A Privacy-Enabled Architecture for Location-Based Services

Martin Werner

Mobile and Distributed Systems Group
Institute for Informatics of LMU Munich, Germany
martin.werner@ifi.lmu.de,
Web: http://www.mobile.ifi.lmu.de/

Abstract. Location-Based Services are emerging fast and the problems with privacy are growing with them. While a platform for Location-Based Services can provide the user with high-quality Location-Based Service browsing and powerful mechanisms to reduce the amount of location data transmitted such a platform is dangerous as it has to manage the location data of the users and the actual service usage. This aggregation of private data is a risk in itself. With this paper we want to show that it is possible to implement most Location-Based Services without such a platform and propose a mechanism enabling fine- grained control of privacy for a Location-Based Service user. We make use of strong cryptographic techniques to enable a real trust relation between individuals and a weaker trust relation between an individual and a company.

Key words: Location-Based Services, Privacy

1 Introduction

Location-Based Services (LBS) are services which depend on the location and other context information (such as the time, weather, environment ...) of the user. This type of service is becoming more and more common for mobile use as most new cellular phones have a GPS chip enabling cheap usage of location information. The most important benefit of Location-Based Services is that a user of a mobile device only gets informations and services relevant for his position. This is very handy compared to searching for some service (e.g. italian food) or a specific information (e.g. wheather) in a classical web search due to the difficulties of typing with an on-screen keyboard.

Currently there are platforms emerging which allow anyone to generate a Location-Based Service without any programming by describing the service in some specified form. One such example is Aloqa [1] which in essence is an intelligent Location-Based Service browser. For the Aloqa case the information is organized in channels which the user can subscribe. These channels include public transport information, restaurants, concerts, health-care services and many others.

All such platforms currently work as an intermediator in the sense of privacy. They collect - on a per user or per session basis - position data of users along

with a description of their interests and then present such users with Location-Based Services which might attract them. This type of platform is very easy to implement and is easy to exploit commercially. The results of user profiling is of great commercial interest and the platform can provide advanced personalized advertising.

Though it is evident that the usage of any Location-Based Services always reveals private information to the provider of the service (e.g. some sort of location data) it is very dangerous to design systems which collect service data and user tracking data. The problem is not the absense of trust in the platform design but that the data of such a platform will be of great interest to traders as well as criminal organisations and that there is a correlation of danger and success in the sense that once such a system is successfull and collects more and more users it will not be able to guarantee for the privacy of the data.

The main argument for the introduction of LBS-platforms is that the platform can be used to intelligently reduce the amount of data transmitted and hence to safe battery life. While this is true for some LBS (e.g. k-next-neighbour) there is no need for a platform in a one-one LBS-connection (e.g. the user and the provider). The calculational power and intelligence of the mobile platform suffices to perform most of these optimizations on the mobile device.

As the usage of Location-Based Services always needs a trust relation, platform provider argue that the the disclosure of location information to a specific known institution (the platform) can be acceptable. We want to propose alternative (though more complicated) ways to provide Location-Based Services in a manner which only needs a trust relation to the actual service itself (e.g. the coffee company or the actual friend I want to use a Friend-Tracker with). This implies some restrictions on the type of service which can be provided in this way but still the introduction of some platform only for those services which really need a platform is possible.

The mechanism we have in mind uses strong cryptographic technology to protect the private data from collection by platforms or carriers. The main idea is based on the observation that a user typically uses only a very small subset of available Location-Based Services and only allows very few of them to proactively notify him. As a result we can allow us more complexity in the way we exchange location information.

The rest of the paper is organized as follows: In the following section we describe two scenaria which we have in mind. The one is a Friend Finder which shall proactively notify the user about friends which are near (in a configurable distance). The second scenario is a Coffee Company which wants to promote his offerings with a coupon service spreading several location based discounts.

2 Scenaria

The following two scenaria show two very different types of Location-Based Services. The Friend Finder is a very private service where the location of users has to be tracked and exchanged permanently and in great detail. The second service

only needs to exchange location information on service invocation. Furthermore the granularity of the location information is not important for the service to work. If the service is only presented with a coarse location (e.g. zip code) it will still work presenting the application with a list of possibly interesting locations which can then be checked locally - on the users phone - for their real distance and importance.

