

# Supporting user privacy in location based services

Anand S. GAJPARIA<sup>†</sup>, Chris J. MITCHELL<sup>†</sup>, and Chan Yeob YEUN<sup>††</sup>, *Nonmembers*

**SUMMARY** 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. We propose the use of constraints, i.e. statements limiting the use and distribution of LI, that are securely bound to the LI, as a means to reduce this threat. 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, ensuring that the LI is only revealed at the discretion of the LIPA.

**key words:** *Multimedia location-based service, mobile security, privacy.*

## 1. Introduction

As devices used for wireless communication become increasingly ubiquitous and mobile, it is becoming apparent that location based services will play an important role in the evolution of ambient networking. Location based services use location information (LI) to allow an LI subject (the entity concerning which LI is being created) or some other entity to exploit this information to support the provision of one or more services. These range from allowing an emergency service to locate an LI subject, as is the case with E911 in North America [1], to an authentication service based on the location of an LI subject [2]. Location based services are also likely to play a significant role as a vehicular technology [3], including the support of navigational services and toll schemes.

Current technologies used to generate LI include various forms of Global Positioning System (GPS) [4], and Enhanced Observed Time Difference (E-OTD) [5] technologies. GPS uses satellites to enable the calculation of the LI of an LI subject. E-OTD calculates LI by observing time differences in transmissions between a user device and a base station. Unfortunately, LI may also be used for malicious purposes. For example, an entity could use LI to stalk an LI subject. Another undesirable use for LI is location-based spam [6]. These are unsolicited messages sent to a device based on its lo-

cation [7]. We define privacy as the controlled distribution of personally identifying information. Securing the privacy of LI is an issue which needs to be addressed in order to gain the trust of the mass market for such services. Only those authorised by the LI subject should be able to gain possession of LI. With this in mind, this paper introduces LI constraints as a means of allowing an LI subject to exert control over the distribution of its LI. In the context of this paper, LI constraints are simply rules associated with a specific piece or set of LI, restricting the ways in which the associated LI may be used and/or disseminated. To be effective, the constraints must be bound to the associated LI, typically by cryptographic means when the LI is in transit, and by access control techniques for stored LI.

LI constraints can be used to help manage the use and distribution of LI. We begin by investigating the possible constraint requirements of an LI subject, and discuss how these may be fulfilled. By looking at various uses for LI we investigate restrictions which may be placed on these uses. Although constraints may allow an end user to have some degree of control over its LI, placing constraints on LI also allows an entity to gain additional knowledge about the LI subject. We discuss the various limitations of using constraints and look at some simple methods to avoid such problems.

The fact that constraints may themselves be regarded as personal information motivates the design of a scheme proposed in this document. This scheme, called the Location Information Preference Authority (LIPA), enables the end user to take advantage of location based services, and also control the way LI is used, stored and distributed. 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 private key, ensuring that unauthorised entities cannot see this information.

Finally, we discuss further work which may aid the wider use of multimedia location based services.

## 2. Previous work

In previous work, a variety of different aspects of security for location based services have been considered.

<sup>†</sup>The author is with the Information Security Group, Royal Holloway, University of London.

<sup>††</sup>The author is with Toshiba Research Europe Limited.

Existing schemes for LI privacy are in many cases geared towards the available wireless technology architectures. These include IEEE 802.11 [8] networks, mobile IP [9] and GSM networks [10].

Myles *et al.* [11] 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.* [12] 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 [13]. 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 logs. By contrast, the primary objective of this paper is the privacy of personal location information.

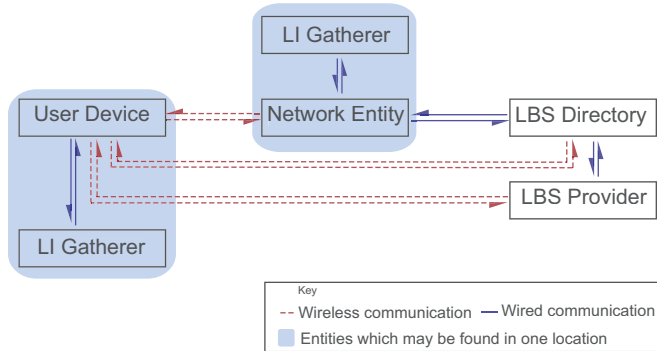
The Internet Engineering Task Force (IETF) geopriv working group is developing a general model for the protection of location information [14]. This model is primarily concerned with securing the Location Object (LO), which encompasses location information and other necessary information which may include 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.

