

A Mechanism for Personal Control over Mobile Location Privacy

Qi He† Dapeng Wu‡ Pradeep Khosla†

†Carnegie Mellon University, Dept. of Electrical & Computer Engineering, Pittsburgh, PA 15213

‡University of Florida, Dept. of Electrical & Computer Engineering, Gainesville, FL 32611

Tel. (352) 392-4954, Email: wu@ece.ufl.edu. URL: <http://www.wu.ece.ufl.edu>.

Abstract—How to protect location privacy of mobile users is an important issue in ubiquitous computing. However, location privacy protection is particularly challenging: on one hand, the administration requires all legitimate users to provide identity (ID) information in order to grant them permission to use its wireless service; on the other hand, mobile users would prefer not to expose any information which enables anyone, including the administration, to get some clue regarding their whereabouts, that is, mobile users would like to have complete personal control of their location privacy. To address this issue, we propose an authorized-anonymous-ID based scheme; this scheme effectively eliminates the need for a trusted server or administration, which is assumed in the previous work. Our key weapon is a cryptographic technique called *blind signature*, which is used to generate an authorized-anonymous-ID that replaces the real ID of an authorized mobile device. With authorized-anonymous-IDs, we design an architecture that is capable of achieving complete personal control over location privacy while maintaining the authentication function required by the administration.

Keywords: security techniques and systems, WLAN, WPAN, mobile computing, ubiquitous computing, location privacy.

I. INTRODUCTION

Information about mobile users' location is a critical and valuable resource which needs to be utilized to fulfill the promise of ubiquitous computing [6]. Many efforts have been made to get it available, as one of the key services in the ubiquitous computing environment. However, the location information service or functionality can act as a double-edged sword – it can make our life more convenient; yet, it could also provide criminals with powerful weapons to compromise privacy of mobile users. Computer scientists have realized that unless the use of this information is strictly controlled, it can be put to a variety of unsavory situations [2], [6].

To address the location privacy issue, an architecture for location privacy control [2] was designed and experimented on the *Wireless-Andrew* network, an IEEE 802.11b wireless network that covers the entire campus of Carnegie Mellon University [4]. As shown in Fig. 1, the architecture [2] has a centralized location server, where a mobile user can register and submit her location information along with her “permission rule set” regarding her privacy preferences. Others can send queries to the server for location information about mobile users whose location information is stored in the server. The server processes the query according to the queried user's preference specified within the set of rules, and then the server

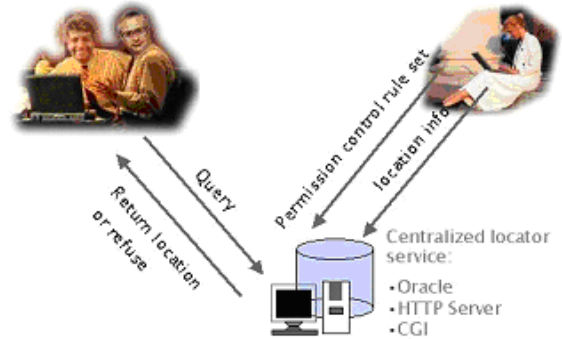


Fig. 1. Architecture of WaveGuard

may return the queried information, deny the query, or return a fake location as described in [2].

This architecture was primitive in the initial phase of the experimental system because it focused on strategic treatments of mobile location privacy rather than the system construction. This simple architecture is essentially identical to the one described in [6]. In [6], it was suggested that a distributed architecture could benefit privacy control of the mobile location, since a centralized architecture has the following drawbacks:

- The location privacy of mobile users is not completely under their own control since the system administration maintains a central server where the location information of mobile users is stored.
- The central server is a single-failure-point; that is, the location privacy of mobile users would be compromised if an attacker successfully hacked into it.
- The centralized architecture is not scalable.

