Towards a Position Sharing Approach for Location-based Services

Frank Dürr, Marius Wernke, Pavel Skvorzov, Kurt Rothermel
Institute of Parallel and Distributed Systems (IPVS), Universität Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany
{frank.duerr,marius.wernke,pavel.skvorzov,kurt.rothermel}@ipvs.uni-stuttgart.de

***Abstract*— In partially-trusted system environments like the WWW, the management of private user positions is a great challenge. On the one hand, location-based services of different partially-trusted providers have to be provided with position information. On the other hand, location servers, which are responsible for the management of user positions, might as well be operated by only partially-trusted providers. Therefore, in order to protect his privacy, a user might only want to store position information of limited precision on location servers, and also provide location-based services with position information of limited precision.**

In this position paper, we sketch a novel position sharing approach that enables the user to tightly control the precision of position information stored on servers and provided to location-based services. This approach is based on the idea of distributing position information among a set of servers of different providers such that a compromised server only reveals information of strictly limited precision.

I. Introduction

Location-based services are becoming more and more popular in today's World Wide Web. This development is driven by the availability of powerful mobile devices, in particular smart phones like the iPhone or mobile phones based on the Android platform, with integrated positioning systems like GPS or network-based positioning technologies like cell ids. Moreover, location-based services are enabled by fast communication technologies and the availability of cheap flat rates. Therefore, mobile clients can provide location-based services with their current geographic position virtually anywhere at at any time.

Obviously, handling private position information of users is a critical issue in particular in open system environments like the WWW with many providers that are only partially-trusted by the user. In particular, we can identify two privacy-critical components. First, *location-based services (LBS)* have to be provided with user position information in order to implement a certain functionality. For instance, a point of interest (POI) finder might use the user's position to display relevant POIs on a map. A more advanced example is a friend finder service that determines all friends in a user's vicinity. Many such LBS might exist in the future that are offered by different service providers. Depending on the trust of the user in the service provider, a user might only be willing to provide the service with more or less precise position information in order to protect her privacy. Moreover, different LBS might have different precision requirements. Therefore, in terms of privacy it is reasonable to provide an LBS with the minimum possible precision that still allows for a decent quality of service. Certainly, different LBS have different requirements, and a user has different trust in different LBS providers. Therefore, it should be a goal to provide LBS with position information of individual precision.

The second privacy-critical component is the *location server (LS)*, which manages the dynamic positions of mobile users. Actually, in simple LBS an LS might not be required since the user position can be managed by the mobile client. Consider for instance the POI finder above. Since a user might explicitly query the POI finder, for instance by using a web-browser, the user can directly send her position to the LBS. At the same time, she can decide about the precision of the position information sent to the LBS. For such simple LBS that only deal with single user positions the W3C Geolocation API [1] is well-suited since it already supports imprecise position information. The browser only has to present the user with a suitable dialog that displays the LBS provider who wants to access the location information, and offer an option to the user to send imprecise information rather than her exact position. However, more advanced LBS need support for more complex functionality that requires an LS in order to be implemented efficiently. In particular, an LS can proactively manage the positions of multiple users and offer advanced query interfaces beyond simple queries for the position of a certain user. As an example consider the friend finder LBS above. In order to find friends in the user's vicinity, a spatial range query can be used that queries for all friends in a certain area. Besides queries, an LS can also support spatial events. For instance, a friend alert service could be implemented by registering an on-meeting-event that notifies a user whenever she gets close to a friend. Multiple LS already exist today, for instance Google Latitude [2], Yahoo Fire Eagle [3], or Trace4You [4]. Since LS are usually offered by third-party providers, the same problems as for third-party LBS arise, i.e. the user might not be willing to store her precise position information on a server of partially-trusted provider that might not be willing or able to protect her data against misuse or theft.

Therefore, we argue that new concepts are required for protecting private position information in such infrastructures of partially-trusted LBS and LS providers. These concepts must *enable the user* to control the access to his private position information rather than relying on the security mechanisms of the LS, which might get compromised by an attacker or a malicious provider. In this position paper, we introduce the concept of *position sharing*. The basic idea of this approach is to decompose an exact user position

