The Location Information Preference Authority: Supporting user privacy in location-based services

Anand S. Gajparia, Chris J. Mitchell and Chan Yeob Yeun

***Abstract*— To offer location-based services, service providers need to have access to Location Information (LI) regarding the users which they wish to serve; this is a potential privacy threat. Constraints, i.e. statements limiting the use and distribution of LI, that are securely bound to the LI, have been proposed as a means to reduce this threat. However, constraints may themselves reveal information to any potential LI user — that is, the constraints themselves may also be a privacy threat. To address this problem we introduce the notion of a LI Preference Authority (LIPA). A LIPA is a trusted party which can examine LI constraints and make decisions about LI distribution without revealing the constraints to the entity requesting the LI. This is achieved by encrypting both the LI and the constraints with a LIPA encryption key. This ensures that the LI is only revealed at the discretion of the LIPA.**

***Index Terms*— Location-based services, constraints, trusted third party, security model, privacy.**

I. Introduction

As the potential for services provided by mobile phones advances [1], it may no longer be appropriate to call such devices mobile phones. Mobile phones already provide far more than the voice communications for which they were originally designed. Text messaging and video download are just two examples of the range of services which are now available to the consumer. We therefore use here the more general term ‘mobile device’.

Amongst the features currently available in mobile devices are location-based services. Location-based services may also be provided to devices which are not mobile, such as desktop PCs. We thus refer here to ‘user devices’, which include both mobile and non-mobile devices. We can then define a location-based service as a service based on the location of a user device [2]. In order to facilitate the provision of such a service, it is necessary that LI is made available to one or more entities; this is at the root of the privacy issues with location-based services.

To provide a location-based service, it may be necessary for LI regarding the user to be passed to an entity with whom the user has little or no basis for a trust relationship. It is unreasonable, however, for a user to be forced to allow its LI to be provided to any entity which requests it, since this would leave the end user with no control over its LI, which is, of course, personal information. It is also unreasonable for a service provider to freely distribute the LI of a user to other entities without permission.

This paper introduces a mechanism designed to enable the end user to take advantage of the convenience of location-based services, and yet also control the way LI is used, stored and distributed.

We begin by introducing constraints [3]. The use of constraints is a technique which allows a user to dictate the way in which LI is managed. We look at some of the disadvantages of constraints which motivate the design of the scheme proposed in this paper.

We next look at the security requirements for methods to enable control of, and privacy for, LI. With this in mind, the notion of a Location Information Preference Authority (LIPA) is introduced. A LIPA is essentially a trusted party which helps control the distribution of LI and accompanying constraints. LI is distributed to service providers in the form of an ‘LI token’. The LI token includes LI securely bound to its constraints. The LI and constraints are also encrypted using the LIPA’s public key, ensuring that unauthorised entities cannot see this information.

We then look at how the LIPA mechanism may be used to address problems with constraints, LI control, and privacy.

II. Previous work

In previous work, a variety of different aspects of security for location-based services have been considered. Existing schemes for LI privacy are in many cases geared towards the available wireless technology architectures. These include IEEE 802.11 [4] networks, mobile IP [5] and GSM networks [6].

Myles *et al.* [7] describe constraints which may be used to control the distribution of location information, although they do not describe cryptographic protection mechanisms to provide privacy. A user registers their privacy requirements with a location server, referred to as LocServ. Entities which require location information make requests to the LocServ, providing their own privacy policies. Based on this, the LocServ can then make a decision whether or not to provide location information. This mechanism does not provide any means for entities to pass on information to other entities.

Aura *et al.* [8] investigate authenticated location information in the Mobile IPv6 protocol. Aura *et al.* see authenticated location information as a defence mechanism against false routing information, which could lead to other forms of attack. The subject of authentic location information is also discussed in [9]. The discussion in this latter paper concerns the location of GSM devices. The motivation is to support location-based access control mechanisms and the inclusion of LI in audit

Anand S. Gajparia and Chris J. Mitchell are with the Information Security Group, Royal Holloway, University of London, Egham, Surrey, UK.

