A Secure Location-Based Service Reservation Protocol in Pervasive Computing Environment

Konidala M. Divyan*, and Kwangjo Kim*

*International Research Center for Information Security (IRIS), Information and Communications University (ICU), Daejeon, Republic of Korea

Abstract

Nowadays mobile phones and PDAs are part and parcel of our lives. By carrying a portable mobile device with us all the time we are already living in partial Pervasive Computing Environment (PCE) that is waiting to be exploited very soon. One of the advantages of pervasive computing is that it strongly supports the deployment of Location-Based Service(s) (LBSs). In PCE, there would be many competitive service providers (SPs) trying to sell different or similar LBSs to users. In order to reserve a particular service, it becomes very difficult for a low-computing and resource-poor mobile device to handle many such SPs at a time, and to identify and securely communicate with only genuine ones. Our paper establishes a convincing trust model through which secure job delegation is accomplished. Secure Job delegation and cost effective cryptographic techniques largely help in reducing the burden on the mobile device to securely communicate with trusted SPs. Our protocol also provides users privacy protection, replay protection, entity authentication, and message authentication, integrity, and confidentiality. This paper explains our protocol by suggesting one of the LBSs namely "Secure Automated Taxi Calling Service".

I. Introduction

Pervasive computing [1][2] or Ubiquitous computing means availability of computing and communication resources whenever and wherever we are. A Pervasive Computing Environment (PCE) is saturated with devices, which compute and communicate "for", "on behalf" and "along with" the users in order to provide some useful services. The user should obtain and make use of such services seamlessly and comfortably, but should never be burdened with instructions and interfaces on how to handle those devices.

An example of a "closed PCE" can be a meeting room (Smart Space) that automatically takes meeting minutes, takes commands from attendees and applies them differently depending on who spoke, and provides security based on face and voice identification, *etc.* [3]. Such an environment involves computations and communications. There are computers and sensors "everywhere" in devices and a high degree of communication among them. It consists of low and high-computing devices, and wired and wireless communications. A user in such an environment need to just concentrate on his work and let the devices do their job seamlessly.

Nowadays mobile phones and PDAs are part and parcel of our lives. We are now able to communicate whenever and from wherever we are. As a result by carrying a portable mobile computing and communication device with us all the time we are already living in partial "open space PCE". Apart from helping us to communicate, these mobile devices would very soon allow us to interact

with other smart devices around us , thus supporting an open PCE. This requires secure integration of trusted devices. One of the advantages of pervasive computing environment is that it would lead to the growth of new breed of service providers (SPs) who would offer Location-Based service(s) (LBSs).

Recently 3G (3rd Generation) [8][9] GPS (Global Positioning Service) enabled mobile phones [12][13][14][15] and PDAs [11] are being introduced in to the consumer market. Such mobile devices greatly assist the open PCE by allowing users to determine their location at the touch of a button, and download location specific information like graphical maps and other useful services. By sending out our current location information, SPs can provide us with services "related to" and "available at" that location.

An example of a "Reserve First and Access Later" type LBS is as follows: You are approaching a food court. SPs or a SP having the knowledge of your current location will offer you with a list of restaurant names, their menus and also an option to reserve a table at the restaurant of your choice (even before you could reach the food court). If you have your diet plan and your food preferences stored in your mobile device the SP can immediately choose the appropriate restaurant thus reducing the amount of interactions needed to select one by yourself. The situation becomes more complicated if you are approaching a location, which has a food court, a movie hall, a discount store and many more shops. The mobile device on behalf of its owner may need to communicate with more than one SP. Communicating with many SPs, identifying and authenticating genuine ones, checking the validity of their digital certificates and signatures if in case they are using Public-key Infrastructure (PKI) [7], securing the entire transaction and protecting the owner's privacy, *etc* cannot be handled alone by the low-computing and resource-poor mobile device. It would create a huge burden on the mobile device and is certainly not user-friendly.

One other feature of PCE is "job delegation" among smart devices. A low-computing device can delegate its job to a trusted high-computing device/entity. By establishing an efficient and a convincing trust model, it would be lot easier for the mobile device to delegate its work to a nearby trusted high-computing and resource-rich entity. This entity behaving like a "proxy" [23] receives the request and preferences and processes the same on behalf of the mobile device. The details of this feature will be discussed later.

Our paper establishes a convincing trust model through which secure job delegation is accomplished. Secure Job delegation and cost effective cryptographic techniques largely help in reducing the burden on the mobile device to securely communicate with trusted SPs. Our protocol also provides users privacy protection, replay protection, entity authentication, and message authentication, integrity, and confidentiality. This paper explains our protocol by suggesting one of the LBSs namely "Secure Automated Taxi Calling Service".

II. LBS: A Secure Automated Taxi Calling Service

1. Motivational Scenarios

In this section we describe three scenarios among many such scenarios, which motivated us to suggest a LBS namely "A Secure Automated Taxi Calling Service".

Scenario 1

Alice is in a very new locality, trying to catch a taxi. She wants to call a taxi but she does not know the nearest taxi call center's telephone number. In big cities there are many taxi call centers which operate area wise, so in order to call a taxi you should be aware of the telephone number of call center operating in that particular area. It will be difficult to remember or have access to such taxi call centers phone numbers.

Scenario 2

Bob is touring a foreign country. He calls a

taxi call center but struggles to inform his current location and destination details, as he cannot speak the local language.

Scenario 3

In the current taxi calling system, the user through a telephone call directly interacts with the call center. As a result some of the call centers in order to provide quick and personalized services to returning customers, maintain travel records (travel history) and detailed profiles of its customers like their phone numbers, names, and addresses of frequently visited places (home, office, shopping malls, etc). But this is in fact privacy intrusion and violation.

2. Protocol Overview

Our simple, efficient and cost effective protocol addresses the above-mentioned concerns. This protocol consists of four entities: *Users* (U), *Mobile Communications Service Provider* (MCSP) like KT, SKT, AT&T, BT, Vodafone, *etc*, *Taxi Control Center* (CC) and the *Taxis* (T). A user using his GPS enabled mobile phone detects his current location. He then securely communicates his current location to *MCSP* and requests for a list of services available at that location. *MCSP* takes responsibility on behalf of users to select, identify, and authenticate the genuine SPs and also maintains a list of services they offer at a particular location. It updates this list as and when required.