### 3. A model for the use of LI

We next define the entities involved in a location based service architecture. The relationships between the various entities are also described.

- **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.
- **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.
- **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. In this paper, when we talk about LI we refer to actual LI. This type of LI is usually generated by a specialist entity, which we call an LI gatherer — see immediately below.
- **LI gatherer.** This is an entity which gathers or possesses LI about an LI subject. A GPS receiver is an example 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 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.
- **Location Based Service (LBS) Provider.** This entity provides a service, based on LI. This could be a multimedia location based service e.g. a vehicular navigation, gaming or advertising service.
- **LBS directory.** This entity provides information regarding the LBS providers which are available for use by a particular user. The LBS directory may itself use LI regarding the service consumer when providing the service. For example, it may show a service requestor lists of LBS providers providing information about particular types of retail premises in the area of the requester.
- **Network Entity.** This is a component which provides a network service to a UD. Two important types of Network Entity are the local base station which provides network access to the UD, and the UD's 'home network' with whom the UD owner has a contract and charging arrangement for the provision of network services.
- **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.

When we talk about the abuse of LI constraints, we define this as *any use, distribution or storage of LI which contradicts the rules defined by the constraints*.



**Fig. 1** Mobile scenario for location-based service provision

Figure 1 shows an example of how entities can interact when the user device has wireless capability. Equally, a non-mobile scenario is possible where entities are connected by wire. The malicious party may compromise any entity and may also compromise the communications medium. The regulator/legal authority may exert its control over any entity.

#### 4. Using constraints with LI

In order to set constraints on LI we must first look at how an LI subject may want to restrict the use and distribution of LI. In addition, these constraints should be in some common format which is automatically processable.

##### 4.1 Constraint types

We first look at the types of LI constraint which may be required.

###### 4.1.1 Storage time constraints

Storage time constraints may be used to limit the duration that an entity can store LI. This can be done in two ways.

One method is to use time stamps. A time stamp can be used to record the time of creation or the existence of information [15, p3]. By adding a validity period, a statement can be made that an entity should not hold LI subsequent to its expiry. For example, the LI subject may state that an entity cannot use LI once one hour after the time set by the time stamp has elapsed. The use of time stamps requires additional security mechanisms. Time stamp protocols require synchronisation and secure time clocks [15, p3]. Also, there should be secure mechanisms to obtain them. The

fields needed for such a constraint would be the issue date/time and the validity period.

Another way of adding time constraints to LI is by stating the time at which LI expires. This may be in the form of a date and time after which LI cannot be held. This would eliminate the need for a secure time stamp. The entity receiving LI will, however, need access to a secure clock in order to learn when LI is invalid. The field necessary for this scheme would be the expiry date/time.

###### 4.1.2 Distribution constraints

The LI subject may also want to constrain the distribution of LI. Distribution constraints can be specified inclusively or exclusively. Inclusive constraints would specify the entities who are permitted to possess LI. Exclusive constraints would indicate the entities who are not permitted to possess LI.

Consideration should also be made with regard to the way in which LI distribution is managed, and which entity is accountable for misuse of LI. This could be the entity which sends LI to an entity who is not permitted to receive it, or it could be the entity which receives and then stores LI when it is not permitted. Of course having both sender and receiver responsible for protecting LI would be most desirable, to ensure that the probability of misuse being prevented, or at least detected, is maximised.

###### 4.1.3 Usage constraints

An LI subject may want to place constraints to restrict the way in which their LI is used. Difficulties when constraining usage arise when attempting to enumerate all the different applications of LI, because of the wide range of possible uses. An attempt to classify the main possible uses of LI is given in section 4.3 below.

###### 4.1.4 Accuracy constraints

In some scenarios, it may not be necessary to provide very accurate LI. Also, an LI subject may not want entities to have particularly accurate LI. For example, a directory service may only need the vicinity of the LI subject to provide information about local restaurants. The accuracy of LI may be degraded prior to it being passed to an LBS provider, if required by the LI subject. The level of degradation may also be set by the LI subject.