However, to achieve complete personal control on location privacy by replacing a centralized architecture with a distributed one is not trivial. For instance, the system administration, for the sake of system maintenance and management, has the privilege to check any access point and obtain a list of IP addresses and corresponding MAC (Medium Access Control) addresses of the mobile devices that are connecting to the checked access point. The administration also has

the data¹ that can indicate a bijection relationship between MAC addresses (or IP addresses) of authorized mobile devices and registered legitimate mobile users. The location information about a mobile user can be easily figured out by the administration. Then, we face such a dilemma: on one hand, the administration would like to require all legitimate users to provide information for authentication in order to grant them permission to use its wireless service; on the other hand, the mobile users would prefer not to expose any of their information (e.g., IDs and MAC addresses) which would enable anyone, including the administration, to get clues regarding their whereabouts.

To resolve the above dilemma, this paper proposes an authorized-anonymous-ID based scheme. In our scheme, an authorized-anonymous-ID generated by a cryptographic technique called *blind signature* [1], is used to replace the real ID (e.g., an MAC address) of an authorized mobile device (e.g., a WaveLAN card). An anonymous ID can tell nothing more than whether the provider of the ID is an authorized user. This authorized-anonymous-ID is then used as the key for packet authentication, and the message authentication code [5] (generated by the key) is used for access control. In this way, the administration can grant authorized mobile users an access to the wireless communication infrastructure, while mobile users need not divulge their real ID during authorization, which could otherwise lead to compromising their location privacy.

Combined with a personal area network (PAN) and a multi-agent structure designed in our previous project named PUMA (Personal Ubiquitous Multi-Agent) [3], our authorized-anonymous-ID based scheme enables the mobile users to have complete control of their location privacy.

The rest of the paper is organized as follows. Section II describes our system architecture. In Section III, we present a set of protocols for the authorized-anonymous-ID based scheme. Section IV concludes the paper.

II. SYSTEM ARCHITECTURE

In this section, we first sketch key components in ubiquitous computing from a security perspective, and then present our agent-based system architecture.

A. A Sketch of Ubiquitous Computing

Fig. 2 shows three key components in ubiquitous computing, which are described as below.

- *Personal Trust Computing Base (PTCB)*: is a personal-held computing device, such as PDA and laptop; a PTCB is under the full control of the owner, and only the owner with proper authentication information such as personal identification number (PIN) or biometrics information, can activate a PTCB to work on behalf of the owner.

¹In the current solution, a mobile user must register her device (wireless LAN card) before she use the card to get connection. In current system, the registration is done by submitting MAC of the wireless card and the user's ID. The MAC is used for access points to decide whether the connection is granted. Packets from an unregistered MAC will get dropped.

- *Personal Area Network (PAN)*: is an architecture that consists of a main home PC, which has a connection to an Internet gateway, and has a wide range of appliances (i.e., PTCBs) connected to the main home PC, by many kinds of means. Each PTCB is associated with some kinds of autonomous software, called agent (or proxy). The agent runs on the PTCB if the PTCB is computational capable of running its agent; otherwise, the agent runs on the main home PC. The communication within a PAN can be secured by symmetric crypto-systems.
- *Internet*: provides a communication channel between a PTCB and a PAN; however, the channel cannot be trusted.

B. Agent-based System Architecture

The fore-mentioned sketch of ubiquitous computing leads to our agent-based system architecture, which is comprised of the following agents acting on the behalf of the players (devices or users).

- *Administrator (A)*: is an agent that acts on behalf of the administration to authenticate legitimate users and grant them access to the wireless infrastructure.
- *Rover (R)*: is an agent running at PTCB and acts on behalf of the owner of the mobile device. It is responsible to work out the location of the mobile device, automatically update the location information stored in the home PC (managed by another agent called *manager*, described below), and interact with the users for privacy permission setting [2].
- *Manager (M)*: is an agent running at home PC and can be delegated to act on behalf of the mobile user. It manages the location information submitted by the Rover and executes the user's control policy [2] for location privacy when it processes location information queries from other users.
- *Connector (C)*: is an agent running at an access point and is delegated by the Administrator agent to authenticate mobile devices and control wireless connections between mobile devices and the access point.
- *Lookup (L)*: is an optional agent facilitating the Internet users with public look-up functionality as a public service. Lookup agents as well-known public service providers, will listen for the location information queries from users and forward the queries to the queried users' Manager agent running at their home.