2.1 Friend Finder

Assume Bob and Alice are friends. They went to school together and now live in two different cities. They both work for big companies and travel very often. As a result they often find out when they meet that they have been in the same city at the same time and just missed a possility to meet. They would like to have a Location Based Service notifying them when they are in the same city at the same time. But as they are careful about their privacy, they do not want to expose their location information to anyone else except each other.

2.2 Coffee Company

Assume a coffe company wants to advertise with location based discount coupons. They want to have a simple way to inform interested customers about discounts on their offerings. For simplicity they do not want to provide a location based service in several special ways for different Location-Based Service platforms but in a generic way through their web-page. They want to provide a web page which one can send his approximate location information to and get a list of active discounts for this area specification. This web page accepts most usual descriptions of locations such as the zip code, a GPS coordinate, a cityname or a street and of course a distance limit. With this generic setup the coffee company can simply use the existing webserver infrastructure and is ready for Location-Based Services. The coffee company can even advertise for their new service with standard tools such as QR-codes showing the URL of the Location-Based Service web page. If this web page is opened in a browser which does not provide the location information it will just show as a standard HTML page where you can search for local discounts in a classical web search. In this way, they instantly support any mobile device equipped with a webbrowser.

3 Related Work

Many commercial Location-Based Services are arising today. Unfortunately the issues with privacy have been ignored in many cases. This is natural due to the fact, that most of the users do not know what data is exchanged and what data is stored in a non-anonymous way and hence accept applications for the individual service experience. We believe that the importance of privacy will grow in the near future when people realize that they are revealing very much information

about themselves to a party that can not guarantee for the protection of the data from abuse.

The privacy threats of Location-Based Services have been brought to public attention such as in the EU directive (2002/58/EC) [2] which essentially requests the explicit consent of a user before the position data is allowed to be processed. In practice such a law does not help much because a one-time acceptance of a checkbox during installation is enough to allow some platform to track and store any private data of its users. As it is not easy to construct better law it is important to inform people about the real danger that lies in using such location based services.

The case study [3] identified the following three important design issues which have to be addressed for good Location-Based Services:

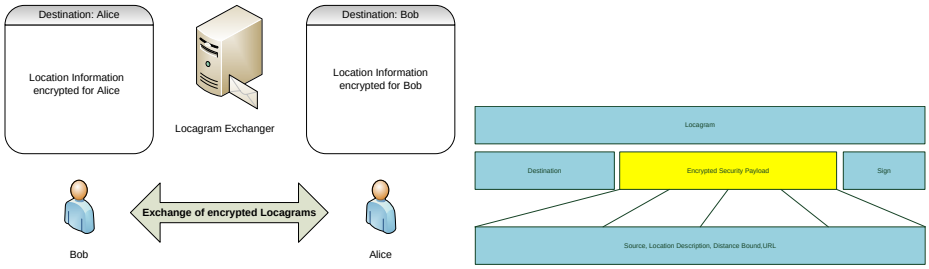
- It is essential that a system provides the user with real-time information about their level of privacy. The basic questions are who gets to know which private information about my position and context.
- Location-Based Services should enable easy short-term deactivation.
- Location-Based Services are more likely to be adopted in closed environments (co-workers or even bigger groups such as the students of a campus)

As most Location-Based Services share similar privacy concerns researcher have proposed several Privacy Enhancing Technologies (PETs) for LBS. Examples range from basic switches disabling the transmission of location information to more sophisticated systems such as area-based filter rules or mechanism related to k-anonymity. In the paper [4] the basic question whether PET's are used by people using LBS is answered positive. The bad news from this research is, that all PET's which need constant awareness of the users fail in practice [4, chapter 6]. Hence we conclude that it is important to protect as much of the location data as possible as it could be accidentally exposed (e.g. due to forgetting to disable a tracker). We will try to support this with mandatory strong end-to-end encryption.

With our generic framework for the exchange of location data we are able to support all of these issues and hence raise the acceptability of such Location-Based Services to a higher degree than current Location-Based Services.