MCSP sends the available services list to the user. User selects "Taxi Calling" service from the list. He detects his current location and also identifies the destination he has to reach on an interactive map displayed in his mobile phone. He securely communicates these details to *MCSP* as an input to the taxi calling service. Coz of this Alice need not remember the phone numbers of many taxi call centers that operate area wise and foreigner Bob need not speak the local language to convey his current location and destination details. The communications between *MCSP* and the user could be via SMS (Short Messaging Service) messages, MMS (Multimedia Messaging Service) messages, XML messages [23] or a more efficient data communication method employed by *MCSP*.

MCSP behaving like a "proxy" processes the request on behalf of the user, thus greatly reducing the burden on the user's mobile phone. *MCSP* identifies and authenticates the genuine *CC* and securely sends only the current location and destination details (but not the identity of the user) to *CC*. This protects the privacy of the user. *CC* cannot maintain the user's travel record and his detailed profile, as it does not know to whom the service is being offered to. *CC*, which keeps track of all its associated taxis, securely communicates with them and dispatches an available taxi closest to the user's current location.

3. Security Requirements

This section describes the various security requirements of our protocol

Users Privacy Protection: privacy is at a greater risk in PCE where users interact with many smart devices around them. Users are prone to revealing their location and identity information to such devices. This information could allow SPs to generate detailed profiles of the user, his buying interests and trace all his actions. As a result restricted access to users personal data [18] should be provided by all protocols executing in PCE.

Replay Protection: In a replay attack an adversary records a communications session and replays the entire session, or a portion thereof, at some later point in time. Replay protection prevents an attacker to impersonate as a genuine entity (user, MCSP, SP, etc) or to waste the resources of an entity by re-initiating old (expired) transactions.

Entity Authentication or Identification: corroboration of the identity of an entity (e.g., a person, a computer terminal, a mobile device, etc.). This allows users to interact with only known or trusted entities in PCE and prevents malicious entities trying to impersonate as genuine ones.

Message Authentication: corroborating the

source of information; also known as data origin authentication. With many smart devices all round us we need to know whether the message came from a trusted source or not.

Message Integrity: ensuring information has not been altered by unauthorized or unknown means.

Message Confidentiality: keeping information secret from all but those who are authorized to see it. Only trusted entities should be able to receive and understand your messages.

III. Protocol Description

1. Notations

We state all the notations used in this paper in Table 1. A brief description of an AVL system, which would be referred in the subsequent sections is as follows:

AVL System: Automatic Vehicle Location system includes Global Positioning System (GPS) with Geographical Information System (GIS). It provides precision time and position data for a vehicle or its trailer to a regional or national control center that operates and manages fleet movements. The GIS element of the AVL system provides fleet managers with on-the-spot information regarding a vehicle and its driver's whereabouts [4].

Table 1: Notations

$MCSP$	Mobile Communications Service Provider. It setups mobile communications infrastructure and provides mobile communication services to its subscribers.
ID_m	Identity of $MCSP$
U	Subscribers of $MCSP$ who have registered for Taxi Calling Service
u	One particular user among U considered for easy explanation of the protocol. $u \in U$
MP_u	GPS enabled mobile phone of u
PN_u	Mobile Phone Number of u
$CLocn_u$	Current location of u
$Dest_u$	Destination to be reached by u
CC	Taxi Control Center, it controls, communicates with, and keeps track of all its associated taxis
ID_c	Identity of CC
T	All the taxis associated with CC
t	One particular taxi among T considered for easy explanation of the protocol $t \in T$
RN_t	Registration Number of t
Tmr_{tu}	Indicates the time that would be taken by t to reach $CLocn_u$.
MK_{um}	Master Secret/Symmetric Key shared between u and $MCSP$
K_{um}	Session Key generated by u and $MCSP$ using MK_{um}
$DCert_m$	Digital Certificate of $MCSP$
SK_m	Private Key of $MCSP$
PK_m	Public Key of $MCSP$
$DCert_c$	Digital Certificate of CC
SK_c	Private Key of CC
PK_c	Public Key of CC
K_{tc}	Secret/Symmetric Key shared between t and CC
Sid_x	Unique Service ID of service x
R_{id}	Unique Random Transaction Reference ID

r	Unique Random Number
ts_x	Timestamp generated by an entity x
Ack_1	Your request is being processed, please wait
Ack_2	The following taxi has been dispatched
$M_x = \{\}$	Message sent in open by an entity x. This message is visible to everyone connected to the network.
$S_x(M)$	Digitally Sign a message M with private key x
$E_x(M)$	Encrypt message M with a shared key or a public key x
$H()$	One Way Hash Function like SHA-1 (Secure Hash Algorithm) [19], [21]
$H(M)$	Hash value or message digest of a message M
$H_x(M)$	Keyed Hash function on message M with shared key x

2. Trust Model and Setup Phase

This section describes the trust model and the setup phase needed to execute our protocol. In PCE many smart devices, which could be genuine or malicious compute and communicate with each other. To ensure secure transactions, establishing an efficient and a convincing trust model is very much required in PCE. Also with existence of such a trust model, it would be lot easier for the mobile device to delegate its work to a nearby trusted high-computing and resource-rich entity (*MCSP*), which processes the request on behalf of the mobile device.

1) Trust between u and $MCSP$

u installs software in his MP_u. The software is required to execute various procedures involved in this protocol. u can either download the software through the $MCSP$'s official website or by approaching the nearest $MCSP$'s licensed customer service center. The software helps to generate MK_{um}, which will be a long-term shared key between u and $MCSP$. MK_{um} and ID_m is stored in the MP_u. MK_{um} is also stored in the database of $MCSP$, probably with PN_u being the index or the reference for such a database entry. As a result for all Us, $MCSP$ generates a unique master shared secret key.

Why Trust $MCSP$?