Chan Yeob Yeun is with Toshiba Research Europe Limited, Telecommunications Research Laboratory, Bristol, UK.

logs. By contrast, the primary objective of this paper is the privacy of personal location information.

The Internet Engineering Task Force (IETF) geopriv working group is developing a general model for the protection of location information [10]. This model is primarily concerned with securing the Location Object (LO), which encompasses location information and other necessary information which may include constraints. They describe a general model which addresses the security requirements for such an object, encompassing a variety of scenarios. Our LIPA model looks at a specific scenario for a generally distributed LI token containing constraints and LI.

III. LI, Constraints and their limitations

A. LI entities

Below are descriptions of the entities in our simple model of a system in which LI is used [3].

- **Location Information (LI).** This is data which provides information regarding an LI subject's location. LI may occur in many forms. In general, we can divide LI into two types, namely *Inferred* LI and *Actual* LI. Actual LI refers to a directly calculated geographical location. This type of data indicates, to some degree of accuracy, the physical location of an LI subject. Inferred LI is, by contrast, obtained by implication. For example, if a user is present on a network, this implies that they are likely to be within an certain vicinity, although no specific calculation of geographical LI has taken place.
- **LI subject.** An LI subject is the entity about whom location information is being gathered, managed and used. This entity is most commonly a human user.
- **Location-Based Service (LBS).** This is a service based on LI, e.g. a vehicular navigation service.
- **Location Information Preference Authority (LIPA).** This entity, discussed in more detail in Section IV, acts like a trusted party on behalf of the LI subject. There may exist many LIPA entities, where the LI subject will typically be able to choose its preferred LIPA. Where an LI subject device has the capability, this device could itself act as the LIPA.
- **Malicious Party.** This is an entity with malicious intent. A malicious party may act as a threat to the confidentiality, integrity or availability of LI for one or more LI subjects.
- **User Device (UD).** This entity is a device with which the LI subject may interact, e.g. to invoke a location-based service. Such a device may either be static, e.g. a desk top computer, or more typically mobile, such as a mobile phone or Personal Digital Assistant (PDA). It is, in fact, this device regarding which LI is generated rather than the user him/herself, since there is typically no way to directly measure the location of individuals. Thus this entity is a key part of the model.
- **LI gatherer.** This is an entity which gathers or possesses LI about an LI subject and then creates an LI token using this information. The LI token is discussed further in section IV. A GPS receiver is an example of part of an LI gatherer, as it obtains location data. An entity in a GSM network which keeps signalling data for a UD is also an example of part of a LI gatherer. Although a GSM network does not normally pass on this LI (except in certain special cases), it certainly possesses such information, and could, in an appropriate environment, be a valuable source of LI for commercial use. Other examples of methods used to generate LI can be found in [11].
- **Regulator/Legal authority.** This is an entity which exerts legal or regulatory control over the management and use of LI. This includes telecommunications regulators, data privacy authorities, law enforcement bodies, and auditors.

B. Privacy and LI

It is becoming increasingly difficult to keep personal information private [12]. It does not help that users have a variety of incentives to surrender it. Shoppers frequently use loyalty cards in exchange for a variety of benefits. Using these cards, information regarding times at which users shop, what they buy, and where they buy from, may be recorded [13]. In this case, shoppers typically have the option of denying access [14] to such information by simply not using these loyalty cards. However, once a customer decides to use a loyalty card, restricting access to any information gathered from it becomes difficult. This problem applies to all forms of personal information, including LI, and does not only apply to loyalty cards.

