Privacy-Preserving Location-based Services for Mobile Users in Wireless Networks

Sheng Zhong*, Li (Erran) Li†, Yanbin Grace Liu‡, Yang Richard Yang*

*Computer Science Department, Yale University, New Haven, CT 06511
†Networking Research Lab, Bell Laboratories, Holmdel, NJ 07733
‡Department of Computer Sciences, The University of Texas, Austin, TX 78712

***Abstract*— Mobility is key to personal freedom. With the increasing availability of mobile devices, many providers begin to offer location-based services. Although these services greatly enrich our mobility experiences, with them also comes the privacy concerns, as a location-based service provider now can continuously track the location of a user. This tracking may allow unauthorized access and cause serious consequences. Although a few solutions have been proposed to address the privacy concerns in various aspects, there has not been any comprehensive study of the problem; furthermore, most of the existing solutions require that a user trust a third party such as a location server.**

In this paper, we investigate privacy-preserving location-based services for the three components involved in providing location-based services: the location-based service component, the localization component, and the communications component. The focus of our study is on the location-based service component, but we also take the other two components into consideration. We identify two major types of location-based services and present novel designs to implement them without using a trusted server. Specifically, we first identify the general location-notification service, whose goal is to transfer location information of users to authorized entities. We design a security protocol to implement the service without trusting the location server. Thus our design uses the efficiency of a location server but does not suffer from associated privacy issues. Next, we investigate the design of an even more challenging location-based service: a location service whose goal is not transfering user location information but computing an outcome that is a function of user locations. We use dating service as an example and illustrate that an efficient protocol can be built such that no extra information about user locations is revealed during the service. For the localization component, we present an impossibility result and propose a privacy preserving localization technique based on directed signals. For the communications component, we propose an anonymous communication protocol. Our extensive evaluations show that our protocols have low overheads and are suitable for personal mobile devices.

***Index Terms*— Privacy, Location-Based Service, Security, Localization**

Sheng Zhong is supported in part by NSF grant ANI-0207399. Yang Richard Yang is supported in part by NSF grants ANI-0207399 and ANI-0238038.

I. Introduction

With the increasing availability of mobile devices, there is a growing demand for location-based applications. In response to such demand, various location-based services emerge recently (*e.g.,* [3], [29], [38]). For example, Nextel is already offering location-based services such as giving driving direction and locating points of interest such as hotels and restaurants within a short distance to a user's location [38]. DoCoMo has been offering dating services in Japan for a few years with much success. Recently Swedish Blue Factory started offering mobile dating service with positioning. This service allows people to send anonymous romantic messages from their mobile phones to people they care about. The positioning functionality allows the receiver to locate the sender.

These applications greatly enrich our lives and drive the demands for mobile and wireless communications services. However, they also raise serious privacy concerns as they enable the continuous tracking of involved users' locations. This tracking may allow improper disclosure or access to the location of a user by a stalker and thus may place a person in physical danger. Given the increasing concerns about location privacy, many governments and organizations are initiating studies on location privacy. For example, the US government has recently initiated the discussion on the Location Privacy Protection Act [28]. The IETF Geopriv working group [12] is also studying the requirements of location privacy.

To protect location privacy, various technical solutions have been proposed recently. However, many challenges still remain. Consider the three components involved in providing location-based services, as shown in Fig. 1. The first component is the localization component, which

determines the location of a user. Existing localization techniques (see [19] for a survey) use either passive measurements (*e.g.*, GPS) or active measurements. The authors of [34] show that active localization has better accuracy but suffers on location privacy. As far as we know, there has not been any study on how to perform active localization which protects location privacy.

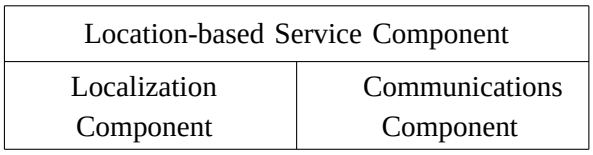

Fig. 1. The components involved in providing location-based services.

Since location-based services need communications support, the second component which will affect location privacy is the communications component. In [18], the authors propose blind signature as a means to protect the identify of a user from her communications provider. However, their proposal is in a college-campus setting, and it is unclear how to design such a scheme in a commercial environment, where financial transactions are involved.

The third component is the location-based service itself. This is the most relevant component and thus the main focus of this paper. In the last few years, various approaches have been proposed to implement this component (*e.g.*, [16], [17], [21], [36], [37]), and the predominant approach is to use a trusted server (*e.g.*, a user agent or a proxy). Although such trusted servers allow the implementation of flexible access control policies, they are undesirable for many reasons. First, with the trust comes the liability. Many providers are reluctant to bear the liability that follows. Second, many users are uncomfortable with trusting a third party. Thus a requirement for a trusted server may deter the adoption of many location-based services. Third, a single trusted party may become a single point of attack. Thus, if the trusted party is compromised, many users' privacy is compromised.

In this paper, we design novel protocols to implement location-based services for mobile wireless users without using a trusted third party. Considering all possible services that depend on user locations, we identify two major types of location-based services: the first type of service directly transfers user location information to authorized entities, and thus the technical challenge is to protect the location information from unauthorized entities, including the service provider itself. The second type of service does not involve transfers of user locations, but it requires computations that take user locations as inputs. The technical challenge for this type of service is therefore how to do these computations without revealing the user locations. Given this classification, we design two novel protocols to provide these two types of location-based services without using a trusted third party.

We also investigate the localization and communications components. For the localization component, we show that it is impossible to hide user locations from radio sensors, if radio-based localization is used. We then present an alternative localization technique based on directed signals, which protects users' location privacy against the localization service provider. For the communications component, we design an anonymous communication protocol that prevents the communication service provider from linking a user's location information with her identity in a commercial setting.

We implement prototypes to validate our design and evaluate the overheads of our protocols. Our evaluations show that our protocols have low overheads and are suitable for personal mobile devices.

Our major contributions can be summarized as follows.

- We propose the study of security solutions to enable location-based services without using a trusted third party.
- We design a novel protocol for a user to control which entities can have access to her location information stored at an untrusted location server.
- We design a very efficient protocol for location-based dating service that do not need to reveal any user's location information to any other party.
- We implement prototypes to evaluate our design and show that the overheads of our protocols are low.

Additional contributions are that we discuss the difficulty of achieving user location privacy against localization and communication service providers, and possible ways to sidestep this difficulty.

The rest of the paper is organized as follows. In Section II, we present our security protocol for location notification service. In Section III, we propose a novel security protocol to enable dating services but does not reveal location information to any party. In Section IV and Section V, we discuss privacy issues about localization providers and communication service providers respectively. We evaluate our protocols in Section VI. We discuss related work in Section VII and conclude in Section VIII.