##### 4.2 Gathering restrictions

Restrictions may be placed on the LI gatherer, preventing it from creating an LI token for a particular subject in specified circumstances. Clearly, this will

prevent any requestor of LI from receiving LI contradicting these restrictions. This type of restriction does not need to be included in any constraints; however it is necessary for the LI gatherer to know the nature of any restrictions specified by the LI subject. Examples of such restrictions are described below.

#### 4.2.1 Time restrictions

The LI subject may want to limit when LI may be gathered. For example during working hours an LI subject may want their employer or colleagues to be able to locate them. However, outside these hours the LI subject may not want to be locatable.

#### 4.2.2 Location restrictions

An LI subject may not want LI to be generated at certain locations. As previously, the LI subject may want his or her employer or colleagues to be able to locate them when they are in the work place; however, when they leave the work environment, they may not want to be locatable.

### 4.3 Uses of constraints

The use of LI can be sub-divided into two main types. LI can be used to:

- provide the LI subject with a service or with location details, or
- provide a service or location details to a separate entity.

The LI subject may, of course, not wish other entities to gain access to its location information, and hence may use constraints to limit uses of LI falling into the second category.

The two main types of constraints may be further divided if an LI subject wishes to be more specific about the purpose for which their LI is to be used.

## 5. Limitations of constraints

Once an entity other than the LI subject has possession of LI, it is difficult to force them to abide by the constraints which have been set.

### 5.1 Difficulties in preventing and detecting constraint abuse

LI is data, and the constraints which may be set on it do not physically prevent the receiving entity from misusing it. What adding constraints does do, however, is to allow entities to know the wishes of the LI subject. A regulatory authority which oversees the way in which other entities handle constraints may go some way towards preventing constraint abuse.

Another problem which arises when considering the use of constraints is proving their abuse. We have already established the difficulties of preventing the abuse of constraints with LI; it is also difficult to prove an entity has abused LI.

### 5.2 LI constraint predicaments

The aim of using constraints with LI is to enable an LI subject to dictate its use. Applying constraints to LI may, however, lead to further security issues.

When a user applies constraints to LI, they give information which indicates how, or how not, to use the LI. Although this information may be necessary to prevent the misuse of LI, applying constraints means that further information is divulged to the receiving entity. Two examples of this are now discussed.

#### 5.2.1 Time constraint predicament

Two potential schemes for time constraints were mentioned in section 4.1.1. One made use of a validity period for the constraints. The other is where the constraints are valid until a specified point in time.

The first scheme makes use of a time stamp which is added to the constraint. This allows a receiving entity to calculate the time at which a user was at the location shown by the LI. This may, in some circumstances, be undesirable. Of course, in most cases, the entities who are likely to receive LI are in all probability trusted by the LI subject, and so the fact that they know that a user was at a location at a particular time should not be a problem.

The difference between this and the second scheme for specifying time constraints is that, in the latter case, a receiving entity is not informed precisely when the LI subject was at a particular location. The delay before the receiving entity obtains the LI may only allow an approximate location of the LI subject to be calculated. In some cases, LI may be used in real time and, in such cases, the second scheme may be inadequate. An example may be for a navigational service, where the location and movement of the LI subject must be calculated in order to provide the required information.

#### 5.2.2 Distribution constraint predicament

If we place the responsibility for enforcing the LI constraints on the receiver of LI, then the presence of non-permitted LI at an entity is evidence that this entity is not acting within the constraints of the LI. Of course the problem with this is that it is not possible to prevent an entity from redistributing LI which it is not permitted to see.

## 6. Combining constraints with auditability

Preventing misuse of LI is inevitably going to be a com-

plex task. For an entity to be able to use LI, they must have access to it. After an entity has seen the LI, they thereafter can use or misuse it as they please. Even when constraints are bound to the LI, an entity may choose to ignore them or decide not to pass them on.

Instead of trying to prevent misuse of LI, which is almost certainly an impossible task, we therefore propose the concept of auditability of LI. The idea is to enable all users of LI to determine where LI originates from, and to make all users accountable for their uses of LI. To work effectively, the majority of LI users must abide by the auditability rules, but this seems a reasonable assumption (otherwise there is little hope of achieving any control over LI). Of course, auditing will not prevent abuse, but it does enable misuse to be detected after the event, thereby acting as a deterrent to misuse.

