Preserving User Anonymity in Context-Aware Location-Based Services: A Proposed Framework

Songpon Teerakanok, Chalee Vorakulpipat, Sinchai Kamolphiwong, and Siwaruk Siwamogsatham

Protecting privacy is an important goal in designing location-based services. Service providers want to verify legitimate users and allow permitted users to enjoy their services. Users, however, want to preserve their privacy and prevent tracking. In this paper, a new framework providing users with more privacy and anonymity in both the authentication process and the querying process is proposed. Unlike the designs proposed in previous works, our framework benefits from a combination of three important techniques: *k*-anonymity, timed fuzzy logic, and a one-way hash function. Modifying and adapting these existing schemes provides us with a simpler, less complex, yet more mature solution. During authentication, the one-way hash function provides users with more privacy by using fingerprints of users' identities. To provide anonymous authentication, the concept of confidence level is adopted with timed fuzzy logic. Regarding location privacy, spatial *k*-anonymity prevents the users' locations from being tracked. The experiment results and analysis show that our framework can strengthen the protection of anonymity and privacy of users by incurring a minimal implementation cost and can improve functionality.

Keywords: Trusted third party, location-based services, anonymity, privacy, context-aware, *k*-anonymity.

I. Introduction

Recently, location-based services (LBSs) have played an important role in technology advancement; this can be seen through the number of LBSs and applications, for example, Google Maps and Four Square. Although the demand for LBSs is continuously and rapidly increasing, their growth and proliferation are still limited due to some privacy concerns.

LBSs are easily and conveniently accessible by mobile users [1], [2]. With the use of GPS-enabled portable devices (for example, smartphones, PDAs, or navigation devices), users can enjoy the LBS of their choice by using their current location to perform an LBS. Regarding the essential or core information that is used in performing an LBS, the user's location (retrieved by GPS, for example) is of the utmost importance. With untrustworthy LBS providers, however, revealing location information may lead to some privacy threats for users [3], [4], for example, location tracking [5]. For instance, assume that a user, Bob, is asking for the nearest gas station by sending his current location to a service provider. If the service provider is untrustworthy, adversaries may learn Bob's location and earn profits by using that information.

Recently, context-aware computing (CAC) moved from the research world into practical deployment scenarios [6], [7]. The term "context" refers to any information that can be used to characterize the situation wherein entities (that is, people, places, temperatures, or objects) are considered relevant to the interaction between user and applications [8]. To perform a context-aware service (CAS), the user's context information must be collected. For example, a traffic reporting application may require user identity, vehicle type, location, and/or activity of the user to ensure the accuracy and efficiency of the application and to produce a suitable result.

Manuscript received May 28, 2012; revised Sept. 8, 2012; accepted Sept. 17, 2012.

Songpon Teerakanok (phone: +66 74 287075, thongpon.te@gmail.com) and Sinchai Kamolphiwong (ksinchai@coe.psu.ac.th) are with the Center for Network Research (CNR), Department of Computer Engineering, Prince of Songkla University, Hat Yai, Thailand.

Chalee Vorakulpipat (corresponding author, chalee.vorakulpipat@nectec.or.th) and Siwaruk Siwamogsatham (siwaruk.siwamogsatham@nectec.or.th) are with the Wireless Information Security and Eco-Electronics Research Unit, National Electronics and Computer Technology Center, Pathumthani, Thailand.

http://dx.doi.org/10.4218/etrij.13.0112.0327

A context-aware LBS (C-LBS) is an intelligent service that offers the user an LBS with richer features [9], [10]. For instance, a user, Alice, asks for the nearest restaurant; Alice is a vegetarian. Using a traditional LBS might get her to a restaurant with a non-vegetarian menu. A C-LBS overcomes this problem.

There are two main privacy issues regarding C-LBSs: disclosing the user's current location and disclosing his or her personal information. In fact, the user sometimes does not want his or her personal information to be collected. Thus, users might choose to turn off their portable devices or temporarily unsubscribe from a service. To preserve privacy in the user's context information, *anonymity* has been adopted in both LBSs and CASs for the purpose of blurring the user's current location and blinding the service provider to the relationship between the user and the user's requests [11]. Moreover, this property can be used to protect personal information during the authentication process. Hence, with anonymity, users can now conveniently request services of their choice anonymously.