In the current mobile communications paradigm we already trust $MCSP$ a lot, as it handles all our voice and data communications. It maintains a record of each subscriber's call details (incoming and outgoing call numbers, talk time, etc), contact information (home and office addresses, etc), social security number, bank account and credit card details, etc. It even has the capability to easily determine our current location and tap in to our communications. But what protects us from $MCSP$ turning hostile is that it has to very strictly adhere to and follow legal, security and privacy policies imposed by the law. Thus so far we have little problems in trusting $MCSP$. Our protocol extends this trust in $MCSP$ to secure LBS transactions. This approach is very practical and easily deployable, as the current mobile communications infrastructure is widely spread and highly stable. This avoids the need to separately setup trusted proxies infrastructure to support LBSs. $MCSP$ can play the roles of both a Trusted Third Part (TTP) and a trusted proxy in open PCE, where smart devices constantly interact via wireless communications.

It is very convenient for MP_u to trust one single entity $MCSP$ rather than validating many SPs and then trusting them. It is

desirable to have our details like preferences

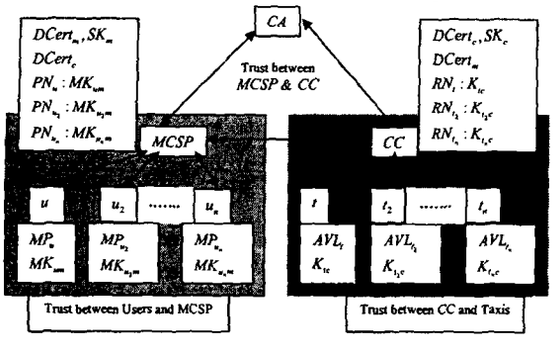

Figure 1: Trust Model and Setup Phase

and requests passed on to one single trusted entity rather than having our details stored with many SPs, who may be genuine or malicious.

2) Trust between $MCSP$ and CC

For commercial gains both $MCSP$ and CC sign a business contract and mutually agree to provide this Automated Taxi Calling Service. A similar deal can be made with other SPs, whom $MCSP$ trusts. To secure the communications between $MCSP$ and CC during the protocol execution we assume the existence of a trusted Public Key Infrastructure (PKI). $MCSP$ obtains $DCert_m$, and SK_m. Similarly CC also obtains $DCert_c$ and SK_c from a Certificate Authority (CA). We assume that $MCSP$ and CC have large computing resources. During the protocol execution they can easily, and very efficiently perform expensive tasks like public-key encryption and decryption, and digital certificate and signature verifications. $MCSP$ stores ID_c and $DCert_c$. Similarly CC stores ID_m and $DCert_m$.

3) Trust between CC and T

CC generates a shared-key K_{tc}, which will be used for securing the communications between CC and t. K_{tc} can be stored in the AVL system available in t and is also stored in the database of CC, probably with RN_t being the index or the reference for such a database entry. As a result all Ts receive a unique shared secret key generated by CC.

Figure 1 depicts the Trust Model and the Setup Phase.

Analysis: One may feel that by solely utilizing PKI implementations throughout the protocol we can avoid the considerable overhead involved in managing the unique shared secret keys of all U and T at $MCSP$ and CC respectively. But since $MCSP$ and CC are assumed to have large computing resources, storing large number of shared secret keys would not be much of a burden on them. Also, this model by using symmetric-key cryptosystem, avoids the expensive PKI implementations at u and at t, as they carry low-computing and resource-poor mobile devices. It is very well proved in [5] that symmetric key implementations are much simpler, faster and less computationally expensive than PKI implementations.

3. Periodic Taxi Information Update Phase

All T via the AVL system periodically and continuously communicate certain details with CC. The frequently sent update information includes RN_t, availability status (vacant or not), and current location, *etc.* Currently most of the taxi control centers employ this method to keep track of their associated taxis. CC stores this update information in its database, probably with RN_t being the index or the reference for such entries.

4. Request Processing Phase

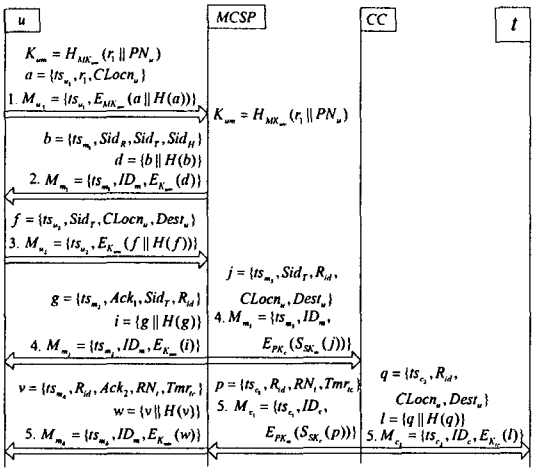

Figure 2: Request Processing Phase

Figure 2 depicts this phase. In this phase u using his MP_u requests $MCSP$ for a list of services available at $CLocn_u$. $MCSP$ responds with a list of services available at $CLocn_u$. u selects Taxi Calling Service and request $MCSP$ for a taxi to be sent to his $CLocn_u$. This request is securely communicated to $MCSP$, which then process the request on behalf of MP_u. $MCSP$ securely sends only $CLocn_u$ details (not the identity of u) to CC thus protecting the privacy of u. CC securely communicates with T in order to dispatch an available t closest to $CLocn_u$.