II. Authorized Location Notification Service

A. Problem Formulation

The first type of location-based services we study directly distributes users' location information. Consider, for example, a person who wants to share her location information with various entities in different time intervals. In her work hours, she is willing to let her employer know where she is; every Saturday, she plays tennis with one of her friends and thus wants that friend to know her location; when she is ill, she would like to be tracked by her doctor — but definitely not after she recovers from the illness. The service studied in this section is called *authorized location notification service*, because it notifies authorized entities of its users' location information.

Formally, each user of the authorized location notification service has a set of entities who are potential receivers of her location information. At each time point, the user authorizes a subset of these entities to retrieve her location information. The user can change the subset of authorized entities at any time. The user wants to ensure that, at any time point, all entities in the *current* authorized subset are able to retrieve her location, while all entities outside this subset learns no information about her location. In particular, even former members of the authorized subset (*i.e.*, those entities who *were* in the authorized subset in the past but are out of it currently) cannot retrieve the user's location information.

B. Design Technique

We let each user store her location information on a location server, encrypted using a key specifically chosen for the subset of entities that are authorized to retrieve the location information. Of course, this encrypted location information will be periodically updated. The basic idea of our privacy-preserving design is that *only* the entities in the authorized subset should be able to derive the key to decrypt the location information. In other words, it must be infeasible for any entity outside the authorized subset, including the location server, to derive the key.

To achieve the above goal, one naive solution is to encrypt the location with a key shared by the user and each entity. This requires the user to update the server with as many encrypted location information as there are authorized entities for a single location update. This solution is obviously undesirable in a wireless setting where both spectrum and energy are limited. Another obvious solution is to encrypt the location information with a group key (*e.g.*, [**?**]). However, this requires distributing a new group key each time when the set of authorized entities changes. To avoid these inefficiencies, we use a cryptographic technique which is motivated by Akl and Taylor's work on hierarchical access control [1] and Fiat and Naor's work on broadcast encryption [11]. Let M be a RSA modulus and K be an element of the multiplicative group $Z_M^\times$ (where $Z_M^\times$ denotes the multiplicative group modulo M). The user keeps K and the factorization of M secret, and gives a pair (N_i, K^{N_i}) to each entity i, where K^{N_i} will work as entity i's secret key. When the authorized subset of entities is D, the corresponding key used for encrypting the location information is $K^{\prod_{i\in D} N_i}$. For each entity in the subset D, it is very easy to derive $K^{\prod_{i\in D} N_i}$ from the entity's own secret key K^{N_i} — all the entity needs to do is to do a modular exponentiation. However, for any entity outside the subset D, it is infeasible to derive the key $K^{\prod_{i\in D} N_i}$ as we shall prove.

C. Protocol Description

Below we specify the protocol in details. Our protocol consists of three phases: Initialization, Location Information Update, and Location Information Retrieval. We describe each of these phases below. For ease of reading, we also illustrate these phases in Fig. 2, Fig. 3 and Fig. 4.

Initialization

The user chooses two large primes P and Q, and computes $M = PQ$. Then she picks $K \in Z_M^\times$ at random.

For each entity i, the user chooses N_i such that all N_is are pairwise co-prime, *i.e.*, $gcd(N_i, N_j) = 1$ for any $i \neq j$. Then she computes $K_i = K^{N_i} \bmod M$.

The user distributes M, N_i, K_i to each entity i.

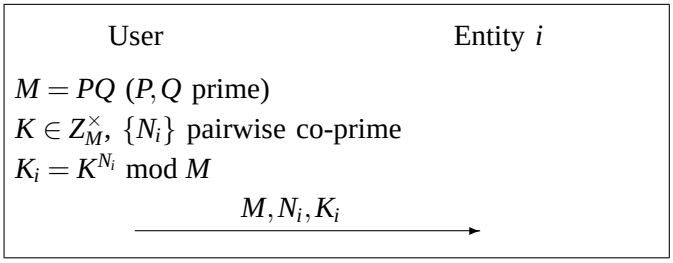

Fig. 2. Authorized Location Notification Service: Initialization Phase.

Location Information Update

Suppose that the user's current location is L and that she wants to authorize a subset D of entities to retrieve this information. The user encrypts L using a secure *symmetric* encryption algorithm and key K_D, where

$$K_D = K^{N_D} \bmod M,$$

and $N_D = \prod_{i\in D} N_i$. Then she uploads the encrypted location, together with N_D, to the location server.

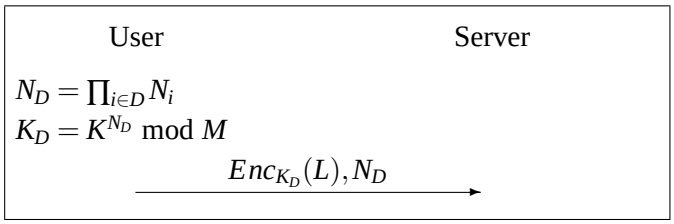

Fig. 3. Authorized Location Notification Service: Location Information Update Phase.

Location Information Retrieval

Any involved entity can download the encrypted location information and N_D from the location server. If $N_i | N_D$, then the entity i is authorized to retrieve the location information. In this case, the entity i derives K_D as follows:

$$K_D = K_i^{N_D/N_i} \bmod M. \quad (1)$$

(To see why this derivation is correct, observe that

$$\begin{aligned} K_i^{N_D/N_i} &\equiv K^{N_i \cdot N_D/N_i} \\ &\equiv K^{N_D} \pmod M) \end{aligned}$$

Then the receiver decrypts the location information using the key K_D.

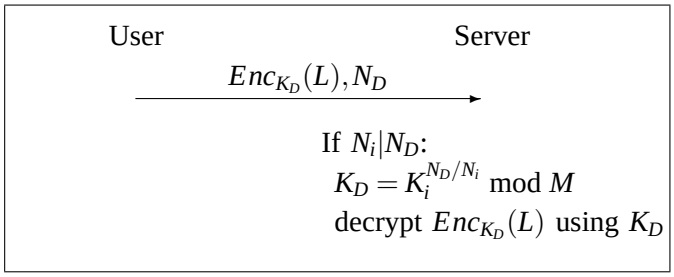

Fig. 4. Authorized Location Notification Service: Location Information Retrieval Phase.

D. Security Analysis

We show that the above design is secure under a standard cryptographic assumption — the strong RSA assumption (see [14] for details about the strong RSA assumption).

Theorem 1: Under the strong RSA assumption, it is computationally infeasible for an entity $j \notin D$ to derive the key K_D.

Proof: Suppose that an entity is able to derive K_D (with non-negligible probability). We will construct an adversary from this entity that can break RSA (with the same probability).

