Selective Location Blinding using Hash Chains

Gabriele Lenzini[1], Sjouke Mauw[1,2], and Jun Pang[2]

[1] Interdisciplinary Centre for Security, Reliability and Trust,
University of Luxembourg
[2] Computer Science and Communications, University of Luxembourg

Abstract. Location-based applications require a user's movements and positions to provide customized services. However, location is a sensitive piece of information that should not be revealed unless strictly necessary. In this paper we propose a procedure that allows a user to control the precision in which his location information is exposed to a service provider, while allowing his location to be certified by a location verifier. Our procedure makes use of a hash chain to certify the location information in such a way that the hashes of the chain correspond to an increasing level of precision.

1 Introduction

The main challenge faced in security protocol design concerns the need to satisfy a number of conflicting security requirements. In the domain of location-based services, this conflict shows in the tension between location assurance and location privacy. On the one hand, service providers must know their clients' location with some level of assurance while, on the other hand, clients do not want to expose more location details than strictly needed for the requested service.

A second factor complicating the design of a security protocol is that its functional requirements and the assumptions concerning the system strongly depend on the intended usage scenario of the protocol. This factor clearly shows in location-based service scenarios, each of them leading to essentially different solutions. In the following, we briefly exemplify three of such scenarios.

A common usage scenario is that of *e-tolling*, in which the location of a car is periodically sensed and used to determine the amount of toll due. In this particular scenario, which gave rise to solutions like VPriv [1] and PrETP [2] assumptions dictate that location data must be stored at the car for offline usage. However, for online usage data must be stored at the server, on the condition that they are used by this service only. Location integrity requires either tamper resistant hardware (a solution which preserves drivers' privacy) or on-road checking spots for a-posteriori assurance (which weakens privacy). Location privacy suggests that cars can hide in the crowd of other vehicles, but service providers need to have enough information to bill correctly.

Another scenario is the use of positioning information to inform the client of the availability of services near his current location, such as the nearest Chinese restaurant or the closest gas station for which the user has a fidelity card. In this scenario location-spoofing by the client does not harm the service, allowing protocol designers to mainly focus on the associated privacy concerns.

Yet another scenario is that a client makes use of a number of different services in a certain area or region. Examples of such services are physical access control, a loyalty program of a filling station, location-dependent congestion or parking charges, etc. Such services have in common that the client's location must be assured in order to prevent *theft of service*, but they differ in the required precision of the client's location data. Thus, the disclosure of the location data has to adhere to the *need-to-know* principle.

The latter usage scenario so far has not received as much attention as several other scenarios, while this particular setting allows for novel approaches. In this paper we will design a simple solution for this particular usage scenario. The main characteristic of the proposed protocol is that the client's location only has to be assured once, while the client can later decide for each particular service used with which precision his location will be revealed. This is achieved by using *hash chains*. We will design our solution under the assumption of a *location verifier*. This is a trusted entity that can assess and certify a user's claimed location, based on a set of raw location data coming from, e.g., a GPS satellite.

This paper is structured as follows. In Section 2 we describe the general architecture that we assume for location-based service provision and location assurance. In Section 3 we develop the basic notations used in our design and we provide our location-blinding protocol, together with an informal validation. In Section 4 we discuss some related work and in Section 5 we summarize our findings and draw some conclusions.

2 General Architecture

We address the problem of balancing location reliability and privacy in the particular infrastructure which relies on the Global Positioning System (GPS) to calculate locations. This choice is motivated as follows. Recently, Harpes et al. propose to fix the absence of assurance in GPS location by introducing a location verifier, which is accountable to verify and sign the user's location [3]. If this solution was implemented, it would be possible to provide assurance that a given device was at a given place at a given time. We will provide a global description of this architecture and the underlying assumptions.