In this paper, we present an anonymity-preserving framework for C-LBSs by combining two important technology advancements in today's services: the LBS and the CAS. The proposed framework is designed for scenarios in which legitimate (authenticated) users want to enjoy available services while their privacy (that is, anonymity) is preserved. Location-based anonymous voting is one example of services that can be applied with our proposed mechanism.

For example, let us consider a scenario in which a conference is held in a hall and all participants are going to vote for Innovation of the Year by using the service available on their smartphones. In this scenario, users allowed to vote should be conference participants and should be present in the conference hall. Users are required to authenticate themselves and then prove their locations to the voting service provider before they can enjoy the voting service. However, the anonymity of the voting service users must be protected. Hence, using our proposed framework ensures that privacy is secure in both the user authentication process and the location identification process.

The rest of this paper is organized as follows. Section II presents the background research of CAC focusing on privacy-preserving techniques in both location-based and context-aware systems. Our proposed framework is described in section III. The experiment results and evaluation of our proposed framework are shown in section IV, followed by a discussion of privacy and security concerns in section V. Section VI presents a discussion regarding the effect of anonymity over quality of service (QoS). In section VII, we provide an analysis of our proposed framework. Finally, we conclude our paper in section VIII by listing the features and stating the applicability of our framework.

II. Research Backgrounds

In a CAS, unification of all context information is a basic procedure in providing a service based on increased knowledge about the user interacting with it. Collecting too much context information, however, might threaten the user's privacy.

1. Overview of Security and Privacy in C-LBS

Many users prefer to enjoy available services without disclosing their identities or personal information. Today, users are more aware of their security and privacy. An international news agency published an interesting article regarding privacy in today's technology climate and associated crimes and threats that have occurred in real life [12].

In LBSs, security and privacy are the major issues; much research has focused on this topic [3]-[5]. For example, a discussion of research regarding location privacy in LBSs and a method to keep the user's location private can be found in [13] and [14], respectively. To perform a service, the user must send his or her current location (obtained by using a GPS locator, for instance) to a service provider, which processes the request and then sends back an appropriate result, satisfying the given location. The user obtains a result depending on his or her current location. The user's location information, however, can lead to location tracking, an attack that threatens the user's privacy. Several methods to prevent location tracking attacks have been designed and proposed in the literature [3], [5].

2. *k*-Anonymity

To provide users with anonymity, *k*-anonymity is one of the most important techniques. In practical application, *k*-anonymity is used to provide anonymity to users by disguising the information intended to be kept secret (R_i), that is, user queries/requests or personal information, within a set of information (I) with the same type that contains k other pieces of information. The generic equation of I can be formed as follows:

$$I = \{R_1, R_2, R_3, \ldots, R_i, \ldots, R_k\}. \tag{1}$$

Spatial *k*-anonymity (SKA) is a well-known technique that is used to keep the user's location private. In this technique, the user's location is hidden in a small region, called the "cloaked region," which contains both the locations of the requesting user and the k–1 other users. Numerous approaches, methods, and implementations have been proposed and discussed in the literature [12], [15]. In a recent work, Gong and others proposed KAWCR, which is a framework for protecting users' privacy, constructed of SKA without the cloaked region [16].

3. Use of Confidence Level

Concerning privacy in CASs, there is also an important issue regarding user anonymity [17]. Generally, when the user attempts to request a service for which authentication is required, the user must reveal his or her identity to the service provider. With an untrustworthy provider, however, revealing one's personal identity subjects the user to tracking of his or her querying process. We have already determined that users sometimes prefer accessing services without revealing any of their personal information (for example, name, email, citizen ID).

To overcome this problem, confidence levels have been proposed to preserve user privacy [18], [19]. A confidence level is the level of trust that can be used as a predefined value that must be reached before the user is able to access his or her desired service. In a context-aware environment, users are instantly provided with context information that can change depending on their surroundings. A framework for context-aware authentication that illustrates the use of a confidence level in authentication processes is found in [20].