1) Step 1

u enters the secret PIN (Personal Identification Number) to authenticate himself to MP_u .This prevents unauthorized communications in the event MP_u is stolen or being tampered with. The above option is currently available in all the mobile phones. u using MP_u detects his $CLocn_u$. MP_u sends a Message M_{u_1} to $MCSP$ requesting a list of available services at $CLocn_u$. MP_u also generates K_{um} using the unique random number r_1.

$$a = \{ts_{u_1}, r_1, CLocn_u\}$$

$$M_{u_1} = \{ts_{u_1}, E_{MK_{um}}(a \| H(a))\} \quad (1)$$

$$K_{um} = H_{MK_{um}}(r_1 \| PN_u) \quad (2)$$

2) Step 2

$MCSP$ receives M_{u_1} and also PN_u as a part of incoming message information. $MCSP$ checks PN_u and retrieves the corresponding MK_{um} from its database and decrypts M_{u_1}. Thus $MCSP$ obtains r_1 and $CLocn_u$. Using r_1 $MCSP$ generates K_{um} and sends message M_{m_1} to MP_u. M_{m_1} contains a list of services available at $CLocn_u$. $MCSP$ takes responsibility on behalf of MP_u to select, identify, and authenticate the genuine service providers and maintain a list of the services they offer at a particular location. $MCSP$ updates this list as and when required

$$K_{um} = H_{MK_{um}}(r_1 \| PN_u) \quad (3)$$

$$b = \{ts_{m_1}, Sid_R, Sid_T, Sid_H\}$$

$$d = \{b \| H(b)\}$$

$$M_{m_1} = \{ts_{m_1}, ID_m, E_{K_{um}}(d)\} \quad (4)$$

3) Step 3

u via MP_u receives message M_{m_1}. MP_u checks ID_m and retrieves the recently

generated K_{um} and decrypts M_{m_1}, which can be displayed as follows:

The list of services available at $CLocn_u$ *is*

Sid_R : *Restaurant Information Service*

Sid_T : *Taxi Calling Service*

Sid_H : *Hotel Information Service*

Please select your choice

Since u requires Taxi Calling Service, he selects Sid_T. u using MP_u detects $CLocn_u$ and identifies $Dest_u$, on an interactive map displayed in his mobile phone. MP_u sends Message M_{u_2} to $MCSP$ selecting the service ID Sid_T.

$$f = \{ts_{u_2}, Sid_T, CLocn_u, Dest_u\}$$

$$M_{u_2} = \{ts_{u_2}, E_{K_{um}}(f \,||\, H(f))\} \quad (5)$$

4) Step 4

$MCSP$ receives M_{u_2} and also PN_u as a part of the incoming message information. $MCSP$ checks PN_u and retrieves the corresponding recently generated K_{um} from its database and decrypts M_{u_2}. $MCSP$, looking at Sid_T creates a R_{id}. Unique R_{id} plays a vital role in identifying one entire Taxi Calling Service transaction for u. $MCSP$ sends message M_{m_2} to MP_u.

Ack_1 = *Your request is being processed, please wait*

$$g = \{ts_{m_2}, Ack_1, Sid_T, R_{id}\}$$

$$M_{m_2} = \{ts_{m_2}, ID_m, E_{K_{um}}(g \,||\, H(g))\} \quad (6)$$

MP_u receives and decrypts M_{m_2} using K_{um} and obtains R_{id}. u can now use R_{id} as a reference to easily and quickly cancel this request or update his $CLocn_u$ at a later stage (before the taxi could reach him).

Also $MCSP$ simultaneously sends message M_{m_3} to CC. It can be noticed that PN_u is never sent and therefore CC can never know whose location details are being sent. This protects the privacy of u.

$$j = \{ts_{m_3}, Sid_T, R_{id}, CLocn_u, Dest_u\}$$

$$M_{m_3} = \{ts_{m_3}, ID_m, E_{PK_c}(S_{SK_m}(j))\} \quad (7)$$

5) Step 5

CC receives M_{m_3} and decrypts it using SK_c and verifies the signature using PK_m. PK_m is obtained from $DCert_m$ which is stored in the database of CC during the setup phase. Now CC knows $CLocn_u$ and $Dest_u$. By comparing $CLocn_u$ and already available current location details of all T (via the Periodic Taxi Information Update Phase), CC detects, selects and dispatches t that is nearest to $CLocn_u$. CC updates its database by including some of the reserved taxi's details like R_{id}, RN_t, driver's name, date, and time. This database entry may be used as a receipt for this particular transaction or for any payment transactions at a later stage. CC

sends message M_{c_1} to $MCSP$.

$$p = \{ts_{c_1}, R_{id}, RN_t, Tmr_{tc}\}$$

$$M_{c_1} = \{ts_{c_1}, ID_c, E_{PK_m}(S_{SK_c}(p))\} \quad (8)$$

Simultaneously CC sends message M_{c_2} to t. M_{c_2} is encrypted using K_{tc}.

$$q = \{ts_{c_2}, R_{id}, CLocn_u, Dest_u\}$$

$$M_{c_2} = \{ts_{c_2}, ID_c, E_{K_{tc}}(q \| H(q))\} \quad (9)$$

t receives M_{c_2}, checks ID_c and retrieves corresponding K_{tc} and decrypts M_{c_2}. Now t knows $CLocn_u$ and $Dest_u$, which are sufficient to pick up u

$MCSP$ receives M_{c_1} (see equation (8)) and decrypts it using SK_m and verifies the signature using the PK_c. PK_c is obtained from $DCert_c$ which is stored in the database of $MCSP$ during the setup phase. $MCSP$ checks for the received R_{id} in its database and retrieves the corresponding PN_u. Using this PN_u, $MCSP$ sends message M_{m_4} to MP_u.

$Ack_2 =$ *The following taxi has been dispatched*

$$v = \{ts_{m_4}, R_{id}, Ack_2, RN_t, Tmr_{tc}\}$$

$$w = \{v \| H(v)\}$$

$$M_{m_4} = \{ts_{m_4}, ID_m, E_{K_{um}}(w)\} \quad (10)$$

MP_u receives M_{m_4} and decrypts it using K_{um}. MP_u stores R_{id}, and RN_t, which can used as a receipt for this particular transaction or for any payment transactions at a later stage. u reads Tmr_{tu} and waits for t.

5. Pickup Phase

t reaches $CLocn_u$. u has already obtained RN_t via the message M_{m_4} (see equation (10)). The message M_{m_4} has been securely communicated to u, and no one other than the u, $MCSP$, CC, and t know RN_t. Looking if registration number of the arrived taxi equals RN_t, u can identify, authenticate and trust the arrived taxi as t else the arrived taxi is not the right taxi dispatched by CC.