We describe our architecture for location-based service provision in terms of roles involved and of communication links among them (see Figure 1). A *User Device* (e.g., a GPS-enabled phone) processes the signals coming from the GPS satellites to calculate its own geographical position. Usually, the calculated position is communicated directly to the *Service Provider*, who uses it to adapt the service on the basis of the user's location. The communication between the user device and the service provider flows over channels for wireless connectivity such as those used in Wi-Fi or in the 3G wireless technologies.

As explained in Section 1, our architecture also includes a *Location Verifier*. Its task is to certify the integrity of the location information that the user is going to send to the service provider (see also [3]). In fact, the user device sends its location claim to the location verifier first. In Figure 1 this link is labeled (1). If the claim is verified to be correct, the location verifier sends back a *location certificate* to the user who, in turn, presents it to the service provider. This link is labeled (2). If the service provider has questions about the proof, he might contact the location verifier directly. This link is labeled (3).

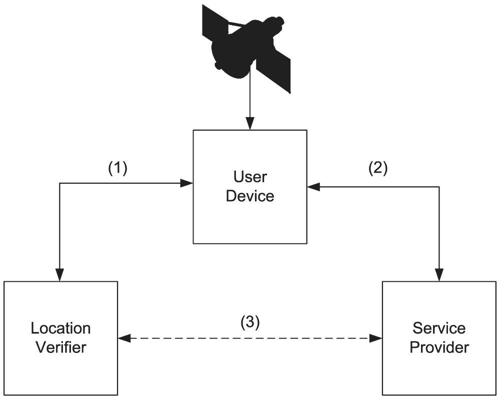

Fig. 1. The architecture for location-based services with location assurance.

There are many reasons to have a location verifier responsible for a high level of location assurance. Many location-based applications, for instance speed-limit enforcement, high valued assets tracking, and forensic reconstruction, actually can function properly only with reliable location data. However, the user device can fail in providing a correct location. First, user devices can be targets of meaconing or spoofing attacks [4]. In fact, satellite information is neither encrypted nor authenticated, thus an adversary can generate counterfeit satellite signals and mislead the user's device over its own location. A device can miscalculate a location, e.g., because the device is compromised by a malware accidentally downloaded while updating the device's software or firmware from the Internet. Finally, a user can intentionally manipulate the algorithm that generates the device's position to deceive the location-based service provider, for instance, to avoid to be billed while driving on toll roads. The presence of a location verifier prevents certain attacks against the integrity of locations that originate from the user device. For example, the location verifier can apply state of the art navigation message assurance mechanisms, or it can assign and revoke localization assurance certificates [3].

Admittedly, the availability of a location verifier is a rather strong assumption. Clearly, the system does not provide the user's location, but the location of his device. A relay attack, in which the user provides location data of a remote device, will indeed be hard to counter and distance bounding techniques will only partially be able to mitigate this type of attacks. Manipulation of the user device can be prevented by assuming that the device is tamper resistant. If we even assume that the antenna and its connection to the device are tamper resistant, direct manipulation of the incoming raw data will even be ruled out. This will require that the attacker has to resort to more complicated and expensive means of attacking the system. A further observation is that location assurance is not a yes-no question. Based on the raw data provided, the location verifier will only be able to assign a trust or assurance level to the location. Despite these possible concerns on how to realize a proper location verification service, we will simply assume the availability of such an entity.

In the following we will further assume that the communication channels between the User Device and the Location Verifier and between the User Device and the Service Provider ensure an authenticated and confidential exchange of messages. This assumption will allow us to concentrate on the information to be exchanged between the parties, without having to give a precise protocol for securing this communication. We consider authentication and confidentiality as services that are provided at another layer in the communication stack.

3 Selective Location Blinding

In this section we explore a solution that helps users to have control on the precision of disclosed location information while still allowing certified location assurance.

3.1 Locations and hash chains

The central concept of our study is the notion of a *location*. We use the word *location* as a generic term for describing a user's position-related parameters, such as latitude, longitude, elevation, velocity, acceleration, orientation, temporal information, etc. An example of a location is $x = (\mathit{lat}, \mathit{long}, t)$, where parameter *lat* is the (normalized) latitude, *long* is the longitude and t is the time.