To illustrate [20], let us assume that the user Bob possesses a set of context data C, which contains attributes that can describe his current activity/situation and also include his personal information. First, Bob chooses a set of context data A, which is a subset of C, as a set of attributes that the service provider must process to be considered trusted (authenticated). For example, the chosen attributes in set A can be represented as follows:

```
Name: Bob
Date of birth: 22 Oct. 1990
Location: Central Park
E-mail: bob@testscenario.com
Membership (Member ID): 0x01
Timestamp: 10:11 PM (4/25/2011)
...
```

In this scheme, each attribute is mapped into variables d_i for $i = 1, 2, 3,\ldots, n$, where n is less than or equal to the number of members in C. Moreover, d_1 is a number representing "`Name`," d_2 represents "`Date of birth`," and so on. Therefore, we can form a generic equation of set A as follows:

$$A = \{d_i \mid i = 1, 2, 3,\ldots, n\} \subseteq C. \qquad (2)$$

In this scheme, an attribute set A is used as a credential of the user. If Bob discloses all of his attributes in A to the service provider, maximum confidence is achieved. Using the benefit of the confidence level, Bob can flexibly choose a subset of attributes in A as his new credential according to the different levels of confidence required by the service provider. For example, if Bob's credential represents only his name and his e-mail address, it might yield "medium" confidence. Basically, techniques based on fuzzy logic are used to determine confidence levels from the user's credential.

4. Timed Fuzzy Logic

As discussed previously, the confidence level is used as an essential predefined value needed to perform the requested service. In practice, a user U sends a request for the service to the provider; the service provider immediately asks for authentication by sending the required confidence level α to the requestor. Using (1), the user U issues his or her credential by choosing some attributes in A that can produce a confidence level equal to or greater than the required level. The user sends his or her credential to the provider to complete the authentication process, and the service provider then calculates the confidence level from the received credential. Hence, the requested service would be delivered from the service provider to the requestor if and only if the calculated confidence level, retrieved from the user's credential, has reached the required level.

To calculate the confidence level from the received credential, Malek and others [20] proposed a technique based on fuzzy logic that integrates time, called "timed fuzzy logic." Generally, fuzzy logic is a form of multivalued logic that was designed to handle the concept of partial truth. Using fuzzy logic allows a gray definition between black and white binary logic, which is normally based on the degree of truth rather than the usual "true or false." This technique gives a greater ability to characterize an undetermined value (for example, tall, short, heavy, or thin). To give an example of combining fuzzy logic with confidence level, the basic fuzzy rule used to characterize the level of confidence is shown below.

```
If: 'Name' is "KNOWN"
 -> Then: Confidence = "MEDIUM"
If: 'Location' is "WITHIN_RANGE"
 -> Then: Confidence = "LOW"
If: 'E-mail' is "RECOGNIZED"
 -> Then: Confidence = "MEDIUM"
If: 'Membership' is "IN_GROUP"
 -> Then: Confidence = "BETTER"
If: 'Timestamp' is "EXPIRED"
 -> Then: Confidence = "LOW"
... (and so on)
```

To obtain more efficiency and greater ability to characterize the confidence level of the user's credential, we can create more complicated fuzzy rules by combining the basic rules together; for example, the combination of basic fuzzy rules can be presented as follows:

```
If: 'Name' is "KNOWN"
AND 'E-mail' is "RECOGNIZED"
AND 'Membership' is "IN_GROUP"
 -> Then: Confidence = "VERY HIGH"

If: 'Name' is "KNOWN"
AND 'E-mail' is "RECOGNIZED"
 -> Then: Confidence = "HIGHER"

If: 'Membership' is "IN_GROUP"
AND 'Timestamp' is "NOT_EXPIRED"
 -> Then: Confidence = "HIGH"
... (and so on)
```

Authentication using the confidence level can provide users with more privacy in their context information. By allowing users to flexibly customize their own credentials, the users can then choose a set of attributes that will be used to prove themselves to the service provider.

5. One-Way Hash Function

In cryptography, there are large quantities of cryptographic tools, methods, and algorithms that are ready to be applied in modern system designs (for example, data encryption and digital signature schemes). Most of them attempt to provide users with more secrecy or confidentiality of the transmitted message through an insecure channel. A one-way hash function, also called "message digest," is a cryptographic hash function that is used to create a fingerprint of the message. Traditionally, a message digest has been used to verify the integrity of the data, that is, to ensure that data has not been changed, modified, or tampered with during transmission [21], [22]. For example, researchers presented the architecture and VLSI implementation of the Whirlpool hash function in [23]. In our proposed framework, a message digest is recommended for use in authentication processes. By using a one-way hash function to create a fingerprint from the user's credential, authentication via a third party in a centralized architecture is more secure and more private.