Because for any $i \in D$, $gcd(N_i, N_j) = 1$, we know that

$$\begin{aligned} gcd(N_D, N_j) &= gcd(\prod_{i \in D} N_i, N_j) \\ &= 1. \end{aligned}$$

Therefore, there exists two integers c_1, c_2 such that

$$c_1 N_D + c_2 N_j = 1.$$

(These two integers can be easily computed using the extended Euclidean algorithm.) Consequently, an adversary controlling entity j, who knows K_j and can compute K_D, can easily derive

$$\begin{aligned} K &= K^{c_1 N_D + c_2 N_j} \\ &= K_D^{c_1} K_j^{c_2}. \end{aligned}$$

Note that this breaks RSA, because K is the N_jth root of K_j modulo M and the adversary does not know the factors P and Q. ■

III. Location-based Dating Service

In the preceding section we have designed a protocol for authorized notification of users' location information. In this section, we study a different class of location-based services. For this class of location-based services, transferring users' location information is *not* the goal of the service. Nonetheless, this class of services is still location-based in that the output of such a service is a function of users' location information. Therefore, it will be ideal if no extra information about users' location information is leaked during the running of such a service. We use a kind of dating service as an example to illustrate how we enable such a service without disclosing the location of any user at all.

A. Problem Formulation

Dating has been an emerging mobile service to wireless users in various parts of the world. In this section, we consider a specific kind of dating service that assists its users to learn whether there is anyone nearby who matches her interest. Obviously, with such a service, users would have a lot more opportunities to find good dates.

To be precise, we divide the service area into many small regions and encode each region with a number. In the sequel, we often refer to a user's region number as her location. A user of the service requests a match by specifying a set of requirements for the person she has interest in; then the service allows the user to learn whether there is another user in the same region or in a nearby region whose profile meets her requirements. For ease of description, we assume that the requestor is only interested in finding matches at her current location. In reality, a requestor might want to find matches in a small area that overlaps multiple locations, including her current location. We remark that, it is easy to extend our

protocol to this case. Due to limit of space, we do not discuss them here.

Suppose that there is a dating service provider, with whom a large number of users are registered. All these registered users provide their profiles to the dating service provider so that they can be matched. However, these registered users are not willing to be tracked either by the service provider or by any other users, *i.e.*, they want to keep their own location private. Similarly, when a user requests a match, although she wants the service provider to search among people around her current location, she does not want to disclose what her current location is.

Clearly, this is a typical problem of secure multi-party computation [**?**]. There have been general solutions to secure multi-party computation problems (see [**?**] for a thorough survey). However, these general-purpose protocols are highly expensive both in computational overhead and in communication overhead. A naive adaptation of a general-purpose secure multi-party computation protocol to our problem would need at least thousands of modular exponentiations in computation and many megabytes in communication. In this section, we present a special-purpose protocol that is much more efficient than those general-purpose protocols. In our protocol, each requestor for a match only needs to do three modular exponentiations, while each matched user only needs to do two. Therefore, our solution is extremely suitable for personal mobile devices. The overall communication overhead of our protocol is low unless too many users match the profile requirements (see Section III-E for efficiency analysis).

B. Design Techniques

Because all profiles and requirements are available to the dating service provider, when there is an incoming request for match, it is very easy for the service provider to find the set of registered users that meet the requestor's requirements. Therefore, the technical problem here is how to decide whether any of these matched users' locations is the same as the requestor's *without revealing either the requestor's or the matched users' locations*.

Below we present novel cryptographic techniques that allow us to compare the locations of the requestor and a matched user without revealing either location. Roughly speaking, this is done in three steps:

- The two involved parties jointly encrypt the quotient of their locations. The private key needed to decrypt this quotient is the sum of these two parties' private keys, and so neither of them is able to decrypt it.
- The service provider raises the encrypted quotient to the sth power, where s is a random exponent.
- The two involved parties jointly decrypt the sth power of the quotient. If this is equal to 1, then the two parties are at the same location; otherwise, they are not at the same location.

Among the above three steps, the second cannot be omitted because we do not want either party to learn the quotient — with the quotient and her own location, she could easily figure out the other party's location. Next, we elaborate the techniques used in each of these steps in details.

Step 1: Jointly Encrypting Quotient

We use the well-known ElGamal encryption scheme. Suppose that the two involved users are i (with private key x_i) and j (with private key x_j), and that we want to encrypt the quotient L_i/L_j such that only with the private key x_i+x_j can this quotient be decrypted. Then the desired encryption is of the format $(L_i/L_jG^{x_i+x_j}, G)$, where G is a random element. To compute this encryption, we can let user i compute $(L_iG^{x_i}, G)$, which can be viewed as her own location encrypted using her own key. Then we let user j to multiply the first component $L_iG^{x_i}$ by G^{x_j}/L_j. The result is exactly the desired encryption.

Step 2: Raising Encrypted Quotient to the sth Power

It is trivial to raise the encrypted quotient to the sth power because ElGamal is multiplicatively homomorphic. All the service provider needs to do is to compute the sth power of the both components of the encrypted quotient. The result is an encryption of the sth power of the quotient.

Step 3: Jointly Decrypting the sth Power of Quotient

The encryption of the sth power of the quotient is $((L_i/L_j)^sG^{s(x_i+x_j)}, G^s)$. To decrypt this ciphertext, we need to divide its first component by the (x_i+x_j)th power of its second component. This is achieved in two substeps: first, user i divides the first component by the x_ith power of the second component; then, user j divides the first component by the x_jth power of the second component. The result is exactly the decryption we want.

C. Protocol Description

Initialization

Let p,q be large primes such that $p=2q+1$. Denote by G_q the quadratic residue subgroup of Z_q^x. Let g be a generator of G_q. Each subscriber i picks a private key $x_i \in Z_q$ and computes the corresponding public key $y_i = g^{x_i} \bmod p$. The subscriber i stores y_i on the dating server.

Each subscriber also uploads her/his profile to the dating server.

Matching

We now describe the matching phase, which is illustrated in Fig. 5.

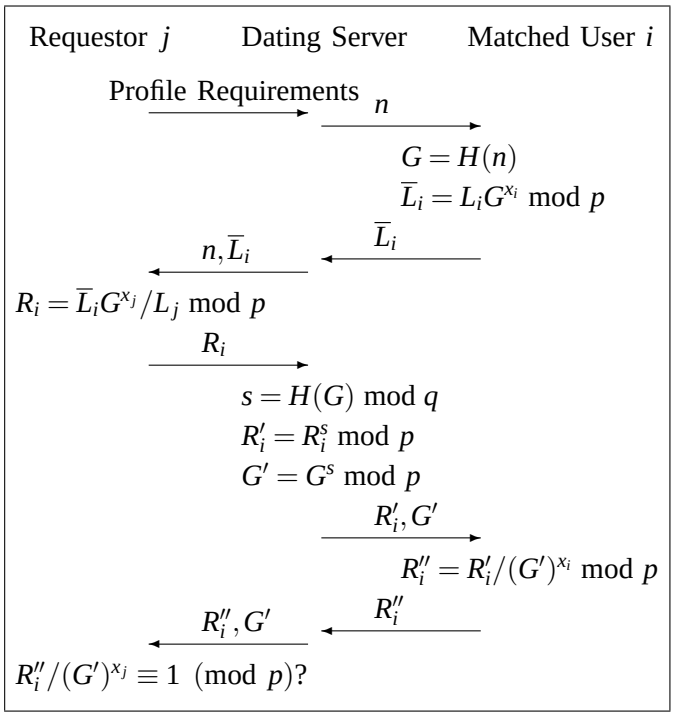

Fig. 5. Location-based Dating Service: The Matching Phase.