Without loss of generality, we assume that a location is represented by n location parameters and that each parameter is a natural number. Location x is therefore defined as a list of natural numbers,

$$x = (x_1, \ldots, x_n) \in \mathbb{N}^n.$$

We assume that each x_i is denoted in base $B > 1$ and that, possibly after padding with zeroes, it consists of d digits. For $1 \leq i \leq n$, we have

$$x_i = x_i^{d-1} x_i^{d-2} \ldots x_i^0,$$

where x_i^{d-1} is the most significant digit of x_i and x_i^0 is the least significant digit.

The accuracy of a parameter x_i can be controlled by hiding its rightmost digits. The granularity of this hiding is determined by the base B in which the numbers are expressed. By $x|_p$ (for $d-1 \leq p \leq 0$) we denote location x of which the p least significant digits of each of the parameters are blinded. Phrased differently, only the $d-p$ most significant digits are exposed. Thus, $x|_d$ contains no information on location x, $x|_{d-1}$ exposes one digit, and $x|_0$ equals x.

We will use a hash chain to selectively hide the least significant bits of a location parameter. Let h be a cryptographic hash function that satisfies *preimage resistance*, *second preimage resistance* and *collision resistance*. We assume that h is publicly known.

For every location parameter x_i ($1 \leq i \leq n$), we construct a hash chain $K_{x_i} = K^d_{x_i}, K^{d-1}_{x_i}, K^{d-2}_{x_i}, \ldots, K^0_{x_i}$, where $K^0_{x_i}$ is a randomly chosen seed and $K^{j+1}_{x_i} = h(x^j_i, K^j_{x_i})$ (for $d-1 \geq j \geq 0$). Thus, the next hash in a chain contains the next more significant digit of the location parameter. Creating the hash chain requires the following calculations: $K^0_{x_i}$, $K^1_{x_i} = h(x^0_i, K^0_{x_i})$, $K^2_{x_i} = h(x^1_i, h(x^0_i, K^0_{x_i}))$, etc. We combine the hash chains for the different location parameters $(x_1, \ldots, x_n)$ by defining $K^j_x = (K^j_{x_1}, \ldots, K^j_{x_n})$ and $K_x = (K^d_{x_1}, \ldots, K^d_{x_n})$.

3.2 A selective location blinding protocol

Next, we show how to use these hash chains to selectively blind certified location data. In our protocol, we will only focus on which information is exchanged and we assume that authentication and confidentiality of the information exchange is dealt with at other levels of the communication stack. Figure 2 shows the entities involved and their interaction.

In our protocol, a *User Device* (e.g., a GPS-enabled phone) processes the signals coming from the GPS satellites to calculate its own geographical position. The calculated position will be verified and certified by the *Location Verifier* and communicated to the *Service Provider*, who offers the service, adapted to the User Device's location.

In full detail, our protocol consists of the following steps. First, based on raw data R (e.g., satellite signals), the User Device u calculates its location x and generates the list of random seeds K^0_x for the hash chain. This information is sent by u to the Location Verifier v. Based on its own context and observations v verifies whether location x corresponds to raw data R. We will not specify this context and observations because they are specific to the positioning system used. We simply assume that v has sufficient information to assess u's calculated location. If the location is correct, v calculates the key chain and offers u a signed certificate $Cert_x = sign_v(u, v, T, K_x)$, where T is a time stamp. Such a certificate is also called a *location proof* [5]. At this point, u can use this certificate in subsequent communications to Service Providers. For each such communication, u determines a precision p (for $d-1 \leq p \leq 0$) and sends the hash corresponding to this precision, K^p_x, together with the partially blinded location $x|_p$ and the location certificate $Cert_x$.