III. Proposed Model

In both context-aware and location-based systems, users' awareness of privacy concerns has increased. Users sometimes do not trust the servers or service providers. They also prefer enjoying available services without revealing their identities to providers. However, to obtain some services, the user still needs to be authenticated, which means revealing his or her credential to the provider for verification purposes. To overcome this, we propose an anonymity-preserving framework for C-LBSs that allows users to enjoy the available services without revealing their identities or personal information to the service providers. In the proposed framework, three important techniques, *k*-anonymity, timed fuzzy logic, and one-way hash function, are adopted. This combination provides privacy regarding the user's current location (location tracking prevention) and querying process (query tracking prevention). With our proposed framework, users can enjoy the available services freely and anonymously.

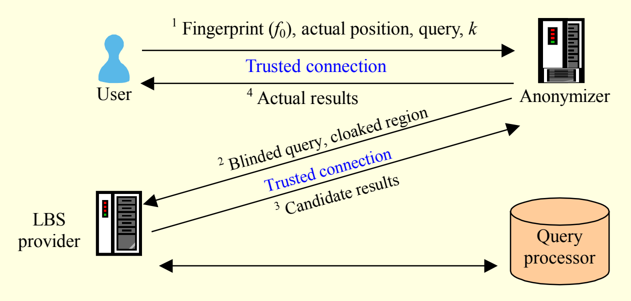

Fig. 1. Overview of proposed framework.

In this section, we divide our proposed framework into two major parts: the user authentication protocol and the querying process. The overall process of the proposed framework is shown in Fig. 1. In this framework, an *anonymizer*, a trusted third party, is adopted to blur and forward requests from the user to the provider. The user then authenticates himself or herself to the anonymizer, which is an agent trusted by the user, instead of to the service provider. Moreover, the anonymizer is used to keep the user's querying process private from the service provider. In our framework, the user requests service from his or her subscribed provider by sending his or her requests to the anonymizer, which is a trusted third party positioned between users and the local server (provider). In this model, the location information is sent along with the request packet. The user requesting the service first needs to be authenticated to the provider. In the following subsections, details of the proposed user authentication protocol and querying process are given.

1. User Authentication Protocol

Generally, a user who attempts to verify himself or herself must send a credential to the server. As a result, the authentication process is successful if and only if all identities sent to the server match those stored in the server's database. With an untrustworthy provider, authentication of the user by sending a credential to the server would threaten the user's security and privacy.

To tackle this problem, in our proposed framework, a user authentication protocol is presented. Inspired by [20], authentication of the user is achieved by using the confidence level, that is, a predefined value indicating the required level of trust, together with timed fuzzy logic, that is, a technique used to calculate the confidence level from the user's credential. Our proposed protocol is divided into four important steps, which are detailed as follows.

A. Step 1: Initialization

In our proposed protocol, we assume that the trusted connections (for example, trusted SQL/SSH connection) between users and the third party has already been established before the user requests services from the local server (that is, service provider).

In this step, the user sends a request packet to the anonymizer. First, the requestor forms a set of attributes, C, which are considered authenticated by the provider. The user chooses a set of attributes, S, which is a subset of C, and consequently obtains the fingerprint f_0 by digesting all attributes in S using a one-way hash function.

A one-way hash function is adopted to provide better privacy protection of the user's credential and personal information. As mentioned before, a set of the user's attributes S, which are considered trusted by the provider, is digested by using a one-way hash function (H) to create the fingerprint of the user's identity. As a result, a bit-string representing a fingerprint of the user's credential is sent to the anonymizer along with the request ID, user's ID, relevant contexts (for example, location), and the list of attribute types (AT) contained in S.

To give an example, let us assume that a user Bob has created his set of attributes S containing three pieces of information, that is, name, member ID, and e-mail address, as

$$S = \{\texttt{Bob}, \texttt{007}, \texttt{Bob@test.com}\}. \tag{3}$$

According to the attributes appearing in S, a fixed-length bit-string (f_0) representing a fingerprint of these attributes is created by using H. Bob can construct a set of AT, as follows:

$$AT = \{\texttt{Name}, \texttt{Member ID}, \texttt{E-mail}\}. \tag{4}$$

The structure of the request packet is shown in Fig. 2.

In our proposed framework, the fingerprint of the user's credential is sent along with his or her request instead of the original credential used in the traditional approach.

B. Step 2: Acquiring Confidence Level

After retrieving the request packet, the anonymizer then classifies the "SERVICE_ID" appearing in the request packet into subcategories. The anonymizer sends the names of the subcategories to the service provider. Using the name of a subcategory, the server replies back to the anonymizer with the confidence level L_0, which is required in performing services in that subcategory. Figure 3 shows the processes of retrieving the confidence level of the requested service.