The notion of auditability introduced here requires use of digital signatures. Every piece of LI, and its associated set of LI constraints, must be accompanied by a digital signature computed over both the LI and its constraints. That is, when any LI is generated by an LI gatherer, then, as well as generating and attaching the LI constraints, the LI gatherer must create a signature over the LI and the associated constraints. The LI gatherer might also be required to include evidence with the LI of how it was obtained, and include this evidence within the scope of the signature.

Any entity receiving LI must verify the accompanying signature, and must log an exception (and must not use the LI) if the signature verification fails or if the signature is not present. Moreover, all LI users must check the constraints accompanying received LI to determine whether they should be in receipt of the LI — again, if they are not then an exception should be generated and the LI should not be used. Finally, the LI and the signature should be retained for auditing purposes for a specified period of time.

We now consider how this combination of rules can prevent (or at least make more difficult) the mishandling of LI. First observe that the mechanism described above does not address the misuse of LI, i.e. the use of LI in ways prohibited by the LI constraints. It is instead intended to address the issue of unauthorised distribution of LI (after all, uncontrolled dissemination of LI is probably the issue of greatest concern to most LI subjects).

Suppose a malicious entity wishes to redistribute LI in a way prohibited by the LI constraints. If the entity simply sends it on as received, then the recipient will detect that the constraints have been violated and the malicious entity can be held responsible for the breach of constraints. Hence the malicious entity will need to change the LI constraints. This, however, will invalidate the original signature, and sending the LI without a signature will also enable the recipient to detect an LI use violation. Hence, if an entity wishes

to disseminate LI with modified constraints, then they must sign the LI and indicate from where it was obtained — this may present a major problem for a fraudulent LI user. It will at minimum enable a subsequent audit to detect exactly which entity was responsible for disseminating unauthorised LI.

A further measure to restrict the ability to fraudulently disseminate LI would be to limit the entities capable of acting as LI gatherers and generating signatures on LI. This is discussed further in section 7. If an LI gatherer required a licence (e.g. in the form of an attribute certificate) to generate signed LI, then a malicious user without such a licence could not falsely disseminate LI, except to other malicious users.

Clearly this notion of auditability is dependent on industry co-operation and a regulatory body to ensure that rules are obeyed.

## 7. A mechanism to provide security for constraints

In this section we introduce the LIPA mechanism, providing privacy control for LI and associated constraints.

### 7.1 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 LIPA is responsible for deciding, based on given constraints, whether an LBS provider is allowed to have the LI of an LI subject. The information sent to the LIPA is encrypted in an LI token so that other entities cannot view it. This allows general distribution of LI within the scope of the LI token. 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 and then decrypts the LI and the constraints and checks if access to LI is permitted for the requesting LBS provider. If access to 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 is sent to the LBS provider. Note that, as discussed in section 4.1.4, the LIPA may choose only to send part of the LI, or a degraded version of it, to the LBS provider, if the constraints so specify. For example, the LBS provider may only be authorised to send LI accurate to at most 100 metres to this particular LBS provider, in which case the LIPA must reduce the precision of the LI before sending it.

Figure 2 shows the LI messages which may be transmitted in a mobile environment where the numbers correspond to the four numbered paragraphs above. Stages 1 and 2 take place at the LI gatherer. Stage 3 indicates the distribution of the LI token. In stage 4, the LI token is sent to the LIPA by the LBS provider, and LI may be provided in return.

## 7.2 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

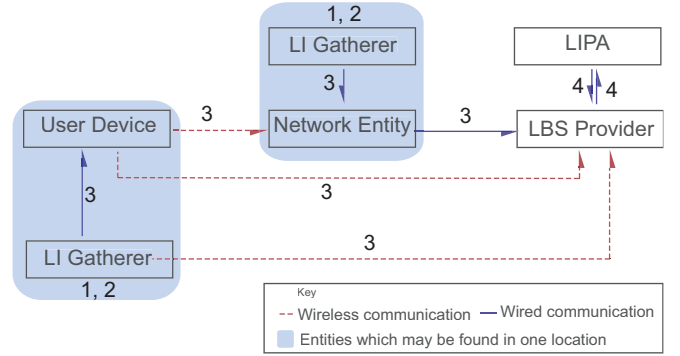


Fig. 2 LI-related transmission for a mobile user device

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.