Using this information, the Service Provider reconstructs that part of the hash chain that follows after K^p_x and compares the final hash value, say K'_x, with the one from the certificate, K_x. In this way the Service Provider receives and

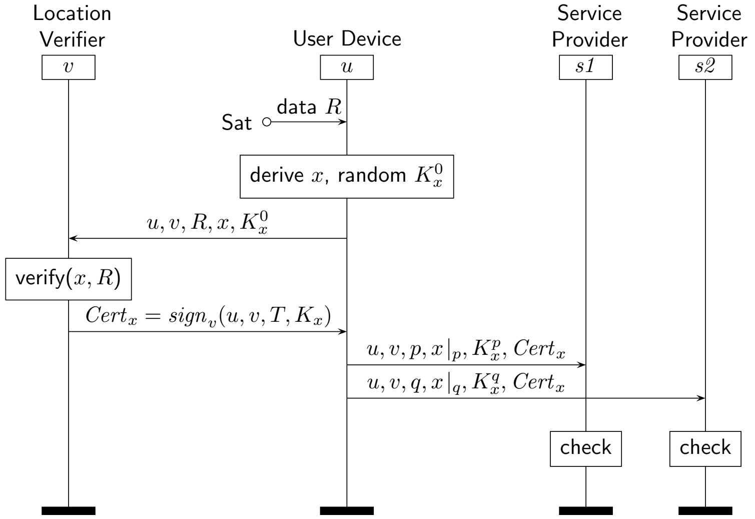

Fig. 2. The location blinding protocol.

verifies the $d - p$ most significant bits of the User Device's location parameters, while the least significant bits remain unknown to him.

An important feature of this protocol is that the User Device (i.e., the user who owns the device) can decide upon the precision p *after* the certification of his location. This will prevent the Location Verifier from having to certify the User Device's location for every possible precision p in advance. The same certificate can be used for different service providers, while adapting the precision to the needs of each of the particular services.

In the above protocol all location parameters are blinded with the same precision. The protocol can be easily adapted to provide a different precision for each location parameter. The only difference is that in the message from the User Device to the Service Provider all location parameters have to be treated separately.

The user device and the service provider should be able to agree upon how much privacy will be disclosed according to the *need-to-know* principle. Here, the user device and the service provider can run a service level agreement (SLA) protocol which supports the two parties in determining the level of privacy satisfying both the user's privacy policy and the service's requirement on location details. In certain cases, the level of privacy can be decided by bilateral agreement between a client and a server at the time of the client's enrollment to the service.

3.3 Validation

In this section, we give a basic reasoning why our protocol can guarantee the secrecy of the least significant p bits of the location parameters and the authenticity of the claimed location. In the previous section we made the assumption that the communication channels between the user device and the location verifier (link (1) in Figure 1) and between the user device and the service provider (link (2) in Figure 1) ensure the authenticated and private exchange of messages. This makes sure that any outside attacker cannot learn any information about, or modify the raw data R, the location x, the randomly generated seeds K_x^0, the hashes K_x and the certificate $Cert_x$, as exchanged between these protocol entities.

From the perspective of an insider, under the assumption that the location verifier is fully trusted, we see two threats: (*i*) a user spoofing his location and (*ii*) a compromised (or malicious) service provider learning more about the user's location than he is allowed to. The first threat is prevented by the certificate. Because the location verifier verified the user's location and because the hash function is second preimage resistant and collision resistant, the service provider can detect a location spoofing attack by checking the hash chain with the provided input. The second threat is prevented because the user only sends the most significant digits of his location to the service provider. Further, because the hash function is preimage resistant, the service provider cannot derive additional location information from the hashed values.

It is possible that from a series of observations a service provider (by conspiring with other service providers) can derive more accurate data on the whereabouts of a user than strictly allowed by the set precision, for example, using a Kalman filter [6]. Another possibility is that conspiring service providers can collaboratively calculate a precision which is better than the best precision for each of them as they can calculate the intersection of the location data that each of them received. To mitigate this type of attacks, we can combine our protocol with techniques providing anonymous usage of services (see Section 4). Since users cannot be identified, the conspiring service providers have no means to link location certificates to a particular user. There is one exception when a service requires a sequence of location data from the *same* user. In this particular case, anonymization cannot help as the service needs to ensure that several location certificates come from the same user.