Almost certainly the main LI security issue is the potential breach of privacy arising from the transmission of LI to entities not trusted by the LI subject. It is important to note that a breach of user privacy only occurs when the relationship between the identity of the LI subject and the LI can be established. Anonymous communication, where a user may use a resource or service without disclosing its identity, or communication using a pseudonym, where a user may use a resource or service without disclosing its user identity but can still be accountable for that use, could overcome this problem. However, in many cases, e.g. for billing, it is difficult to use anonymous or pseudonymous communication. Moreover, whilst many proposals for protecting location privacy rely on anonymisation of the LI subject, this does not seem as if it will be a solution of general applicability – many, conceivably most, location-based services will require the service provider using LI to be able to associate the information with a particular LI subject. Thus we throughout assume that the (authorised) user of LI is permitted to learn the association between the LI and the LI subject.

Another privacy issue is analogous to the problem of 'spam', i.e. the receipt of unsolicited messages. This already poses a huge problem in email systems [15], and has also started to become an issue in other domains, e.g. mobile text messaging. This is a problem which may also migrate to location-based services and thereby become even more intrusive. For example, service providers wishing to advertise their services [16] may use LBSs to send unsolicited messages to LI subjects in a given area.

To resolve these issues, LI should only be provided to entities authorised by the LI subject.

C. Constraints

Constraints are simply statements, bound to LI, which may be used to help control the use, storage and distribution of this LI [3].

An LI subject may, for example, want to limit the period of time an entity stores their LI. This will prevent entities collating data to provide information about the LI subject's travel habits. Storage time may be limited either by stating in the constraints the amount of time that the LI may be kept from a specified start point, or by stating a point in time after which the LI must be deleted. In the first case, the start point may be indicated by including a time stamp in the constraints, e.g. the time at which the LI was generated. However, as previously discussed in [3], placing a time stamp in the constraints allows receiving entities to learn the time at which LI was generated, and so the time when the LI subject was at a particular location. By contrast, a mechanism stating the time when the LI expires will limit the information revealed, as the time at which the LI subject was at a location cannot be precisely determined.

Limiting the distribution of LI ensures that LI is only sent to entities authorised by the LI subject. Restrictions on LI distribution may be specified either by stating the entities who are authorised to receive the LI, or by listing the entities not authorised to receive the LI. However, statements about permitted distribution give a receiving entity knowledge about relationships between the LI subject and other entities. For example, it enables entities to know which other entities are trusted by the LI subject and those which are not.

LI use may be restricted by stating how LI is or is not to be used. For example, an LI subject may only want their LI used for navigation purposes, and the constraints could state this. Conversely, the constraints could contain a negative statement indicating that, for example, the LI is not to be used for advertising purposes. These types of statement also provide information about the preferences of an LI subject, i.e. they are themselves a potential breach of user privacy.

Thus, providing information about how LI is to be managed allows personal information to be divulged. This is because the preferences of an LI subject are themselves personal information. Thus, in order to fully protect user privacy, the statements in the constraints must somehow be enforced without divulging the contents of the constraints to the LI consumers.

IV. A MECHANISM TO PROVIDE SECURITY FOR CONSTRAINTS

In this section the LIPA-based mechanism, providing privacy control for LI and associated constraints, is described.

A. Overview of the mechanism

In order to ensure that the information held within the constraints remains private, we propose the use of a trusted party which we call a Location Information Preference Authority (LIPA). The LI gatherer is assumed to be in possession of the list of preferred LIPAs for each LI subject for which it generates LI. This is an indication of the LIPAs trusted by the LI subject. The LI gatherer must be trusted by the LI subject to act according to its wishes.