## 7.3 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:

$$\begin{aligned}
 & E_{e_L}(LI||C)|| \\
 & I_L||I_S||TokenID||I_G|| \\
 & S_G(E_{e_L}(LI||C)||I_L||I_S||TokenID||I_G)|| \\
 & [Cert_G]
 \end{aligned}$$

where:

$e_X$  represents the public encryption key of entity  $X$ ;  $E_{e_X}$  denotes asymmetric encryption using the

public key  $e_X$ ;  $X||Y$  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 consists of  $LI$  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 than 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, [15].

#### 7.4 LI distribution

Section 7.3 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  $\rightarrow P$ :

$$\begin{aligned} & E_{e_L}(LI||C)|| \\ & I_L||I_S||TokenID||I_G|| \\ & S_G(E_{e_L}(LI||C)||I_L||I_S||TokenID||I_G)|| \\ & [Cert_G] \end{aligned}$$

where:

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

#### 7.5 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:

$$\begin{aligned} & E_{e_L}(LI||C)|| \\ & I_L||I_S||TokenID||I_G|| \\ & S_G(E_{e_L}(LI||C)||I_L||I_S||TokenID||I_G)|| \\ & [Cert_G]||[Cert_P] \end{aligned}$$

The above indicates the LBS provider sending 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 provider 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 LI, 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  $\rightarrow P$ :

$$\begin{aligned} & E_{e_P}(LI||Expiry||TokenID) \\ & S_L(E_{e_P}(LI||Expiry||TokenID)) \end{aligned}$$

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*

## 8. Billing

There are numerous ways 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 may be billed for the LIPA service are the LI subject and the LBS provider. Billing the LI subject may result in a scenario where LBS providers 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 desirable. Alternatively, billing the LBS provider appears a more appropriate solution since the LBS provider can recover the cost of obtaining the LI.

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 the charges they make to LBS providers to the LI gatherers.

## 9. Performance analysis

The scenario for our analysis in this section involves an LI subject with a wireless UD who wants to receive a

service based on his or her location. We assume the existence of a wireless network where the LI gatherer is a network entity. The UD and the LI gatherer are capable of both transmitting and receiving wireless transmissions.

### 9.1 Assumptions

We suppose that encryption of LI token contents is achieved by using a ‘digital enveloping’ technique. This involves first generating a random secret key,  $K$  say, which is used to encrypt the data (using a symmetric encryption algorithm). The secret key  $K$  is then itself encrypted with an asymmetric encryption algorithm (using the public key of the intended recipient), and sent with the encrypted data. For the purposes of our analysis here we suppose that  $K$  contains 16 bytes, that symmetric encryption takes place using a stream cipher, e.g. AES in counter mode [16], and that asymmetric encryption uses 1024-bit RSA (e.g. using OAEP) [16]. Hence, since we are supposing that symmetric encryption leaves the length of the data unaltered, enciphered data strings will always contain 1024 bits (128 bytes) more than the corresponding plaintext strings.

The digital signature scheme used is also RSA with a 1024-bit key see, for example [15], [16]). To generate the RSA digital signature, the data is first hashed using the SHA-1 hash algorithm [16]. This outputs a 160-bit (20-byte) hash with any given input.

The LI gatherer is capable of generating RSA digital signatures, and performing RSA encryption. It is also capable of generating a hash using the SHA-1 algorithm. The LBS provider is capable of verifying digital signatures and decrypting data. The LIPA is capable of verifying and generating digital signatures, and also encrypting and decrypting data. The LI gatherer and the LIPA can also generate public and private key pairs for digital signatures. The LBS provider and the LIPA should be able to generate a public and private key pair for encryption.

The length of the LI can be assumed to be 15 bytes. This is a reasonable assumption as the LI in a 3G network is 11 bytes, referred to as LOCI [17]. The LI subject identifier,  $I_S$ , is assumed to contain 10 bytes. The International Mobile Subscriber Identifier (IMSI) [17] in a 3G network, used to uniquely identify mobile subscribers, contains 9 bytes, so this is also a reasonable assumption. The LIPA identity  $I_L$ , the token identity *TokenID*, and the LI gatherer identity,  $I_G$  are all assumed to be 5 bytes in length, which allows a large number of unique identifiers. The constraints are assumed to be 800 bytes long. This is approximately the size of an XML schema with 17 lines. As we see below, this means that the total size of the LI token in this scenario is 1096 bytes (943+25+128).