6. Request Cancellation Option

This option allows u to cancel his request. u obtains R_{id} right at the beginning of the protocol via message M_{m_2} (see equation (6)). At anytime before u receives the acknowledgement message M_{m_4} (see equation (10)), he can send message M_{u_3} to $MCSP$ to cancel his request.

$$y = \{ts_{u_3}, R_{id}, Cancel\}$$

$$M_{u_3} = \{ts_{u_3}, E_{K_{um}}(y \| H(y))\} \quad (11)$$

$MCSP$ receives M_{u_3} and securely communicates it to CC and CC in turn securely communicates M_{u_3} to t thus terminating this particular Taxi Calling Service request from u. If u cancels the request after receiving M_{m_4}, he may have to pay cancellation charges as per the terms and

conditions of CC. In such a situation $MCSP$ pays the cancellation charge to CC and u settles this amount with $MCSP$ via his monthly mobile phone bill.

IV. Analysis

This section describes the security strength of our protocol and shows how it satisfies the security requirements (see Sect. 2.3) mentioned above. Before proceeding with security analysis this section presents some of the cryptographic primitives utilized in our protocol

1. Cryptographic Primitives

1) Key freshness [24]:

A key is fresh if it can be guaranteed to be new, as opposed to possibly an old key being reused through the actions of either an adversary or authorized party.

2) Timestamps [16]:

Timestamps may be used to provide timeliness and uniqueness guarantees, to detect message replay. They may also be used to implement time-limited access privileges, and to detect forced delays. Timestamps function as follows: The party originating a message obtains a timestamp from its local (host) clock, and cryptographically binds it to a message. Upon receiving a time-stamped message, the second party obtains the current time from its own (host) clock, and subtracts the timestamp received. The received message is valid provided: (1) The timestamp difference is within the acceptance window (a fixed-size time interval, e.g., 10 milliseconds or 20 seconds, selected to account for the maximum message transit and processing time, plus clock skew); and (2) The latest (valid) timestamp is used by the sender (in this case the verifier accepts only strictly increasing time values). The security of timestamp-based verification relies on use of a common time reference. This requires that host clocks be available and both "loosely synchronized" and secured from modification.

3) Hash function [6][19][20][21]:

A hash Function is a computationally efficient function mapping binary strings of arbitrary length to binary strings of some fixed length, called hash-digests. The basic idea is that a hash-value serves as a compact representative of an input string. To be of cryptographic use, a hash function H is typically chosen such that it is computationally infeasible to find two distinct inputs which hash to a common value (i.e., two colliding inputs x and y such that $H(x)=H(y)$), and that given a specific hash-value y, it is computationally infeasible to find an input (pre-image) x such that $H(x)=y$.

4) Message Authentication and integrity using a MAC [6]:

Message Authentication Codes (MACs) are designed specifically for applications where data integrity (but not necessarily privacy) is required. The originator of a message x computes a MAC $H_k(x)$ over the message using a secret MAC key k shared with the intended recipient, and sends both (effectively $x \| H(x)$). The recipient determines by some means (e.g., a plaintext identifier field) the claimed source identity, separates the received MAC from the received data, independently computes a MAC over this data using the shared MAC key, and compares the computed MAC to the received MAC. The recipient interprets the agreement of these values to mean the data is authentic and has integrity that is, it originated from the other party, which knows the shared key, and has not been altered in transit.Hash functions are considered to be cheap i.e., they are computationally less expensive. But they provide good security features. As a result they can be easily implemented and executed in low-computing and resource-poor mobile devices

5) Message Authentication and integrity using encryption and a MDC [6]:

If both confidentiality and integrity are required, then the following data integrity technique employing an m-bit MDC (Manipulation Detection Code) H may be used. The originator of a message x computes a hash value $h = H(x)$ over the message, appends it to the data, and encrypts the augmented message using a symmetric encryption algorithm E with shared key k, producing ciphertext $c = E_k(x \| H(x))$. This is transmitted to a recipient, who determines (e.g., by a plaintext identifier field) which key to use for decryption, and separates the recovered data x' from the recovered hash h'. The recipient then independently computes the hash $H(x')$ of the received data x', and compares this to the recovered hash h'. If these agree, the recovered data is accepted as both being authentic and having integrity. The intention is that the encryption protects the appended hash, and that it be infeasible for an attacker without the encryption key to alter the message without disrupting the correspondence between the decrypted plaintext and the recovered MDC.

2. Security Analysis

This section describes the security analysis of the "Request Processing Phase" (see Sect. 3.4) described above.

1) Step 1 (see Sect. 3.4.1)

In this step we used Message Authentication and integrity using encryption and a MDC (see Sect. 4.1.5). M_{m_1} (see equation (1))is encrypted using MK_{um}. Only $MCSP$ and u know MK_{um} and no one else can encrypt or decrypt M_{m_1} other than them. Based on this fact u and $MCSP$ can identify and authenticate each other and also provide message authentication and confidentiality. The use of one-way hash function in M_{m_1}, which is described in detailed below, provides message integrity.

Unlike wire-based communication systems where lines can be put up until they obliterate the sky, each wireless system requires its own unique slice of the limited radio spectrum. For practical use, this spectrum is physically limited from a few hundred kHz to more than 1000 MHz. In order to get the most out of its assigned slice of the radio spectrum, a wireless system must be carefully timed and synchronized. The timing required can be sub-microsecond precision for base-stations located across very large geographical areas [22]. In other words, the mobile communications infrastructure (like TDMA technology) has the clocks of the mobile phones synchronized with the clock of $MCSP$. This aspect greatly supports the use of timestamp (see Sect. 4.1.2) as nonce to prevent replay attacks (see Sect. 2.3).

Long-term master shared key MK_{um} is used only once at the beginning of the session to prevent key compromise due to extensive use. If we use the same key again and again to encrypt the communications an attacker can maintain a table of ciphertexts generated under the same key and can mount cryptanalytic attack. Instead MK_{um} is used to generate a short-term session key K_{um}. K_{um} is used to encrypt rest of the communications between u and $MCSP$. This provides entity authentication and message authentication and confidentiality. K_{um} is unique for each session, thus providing key freshness (see Sect. 4.1.1). K_{um} is generated from keyed hash function (MAC, see Sect. 4.1.4) of a unique random number r_1 concatenated with PN_u. In every session or periodically u generates a new random number and sends it to $MCSP$. Whenever $MCSP$ receives a new r it knows

that it has to generate a new session key. Even if K_{um} is compromised, no attacker can derive or generate MK_{um}, since the function $H_{MK_{um}}(r_1 \| PN_u)$ is non-reversible (see Sect. 4.1.3 and 4.1.4). And also no one else other than u and $MCSP$ can generate K_{um} as they only know MK_{um} .