The multi-agent system architecture is illustrated in Fig. 3, where the protocols used for communication between agents are numbered with 1, 2, and 3; the three protocols are the registration protocol, the controlled connection protocol, and the location query/response protocol, respectively. In the next section, we present the registration protocol and the controlled connection protocol; the description of the location query/response protocol can be found in Ref. [2].

III. AUTHORIZED-ANONYMOUS-ID BASED SCHEME

This section presents our authorized-anonymous-ID based scheme.

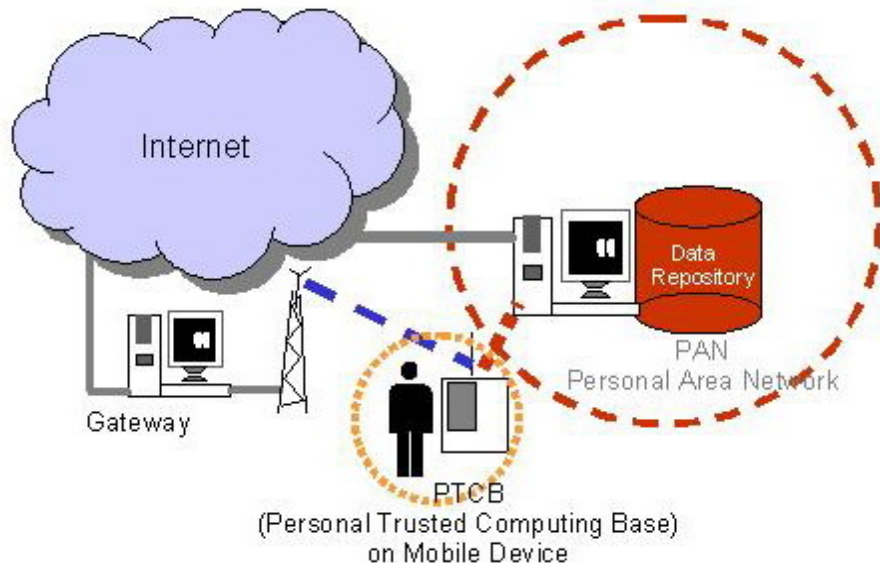


Fig. 2. A sketch of ubiquitous computing

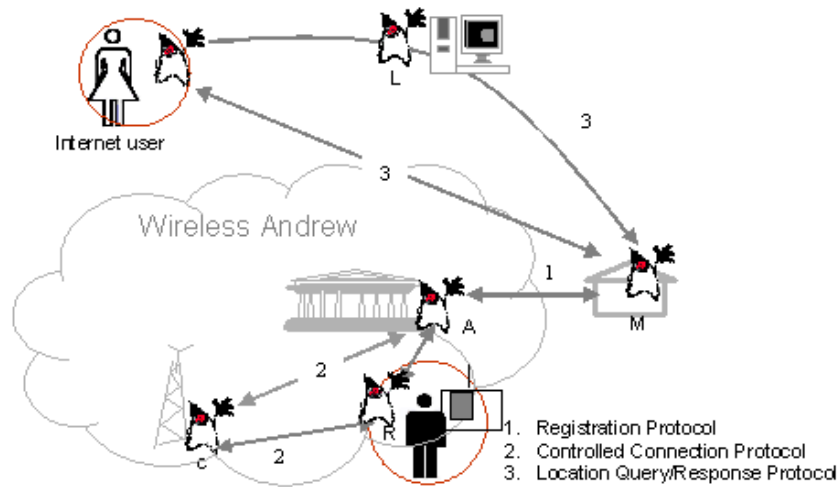


Fig. 3. Agent-based system architecture

There are two phases in our scheme: 1) the registration phase specified by a registration protocol, and 2) the controlled connection phase specified by a controlled connection protocol. In the first phase, the Manager (or Rover) of a mobile user applies for an authorized-anonymous-ID from the Administrator of the wireless infrastructure. After the first phase, the obtained authorized-anonymous-ID is carried by the Rover of the mobile user and will be presented when the mobile device is requesting for connection through an access point. In the second phase, the Rover presents the ID to request for connection and the ID is also used by the access point to authenticate the packets from the mobile device thereafter for the purpose of access control.