Suppose that user j would like to request a match. He sends the dating server her profile requirements.

The dating server assigns a session number n for this request and searches all profiles to find the set I of matched users. For each matched user $i \in I$, the dating server sends n to i.

Upon receiving the session number n, a matched user i computes:

$$\overline{L}_i = L_i G^{x_i} \bmod p,$$

where L_i is her current location, and $G = H(n) \in G_q$ for a cryptographic hash function H. Then user i sends $\overline{L}_i$ back to the dating server.

The dating server forwards all $\overline{L}_i$s back to the match requestor j, together with the session number n.

Upon receiving $\overline{L}_i$s and n, the requestor j computes:

$$R_i = \overline{L}_i G^{x_j} / L_j \bmod p,$$

where L_j is the current location of j. Note that, in the above computing of R_is, for efficiency optimization, the requestor j should compute G^{x_j}/L_j *only once* and *reuse* it to compute all R_is. The requestor sends all R_is back to the dating server.

The dating server computes $s = H(G) \bmod q$ and

$$R'_i = R_i^s \bmod p,$$

$$G' = G^s \bmod p.$$

It sends (R'_i, G') to each matched user i.

Upon receiving (R'_i, G'), the user i computes

$$R''_i = R'_i / (G')^{x_i} \bmod p,$$

and sends R''_i back to the dating server. The dating server forwards R''_i to j. It also sends G' to j.

Upon receiving R''_is and G', the user j tests whether

$$R''_i / (G')^{x_j} \equiv 1 \pmod p.$$

If yes, then there is a match. Note that, in the above tests, for efficiency optimization, user j should compute $(G')^{x_j}$ *only once* and *reuse* it to test all R''_is.

D. Security Analysis

We show that our system is secure under a standard cryptographic assumption — the Decisional Diffie-Hellman Assumption (see [5] for details about the Decisional Diffie-Hellman Assumption).

Theorem 2: Under the Decisional Diffie-Hellman Assumption, it is infeasible for a malicious requestor to compute any matched user's location.

Proof: Under the Decisional Diffie-Hellman Assumption, the ElGmal encryption scheme is semantically secure [5]. Suppose that a malicious requestor can compute any matched user's location (with a non-negligible probability); we will show that this malicious requestor can break the ElGamal encryption scheme (with the same probability).

During the matching phase, the requestor first receives $\overline{L}_i$ and the session number. It is easy to see that $(\overline{L}_i, G)$ is an ElGamal encryption under the key pair (x_i, y_i), where the private key x_i is unknown to the requestor. Next, the requestor receives R''_i and G', where $G' = G^s \bmod p$ and $R''_i = R_i^s / (G')^{x_i} \bmod p$ for a random s. Note that $((G')^{x_i} \bmod p, G')$ is an ElGmal encryption of 1 under the key pair (x_i, y_i), and that the messages received by the requestor in this round (R''_i and G') can be derived from this encryption. Therefore, if the requestor is able to compute the location information L_i, then essentially she is able to decrypt the ciphertext $(\overline{L}_i, G)$ after observing one single encryption of 1. This breaks the ElGamal encryption scheme. ■

Theorem 3: Under the Decisional Diffie-Hellman Assumption, it is infeasible for a malicious matched user to compute the requestor's location.

Proof: All a matched user i receives is the session number and (R'_i, G'), where $R'_i = (\overline{L}_i / L_j)^s G^{sx_j} \bmod p$ and $G' = G^s$. Because $\overline{L}_i$ is sent by the user i himself, computing L_j is equivalent to computing $\overline{L}_i / L_j)^s G^{sx_j}$. It is easy to see that (R'_i, G') is essentially an ElGamal encryption of $\overline{L}_i / L_j)^s G^{sx_j}$ under the key pair (x_j, y_j). Therefore,

computing the requestor's location L_j is equivalent to breaking the ElGamal encryption scheme. ■

Theorem 4: Under the Decisional Diffie-Hellman Assumption, it is infeasible for a malicious dating server to compute either the requestor's or any matched user's location.

Proof: Besides the profiles and the public profile requirements, all the dating server receives is $\overline{L}_i, R_i, R_i''$, where $R_i'' = R_i'/(G')^{x_i} \bmod p$. Because R_i' is sent by the dating server itself, R_i'' can be derived from $((G')^{x_i} \bmod p$. Therefore, we only need to show that it is infeasible to compute either L_i or L_j from $\overline{L}_i, R_i, ((G')^{x_i} \bmod p$. Note that $\overline{L}_i$, R_i, $((G')^{x_i} \bmod p$ are all ElGamal encryptions. Because ElGamal encryption is secure under the Decisional Diffie-Hellman Assumption, it is infeasible for a malicious matched dating server to compute either the requestor's or any matched user's location. ■

E. Efficiency Analysis

As we have mentioned in Section III-A, our protocol is very efficient. Here we give a brief theoretical analysis of efficiency. Experimental evaluations are given in Section VI.

We measure the computational efficiency of our protocol by the number of modular exponentiations because the time used by other operations (modular multiplications and divisions, hashing, etc.) can be ignored if compared to that used by modular exponentiations. In our protocol, each requestor for a match only needs to do three modular exponentiations, while each matched user only needs to do two. In addition, the dating server needs to do $k+1$ modular exponentiations if there are k matched users. Such computational overheads are very low for a secure multi-party computation protocol.

In terms of communication overhead, each involved user (either requestor or matched user) needs to send two messages and receives two messages. Most of these messages are very short, containing only one or two variables. Typically, each of these variables is of 256 or 512 bits, namely 32 or 64 bytes. Therefore, the length of a typical message is about 256 – 1024 bytes. The only exception is the first message sent by the requestor, which contains all the profile requirements. The length of this message depends on how fine-grained the dating service is with respect to user profiles and how the requirements are encoded. Our estimate for a typical message of profile requirements is below one hundred kilobytes. Consequently, if the number of matched users is not too large (*e.g.*, below 10), the *overall* communication overhead is most likely below 100 kilobytes.

IV. Privacy-Preserving User Localization

In Sections II and III, we have studied higher-level location-based services whose goals are or are not directly transferring user location information. Starting from this section, we study lower-level services. The service we consider in this section is user localization.