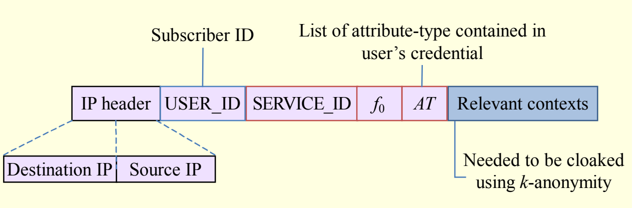
Subscriber ID
List of attribute-type contained in user's credential
IP header
USER_ID
SERVICE_ID
f_0
AT
Relevant contexts
Destination IP
Source IP
Needed to be cloaked using k-anonymity

Fig. 2. Structure of requested packet.

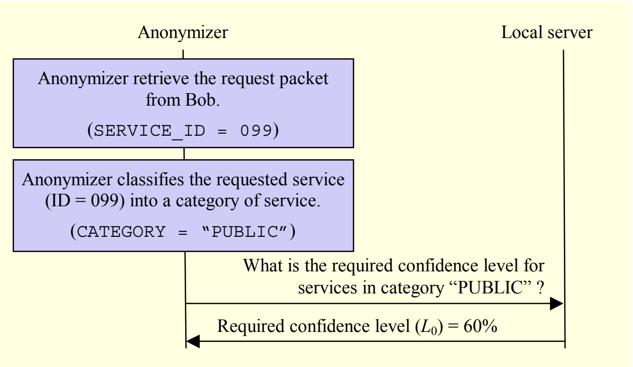
Anonymizer
Local server
Anonymizer retrieve the request packet from Bob.
(SERVICE_ID = 099)
Anonymizer classifies the requested service (ID = 099) into a category of service.
(CATEGORY = "PUBLIC")
What is the required confidence level for services in category "PUBLIC" ?
Required confidence level (L_0) = 60%

Fig. 3. Acquiring confidence level.

C. Step 3: k*-Anonymity*

In this step, the anonymizer acquires fingerprints of users' credentials from the service provider by using k-anonymity. First, the anonymizer asks the server for the range of all subscribers' ID numbers. Upon retrieving the range of all registered users' ID numbers, which ranges from X to Y, the anonymizer randomly chooses k ID numbers ($I_1, I_2, \ldots, I_k$) within $[X, Y]$, which also includes the ID of the requesting user. Regarding the k-chosen ID numbers, the anonymizer then asks the server for a set of fingerprints of the credentials that belong to the users whose IDs were chosen by the anonymizer in the previous process.

Regarding the creation of fingerprints from users' credentials at the server side, a list of AT is used to define which attributes would be digested using the one-way hash function. Then, the server sends the set of fingerprints (F) requested in the previous process back to the anonymizer, at which time F can be presented as follows:

$$F = \{f_1, f_2, f_3, \ldots, f_k\}. \tag{5}$$

To give an illustration, let us assume that a user Bob, whose ID number is equal to "007," is requesting a service S_i. Figure 4 shows the overall process of this step.

D. Step 4: User Verification Using Confidence Level

Regarding the received set of fingerprints (F), let us assume that f_g is a fingerprint within F, which is created by digesting the requestor's credential stored in the server's database. The anonymizer compares f_0 (the requestor's fingerprint) with f_g to

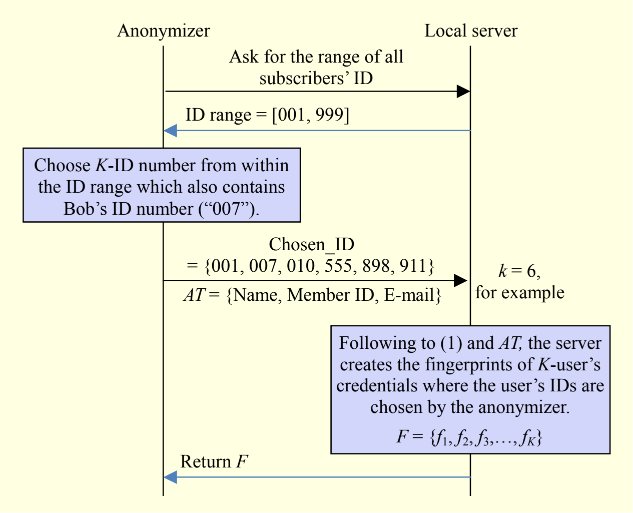
Anonymizer
Local server
Ask for the range of all subscribers' ID
ID range = [001, 999]
Choose K-ID number from within the ID range which also contains Bob's ID number ("007").
Chosen_ID = {001, 007, 010, 555, 898, 911}
AT = {Name, Member ID, E-mail}
k = 6, for example
Following to (1) and AT, the server creates the fingerprints of K-user's credentials where the user's IDs are chosen by the anonymizer.
F = {f1, f2, f3,…, fK}
Return F