2) Step 2 (see Sect. 3.4.2)

In this step it is very essential to receive the users phone number PN_u as a part of the incoming message information, if not $MCSP$ cannot fetch the corresponding MK_{um} and thus rejects M_{u_1} (see equation (1)). This procedure partially prevents an adversary attempting to mount replay (see Sect. 2.3) attack using spoofed phone numbers. $MCSP$ first checks whether the received spoofed phone number has registered for LBSs, if not it rejects M_{u_1}. By chance if the spoofed phone number is registered, then its corresponding shared key cannot decrypt M_{u_1} correctly. Reason being M_{u_1} is encrypted by MK_{um} and also every unique phone number has it corresponding unique master shared key and generated session keys. It is very difficult for an attacker to obtain spoofed phone numbers and as well as their corresponding master shared keys and session keys.

Consider the equation (1) below:

$$a = \{ts_{u_1}, r_1, CLocn_u\}$$

$$M_{u_1} = \{ts_{u_1}, E_{MK_{um}}(a \| H(a))\} \quad (1)$$

$MCSP$ decrypts M_{u_1} using MK_{um}, this proves that M_{u_1} has indeed come from u (entity authentication and message authentication). $MCSP$ Computes $H(a)$ and verifies the same with the received hash value. If both the hash values equal then M_{u_1} would be accepted else it'll be rejected. This shows whether M_{u_1} has been modified or not (by an attacker) on its way to $MCSP$ (message integrity, see Sect. 4.1.3 and 4.1.5).

$MCSP$ verifies whether ts_{u_1} is the latest and within the acceptance window (see Sect. 4.1.2), if yes M_{u_1} is accepted else rejected. $MCSP$ also verifies whether ts_{u_1} inside a equals ts_{u_1} sent in open (see equation 1). If both match then M_{u_1} would be accepted else it'll be rejected. This prevents replay attack (see Sect. 2.3). An attacker can modify ts_{u_1} to the latest timestamp say ts'_{u_1} and reply the rest of the message. But this can be detected since ts'_{u_1} does not match with ts_{u_1} inside a. An attacker cannot modify ts_{u_1} inside a to ts'_{u_1} as he does not know MK_{um} to decrypt and obtain a. Thus this kind of replay attack fails.

Using r_1, $MCSP$ generates K_{um} and sends an encrypted message M_{m_1} to MP_u. K_{um} has a short life span. If further communications from u (after step 2 (see Sect. 3.4.2)) do not reach $MCSP$ within a desired threshold time limit (calculated from ts_u), K_{um} is discarded. This fact is duly notified to MP_u. u has to start a new session and create a new K_{um} using a new r. This increases the protocol security and prevents key compromise.

3) Step 4 (see Sect. 3.4.4)

According to our Trust Model, $MCSP$ and CC having enough computing resources can easily perform PKI implementations. $MCSP$ sends message M_{m_3} (see equation (7))to CC. M_{m_3} is first digitally signed by $MCSP$ using its SK_m, this provides message authentication and message integrity. Only $MCSP$ can sign M_{m_3}, since no one other than $MCSP$ possesses SK_m. CC verifies the above signature using PK_m. PK_m is obtained from $DCert_m$ which is stored in the database of CC during the setup phase. M_{m_3} is then encrypted with the public key PK_c of CC. PK_c is obtained from $DCert_c$ which is stored in the database of $MCSP$ during the setup phase. Only CC can decrypt M_{m_3}, since no one other than CC possesses SK_c, this provides message confidentiality.

4) Step 5 (see Sect. 3.4.5)

CC sends message M_{c_2} (see equation (9)) to t.

$$M_{c_2} = \{ts_{c_2}, ID_c, E_{K_{tc}}(q \| H(q))\}$$

M_{c_2} is encrypted using shared-key K_{tc}. In here, we can also consider the concept of long-term master shared-key and short-term session keys as in the case of securing communications between u and $MCSP$. This would prevent key compromise and provides key freshness as described above.

The communications between u - $MCSP$, $MCSP$ - CC and CC - T, fall under the above-mentioned security analysis.

V. Related Works

Most of the pervasive computing projects [3] [27-30] being carried out at various universities and research institutes deal in closed pervasive computing environments (PCE) like home networking or Smart Spaces in buildings. In such closed environments, interacting smart devices are mostly under the control of a trusted server (for e.g., a home server) and with establishment of proprietary trust model every device can easily trust and communicate with every other device. Key distribution, access control, privacy protection and security policies for securing the communications can be easily accomplished in such closed environments.

But in an open PCE (for e.g., streets, highways, etc.) a mobile device on behalf of its owner may need to interact with many smart devices or service providers (SPs) available around it. These SPs may be genuine or malicious. Communicating with many SPs, identifying and authenticating genuine ones, checking the validity of their digital certificates and signatures if in case they are using Public-key Infrastructure (PKI), securing the entire transaction, and protecting the owner's privacy, etc cannot be handled alone by the low-computing and resource-poor mobile device. It would create a huge burden on the mobile deviceand is certainly not user-friendly. And also it is very difficult to establish a convincing trust model needed for key distribution, privacy protection, trusted entity authentication, job delegation, and secure communication. Our protocol suggests a convincing trust model in such environments.