## 9.2 Storage requirements

For this scenario, we assume that LI tokens are generated upon request. This means that the LI gatherer is not required to store LI tokens. The LI gatherer must, however, obtain the 1024-bit public keys of LIPA entities to which it sends LI tokens. These may be stored by the LI gatherer or acquired when necessary. The LI gatherer must hold the constraints for each LI subject to which it provides a service. This is necessary to generate the LI token.

When the LBS provider receives the LI token it can decide if it requires the LI from the LI subject based on its stored policies. These policies may list the LI subjects which have subscribed to a service; they may also specify the LI subjects to which a service should not be provided. If the LBS provider successfully receives LI from the LIPA it may also store the LI until the specified expiry time.

The LIPA may also store policy information. This could include information about the LI subjects to which it provides service. These stored policies may also hold information about the activities of various LBS providers. This helps the LIPA's decision making process when deciding if it should send LI to an LBS provider.

## 9.3 Message exchanges

The processes and data transfer requirements are summarised below. We start by describing the LI gatherer. After receiving a request for LI, it must perform the following tasks.

1. Encrypt the 815 bytes of data (800 bytes for the constraints and the 15 bytes of location data) using the RSA-based digital enveloping technique with the 1024-bit public key of the LIPA. This will result in an encrypted block of data containing  $815+128=943$  bytes.
2.  $I_L$ ,  $I_S$ ,  $TokenID$  and  $I_G$ , a total of 25 bytes of data (10 bytes for  $I_S$  and 5 bytes for each of  $I_L$ ,  $TokenID$  and  $I_G$ ), are then concatenated with the encrypted data described above. This will result in a data string containing 968 bytes.
3. Generate the SHA-1 hash of the encrypted data and the identifiers, and then generate a digital signature over this hash using the 1024-bit signature key of the LI gatherer. The resulting signature will contain 128 bytes. The concatenation of this with the identifiers and the encrypted data will result in a 1096-byte LI token.

The LI token is then sent to the LBS provider. When the LBS provider receives the LI token it decides if it requires LI from this LI token. If this is required it then sends the 1096-byte LI token to the LIPA.

At the LIPA:

1. The LIPA first verifies the signature contained in the LI token.
2. The ciphertext in the LI token is then decrypted. Based on the constraints found in the resulting plaintext, the LIPA then decides if the requesting LBS provider is permitted to receive LI. If permitted, it may be necessary to degrade the accuracy of the LI, depending on the constraints.
3. The LIPA must then encrypt the data to be sent to LBS provider. The data to be encrypted in this case will be the 15 bytes of LI, and approximately 100 bytes for the constraints, as the expiry time will be the only data from the constraints to be sent. This is sent together with the 5 byte  $TokenID$ . This is a total of 120 bytes of data to be encrypted, resulting in 248 bytes of ciphertext.
4. The above 248 bytes are then signed, resulting in a 128 byte signature. The resulting 376 bytes is then sent to the LBS provider.

When the LBS provider receives the data, it first verifies the signature and then decrypts the data. The LI from this data is then used to provide service to the LI subject. This data may be stored by the LBS provider until the specified expiry time.

## 10. 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. 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 protection could be enhanced by a regulatory authority, which would ensure that rules are being adhered to.

Auditability should allow the identification of entities acting in violation of the rules set by the constraints. 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.

## 11. Conclusion

This document 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.

Although attaching constraints has the advantage of allowing entities to see the requirements of the LI subject regarding LI, in doing so it also allows them to see additional information which may breach the privacy of the LI subject. It is also difficult to ensure that entities abide by the constraints which are set by the LI subject, and to prove when the constraints have been abused. Finding ways to address such issues is an important research challenge.

In order to enable a wide use of location based services it is important to have a single language for the specification of LI. This should allow LI to be generated, transferred and used on a wide variety of platforms.