1) **LI gathering.** The first step in our mechanism involves the provision of LI by the gatherer. The LI gatherer may be at any location, including in the UD itself. The LI gatherer may obtain LI in response to a request by an LBS provider or an LI subject, or it may constantly collect LI for a large number of LI subjects.
2) **LI token generation.** The LI gatherer then creates what we refer to as an LI token. This includes both LI and accompanying constraints. The LI and constraints are encrypted by the LI gatherer using the public key of the LIPA. This ensures that only the LIPA is able to view this information. Also contained within the scope of the token is information which helps to identify both the LI subject and the LIPA, together with a unique token identifier. The LI token includes the signature of the LI gatherer, guaranteeing the integrity of the LI token. This also provides evidence to receiving entities regarding the identity of the LI gatherer. An LI gatherer may generate several tokens for the same LI, e.g. if an LI subject uses two or more LIPAs. There is also provision for the inclusion of an optional public key certificate for the LI gatherer's public key.
3) **LI token distribution.** When LI is required, an LI token is provided to the LBS provider wishing to use the LI for service provision. This could occur in a variety of ways, e.g. by using third party LI token repositories, by sending the LI token via the UD, or by direct transfer from the LI gatherer to the service provider.
4) **LI token verification and decryption.** Once an LBS provider wishing to use LI receives an LI token, it must submit it to the appropriate LIPA. From the LI token the LBS provider can establish the identity of the LI subject, the identifier for the LI token and the identity of the LIPA, but not the LI or constraints since they are encrypted.

   Upon receiving the LI token, the LIPA verifies the signature, then decrypts the LI and the constraints, and checks if access to this LI is permitted for the requesting LBS provider. If access to the LI is permitted by the constraints, the LIPA returns the LI, the date/time of expiry of the LI, and the identifier of the LI token, all encrypted with the public key of the LBS provider, and signed by the LIPA. If permission is denied, a message stating this, together with the identity of the LI token, is returned to the LBS provider.

There are numerous ways that the LIPA may generate income for the provision of its service. The LIPA may charge for each request for LI which it receives, or each successful request for LI, i.e. when LI is sent to a LBS provider by a LIPA. Also, billing may be per LI token or per individual request. The entities which could potentially be billed for the LIPA service are the LI subject and the LBS provider. Billing the LI subject may result in a scenario where LBSs could request LI from the LIPA, which will charge the LI subject whether or not the LBS provider gives any service to the subject, and this is clearly not a desirable scenario. Alternatively, billing the LBS provider appears a more appropriate solution since the LBS provider can potentially recover the cost of obtaining the LI by including it in the charge for services provided.

The LI gatherer (unless it is the LI subject him/herself) will also typically require a means of obtaining payment for providing LI tokens. However, the LI gatherer may have no obvious party to charge except for the LI subject. In cases where the LI gatherer provides LI tokens for use by LBS providers not providing services to the LI subject, this is probably unviable. Another possibility might be for the LIPA entities to pass on a percentage of charges they make to LBS providers to the LI gatherers.

B. Requirements for use of the mechanism

This section describes the requirements on the entities involved in use of the mechanism.

The LI gatherer is the entity responsible for creating LI. It must possess a signature key pair. It must also possess a trusted copy of the public encryption key for all the LIPAs used by the LI subjects for which it generates/collects LI. These keys are used to encrypt the LI and the constraints in the LI token. The LI gatherer must also be in possession of a reliable copy of the constraints and LIPA preferences for each LI subject for which it generates LI.

The LIPA entity must possess both a signature key pair and an asymmetric encryption key pair. It must also possess a trusted copy of the verification key of every LI gatherer whose LI it needs to process, and a trusted copy of the public encryption key of each service provider to whom it might wish to provide decrypted LI. (The need for LIPAs to hold public keys of LI gatherers and LBS providers can be obviated by requiring LI gatherers and LBS providers to obtain and distribute public key certificates).

Each LBS provider must possess a trusted copy of the public signature verification key of each LIPA with which it interacts. It must also possess an asymmetric encryption key pair.

It is assumed that all the necessary encryption and signature algorithms have been globally agreed before use of the scheme.

C. LI creation

The entity responsible for generating LI is also responsible for creating what we refer to as an LI token. At the time of creation (or acquisition) of the LI, we suppose that the LI gatherer generates accompanying constraints C based on pre-specified LI subject preferences. The structure of the LI token is described below.