In a wireless network, a mobile user needs to determine her location in order to make use of the location based services. The user can determine her location passively by receiving signals, or determine it actively by sending signals (*e.g.*, [34]). There is no privacy concern in localization if the location is determined passively, *e.g.*, using a GPS unit. Nevertheless, GPS may not work all the time, and some users may need finer-grained location information than what is offered by GPS today. Thus, a user may subscribe to a localization service using an active localization technique. However, the localization service provider may be able to violate the user's location privacy in this case.

In this section, we first show in Subsection IV-A that, in principle, it is impossible for a user to hide her location from four nearby radio sensors. The practical implication of this result is that wireless users using radio-based active localization techniques are at higher risk of having their location privacy violated. Given this result, how do we protect users' location privacy against a localization service provider? There are two possible ways. The first possibility is that we replace radio-based localization techniques with alternative techniques. In particular, we present a privacy-preserving localization technique based on directed signals in Subsection IV-B. The other possibility is that we weaken the requirement of privacy so that it is achievable. In particular, if we are satisfied with anonymous localization, we can use a protocol presented in Subsection IV-C.

A. Impossibility of Hiding Location from Four Radio Sensors

Theorem 5: In a region that is monitored by four or more sensors of radio waves, any user who sends radio signals cannot hide her location from an authority that controls these sensors.

Proof: Suppose that, in such a region, a user sends a radio signal at power P_0. The power at which a sensor i receives this signal is

$$P_i = P_0 \mathcal{K}/d_i^{\alpha},$$

where $\mathcal{K}$ is a constant, d_i is the distance from the sensor i to the user, and α is the distance-power gradient. Therefore, for any two different radio sensors i, j ($i \neq j$),

we can easily get

$$P_i/P_j = (P_0\mathcal{K}/d_i^{\alpha})/(P_0\mathcal{K}/d_j^{\alpha}) \quad (2)$$
$$= (d_i/d_j)^{\alpha}, \quad (3)$$
$$\Rightarrow d_i/d_j = (P_i/P_j)^{\frac{1}{\alpha}}. \quad (4)$$

Let us set up two-dimensional Cartesian coordinates. Assume that the user location is (x_0,y_0), and that the sensor i's location is (x_i,y_i). Equation (4) can be rewritten as

$$((x_0-x_i)^2+(y_0-y_i)^2)/((x_0-x_j)^2+(y_0-y_j)^2) = (P_i/P_j)^{\frac{1}{\alpha}}.$$

This essentially means that the user's location is on a *quadratic curve* given by the following equation:

$$((x-x_i)+(y-y_i))^2 = (P_i/P_j)^{\frac{1}{\alpha}}((x-x_j)+(y-y_j))^2.$$

Assume without loss of generality that the authority controls sensors 1, 2, 3, and 4. Then the user's location is determined by

$$\begin{cases} ((x-x_1)^2+(y-y_1)^2) = (P_1/P_2)^{\frac{1}{\alpha}}((x-x_2)^2+(y-y_2)^2) \\ ((x-x_1)^2+(y-y_1)^2) = (P_1/P_3)^{\frac{1}{\alpha}}((x-x_3)^2+(y-y_3)^2) \\ ((x-x_1)^2+(y-y_1)^2) = (P_1/P_4)^{\frac{1}{\alpha}}((x-x_4)^2+(y-y_4)^2) \end{cases}$$

(Note that two quadratic curves have at most four intersections. With an additional third curve there is only one intersection in general.) ■

The above theorem assumes an ideal radio propagation model. In reality, if the radio sensors are far away, three sensors may not be sufficient to determine a user's location. For example, the current accuracy of triangulation from different cellular base stations is in the order of hundred meters, which is not very high. However, with increased density of radio sensors, Niculescu and Nath [**?**] shows that the accuracy of localization can be increased such that the median error is between 2.1 and 4 meters. Therefore, in practice, the potential privacy violation by radio-based localization techniques cannot be ignored.

B. Privacy Preserving Localization Using Directed Signals

In this subsection, we present a localization technique using directed signals. This technique is immune to the triangulation attack given in the proof of Theorem 5. Consequently, it has better privacy protection than the radio-based localization techniques.

Suppose that there are n sensors of directed signals. Here by "directed" we mean that the signal propagates in one direction such that only the sensor in this direction can sense it. A user attempting to localize herself chooses $n-1$ random time lengths $\delta_1,\ldots,\delta_{n-1} \in [-T,T]$. Then she sends a signal to each of these sensors, where the time difference between sending signals to the ith and to the $i+1$st sensors is $T+\delta_i$.

The localization service provider computes the time difference based on the time the sensors heard the user's signal and sends it back to the user. Assume that the time difference between receiving signals at the ith and at the $i+1$st sensors is t_i. Then the user solves the following over-determined equation system to get her own location:

$$\begin{cases} \sqrt{(x-x_2)^2+(y-y_2)^2}-\sqrt{(x-x_1)^2+(y-y_1)^2} \\ = v(t_1-T-\delta_1) \\ \ldots\ldots \\ \ldots\ldots \\ \sqrt{(x-x_{i+1})^2+(y-y_{i+1})^2}-\sqrt{(x-x_i)^2+(y-y_i)^2} \\ = v(t_i-T-\delta_i) \\ \ldots\ldots \\ \sqrt{(x-x_n)^2+(y-y_n)^2}-\sqrt{(x-x_{n-1})^2+(y-y_{n-1})^2} \\ = v(t_{n-1}-T-\delta_{n-1}), \end{cases}$$

where (x_i,y_i) is the location of the ith sensor and v is the velocity of the signal.

It is easy to see that a localization service provider controlling all these sensors cannot figure out the location of the user because she does not know δ_is. (Of course, the localization service provider still knows that a user is within a certain distance to the sensors that sense the directed signals, because otherwise the sensors would not be able to sense the signals. However, this is unavoidable for any active localization techniques.) Note also, that our result have assumed an idealized directional antenna. In reality, there is a non-zero beam width and there may be side lobes. if there are more than 4 sensors located inside the beam and the side lobes, then we may not be able to preserve a user's location privacy.

C. Anonymous Localization

If it is hard to hide users' locations from the localization service provider, we may be satisfied with keeping the serviced users anonymous to the service provider. That is, although the service provider knows that there is a user at some location, it does not know which user is there. In some practical situations, this is sufficient protection for users' location privacy.

Using anonymity to protect location information was first proposed by He, Wu, and Khosla [18]. Specifically, they proposed to use blind signatures to achieve anonymity. Here we present a protocol for anonymous localization, which is also based on blind signatures. However, compared to [18], our protocol uses blindly signed coins instead of blindly signed pseudo-identities and thus enables per-use charging of the service.