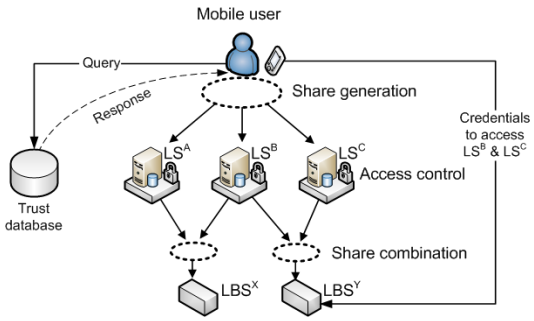

Fig. 1. Position Sharing System Model

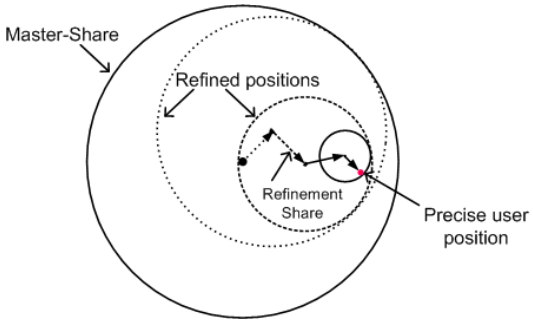

Fig. 2. Share Combination

into position shares of user-defined limited precision. These position shares are then distributed among a set of LS of different providers depending on the trust level of an individual provider. That is, a provider with a higher trust level might get more precise shares than a provider on a lower trust level. With that approach, a compromised LS only reveals position information of strictly limited precision, and therefore limits the impact of attacks. Moreover, position sharing allows for the combination of multiple shares into information of higher precision. In order to provide LBS with position information of a certain precision, it gets rights to access a certain set of position shares from different LS. This set of shares can be configured individually by the user in order to account for the different quality requirements and trust levels of different LBS. The main contributions of this paper are: (1) An architecture for position sharing in partially-trusted environments. (2) A sketch of an algorithm for secure share generation and combination.

The rest of this paper is structured as follows. In Section II we present the architecture of our position sharing system. In Section III we outline an algorithm for share generation and combination, before we conclude this paper with a short summary and outlook onto future work in Section IV.

II. System Model and Components

Our system model consists of four different components: A *mobile user*, *location server (LS)*, *location-based services (LBS)*, and a *trust database*. All these components are described in this section and shown in Figure 1:

The *trust database* calculates users trust in different LS providers and LBS providers. The trustworthiness of these components can be ranked by a trust model as presented in [5]. The database answers queries with the trust value for the queried components.

The *mobile user* uses location-based services and therefore stores his position on location servers. The mobile user is aware of his current position, determined by a positioning system like the global positioning System (GPS). By querying the trust database, the trust values of the LS that should store position shares are determined. The exact user position is then used to create the required position shares by using the share generation algorithm described in Section III. The created shares are distributed to different LS, depending on their trust values. The trust value can also be used to specify the number of shares the LS has to store. A server on a higher trust level might store more position shares or more precise position shares than a server on a lower trust level.

A *location server* is responsible to manage the position shares received from the mobile client. To protect user privacy, only LBS with the corresponding permissions can access the predefined position shares of the user. The access control is managed by the LS.

The stored position shares from the LS are queried by *location-based services*. Therefore, a LBS requires user's credentials to get access to the corresponding LS. The queried shares can then be used to increase the position precision. This is done by the share combination algorithm presented in the next section.

III. Share Combination and Generation Algorithm

The current position of a user can be expressed by a circle that contains the real user position. This is a widely known approach to increase the uncertainty about the true user position. The circle is defined by the center and the corresponding radius. The size of the radius depends on the privacy preferences of the user and the trust value of the LS storing the current position. A circle that could be revealed without raising privacy concerns represents the user's *master share* of the current position. This coarse position can be refined by additional *refinement shares*. Each refinement share describes a shift of the center and a reduction of the radius. The combination of the refinement shares is done by vector addition until the exact position of the user is determined or no further refinement shares could be collected by the LBS. Therefore it is possible to regulate the position precision which a LBS is allowed to collect by granting access to a predefined number of shares on different LS. The combination of four refinement shares that reduce the inaccurate position description to the precise user position is shown in Figure 2.

For share generation it is required to randomly select shares in an unpredictable manner. Currently, we are analysing the security of different share generation algorithms that make it impossible to an attacker to predict additional shares without having the required access rights.

IV. Summary and Future Work

In this position paper, we sketched a novel position sharing approach that enables the user to tightly control the precision of position information stored on location servers and provided to location-based services. This approach is based on the idea of distributing position information in form of position shares among a set of location servers of different providers such that a compromised server only reveals information of strictly limited precision. Location-based services can be provided with position information of configurable precision by combining position shares from different location servers. This allows for a very flexible configuration of the level of privacy to be achieved.

As part of our future work, we will elaborate on the share generation algorithm outlined in this paper. In particular, a detailed analysis of the share generation algorithm is required to formally prove the security of the approach. Moreover, suitable privacy metrics for position information are required that enable a user to asses and configure the achieved and desired level of privacy, respectively.

References

[1] W3C, "Geolocation api specification," http://www.w3.org/TR/geolocation-API, July 2010.
[2] Google Latitude, www.google.com/latitude, July 2010.
[3] Yahoo!, "Fire eagle," http://fireeagle.yahoo.net, July 2010.
[4] FALCOM, "Trace4you," http://www.trace4you.com, July 2010.
[5] A. Gutscher, "A Trust Model for an Open, Decentralized Reputation System," in *Proceedings of the Joint iTrust and PST Conferences on Privacy Trust Management and Security (IFIPTM 2007)*, August 2007.