4 Related Work

A major concern for large-scale deployment of location-based services is the potential abuse of users' locations, a recognized sensitive information. Papers on achieving location privacy can be classified into two classes: one uses location cloaking and the other studies user anonymity.

Location cloaking [7,8,9] is a popular approach for providing location privacy – a user's location is cloaked (by a third party or the user's device) before it is given to a service provider. For instance, Gruteser and Grunwald [7] develop an algorithm to adjust the resolution of location information based on the entities who may be using services in a given area. Cheng et al. [8] study a system model, which can be used to find the balance between privacy and the quality of location-based services. In their model, users can specify their location, service request and privacy requirements to the cloaking agent, which in turn produces the cloaked location and an "imprecise" service request. In this way, the service provider only knows the region where the user is, but does not know where exactly. In order to achieve location privacy based on the notion of k-anonymity, Zhong and Hengartner [10] develop a protocol based on homomorphic encryption which can cloak a user's location in a way that there are at least $k-1$ other people within the cloaked area. The system Casper* [9] uses a location anonymizer to blur a user's exact location information into a cloaked area to satisfy user specified privacy requirements. Our solution differs from these papers in several aspects. First, we allow the users to have control on their location privacy using hash chains – the user can provide an appropriate decryption key depending on which level location information is required by the service provider. Second, the above solutions require a trusted third party (cloaking agents or location-anonymizers) to protect the user's location privacy. Instead, we use a trusted third party, a location verifier, only to certify a user's location. Location cloaking is performed and controlled by the users.

Anonymous usage (of a service) is another important feature in LBSs. It is closely related to location privacy, in the sense that in LBSs it is desirable to make sure that the precision of location information cannot be used to identify a user. For this purpose, the algorithm developed by Gruteser and Grunwald [7] can be used to achieve a certain degree of anonymity for users by decreasing the accuracy of the revealed location information. Li et al. [11] propose a method to prevent an adversary to track the location of users, by allowing users to change their pseudonyms. Through a game theoretical analysis, Freudiger et al. [12] suggest some improvements on this protocol by Li et al. [11]. Our proposed solution has its focus on location privacy, it can be extended to satisfy more properties like user anonymity.

Papers on verifying the correctness of location proofs provided by a user are also related. Sastry et al. [13] present a simple protocol to securely check a user's location to be at some location within a region. By increasing the number of verifiers in the protocol, it can verify the user's location more precisely. Køien and Oleshchuk [14] develop a protocol which can be used to check whether the location reported by a user is inside of a polygon. Graham and Gray [15] propose a protocol for verifying location claims using proofs gathered from the neighboring devices/users. In our solution, we assume the existence of a location verifier, which is in charge of certifying location information. This can be achieved, for example, using techniques proposed in [3].

5 Conclusion

We proposed a procedure for the selective blinding of location information. The underlying idea is to hide the least significant digits of location data. This is a rather simple idea, but the problem becomes more complex in the context of a location verifier, when we require that the certificate does not depend on the precision which will be set later by the user. Our solution based on hash chains neatly enables the independence of a certificate from the required precision.

Because the procedure is based on hiding digits, the granularity of precision is determined by the number base used to represent the location data. In our current solution, the location verifier's certificate depends on this base, which implies that the granularity has to be decided upon before verification. An interesting question is whether we can use homomorphic encryption to allow the user to change the base of his certified location without requesting a new certificate.

In this paper we mainly focused on location data represented by natural numbers. Other flexible and versatile means to represent location information can also be supported. A wide range of solutions for representing location in Internet protocols are proposed in RFC 5491 [16]. Furthermore, our results can be extended to other types of contextual data that support a notion of precision, such as a person's age.