The basic idea of our design is that, before using the localization service, each user needs to buy digital coins from the service provider. These digital coins are

blindly signed such that, when they are spent by the user to get localization service, it is infeasible for the service provider to trace the coins back to their buyer. Therefore, when a user uses the localization service, she is anonymous to the service provider.

To avoid double spending of coins, the service provider needs to keep track of the coins already spent. However, searching in the database of already-spent coins is an expensive task. To mitigate this problem, we let each coin expire after a certain amount time. Consequently, the service provider only needs to keep track of the already-spent coins that were bought not too long ago.

Initialization

The localization service provider chooses $N = PQ$, where P, Q are large primes. For the ith day of service, the provider chooses $e_i, d_i \in Z_N^\times$ such that $e_i d_i \equiv 1 \pmod{\Phi(N)}$, where e_i is made public and d_i is kept private. In the sequel, we assume that $H(\cdot)$ is a cryptographic hash function. We also assume that each coin is valid for t days after purchase.

Payment Phase

Suppose that today is the ith day. This session is through the Internet.

- User: pick $r_1, r_2 \in Z_N$ at random; compute $m_1 = H(r_1) \cdot r_2^{e_i} \bmod N$.
- User $\rightarrow$ Service Provider: credit card information, m_1.
- Service Provider: verify the credit card information and charge the user; compute $m_2 = m_1^{d_i} \bmod N$.
- Service Provider $\rightarrow$ User: m_2.
- User: compute the signature $s = m_2/r_2 \bmod N$; the coin is $c = (r_1, s)$.

Service Phase

Suppose that today is the jth day, and that $c = (r_1, s)$ is an unused coin of the user's, purchase on the ith day. This session is through a wireless network.

- User $\rightarrow$ Service Provider: c, i, and the information needed by the localization algorithm.
- Service Provider: verify $j \leq i + t$ and $s \equiv H(r_1)^d \pmod{N}$; verify that c has not been used; record c in the database of used coins; use the localization algorithm to compute L, the location of the user.
- Service Provider $\rightarrow$ User: c, L.[1]

[1]Here, c is used to identify the session to the user. Note that wireless communications have a nature of multicast. If several users in a neighborhood are using the service concurrently, when they hear L from the service provider, they need to distinguish whose location L is.

V. Privacy-Preserving Wireless Communication

Recall that Theorem 5 shows the difficulty of hiding users' location from an authority controlling radio sensors. This authority can be a localization service provider as we study in the previous section. It can also be a wireless communication service provider, because base stations are essentially radio sensors. To protect users' location privacy against wireless communication service providers, in this section we propose an anonymous communication protocol which prevents the communication service provider from knowing the identity of the user who is communicating. The techniques we use in designing this protocol is very similar to those used in designing the anonymous localization protocol in Subsection IV-C. We still use blindly signed coins to protect users' anonymity. The major new idea in this protocol is that we assign a temporary address to each user that requests an anonymous session. This address works as a pseudo-identity, so that the user can receive messages during the session. Compared to [18], again, our protocol allows per-use charging because we have digital coins. Therefore, it is particularly suitable for use in a commercial setting.

We describe our protocol in details as follows.

Initialization

We need an initialization similar to the initialization phase in Subsection IV-C. In addition, we assume that the communication service provider reserves an address space of reasonable size for the temporary use of anonymous users. Note that this space does not need to be too large, because we only require that, if two anonymous users are communicating with the *same external node simultaneously*, then the probability that they are using the same temporary address is low.

Payment Phase

This is also similar to the payment phase in Subsection IV-C.

Service Phase: Requesting an Anonymous Session

Suppose that today is the jth day, and that $c = (r_1, s)$ is an unused coin of the user's, purchase on the ith day.

- The user picks a random temporary address in the reserved space.
- The user chooses an RSA modulus N_u and a key pair (e_u, d_u), where e_u is the encryption key and d_u is the decryption key.
- The user sends out the first message together with c, i, N_u, e_u, encrypted using e, the service provider's public key. The source address of this message is set to the above temporary address.

- The service provider decrypts the first message, and verifies $j \leq i+t$, $s \equiv H(r_1)^d \pmod N$, and that c has not been used. The service provider records c in the database of used coins and and forwards the message to the destination. The service provider records N_u, e_u and the base station that receives the first message of the session, where the session is identified by the temporary source address and the destination.

Service Phase: Sending a Message

- The user sends out the message encrypted using e, the service provider's public key. The source address of this message is set to the temporary address selected for this session.
- The service provider decrypts the message and forwards it to the destination.

Service Phase: Receiving a Message

- The service provider checks the incoming message's destination address. If it is a temporary address, the service provider chooses the appropriate encryption key e_u and base station for the corresponding session, where the session is identified by the temporary destination address and the source address.
- The service provider encrypts the message using e_u and sends it out through the above chosen base station.
- The user decrypts the message using key d_u.

Service Phase: Expiration of a Session

A session expires if there is no related traffic for a certain amount of time.

VI. Evaluation

In this section, we evaluate the overheads of the protocols we have presented for privacy-preserving location-based services. We focus on evaluating the overheads introduced by the cryptographic operations for protecting the location privacy of the user.

We implement prototypes of the protocols using Crypto++5.2 [**?**]. The implementation can run over a wide range of platforms such as Linux and Win32. We collect overhead data by running the prototype on an Intel Pentium III Processor at 700MHz. The data shown in this section are the average of 100 runs.

First, we evaluate the computational overhead of the protocol implementing the authorized location notification service. We use 256-bit primes for P and Q. The overheads of location retrieval and recomputing K_D (for change of authorized subset) are shown in Table I. We can see that they are pretty low. A location retrieval only takes 57.1ms. This indicates that a location service provider is capable of retrieving more than 1,000 (60*1000/57.1) users' location information in one minute. The overhead of the initialization phase is about 200ms when there are 10 entities. This is not too low but still acceptable, because we only need to initialize *once*. The overhead of location update is extremely low since we only need to do a symmetric encryption with a pre-computed key. Such low overhead makes our protocol suitable for mobile users who may update her location frequently.

	location retrieval	adding entity	removing entity
overhead	57.1 ms	54.1 ms	48 ms

TABLE I
Computational Overheads of Location Retrieval and Recomputing K_D.