Currently the most promising means of achieving a universally recognised means of specifying LI would be to employ an appropriately devised XML schema. XML (Extensible Markup Language) [18] is a language for data exchange between different devices. It allows data to be shared regardless of programming language or operating system, making it a strong candidate for use with location based services and to describe LI. If LI is described in XML, it should be possible to also describe constraints in XML, giving similar advantages. XML can also be used to create digital signatures, which may be used to support the auditing scheme mentioned above.

Location is just one aspect of a context-based service. A context-based service is one in which the context of an application automatically initiates some activity. Examples of possible contexts other than location include temperature and special events. Of course different forms of context have different security as-

pects. For example, the temperature of a subject's environment may not be private data; however, the end user's personal blood temperature may be private. As with location information, it would be necessary to subject such data to distribution and use constraints. This would mean extending the constraints described here to different contexts.

## Acknowledgments

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

## References

- [1] Federal Communications Commission, Revision of the Commission's Rules to Ensure Compatibility with Enhanced 911 Emergency Calling Systems ORDER (DA 02-2423). Wireless Telecommunications Bureau, 2002.
- [2] D.E. Denning and P.F. MacDoran, "Location-based authentication: Grounding cyberspace for better security," in *Internet Besieged, Countering Cyberspace Scofflaws*, ed. D.E. Denning and P.J. Denning, ch. 12, pp.167-174, ACM Press Books, 1st ed., February 2001.
- [3] C. Schwingschogl and T. Kosch, "Geocast enhancements of AODV for vehicular networks," *ACM Mobile Computing and Communications Review*, vol.6, no.3, pp.96-97, July 2002.
- [4] U. S. Department of Defense, Global Positioning System Standard Positioning Service Signal Specification, 2nd ed. U.S. Department of Defense, 1995.
- [5] 3rd Generation Partnership Project, "Technical specification group services and system aspects; location services (LCS); functional description; stage 2 (release 1999)," *Technical Specification 3GPP TS 03.71 V8.9.0*, 3GPP, June 2004.
- [6] Lorrie Faith Cranor and Brian A. La Macchia, "Spam!," *Communications of the ACM*, vol.41, no.8, pp.74-83, August 1998.
- [7] E. Kaasinen, "User needs for location-aware mobile services," *Personal and Ubiquitous Computing*, vol.7, no.1, pp.70-79, May 2003.
- [8] M. Gruteser and D. Grunwald, "Enhancing location privacy in wireless LAN through disposable interface identifiers: A quantitative analysis," *First ACM international workshop on Wireless mobile applications and services on WLAN hotspots*, pp.46-55, ACM Press, September 2003.
- [9] J. Zao, J. Gahm, G. Troxel, M. Condell, P. Helinek, Nina Yuan Isidro Castineyra, and S. Kent, "A public-key based secure mobile ip," *Wireless Networks*, vol.5, no.5, pp.373-390, October 1999.
- [10] Chii-Hwa Lee and Min-Shiang Hwang and Wei-Pang Yang, "Enhanced privacy and authentication for the global system for mobile communications," *Wireless Networks*, vol.5, no.4, pp.231-243, July 1999.
- [11] 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.
- [12] T. Aura, M. Roe, and J. Arkko, "Security of internet location management," *18th Annual Computer Security Applications Conference*, pp.78-87, IEEE Computer Society, December 2002.
- [13] C. Willems, M. Looi, and A. Clark, "Enhancing the security of internet applications using location: A new model

- for tamper-resistant GSM location,” Eighth IEEE International Symposium on Computers and Communications, pp.1251–1258, IEEE Press, June 2003.
- [14] J. Cuellar, J. Morris, D. Mulligan, J. Peterson, and J. Polk, “Geopriv requirements,” RFC 3693, IETF, February 2004.
  - [15] Alfred J. Menezes and Paul C. van Oorschot and Scott A. Vanstone, Handbook of applied cryptography, CRC Press Series on Discrete Mathematics and Its Applications, CRC Press, Boca Raton, Florida, 1997.
  - [16] Alexander W. Dent and Chris J. Mitchell, User’s Guide to Cryptography and Standards, Artech House, 2004.
  - [17] Technical Specification Group Terminals, Characteristics of the USIM application, v6.7.0 ed. 3rd Generation Partnership Project, September 2004.
  - [18] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau, “Extensible markup language (XML) 1.0 (third edition),” w3c recommendation, World Wide Web Consortium, February 2004.