Table I lists the notations used in the description of the

protocols. Note that since both R_u and M_u have the private key of U , both of them can ‘speak for’ U . If R_u and M_u are interchangeable in the protocol, we use U to represent them. In other words, U in the following protocols can be replaced by either R_u or M_u .

A. Registration Protocol

Our registration protocol is based on authorized-anonymous-ID. With an authorized-anonymous-ID as a digital token, a legitimate mobile device can be granted permission to access the wireless infrastructure after a successful authentication; yet the association between the token and the real ID of a legitimate user is eliminated. The registration protocol is outlined in Fig. 4.

TABLE I
NOTATIONS.

U	: A mobile user, identified by her public key. The corresponding private key is held by her Rover running in her PTCB and Manager in home-PC of PAN.
R_u	: Rover of mobile user U .
M_u	: Manager of mobile user U .
E_x	: Public key of X .
D_x	: Private key of X .
$K_{xy}(m)$: Encrypt m by using symmetric crypto-system with a key shared by x and y .
$K_{xy}^{-1}(c)$: Decrypt c by using symmetric crypto-system with a key shared by x and y .
$H(x)$: One-way hash function with input x .
$E_x(m)$: Encrypt m by using asymmetric crypto-system ² with the public key of x .
$D_x(c)$: Decrypt a cipher c with the public key of x .
r_0, r_1	: Random numbers.
ack	: Acknowledgement for the last received message.