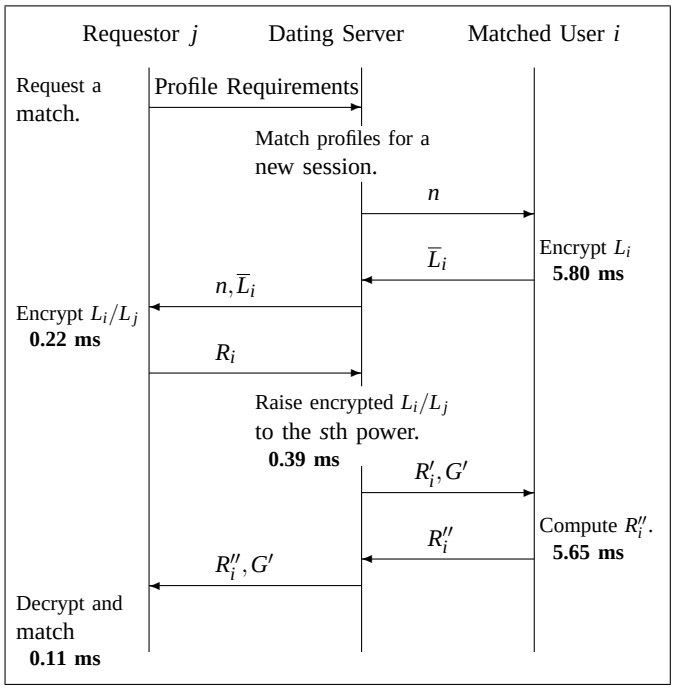

Fig. 6. Computational Overheads of the Major Steps of Dating Service Protocol.

Next we evaluate the overhead of our protocol implementing the dating service. We use 512-bit primes for p and q, and MD5 for the hash function $H()$. Fig. 6 shows the protocol flow and labels the computational overhead of each major step. We assume that there are 50 matched users in one session. Thus, the overhead of this computation is amortized over 50 operations. We can see that the overhead of our protocol is very low. Among these data, the overhead on the requestor and server side is smaller than that of the user side because the requestor

and the server can reuse one modular exponentiation computation. We see that, a user can request more than 3,000 matches a minute. A user can be matched by more than 80 users in one second. Again, this is very scalable.

To evaluate the communication overhead of the dating protocol, we shall check the lengths of the transferred messages. Given 512-bit p and q, almost all involved variables (*e.g.*, $\overline{L_i}, R_i, R_i', G', R_i''$) are 64 bytes. The session number n is typically of a similar length. All messages in the dating protocol contain one or two variables, except the message containing profile requirements (which depends on the design of dating service and is typically not too long). Therefore, most of the messages transferred are short messages of either 64 bytes or 128 bytes.

	Payment phase	Service phase	
		request a session	send a msg
user side	0.6 ms	33.0 ms	33.0 ms
server side	0.6* ms	265.5* ms	264.2 ms

TABLE II
COMPUTATIONAL OVERHEADS OF THE MAJOR STEPS OF PRIVACY-PRESERVING WIRELESS COMMUNICATION. *THE OVERHEAD OF THE SERVER DOES NOT INCLUDE THE OVERHEAD OF THE VERIFICATION OF THE CREDIT CARD INFORMATION OR THE OVERHEAD OF SEARCHING AND STORING DATA IN A DATABASE

At last, we evaluate the overhead of our protocol implementing privacy-preserving wireless communications, as shown in Section V. We use RSA with a modulus of 1024 bits. Since the protocol includes the protocol to implement anonymous localization, the overhead of the protocol in Section IV is given too. Table II shows the computational overheads of the steps in the protocol. We can see that the overhead in the payment phase is very small. The overhead in the service session is relatively high. However, the main part of the overhead is due to RSA. The remaining overhead excluding RSA overhead is negligible.

VII. RELATED WORK

Location privacy has been receiving considerable attention recently; see [15] for a survey. As we discussed in introduction, three components are involved in obtaining, transferring and accessing location information. They all impact on user location privacy. In this section, we review related work addressing privacy issues on each of these components.

A. Location-based Services

Location-based service using a trusted third party: This class of previous work relies on a trusted third party, which we call a location management server, to enforce privacy in location-based services. One possibility of a trusted third party is a user agent. In [36], Spreitzer and Theimer use a user agent to collect and control all personal information pertaining to its user, and any request for such information must be routed through the user agent, which enforces predetermined access policies. Confab [21] also takes this approach and extends it with more privacy mechanisms, including notifications, tags, logging, and interactive requests. Another possibility is a trusted proxy. In [16], [17], Gruteser and Grunwald propose spatial and temporal cloaking, in which a trusted proxy is used to adjust the resolution of location reported to services based on the density of users in a region. If enough number of users report their location through the proxy, the proxy can provide k-anonymity [37]. Yet another way is a mix network,[2] *e.g.*, mixes [10] and mix zone [4]. In these networks, the infrastructure provides an anonymity service using a mix network. The infrastructure delays and reorders messages from subscribers within a mix zone to confuse an observer. A problem with this system is that there must be enough subscribers in the mix zone to provide an acceptable level of anonymity. The authors of [4] conducted statistical attacks against these systems and found the security to be quite limited. Overall, the problem of using a trusted third-party or infrastructure is that a user has to trust and ensure the security of the third party; if the party is comprised, the user's privacy is compromised as well. In contrast, we focus on providing location-based services without the assumption of a trusted third party.

User access control policies on location information: To facilitate users to specify policies on accessing their location information, one thread of research on location privacy is to design policy languages to specify the privacy requirement of location-based services. In [26], [27], Langheinrich proposed pawS, a system based on P3P [7] which specifies policies of what data is being collected, and offers database support for enforcing those policies. In [35], Snekkens presents another conceptual framework based on lattice to specify personal location privacy policy. In [30], Myles et al. present another framework to specify location privacy based on location, time, and institution etc. These studies on privacy policy focused mainly on general system objectives, and thus are complementary to our project. This line of research is orthogonal to our investigation on providing location-based services without trusted third parties.

[2]A mix network can be designed under a *threshold trust* assumption. However, if we view the entire mix network as one party, it still has to be trusted.

B. Localization

A building block of a location-privacy system is how to measure the location of mobile devices. This is an active field; see [19] for a survey, and [9], [22], [31] for some recent advance. Roughly, we can partition the location measurement techniques into two categories [34]: active measurement (*e.g.*, [8], [13], [25]), in which a mobile node actively sends out measurement signal, and passive measurement (*e.g.*, GPS [20] and [2], [6], [33], [32]), in which a mobile node does not send any measurement signal and determines its position by receiving the signal from the infrastructure. It is possible that some techniques can be applied to block transmission to avoid unintentional active location, *e.g.*, [23].

C. Communications service

The inherent broadcast nature of wireless communication introduces privacy issues. To protect user identity in wireless communication in infrastructure wireless networks, He et al. [18] propose a blind signature based scheme to prevent communication service providers from knowing the user identity during communication. To address the same issue in wireless ad hoc networks, Kong and Hong [24] propose an anonymous routing protocol which enables routing without revealing the sender and receiver's identity. Our protocol for anonymous communication is similar to [18] but allows per-use charging.

