);
    deleter void removeItem(DOMString key);
    void clear();
};
```

The DOM Storage mechanism is a means through which string key/value pairs can be securely stored and later retrieved for use. The goal of this addition is to provide a comprehensive means through which interactive applications can be built (including advanced abilities, such as being able to work "offline" for extended periods of time).

User agents must have a set of local storage areas, one for each origin. User agents should expire data from the local storage areas only for security reasons or when requested to do so by the user. In our projects Local Storage keeps identification data. It is a typical key-value system. Key is user's ID, and value is user's name (nick, alias).

Now we are ready to describe the whole algorithm. Each user automatically obtains own ID. For the first time visitors, web service generates a new ID, for the returning users ID will be extracted from user's local storage. As soon as ID is obtained, user can share his location information. There is no registration system, so "share location" requests could be addressed to any person with valid email address. Technically, "Share my location" request is just an ordinary email with the link to WATN service. This link contains an ID for the person who is going to share location. It is an ordinary email (or SMS) and WATN service is completely unaware about the target address. Actually, it is one of the main features. The communications in WATN are completely separated from the service.

As soon as the email is received, the recipient can open the shared link. It is a new request (hit) to WATN service. As any other request to WATN it caused user's ID detection (assign a new one or select existing from local storage – see above). It means that for such kind of requests ('share location') the service collects two IDs. One of them is user's ID, and the second one is passed as a parameter. It shows who is sharing location info now.

As the next step for processing 'share location' request, WATN service asks recipient for two things:

a) accept (do not accept) request
b) assign a new name (nick name, alias) for ID from accepted request

This name choise is completely up to the recipient. Conceptually, it is based on the fact that 'share location' request was opened right from some message (email, SMS). So, the target party in 'share location' request (email's recipient) is aware about the context. Simply, he knows who is an author. He can see email's header ('From' inormation) or phone number (address book info) for SMS. Based on this information, the recipient can choose some name for this correspondent. It is illustrated in Figure 3.

Here is a small additional technical trick. Of course, any name could be assigned. But for names, that corresponds to Twitter's (Facebook's) account it is possible to pull profile picture also. Note, that we cannot use some services like Gravatar, because the email address is unknown for WATN service.

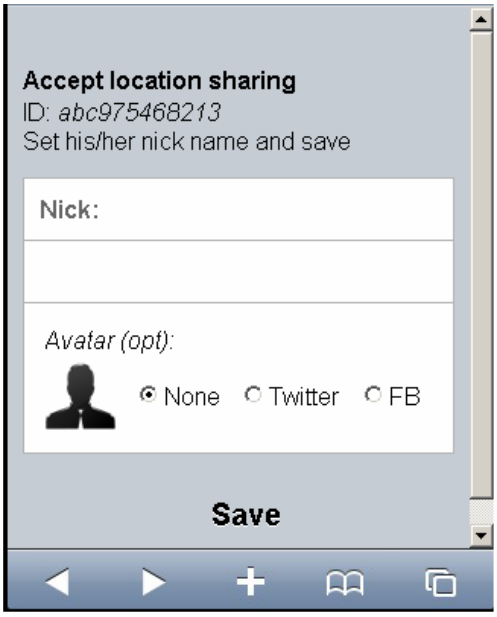

Figure 3. Share location

It explains the process of collecting (filling) local identification database. Each user simply assigns own names (nicks) to IDs provided in 'share with you requests'.

In the same time, if 'share location' request is accepted, we can fix this fact on the server. We have two IDs and can use them for creating social graph record: ID_1 shares location info with ID_2

This information is centralized, but it completely dumb. Our server is completely unaware about the names behind IDs. 'Share location' processing works like a typical two phase commit in distributed database. It saves social graph info on the server and saves identification info locally. It is how our distributed database within WATN is organized. It has centralized store for social graph (who is sharing location info with whom) and local store with identity data.

Now we can go back to the whole algorithm. Lets us return to the first step. As soon as any user hits WATN web site, mashup detects his ID. Having ID, we can pull data from the social graph. Actually, WATN engine returns extracted social graph info as JSON array. At the first hand, this array contains a list of IDs for participants who shared location info with the given user. Simultaneously, mashup records a new check-in (saves location info) for the current user. This information is anonymous again and contains meaningless ID, time and geo coordinates. Check-ins let us detect the current (more precisely – last known) location for the each participant. It let us add location into to the above mentioned extraction from the social graph. So, WATN

 ISBN: 978-1-61208-262-2

engine returns to the client IDs and locations. This JSON array could be mapped (on the client side) against locally saved identity info. This mapping replaces IDs with saved names (nicks). And this information could be simply visualized (Figure 4):

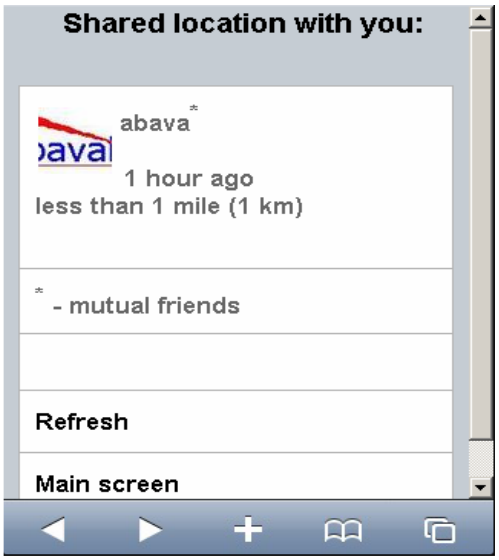

Figure 4. WATN information screen

On this screen ID for the user has been replaced with the mnemonic name (abava) and avatar fom Twitter.