4 Locagrams

A system which can provide users with full control about their privacy and the way they export their location information to LBS automatically needs a trust relation between the endpoints, namely between the user of a Location-Based Service and the Service itself. We decided to allow the usage of modern technologies such as strong encryption and some sort of microblogging to enable anonymous information exchange to the maximum extent possible. The basic LBS communication is done via so called Locagrams which stands for Location-describing Telegrams. These shall be short messages which are constructed as depicted in figure 1(b) from the following data:



(a) Locagram Exchanger (b) The general structure of a locagram

Fig. 1. Communication is done using a common known locagram exchanger

1. Destination Identifier
2. Time-To-Live
3. Source Identifier
4. Location Description
5. Distance bound
6. URL for further communication
7. signature

where 2) to 6) can be encrypted by some public key encryption and 1) to 6) should be cryptographically signed - the sign being stored in 7). In this way everything is protected against modification and all information except the destination description can be encrypted.

Figure 1(a) illustrates how locagrams shall be exchanged. They are stored on a server (Locagram Exchanger) and can be downloaded by the users. There is no need to introduce sessions or passwords or any identification here unless for scalability reasons or to prevent abuse of the exchanger. One could for example protect the locagram exchanger from abuse by using a captcha on the first connection and sharing a cookie for some time after correct captcha translation.

4.1 Destination Identifier

The Destination Identifier should be any unique identifier identifying the person which shall receive this locagram. It can but need not include information to contact the destination. Possible choices include a cryptographic public key, an account name or some synonym.

4.2 Time-To-Live

The Time-To-Live field contains an integer specifying the duration (in seconds) that a locagram exchanger is requested to keep a locagram. This field must be used by the locagram exchanger as the maximum time to keep a locagram. In this way a basic deactivation of the software leads to the removal of all location information within the time specified in this fields.

4.3 Source Identifier

The Source Identifier shall be the same type of identification as the destination identifier except that it shall describe the source of the message. In this way it is possible for the destination to answer to locagrams with another locagram.

4.4 Location Description

The location description should be a textual representation of the current location. It could contain one of the following information:

- WGS84-coordinate (possibly obtained from GPS or some coarse network localization)
- zip code
- address description(either complete or only a city name)

4.5 Distance Bound

A distance bound is introduced to describe the area for which the locagram is relevant. This is currently stored as a string and might contain either a floating point number in a predefined unit or a string containing a well-known unit string (e.g. "1 km").

4.6 URL for further commication

This field can contain any URL. We think of web URL's for allowing enterprises to export location based services in a simple way through their webpage and special values such as

- about:return indicating that the same locagram exchanger shall be used to answer to this locagram with another locagram
- phonecall:number indicating that in case of relevance the user should be prompted to call the given phone number
- sms:number indicating that the locagram can be answered with a SMS

4.7 Signature

The Signature field contains a cryptographical message authentication code for the complete Locagram protecting it from changes.

5 The usage of Locagrams

Now we want to describe how the information which can be exchanged using Locagrams can be used to implement both types of Location-Based services we described earlier in section 2.