VIII. Conclusion

In this paper, we designed novel protocols to provide location-based services which do not require a user to trust a third party. Specifically, we designed a novel protocol for a user to control which entities can have access to her location information stored at an untrusted location server. We also studied a class of location-based services that do not directly transfer user locations. We used dating service as an example and showed that the service can be provided without disclosing any users' location. Furthermore, we discussed the difficulty of achieving user location privacy against localization and communication service providers, and possible ways to sidestep this difficulty. We implemented prototypes of our protocols. Our evaluations showed that our protocols have low computation and message overheads and are suitable for personal mobile devices.

References

[1] Selim G. Akl and Peter D. Taylor. Cryptographic solution to a problem of access control in a hierarchy. *ACM Transactions on Computer Systems*, 1(3):239–248, 1983.

[2] Paramvir Bahl and Venkata N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of IEEE INFOCOM '00*, pages 775–784, Tel Aviv, Israel, March 2000.

[3] J. Bailey. Internet price discrimination: Self-regulation, public policy, and global electronic commerce. `http://www.tprc.org/abstracts98/bailey.pdf`.

[4] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 1:46–55, 2003.

[5] Dan Boneh. The decision Diffie-Hellman problem. In *ANTS-III*, volume 1423 of *Lecture Notes in Computer Science*, pages 48–63, 1998.

[6] P. Castro, P. Chiu, T. Kremenek, and R. Muntz. A probabilistic room location service for wireless networked environments. In *Proceedings of Ubicomp*, 2001.

[7] L. Cranor, M. Langheinrich, M. Marchiori, M. Presler-Marshall, and J. Reagle. The platform for privacy preferences 1.0 (P3P1.0) specification. `http://www.w3.org/TR/P3P/`.

[8] E911. Available at `http://www.fcc.gov/911/enhanced/`.

[9] T. Eren, D. Goldenberg, W. Whitley, Y.R. Yang, S. Morse, B.D.O. Anderson, and P.N. Belhumeur. Rigidity, computation, and randomization of network localization. In *Proceedings of IEEE INFOCOM '04*, Hong Kong, China, April 2004.

[10] H. Federrath, A. Jerichow, and A. Pfitzmann. MIXes in mobile communication systems: Location management with privacy. In *Information Hiding*, pages 121–135, 1996.

[11] Amos Fiat and Moni Naor. Broadcast encryption. In *Advances in cryptology, CRYPTO '93*, pages 480–491, 1993.

[12] Geographic location/privacy (geopriv) group. `http://www.ietf.org/html.charters/geopriv-charter.html`.

[13] L. Girod and D. Estrin. Robust range estimation using acoustic and multimodal sensing. In *IEEE/RSI Int. Conf. on Intelligent Robots and Systems (IROS)*, 2001.

[14] S. Goldwasser and M. Bellare. Lecture notes on cryptography. Summer Course Lecture Notes at MIT, 1999.

[15] Andreas Gorlach, Andreas Heinemann, and Wesley W. Terpstra. Survey on location privacy in pervasive computing. In *Workshop on Security and Privacy in Pervasive Computing*, April 2004.

[16] Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of The First International Conference on Mobile Systems, Applications, and Services (MobiSys)*, June 2003.

[17] Marco Gruteser and Dirk Grunwald. Enhancing location privacy in wireless lan through disposable interface identifiers: A quantitative analysis. In *WMASH '03*, San Diego, CA, September 2003.

[18] Q. He, D. Wu, and P. Khosla. Quest for personal control over mobile location privacy. *IEEE Communications Magazine*, 42(5):130–136, May 2004.

[19] Jeffrey Hightower and Gaetano Borriella. A survey and taxonomy of location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66, August 2001.

[20] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning System: Theory and Practice, Fourth Edition*. Springer-Verlag, 1997.

[21] Jason I. Hong and James A. Landay. An architecture for privacy-sensitive ubiquitous computing. In *Proceedings of The Second International Conference on Mobile Systems, Applications, and Services (MobiSys)*, Boston, MA, June 2004.

[22] Xiang Ji. Sensor positioning in wireless ad-hoc sensor networks with multidimensional scaling. In *Proceedings of IEEE INFOCOM '04*, Hong Kong, China, April 2004.

[23] Ari Juels, Ronald L. Rivest, and Michael Szydlo. The blocker tag: Selective blocking of RFID tags for consumer privacy. In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, 2003.

[24] Jiejun Kong and Xiaoyan Hong. ANODR: ANonymous On Demand Routing protocol with untraceable routes for mobile ad-hoc networks. In *Proceedings of The Fourth ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 291–302, Annapolis, MD, June 1-3 2003.

[25] Andrew M. Ladd, Kostas E. Bekris, Algis Rudys, Lydia E. Kavraki, Dan S. Wallach, and Guillaume Marceau. Robotics-based location sensing using wireless Ethernet. In *Proceedings of The Eighth International Conference on Mobile Computing and Networking (Mobicom)*, Atlanta, GA, November 2002.

[26] M. Langheinrich. Privacy by design – principles of privacy-aware ubiquitous systems. In G. D. Abowd, B. Brumitt, and S. A. Shafer, editors, *Ubicomp*, volume 2201 of *Lecture Notes in Computer Science*, pages 273–291. Springer, 2001.

[27] M. Langheinrich. A privacy awareness system for ubiquitous computing environments. In G. Borriello and L. E. Holmquist, editors, *Ubicomp*, volume 2498 of *Lecture Notes in Computer Science*, pages 237–245. Springer, 2002.

[28] Location Privacy Protection Act. `http://www.techlawjournal.com/cong107/privacy/location/s1164is.asp`, 2001.

[29] Mobiloco - location based services for mobile communities. `http://www.mobiloco.de`.

[30] Ginger Myles, Adrian Friday, and Nigel Davies. Preserving privacy in environments with location-based applications. *IEEE Pervasive Computing*, 2003.

[31] Yi Shang and Wheeler Ruml. Improved MDS-based localization. In *Proceedings of IEEE INFOCOM '04*, Hong Kong, China, April 2004.

[32] A. Smailagic, D. P. Siewiorek, J. Anhalt, D. Kogan, and Y. Wang. Location sensing and privacy in a context aware computing environment. In *Pervasive Computing*, 2001.

[33] Asim Smailagic and David Kogan. Location sensing and privacy in a context-aware computing environment. *IEEE Wireless Communications*, 9:10–17, October 2002.

[34] Adam Smith, Hari Balakrishnan, Michel Goraczko, and Nissanka B. Priyantha. Tracking moving devices with the cricket location system. In *Proceedings of The Second International Conference on Mobile Systems, Applications, and Services (MobiSys)*, Boston, MA, June 2004.

[35] E. Snekkenes. Concepts for personal location privacy policies. In *Proceedings of the ACM Symposium on Electronic Commerce (EC'01)*, pages 48–57, Tampa, FL, October 2001.

[36] M. Spreitzer and M. Theimer. Providing location information in a ubiquitous computing environment. In *sosp93*, 1993.

[37] L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

[38] Telenav. `http://www.telenav.com`, 2004.