LI Token: $E_{e_L}(LI\|C)\|\ I_L\|I_S\|TokenID\|I_G\|$
$S_G(E_{e_L}(LI\|C)\|I_L\|I_S\|TokenID\|I_G)\|\ [Cert_G]$

where: $E_K(X)$ denotes the asymmetric encryption of data string X using the public key K; $S_A(X)$ denotes a digital signature (not providing message recovery) computed on data string X using the private key of entity A; e_X represents the public encryption key of entity X; $X\|Y$ represents the concatenation of data items X and Y; L represents the LIPA; S represents the LI subject; G represents the LI gatherer; I_X represents an identifier for entity X, e.g. I_G denotes an identifier for the LI gatherer G; $Cert_G$ is the public key certificate of the LI gatherer; [...] represents an optional data item.

The LI token is divided into four parts: the encrypted part, the plaintext part, the digital signature, and the (optional) public key certificate of the LI gatherer. The encrypted section contains the LI and the constraints, C. These are encrypted using the public key of the LIPA, e_L. This ensures that entities other than the LIPA cannot see this information. The plaintext part consists of I_L, I_S, $TokenID$ and I_G. The identifier I_L identifies the LIPA whose public key has been used to encrypt the LI and the constraints. This enables any entity wishing to gain access to the contents of an LI token to determine which LIPA it can be requested from. This identifier could take a variety of forms, e.g. a URL or an IP address. The identifier I_S allows any entity to identify the LI Subject to which the LI in the token relates. This identifier may be a pseudonym. The $TokenID$ is an identifier which, in conjunction with I_G, enables an LI token to be uniquely identified. The identifier I_G allows any entity to determine which entity generated the LI token. This also enables entities to decide which public key to use to verify the digital signature. This identifier may also be a pseudonym. The digital signature is computed over both the encrypted and plaintext parts of the LI token. This provides assurance that the LI Token has not been tampered with, and authenticates the entity which created the LI. The certificate $Cert_G$ may be optionally included in the LI token. This makes it easier for LIPAs which communicate with many LI subjects to obtain the necessary public keys.

Before proceeding, note that the encrypted part of the LI token could alternatively be encrypted using a symmetric encryption scheme with a shared secret key. The major advantage of such an approach would be that a symmetric encryption algorithm is typically much less computationally intensive that an asymmetric scheme. The main disadvantage is the key management overhead, since such an approach would require each LI gatherer to share a secret key with every LIPA with which it ‘does business’. A variety of different mechanisms exist to provide the necessary key management functions — see, for example, [17].

D. LI distribution

Section IV-C describes the structure of an LI token. When there is a request for LI or, when an LI subject requests a service, the LI token is sent to the relevant LBS provider.

LI Gatherer → P:
$E_{e_L}(LI\|C)\|\ I_L\|I_S\|TokenID\|I_G\|$
$S_G(E_{e_L}(LI\|C)\|I_L\|I_S\|TokenID\|I_G)\|\ [Cert_G]$

where:

$A \rightarrow B$ represents the communication of a message from entity A to entity B; and P represents the LBS provider.

LI should always be distributed within an LI token, regardless of who is sending the LI. The message above describes direct communication of the LI token from the LI gatherer to the LBS provider; however, as mentioned earlier, LI tokens may also be distributed via third parties and between LBS providers.

E. LI use

This section describes how an entity uses an LI token. When a LBS provider decides that it want to gain access to the LI within an LI token, it must send the LI token to the LIPA whose identifier is in the token, and hence whose public key was used to encrypt the LI in the token.