[18] [23] [25] [26] [30-34] describe the need and importance of privacy protection in PCE and LBSs and also suggest privacy protection methods, which can be broadly categorized as follows:

1. Identity Management [26]:

In this method users interact with other smart devices through pseudonyms. [31]

describes the drawbacks of this method. Thus, a possible SP can see the user only under one of its virtual identities (VIDs) with the personal information, disclosed for the respective pseudonym. By this mechanism, users can tune their level of anonymity. But the user has to choose carefully, towards which party he uses which VID and when he has to change this VID, on too much disclosed information in the VID's context. This approach is certainly not user friendly as it involves lot of pre-settings and user intrusion to resolve a situation not considered in the pre-settings. It also creates burden on the user's mobile device to decide and choose the appropriate VID depending on the interacting SP. Another critical point of the usage of VIDs is the maintenance of system scalability in view of duplicated user contexts. It is, e.g., not clever to register each VID separately at the Location Service. Firstly, the scalability of the Location Service would suffer an exploding database and secondly the Location Service knows the link between the VIDs anyway, because of the exact similar position trace. As a result we need to trust the SPs to a certain degree. But can we trust all the SPs?The answer is "No". Initially, how would a user know whether he is interacting with a genuine SP? How can a user's low-computing and a resource-poor mobile device identify and authenticate a genuine SP among so many SPs? And also if users are only known bypseudonyms, questions about accountability and non-repudiation will be raised. Although our protocol protects user's privacy from SP, but upon a legal inquiry the entire transaction between the user and the SP

can traced using and with the intervention of MCSP

2. Adhering to the privacy policies issued by the law [32] [34]:

[35] describes the drawbacks of this approach. Policies such as FTC's five principles for Fair Information Practices, the EU directive on the Protection of Personal Data, and US Privacy Act of 1974, have been set to protect consumer's privacy. W3C's Platform for Privacy Preferences Project (P3P) makes transparent use of above-mentioned privacy policies possible. P3P is only able to provide a technical mechanism by which services and their use of personal information are described. It does not provide mechanisms by which policies are enforced. Nor can policies be used to verify or prove thatthe services accurately reflect the stated policy. Even if this approach were implemented it would be very difficult and burdensome for the user's low-computing and a resource-poor mobile device to verify such policies from different SPs and to act accordingly.

3. Use of Mix Networks [30] [33]:

In a mix network, messages are exchanged through a chain of one or more intermediaries called "mixes". The purpose of a mix is to hide the correspondences between the items in its input and those in its output. The mainfunction of a mix is to: receive and decrypt messages, buffer messages until a defined number of messages has been received, change the sequence of the received messages in a random manner and encrypt and forward the messages to the next mix or to the receiver [23]. But installing and maintaining trusted mix networks in an open PCE is very difficult and expensive. Generally a mix network is based on public key cryptography. As a result key management becomes complex and difficult. It would be a great burden on the user's low-computing and a resource-poor mobile device to carryout expensive PKI implementations like encryption, decryption, and digital certificate and signature verification, etc. This approach is computationally expensive.

4. Use of Proxies:

[23]describes the role(s) of proxy in LBSs. Location data, service requests, and privacy policies are encoded in XML by the mobile device and forwarded to a proxy server placed between the mobile terminal and the LBSs. Their approach hides the network location of the mobile device, hides the identity of the user of the mobile device, and in some cases even provide misleading physical location(s)

(by using mix network method) for the mobile device. The functionality of the LBS Proxy Server is to process SOAP (message envelope) requests and generate responses. The proxy server acts as a SOAP Dispatcherand an intermediary between a SOAP client and the requested SP. Here the privacy achieved by the following

(1) The proxy can conceal from the SP the mobile device's mobileID and protect its identity as this information may not be required to obtain the requested service. But the paper fails to mention about how to establish or envisage such a trusted proxy. Our paper clearly justifies the consideration of MCSP to be such a trusted proxy. MCSP apart from concealing the identity of the user from SP, also takes responsibility on behalf of users to process their requests, select, identify, and authenticate the genuine SPs and also to maintain a list of services SPs offer at a particular location.This reduces a great burden on user's mobile device to identify and authenticate a genuine SP.

(2) The location information is encrypted using a public key of the service provider (public key encryption scheme with multiple private keys), embedded in a XML message and transmitted to a proxy. Thus the location record is only reveled to the corresponding SP and concealed from eavesdroppers and from the proxy itself. This approach assumes that the user already trust the SP or knowsa list of trusted SPs. But how can this trust in a SP be established initially? And in reality open PCE is very dynamic in nature. New SPs may enter with better services and old SPs may change or cease to exist. As a result this assumption is not a scalable one. In our protocol, mobile device interacts with SPs more freely with the help of MCSP identifying and authenticating the genuine ones. Our approach certainly helps a user in a new PCE that is saturated with new SPs. Also maintaining a list of trustedSPs and their corresponding public keys or shared keys induces a huge burden on the mobile device. Key distribution and update between user and trusted SPs is very difficult to accomplish. PKI implementations are very expensive to carryout in a mobile device. In our protocol the mobile device neither stores a list of trusted SPs nor their corresponding keys. It needs to store only one shared secret key that is established between mobile device and MCSP during the setup phase. Using this key and cheap cryptographic techniques like hash functions we provide a secure job delegation and communication channel between the mobile device and MCSP. This reduces the burden on the mobile device to a great extent.

Finally [23] depends on lower-layer (transport layer) security mechanisms like TLS and SSL. But our protocol provides cost effective application layer security for LBSs availing.

VI. Conclusion

The advantages of this protocol are as follows: Simple, involves less user interactions, involves secure delegation of duties among the entities, automation leads to speedy taxi calling service as it involves less human interactions, in case of a legal inquiry the entire transaction can traced using R_{id}, the taxi control center cannot maintain users travel record and their profiles because they would never know the users phone number, this protects users privacy, R_{id} speeds up the process of cancellation of a request and current location update of walking users, avoids expensive PKI implementations at user end and at taxis end as they have low-computing and resource-poor mobile devices, automation reduces the cost involved in establishing taxi call centers and manpower to manage them.

There are many advantages of Secure Automated Taxi Calling Service and could be a killer application as it speeds up the call taxi process. This facility would be greatly sought by both public and taxi drivers. For public it could be a quick, useful and convenient service, which also protects their privacy. For the taxi drivers, they would never pass by a waiting customer unnoticed. This could mean more money to them. It could be a good revenue generator for the mobile

communications service provider and the taxi call center through commissions for every transaction.