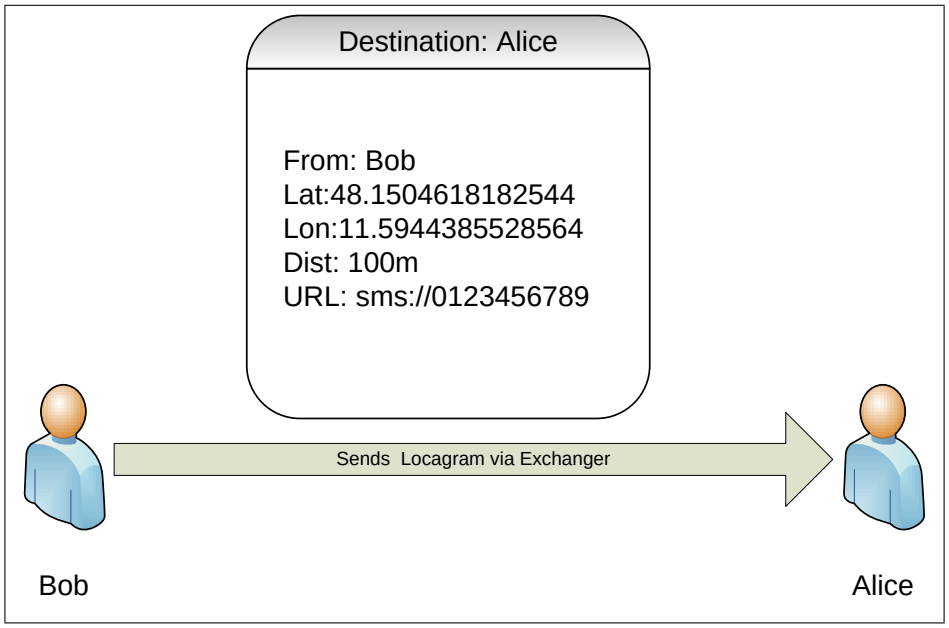


Fig. 2. A Locagram example for the Friend-Finder

5.1 A Friend-Finder with locagrams

For the Friend-Finder locagram implementation we use a server where we can store locagrams and recieve locagrams by identifier. This server of course has to be protected from spamming and denial of service attacks by some technology. There are very different ways to achieve something like that and as it depends very much on the actual implementation and the network location of the Locagram Exchanger, we do not want to go into detail with this.

Now the basic functionality of the Friend-Finder is the following: The Friend Finder has to be configured with the following data:

- Identifiers of Alice and Bob
- A public key of the friend
- The own private and public keys
- A Locagram Exchanger address

The key exchange is done via SMS or via QR-Barcode when the two friends meet. In the future we could also use NFC for key exchange. It is of course also possible to use any other mechanism to exchange keys, but we believe that our proposal is the best mixture of usability and security.

Now Bob activates its Friend-Finder. As Bob knows Alice (e.g. Alice's public key) Bob sends a locagram for Alice containing its current position and a given configured distance bound where Bob wants Alice to inform him. (An Example Locagram can be seen in figure 2). The URL is set such that Alice is notified if

near enough to Bob and told to send an SMS to Bob in reaction to a proximity event. In this way, she is able to contact Bob in the way Bob prefers: Proactively via SMS. Bob could alternatively have set the URL to the Locagram Exchanger in the Locagram's URL field. Then Bob's Friend-Finder would have to regularly check for incoming locagrams and notify him if a locagram answer is incoming within the given distance bound.

This is a very simple approach which can be optimized in many ways. The first optimization would be to let Alice answer to Bob's locagram with a locagram of it's own position and distance information such that Alice and Bob have the possibility to estimate the time until they will update each other with location information. If the distance between Alice and Bob is big, it is not important for the Friend Finder to exchange location information. Another important optimization at this point is to start with Locagram Exchange with very coarse location information and only if they do not conflict to send finer locagrams. In this way we can even safe battery by using some localization mechanism which is more efficient than enabling the GPS receiver.

To allow even more privacy, Alice and Bob could configure HTTP proxies to exchange locagrams with the locagram server or even setup their own locagram server on their private home page. It is of course also possible to use existing internet technologies such as a microblog (e.g. Twitter) or the Internet Relay Chat for the exchange of locagrams.

If the Friend-Finder is implemented in this way, we enhance the privacy of Alice and Bob in several ways. The first enhancement is, that no one except Alice and Bob can get any location information as it is encrypted per default. Another enhancement is, that all information is kept physically on Alice and Bob's devices. So no one will have any interest in collecting locagram data. Moreover the system does not have a central element (such as a platform) but can use any distributed data exchange mechanism (Microblog, Internet Relay Chat, ...).

5.2 The Coffee-Company Location Based Voucher System

But how can a coffee company export a location based service within this framework? The coffee company exports its location based service using a web page which takes location information, a user identification and public key and in answer to the request sends a list of possibly applicable locagrams.

Assume Bob wants to get a coffee. It then actiates the Coffee-LBS for its actual position. Bob has configured its location based service browser to export only the city-name to the coffee-company and thus the application starts with mapping its actual GPS-coordinate to city names (which Bob - as he really likes privacy - has downloaded a list for). Bob then provides the Coffee-Shop's locagram page with this city name and in turn gets many locagrams as answers. Which locagram is the most applicable for him now can then be checked locally on his phone and a map application is opened showing the coffee shops and vouchers.

The main advantage of such an approach is, that the data is exchanged only between the actual service provider and the service user. And due to the integration of web publishing for Locagrams, we achieve a simpler integration into existing infrastructures.

6 Prototype

The usage of locagrams as described before is a very generic approach to implementing a location based service. We did not and do not want to specify the usage of one specific communication system. For our prototypical implementation we decided to use a web-page as the locagram exchanger and some string as the destination identification. If one would use the public key as the identification one could extend the locagram exchanger to check the identity of the requesting person. For readability we however decided to keep the names Bob and Alice (or anything else you configure).