In our procedure, security properties of location certificates, such as secrecy and integrity, are easily achieved as we employ a trusted location verifier. Combining our protocol with other techniques (e.g., see Section 4) to provide users with properties like anonymity and untraceability is part of our future work.

As part of a project with industry, we plan to experimentally implement the architecture from Section 2. Our procedure will be validated by a prototype application using a mobile network with existing GPS or the upcoming Galileo receiver. The main challenge will be how to combine the different techniques concerning location verification, location privacy and user anonymity.

References

1. Popa, R.A., Balakrishnan, H., Blumberg, A.J.: VPriv: Protecting privacy in location-based vehicular services. In: Proc. USENIX Security Symposium, USENIX (2009) 335–350
2. Balasch, J., Rial, A., Troncoso, C., Geuens, C.: PrETP: Privacy-preserving electronic toll pricing. In: Proc. USENIX Security Symposium, USENIX (2010) 63–78
3. Harpes, C., Jager, B., Gent, B.: Secure localisation with location assurance provider. In: Proc. European Navigation Conference - Global Navigation Satellite Systems. (2009)
4. Warner, J.S., Johnston, R.G.: A simple demonstration that the global positioning system (GPS) is vulnerable to spoofing. Journal of Security Administration **25** (2002) 19–28
5. Saroiu, S., Wolman, A.: Enabling new mobile applications with location proofs. In: Proc. 10th Workshop on Mobile Computing Systems and Applications, ACM Press (2009)
6. Kalman, R.E.: A new approach to linear filtering and prediction problems. Journal of Basic Engineering **82**(1) (1960) 35–45
7. Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: Proc. 1st Conference on Mobile Systems, Applications, and Services, USENIX (2003)
8. Cheng, R., Zhang, Y., Bertino, E., Prabhakar, S.: Preserving User Location Privacy in Mobile Data Management Infrastructures. In: Proc. 6th Workshop on Privacy Enhancing Technologies. Volume 4258 of LNCS., Springer (2006) 393–412

9. Chow, C.Y., Mokbel, M.F., Aref, W.G.: Casper*: Query processing for location services without compromising privacy. ACM Transactions on Database Systems **34**(4) (2009) 1–48
10. Zhong, G., Hengartner, U.: Toward a distributed k-anonymity protocol for location privacy. In: Proc. 7th ACM Workshop on Privacy in the Electronic Society, ACM Press (2008) 33–38
11. Li, M., Sampigethaya, K., Huang, L., Poovendran, R.: Swing & swap: user-centric approaches towards maximizing location privacy. In: Proc. 5th ACM Workshop on Privacy in the Electronic Society, ACM Press (2006) 19–28
12. Freudiger, J., Manshaei, M.H., Hubaux, J.P., Parkes, D.C.: On non-cooperative location privacy: a game-theoretic analysis. In: Proc. 16th ACM Conference on Computer and Communications Security, ACM Press (2009) 324–337
13. Sastry, N., Shankar, U., Wagner, D.: Secure verification of location claims. In: Proc. ACM Workshop on Wireless Security, ACM Press (2003) 1–10
14. Køien, G.M., Oleshchuk, V.A.: Location privacy for cellular systems; analysis and solution. In: Proc. 5th Workshop on Privacy Enhancing Technologies. Volume 3856 of LNCS., Springer (2005) 40–58
15. Graham, M., Gray, D.: Protecting privacy and securing the gathering of location proofs - the secure location verification proof gathering protocol. In: Proc. 1st ICST Conference on Security and Privacy in Mobile Information and Communication Systems. Volume 17 of LNICST., Springer (2009) 160–171
16. Winterbottom, J., Thomson, M., Tschofenig, H.: GEOPRIV Presence Information Data Format Location Object (PIDF-LO) Usage Clarification, Considerations, and Recommendations. Technical Report RFC 5491, IETF Network Working Group (March 2009)