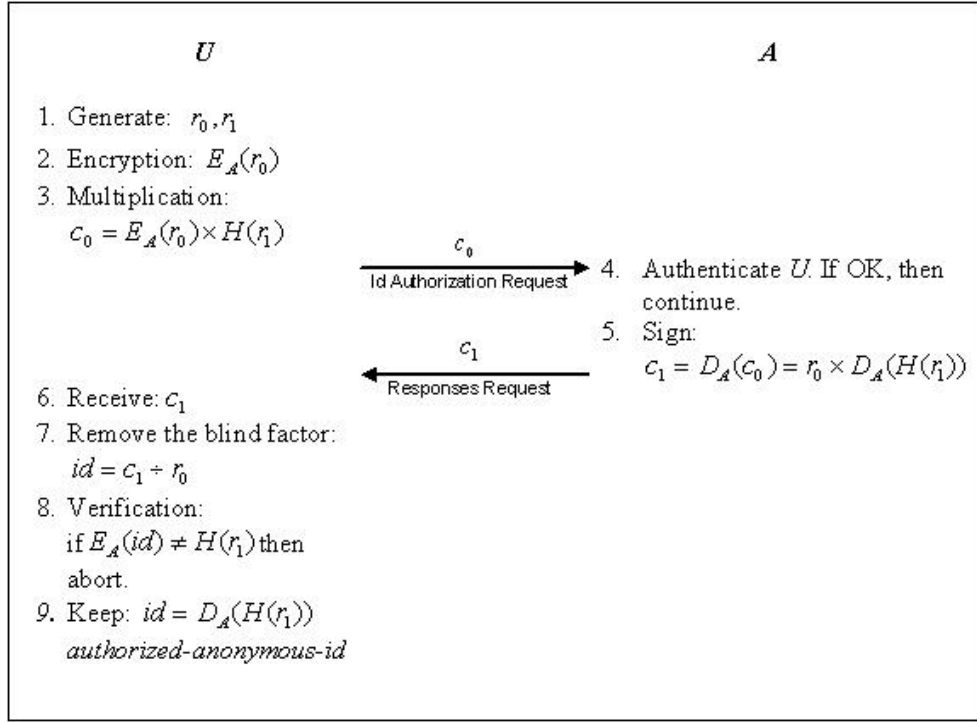


Fig. 4. Registration protocol

As we mentioned previously, the role of U in the registration protocol could be played by either R_u or M_u , which depends on the environment where U is currently staying. Usually, when U is at home with her mobile device, she can have M_u initiate the protocol to get the authorized-anonymous-ID ($r_1, D_A(H(r_1))$), and then convey the authorized-anonymous-ID to R_u via a secure channel between R_u and M_u , which is protected by a symmetric crypto-system as mentioned in Section II-A. In case that the mobile device already has a connection to the Administrator, the Rover can also initiate

the registration procedure to get an ID in order to make the mobile user ‘disappear’ with the new ID (refer to re-confusing protocol in Section III-C). Whoever initiates the protocol, the ID must be passed to the Rover in order for the mobile device to get authenticated at access points.

B. Controlled Connection Protocol

Once an R obtains an ID, the mobile device can use the controlled connection protocol to get access to the wireless infrastructure via an access point. The procedure is the following. First, the R sends an access request by presenting its