It is possible for the user to generate a new key-pair for every invocation of the coffee-shop service and hence being very private. It would be nice if the locagrams would not have to be stored somewhere but transported directly to the clients. We did not implement something like that, because at the time being cellular service providers (at least in germany) do not allow much more types of traffic than HTTP-client requests. We also hope that at some day a cellular service provider could allow the direct exchange of locagrams via SMS or similar service.

Our prototype consists of four components: A *Locagram Exchanger* implemented as a web page, a *LBS Service* consisting of some Java classes implementing the basic LBS functionality and a *Friend-Finder LBS* and a *Coffee-Company LBS web-page*. The Locagram Exchanger and the Coffee-Company LBS web-page have been implemented in PHP while the LBS Service and the Friend-Finder have been implemented in Java using a basic and simple RSA implementation for encryption. Unfortunately the Java Crypto API does not allow for generation of public / private key pairs and hence we use our demonstrative implementation. This implementation is weak and does only serve as a proof of concept.

6.1 The LBS-Service

The LBS service is implemented as a Java class which holds a list of different locagram exchanger URLs and some basic configuration for polling locagram exchanger.

The LBS-Service exports the basic functionality of

- addition and removal of locagram exchangers
- query/poll locagram exchangers for new locagrams
- Generate and Exchange Public Keys via SMS and QR-Code
- Manage a storage of public keys

The list of locagram exchangers and the list of public keys is stored in seperate files. Due to the simplicity of data we do not need the functionality of a SQLite database. The file formats are basic CSV and no escaping is done. This leads to the restriction that the identifier do not contain a semicolon.

7 The Friend-Finder

The Friend Finder is implemented as an Android application and works just as described in the basic scenario. On startup, it shows a menu where you can either start a key exchange via SMS or QR-Code or run the Friend-Finder. Once running, the Friend-Finder contacts the configured Locagram Exchanger and requests locagrams using the LBS-Service. Once it receives a locagram, it will check, whether the distance and position apply and then will inform the user. It also contains a list of friends which are regularly informed about the position by sending out locagrams.

8 The Coffee-Company LBS Web Interface

The Web Interface for the Coffee-Company LBS Web Interface is implemented as a website which needs the following text variables and returns a status code to the client requesting a service.

- **ident** which is the identification to be used as the destination of the locagram
- **pubkey** which is the public key of the client requesting the locagrams
- **position** which is some description of the position as explained before
- **distance** which is the distance limit for which locagrams shall be received
- **locagramexchanger** which is the URL of the locagram exchanger to be used

This website will then send the locagrams via the given locagramexchanger or - if the locagram exchanger is not given - on the webpage itself. It can return status codes which might indicate that it refuses to encrypt data (e.g. for scalability reasons), that there were no results to the given search or that other error conditions occured. In absence of errors the locagrams will be transmitted to the given locagram exchanger where the user can download them and analyze them further.

9 Outlook

With this paper we have presented a new approach to protecting privacy in LBS which does remove the need of a trusted party intermediating between the service provider and the service user. This is important because a market-place for Location-Based services which manages the tracking of clients as well as the service discovery can collect too much private information. It is of course clear,

that a location based service user does expose location information but it does not make sense to expose this information to anyone else but the entity providing the actual service.

This approach can be extended to cover almost any type of location based service and reduces the amount of private data exchange to an absolute minimum. It is even possible to generate a new key-pair for every invocation of e.g. the Coffee-Shop-Service. It now depends on the user to decide who gets what information in what granularity. With this framework it is even possible to implement Location-based Services in closed high-security environments.

Especially in situations where the usage of GPS and network does not imply problems with power consumption (e.g. in a car with a navigation system) the usage of locagrams can very efficiently protect the privacy of the user against e.g. a gas station operator.

References

1. Aloqa GmbH, http://www.aloqa.com/ (2010)
2. European Parliament: Directive 2002/58/ec of the european parliament and of the council. (2002)
3. Barkhuus, L.: Privacy in Location-Based Services, Convern vs. Coolness. Proceedings of Workshop paper in Mobile HCI (2004)
4. Burghardt, T.: Understanding User Preferences and Awareness: Privacy Mechanisms in Location-Based Services (2009)