V. CONCLUSION

This paper discussed several new models for sharing location information without disclosing identity data to some third party server. We described existing approaches as well as propose several new implementations. All proposed services share the common background idea about the separation the location and identity data. They could be described as a safe location sharing. It combines server-side (centralized) location info with the locally-based distributed identity info. In this distributed data store identity info is always either saved locally or borrowed from the external services (e.g., messages). The proposed approach eliminates one of the biggest concerns for location based systems adoption – privacy.

REFERENCES

[1] D. Namiot and M. Sneps-Sneppe, "Where Are They Now – Safe Location Sharing. A New Model for Location Sharing Services, Internet of Things", Smart Spaces, and Next Generation Networking, Lecture Notes in Computer Science, 2012, Vol. 7469/2012, pp. 63-74, DOI: 10.1007/978-3-642-32686-8_6

[2] G. Schilit and B. Theimer, "Disseminating Active Map Information to Mobile Hosts", IEEE Network, 8(5) (1994) pp. 22-32

[3] N. Hristova and G. M. P. O`Hare, "Ad-me: Wireless Advertising Adapted to the User Location, Device and Emotions," in Thirty-Seventh Hawaii International Conference on System Sciences (HICSS-37), 2004, pp. 1-10

[4] M. Scipioni, "A privacy-by-design approach to location sharing." Proceedings of the 2012 ACM Conference on Ubiquitous Computing. ACM, 2012, pp. 580-583

[5] L. Palen and P. Dourish, "Unpacking "Privacy" for a Networked World", CHI Letters, 2003. 5(1): pp. 129-136.

[6] J. Hong and J. Landay, "An Architecture for Privacy-Sensitive Ubiquitous Computing", MobiSys'04, Jun. 6–9, 2004, Boston, Massachusetts, pp. 177-189

[7] D. Wagner, M. Lopez, A. Doria, V. Kostakos, I. Oakley, and T. Spilitiopoulos, "Hide and seek: location sharing practices with social media", Proceedings of the 12th international conference on Human computer interaction with mobile devices and services, September 07-10, 2010, Lisbon, Portugal, pp. 55-58

[8] J. Tsai, P. Kelley, P. Drielsma, L. Cranor, J. Hong, and N. Sadeh ,"Who's viewed you?: the impact of feedback in a mobile location-sharing application", Proceedings of the 27th international conference on Human factors in computing systems, April 04-09, 2009, Boston, MA, USA, pp. 2003-2012

[9] D. Namiot, "Geo messages", Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2010 International Congress pp. 14-19 DOI: 10.1109/ICUMT.2010.5676665

[10] D. Namiot and M. Sneps-Sneppe "Customized check-in Procedures ", Smart Spaces and Next Generation Wired/Wireless Networking Lecture Notes in Computer Science, 2011, Volume 6869/2011, pp. 160-164, DOI: 10.1007/978-3-642-22875-9_14

[11] Y. Daradkeh, D. Namiot, and M. Sneps-Sneppe, "Spot Expert as Context-Aware Browsing", Journal of Wireless Networking and Communications, vol.2, N. 3, 2012, pp. 23-28

[12] D. Namiot, "Context-Aware Browsing -- A Practical Approach", Next Generation Mobile Applications, Services and Technologies (NGMAST), 2012 6th International Conference on, pp. 18-23, DOI: 10.1109/NGMAST.2012.13

[13] M. Duckham and L. Kulik. A FORMAL model of obfuscation and negotiation for location privacy. In Proceedings of Pervasive 2005, Munich, Germany, 2005. Springer. pp. 152–170

[14] J. Krumm, "Inference attacks on location tracks", In Proceedings of the Fifth International Conference on Pervasive Computing (Pervasive), volume 4480 of LNCS, Springer-Verlag, 2007, pp. 127–143

[15] C. Chow, M. Mokbel, and X. Liu, "A peer-to-peer spatial cloaking algorithm for anonymous location-based service", GIS '06 Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems, 2006, pp. 171 – 178 DOI:10.1145/1183471.1183500

[16] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In MobiSys '03: Proceedings of the 1st international conference on Mobile systems, applications and services, New York, USA, 2003., pp. 31–42

[17] M. Langheinrich, "Privacy in ubiquitous computing", Ubiquitous Computing, CRC Press, 2009, pp. 95–160.

[18] J. Xu, J. ZpP, M. Xu, and N.Zheng, "Mobile-Aware Anonymous Peer Selecting Algorithm for Enhancing Privacy and Connectivity in Location-Based Service", e-Business Engineering (ICEBE), 2010 IEEE 7th International Conference on Nov. 2010 pp. 172 – 177 DOI: 10.1109/ICEBE.2010.32

[19] M. Casario, P. Elst, C. Brown, N. Wormser, and C. Hanquez, "HTML5 Solutions: Essential Techniques for HTML5 Developers", 2011, pp. 281-303, DOI: 10.1007/978-1-4302-3387-9_11

 ISBN: 978-1-61208-262-2