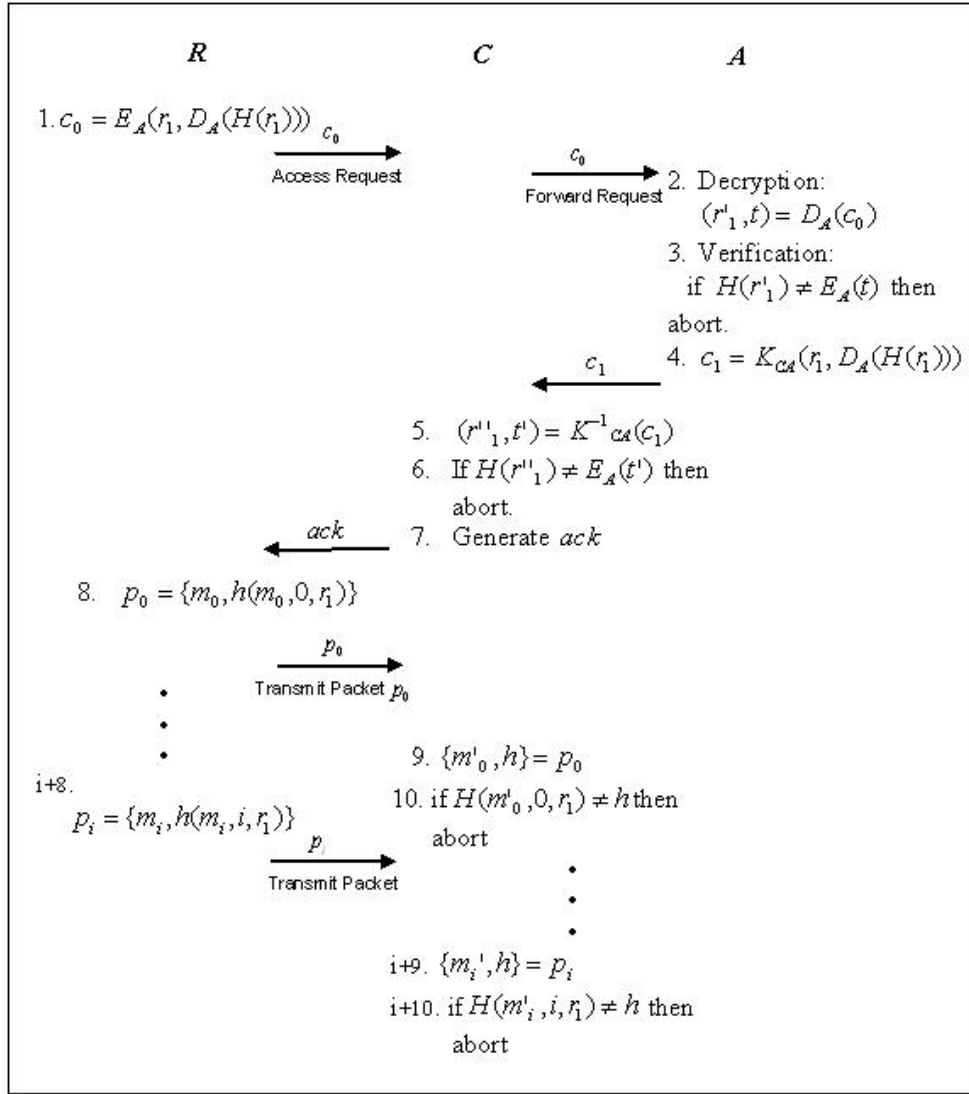


Fig. 5. Controlled connection protocol

authorized-anonymous-ID (encrypted with the Administrator's public key) to the C at the access point. Then the C forwards the message to its A for verification. The A decrypts the message, verifies the authenticity of the embedded authorized-anonymous-ID, signs the ID (if it is a valid one), encrypts the ID and the signature with the key shared by the C , and sends the encrypted message back to the C . Once the C receives the encrypted message from the A , it decrypts it and checks the signature signed by the A and send an "ack" to the R if the signature is valid. Thereafter, the R and the C share the ID as a secret for packets authentication, and only successfully authenticated packets can get through the access point to the Internet. This protocol is outlined in Fig. 5.

C. Improvements

The basic protocols presented in Sections III-A and III-B can be improved by the following methods.

- **Re-confusion:**

It is known that the longer an ID exists, the higher the probability of exposing the association between the ID and the corresponding mobile user. To mitigate this problem, we propose a method called "re-confusion", the objective of which is to generate a new authorized-anonymous-ID to replace the old authorized-anonymous-ID. Fig. 6 outlines our protocol for the re-confusion method.

Specifically, the process of re-confusion works as below. First, an R sends the Administrator a request (encrypted with the public key of the Administrator) for a new authorized-anonymous-ID. Different from the registration protocol, in which a real identification (e.g., a public key certificate) is required to be presented, a request for a new authorized-anonymous-ID in re-confusion contains 1) one of the mobile user's current or previous authorized-anonymous-IDs, 2) a random number multiplied by a