Finally this protocol adheres to pervasive computing requirements, as there is a secure delegation of work between low-computing devices and high-computing devices. The user passes on his request to the trusted MCSP, which then identifies and authenticates the genuine service providers, and establishes a secure communication with them on behalf of the user. As a result the mobile device can securely handle many service providers at a time. The load on the users mobile phone is greatly reduced by employing cheap yet strong cryptographic techniques like message authentication and integrity using encryption and a MDC in order to secure the communication channel. This protocol would certainly serve the purpose for most of the Location-Based Services in open pervasive computing environment like hotel room, restaurant table, movie tickets reservations, and calling emergency services.

References

[1] M. Weiser, "The Computer for the 21st Century", *Sci. Amer.*, Sept., 1991

[2] M. Satyanarayanan, "Pervasive Computing: Vision and Challenges", *IEEE Personal Communications*, August 2001

[3] National Institute of Standards and Technology (NIST), "Pervasive Computing SmartSpace Laboratory", http://www.nist.gov/smartspace/

[4] S.M. Dye, Dr. F. Baylin, "Mobile Positioning", published by *Mobile Life streams*, 1999

[5] A.M. Basyouni and S.E. Tavares, "Public Key versus Private Key in Wireless Authentication Protocols", *Proceedings of the Canadian Workshop on Information Theory*, pp. 41-44, Toronto, June 1997

[6] A.J. Menezes, P.C. vaz Oorschot, S.A. Vanstone, *Handbook of Applied Cryptography*, Chapter 9, Hash Functions and Data Integrity, CRC Press.

[7] Whitfield Diffie and Martin E. Hellman, New Directions in Cryptography, *IEEE Transactions on Information Theory*, IT-22(6):644-654, Nov 1976.

[8] Introduction to 3G, http://www.3g.co.uk/All About 3G.htm

[9] 3rd Generation Partnership Project (3GPP), http://www.3gpp.org/

[10] Global positioning system overview and bibliography, http://www.colorado.edu/geography/gcraft/notes/gps/gps_f.html

[11]Garmins iQue 3600, the first PDA to include integrated GPS technology, http://www.garmin.com/products/iQue3600/

[12]Samsung GPS-enabled SPH-N300 Wireless Phone, http://www.samsungusa.com/cgi-bin/nabc/prod/hhcommerce/telecommunications/sph_n300_features.jsp

[13]Motorolas GPS enabled i205 Handset, http://idenphones.motorola.com/iden/products/phones/phones_home.jsp

[14]NTT DoCoMos first Global Positioning Service (GPS)-compatible handset F661i, http://www.nttdocomo.co.jp/p_s/f/f661i.html, http://www.i4u.com/article263.html

[15]SiRFs GPS chip sets, http://www.sirf.com/products.html

[16]A.J. Menezes, P.C. vaz Oorschot, S.A. Vanstone, *Handbook of Applied Cryptography*, Chapter 10, Identification and Entity Authentication, CRC Press.

[17]David Youd, "An introduction to Digital Signatures", http://www.youdzone.com/signature.html

[18]H.T. Tavani, and J.H. Moor,"Privacy Protection, Control of Information, and Privacy-Enhancing Technologies", *ACM SIGCAS Newsletter* 31(1), 6-11, 2001.

[19]William Stallings, "Secure Hash Algorithm", in *Cryptography and Network Security: principles and practice* Second Edition, Prentice-Hall.,pp.193-197,1999.

[20] Bruce Schneier, "Using One-Way Hash Functions", in Applied Cryptography Second Edition, John Wiley & Sons,Inc., pp. 351-354, 1996.

[21]National Institute of Standards and Technology, "Secure Hash Standard", *FIPS Publication* 180-1, 1995. in Computer Science from Chonnam National.

[22] Peter Kuykendall and Dr. Peter V. W. Loomis, Trimble Navigation, "In Sync with GPS: GPS Clocks for theWireless Infrastructure", http://www.gfec.com.tw/english/service/content/gps_ec.htm

[23]Escudero A., Maguire G.Q.,"Role(s) of a proxy in location based services". 13 th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications. PIMRC2002. Lisbone. Portugal. September 2002.

[24] A.J. Menezes, P.C. vaz Oorschot, S.A. Vanstone, *"Handbook of Applied Cryptography"*, Chapter 12, Key Establishment Protocol, CRC Press

[25] S. Lederer, J. Mankoff, A. Dey," Towards a Deconstruction of the Privacy Space", *Workshop on Privacy In Ubicomp'2003.*

[26]Uwe Jendricke, Michael Kreutzer and Alf Zugenmaier,"Pervasive Privacy with Identity Management", *Workshop on Security in Ubiquitous Computing, UBICOMP 2002.*

[27] MIT Project Oxygen, http://oxygen.lcs.mit.edu/.

[28] Easy Living, Microsoft Research, http://research.microsoft.com/easyliving/.

[29]The Aware Home, Georgia Institute of Technology http://www.cc.gatech.edu/fce/ahri/.

[30] GAIA - Active Spaces for Ubiquitous Computing, University of Illinois at Urbana-Champaign, http://choices.cs.uiuc.edu/gaia/

[31] Christian Hauser, "Privacy and Security in Location-Based Systems With Spatial Models", *PAMPAS '02 - Workshop on Requirements for Mobile Privacy & Security*

[32] G. Myles, A. Friday, and N. Davies, "Preserving Privacy in Environments with Location-Based Applications", *IEEE Pervasive Computing - Security and Privacy, journal, Jan-Mar 2003.*

[33] A.R. Beresford and F. Stajano, "Location Privacy in Pervasive Computing", *IEEE Pervasive Computing - Security and Privacy, journal, Jan-Mar 2003.*

[34] Marc Langheinrich , "A Privacy Awareness System for Ubiquitous Computing Environments", http://www.inf.ethz.ch/vs/publ/papers/privacy-awareness.pdf

[35] Moamo Wu, Adrian Friday, "Integrating Privacy Enhancing Services in Ubiquitous Computing Environments", *UBICOMP2002 - Workshop on Security in Ubiquitous Computing*