$P \rightarrow$ LIPA entity:
$E_{e_L}(LI\|C)\|\ I_L\|I_S\|TokenID\|I_G\|$
$S_G(E_{e_L}(LI\|C)\|I_L\|I_S\|TokenID\|I_G)\|$
$[Cert_G]\|[Cert_P]$

The above indicates that the LBS provider sends the LI token to the LIPA entity. The LBS provider may also optionally include a certificate for its public key, to avoid the need for the LIPA to possess a trusted copy of every LBS provider's public key. When the LIPA receives the LI token, it must first verify the signature and decrypt the enclosed LI and constraints. If the signature is invalid, or the token syntax is not as expected, then the LBS provider must be sent the 'Permission Denied' message (see below). The LIPA must then check that the LBS is permitted by the constraints of the LI subject to receive this LI. The LIPA must also check the authenticity of the LBS provider, which may be based on the certificate provided by the LBS provider. Details of a mechanism to provide this check for authenticity are not discussed further in this document. If the LBS provider is permitted to have access to the LI in the token, then it may be sent. The structure of the message used to send the LI back to P is described below. The LIPA also keeps a record of the LI token and the entity to which it is providing LI.

LIPA entity $\rightarrow P$:
$E_{e_P}(LI\|Expiry\|TokenID)$
$S_L(E_{e_P}(LI\|Expiry\|TokenID))$

The message from the LIPA to the service entity contains two parts: the encrypted part, which contains LI, $Expiry$ and the $TokenID$, and the signature. The encrypted part is encrypted with the public key of the service entity requesting the LI. This ensures that only the service entity can read this information, preventing malicious parties intercepting data while in transit. $Expiry$ is a time-stamp extracted from the constraints and specifies when the LI expires, i.e. when the LI should be deleted. This is the only information from the constraints which needs to be sent to the service entity. The $TokenID$ allows the LI subject to relate the LI received from the LIPA to the LI token from which it has been taken. The digital signature allows the receiving entity to check whether the message has been tampered with during transit.

If the requesting entity is not permitted to have access to the LI in the token then the following $PermissionDenied$ message is sent to the requesting entity:

LIPA entity $\rightarrow P$:
$TokenID\|PermissionDenied$

V. Security analysis

In this section we describe how our mechanism addresses control and privacy issues for LI. We also describe certain remaining issues with the mechanism. These could provide suitable topics for further research.

The primary aim is to provide a mechanism which enables the control of access to LI and constraints, enabling a greater degree of privacy without divulging extra personal information. By enabling the LIPA to make decisions based on constraints, untrusted entities do not gain access to the information found in constraints or LI. However, this does mean that the LIPA has access to both the constraints and the LI. Should the LIPA be compromised, the malicious party would have access to both the LI and the constraints of any LI subject using its services.

Once an entity is in possession of LI, maintaining control of this information is a difficult task. Ensuring that LI is managed according to the preferences of the LI subject once an entity possesses it, can only be based on trust. A problem inherent to LI is that when an entity has plaintext LI, they are free to do with it as they please. Our mechanism aims to provide LI only to entities which can be trusted, giving the LI subject control over their LI. Of course, even trusted entities cannot be trusted all the time and once these trusted entities have this LI, the LI subject can only rely on a regulatory or legal authority to ensure that messages are being transmitted in the manner which has been previously agreed. If an entity wishes to redistribute the LI of an LI subject, it should only distribute the LI token. If it chooses to redistribute LI in other forms, then this can only be addressed by some form of policing, e.g. through peer enforcement. Of course this could enhanced by a regulatory authority which ensures that rules are being adhered to.

Auditability should allow the identification of entities acting in violation of the rules set by the constraints. Identifying these entities is difficult, and is a desirable property. The use of peer pressure to enable auditability was introduced in [3]. To prevent unauthorised distribution of LI, its origin, i.e. the entity responsible for generating the LI token, must be verifiable. In addition, users of LI must be accountable for its use. Therefore, if a malicious entity redistributes LI in a way prohibited by the LI constraints, the recipient will detect this, and the malicious entity can be held responsible for the breach of constraints.

An additional concern is the potential for overloading the LIPA with requests for access to LI. This entity is of course, the central point for LI requests from service providers. This problem can be addressed by distributing the LIPA service across multiple servers, thereby removing the potential bottleneck and the single point of failure.