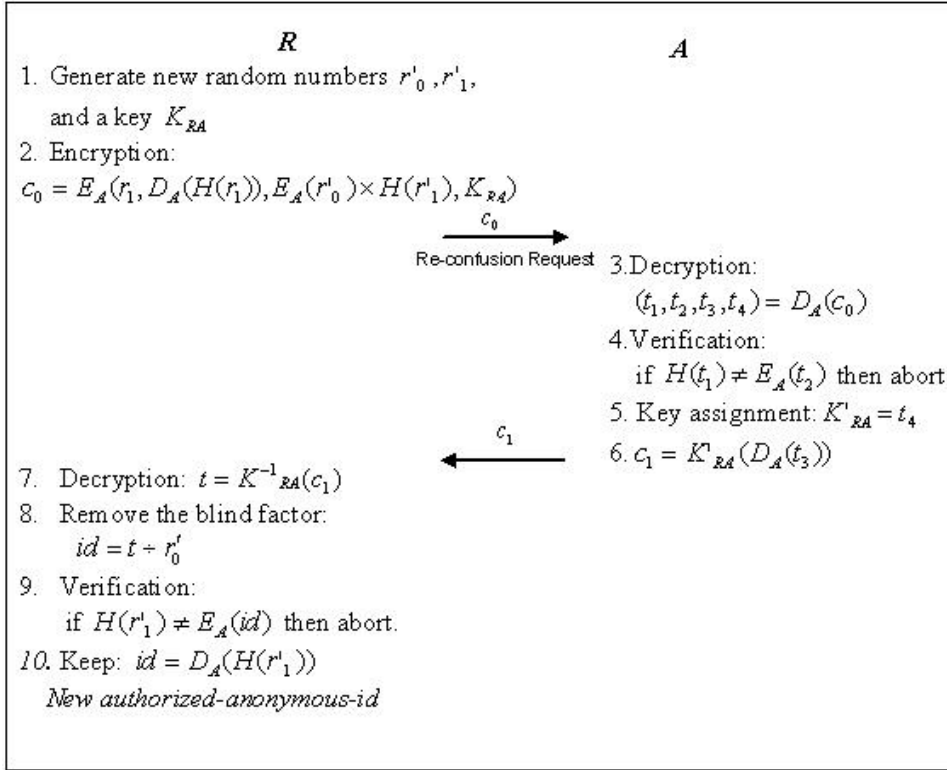


Fig. 6. Re-confusion protocol

blind factor, and 3) symmetric encryption key suggested for this communication session between the R and the A . After a successful verification of the presented ID, the A signs blind signature on the random number, encrypts it with the suggested key, and sends it back to the requesting Rover R . The R decrypts the message from A , removes the blind factor, and gets the new authorized-anonymous-ID. The nice feature of our re-confusion protocol is that any disclosure of the previous ID wouldn't compromise the anonymity of the new ID.

- **Access Authorization Revocation:**

It is not desirable from the administration perspective, that an authorized-anonymous-ID enables a mobile device to have an *eternal* right to access the infrastructure. Hence, an administration may want to have a function that can revoke or invalidate an issued authorized-anonymous-ID. One way to add this revocation function to our protocol family is that the Administrator A periodically expires and changes its own keys for access authorization. The anonymous IDs signed by the revoked keys will no longer valid for authentication. But this solution has a drawback: the mobile users need to periodically update their anonymous IDs, which introduces much communication overhead if the keys of the Administrator expire too fast. Another solution is to attach an expiration time-stamp with the ID. However, the expiration time-stamp should not be unique to the mobile user; otherwise, the

unique association between the expiration time-stamp and the ID, can reveal the identity of a mobile user.

IV. CONCLUDING REMARKS

In this paper, we investigated the problem of protecting location privacy of mobile users in the setting of ubiquitous computing. We pointed out that location privacy protection is particularly challenging due to the different requirements imposed by the administration and mobile users. To address this issue, we proposed an authorized-anonymous-ID based scheme. In our scheme, an authorized-anonymous-ID is created by the 'blind signature' technique and is used to replace the real ID of an authorized mobile device. With authorized-anonymous-IDs, we designed an architecture that is able to provide the mobile users with complete control over their location privacy while yet allowing the administration to authenticate the legitimate mobile users.

Our future work will focus on theoretical analysis of the security of this set of protocols. In addition, the system has been built on the Wireless-Andrew network and we plan to generalize the protocols for heterogeneous networking environments to accommodate various networking technologies.

REFERENCES

- [1] D. Chaum, "Blind signatures for untraceable payments," in *Proc. of Crypto'82*, 1982.
- [2] Q. He, B. Liu, A. Pennington, D. Siewiorek, P. Khosla, and Z. Su, "WaveGuard: secure location service for wireless andrew," *International Conference on Wireless Communications*, 2001.

- [3] Q. He, P. Khosla, and Z. Su, "A Practical Study on Security of Agent-based Ubiquitous Computing," in *Proc. AAMAS'02 Deception, Fraud, and Trust in Agent Societies workshop*, 2002.
- [4] A. Hills, "Wireless andrew," *IEEE Spectrum*, vol. 36, no. 6, pp. 49–53, June 1999.
- [5] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: keyed-hashing for message authentication," *IETF RFC 2104*, Feb. 1997.
- [6] M. Weiser, "Some computer science issues in ubiquitous computing," *Communication of The ACM*, July 1993.