VI. Conclusion

This paper addresses the issue of control and privacy of LI and associated usage constraints by introducing a Trusted

Third Party based framework. We have introduced a mechanism which gives the end user the ability to control their LI without having to divulge additional personal data.

Acknowledgment

The research presented in this paper is being supported by sponsorship from Toshiba Telecommunications Research Laboratory, UK.

References

[1] U. Varshney and R. Vetter, “Mobile commerce: framework, applications and networking support,” *Mobile Networks and Applications*, vol. 7, no. 3, pp. 185–198, June 2002.

[2] E. Kaasinen, “User needs for location-aware mobile services,” *Personal and Ubiquitous Computing*, vol. 7, no. 1, pp. 70–79, May 2003.

[3] A. S. Gajparia, C. J. Mitchell, and C. Y. Yeun, “Using constraints to protect personal location information,” in *Proceedings of VTC 2003 Fall, IEEE Semiannual Vehicular Technology Conference*, vol. 3. IEEE press, 2003, pp. 2112–2116,.

[4] M. Gruteser and D. Grunwald, “Enhancing location privacy in wireless LAN through disposable interface identifiers: A quantitative analysis,” in *First ACM international workshop on Wireless mobile applications and services on WLAN hotspots*. ACM Press, September 2003, pp. 46–55.

[5] J. Zao, J. Gahm, G. Troxel, M. Condell, P. Helinek, N. Y. I. Castineyra, and S. Kent, “A public-key based secure mobile IP,” *Wireless Networks*, vol. 5, no. 5, pp. 373–390, October 1999.

[6] C.-H. Lee, M.-S. Hwang, and W.-P. Yang, “Enhanced privacy and authentication for the global system for mobile communications,” *Wireless Networks*, vol. 5, no. 4, pp. 231–243, July 1999.

[7] G. Myles, A. Friday, and N. Davies, “Preserving privacy in environments with location-based applications,” *IEEE Pervasive Computing*, vol. 2, no. 1, pp. 56–64, 2003.

[8] T. Aura, M. Roe, and J. Arkko, “Security of internet location management,” in *18th Annual Computer Security Applications Conference*. IEEE Computer Society, December 2002, pp. 78–87.

[9] C. Wullems, M. Looi, and A. Clark, “Enhancing the security of internet applications using location: A new model for tamper-resistant GSM location,” in *Eighth IEEE International Symposium on Computers and Communications*. IEEE Press, June 2003, pp. 1251–1258.

[10] J. Cuellar, J. Morris, D. Mulligan, J. Peterson, and J. Polk, “Geopriv requirements,” IETF, RFC 3693, February 2004.

[11] J. Hightower and G. Borriello, “Location systems for ubiquitous computing,” *Computer*, vol. 34, no. 8, pp. 57–66, August 2001.

[12] L. Palen and P. Dourish, “Unpacking “privacy” for a networked world,” in *Proceedings of the conference on Human factors in computing systems*, G. Cockton and P. Korhonen, Eds. ACM Press, April 2003, pp. 129–136.

[13] A. Adams, “A whole picture is worth a thousand words,” *ACM SIGCHI Bulletin*, vol. 35, no. 3, p. 12, May/June 2003.

[14] J. H. Moor, “Towards a theory of privacy in the information age,” *ACM SIGCAS Computers and Society*, vol. 27, no. 3, pp. 27–32, September 1997.

[15] L. F. Cranor and B. A. L. Macchia, “Spam!” *Communications of the ACM*, vol. 41, no. 8, pp. 74–83, August 1998.

[16] A. Ranganathan and R. H. Campbell, “Advertising in a pervasive computing environment,” in *Proceedings of the second international workshop on Mobile commerce*. ACM Press, September 2002, pp. 10–14.

[17] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*, ser. CRC Press Series on Discrete Mathematics and Its Applications. Boca Raton, Florida: CRC Press, 1997.