Fig. 4. Acquiring users' fingerprints by using *k*-anonymity.

prove equality between the user's credential (received from the requestor) and the credential stored in the database on the server side.

If f_0 matches f_g, then the anonymizer calculates the confidence level of the requestor's fingerprint by using *AT*. To calculate the confidence level from *AT*, timed fuzzy logic is adopted. As a result, the level of confidence, L_1, is produced. Figure 5 shows the processes of verifying the user's fingerprint. Hence, examples of fuzzy rules that are used to calculate the confidence level can be shown as follows:

```
If: 'Name', 'Member-ID', and 'E-mail' are
"MATCHED"
 -> Then: Confidence (L1) = "VERY HIGH"
Else If: 'Name' and 'Member-ID' are "MATCHED"
 -> Then: Confidence (L1) = "MEDIUM"
Else If: 'Member-ID' and 'E-mail' are "MATCHED"
 -> Then: Confidence (L1) = "MEDIUM"
Else If: 'Name' and 'E-mail' are "MATCHED"
 -> Then: Confidence (L1) = "QUITE LOW"

Else If: 'Member-ID' is "MATCHED"
 -> Then: Confidence (L1) = "LOW"
Else If: 'Name' is "MATCHED"
 -> Then: Confidence (L1) = "VERY LOW"
Else If: 'E-mail' is "MATCHED"
 -> Then: Confidence (L1) = "VERY LOW"
```

As mentioned earlier, L_0 is the confidence level retrieved from the service provider and needs to be reached before the user can enjoy the requested service. User authentication succeeds if and only if L_1 is larger than or equal to L_0. The anonymizer notifies both the requesting user and the service provider about the result of the authentication (success or failure); if authentication succeeds, then the requested service is delivered to the user through the anonymizer.

To summarize, the anonymizer, a trusted third party, is used to authenticate the user's credential instead of the provider; the anonymizer notifies the result of the authentication to both user and service provider without disclosing any details of the user's credential. Using the one-way hash function, the anonymizer obtains only fingerprints of the users' credentials and is thus unable to disclose any of the credentials belonging to the users. As a result, each user's privacy is preserved.

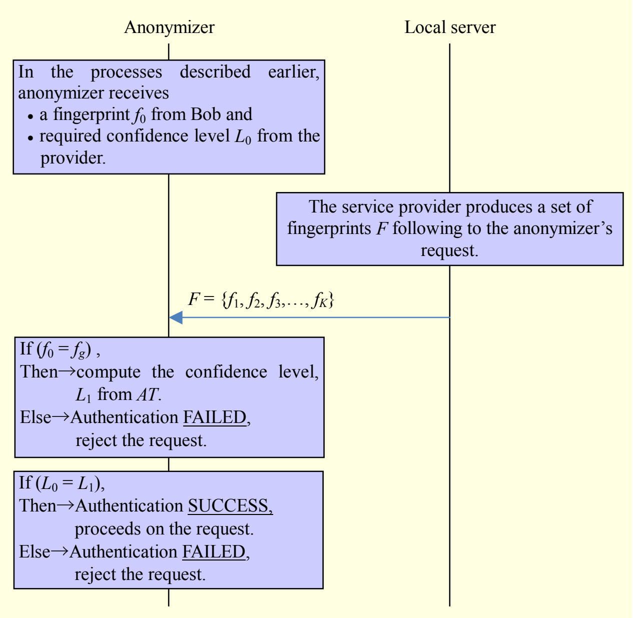
Anonymizer
Local server
In the processes described earlier, anonymizer receives
• a fingerprint f0 from Bob and
• required confidence level L0 from the provider.
The service provider produces a set of fingerprints F following to the anonymizer's request.
F = {f1, f2, f3,…, fK}
If (f0 = fg) ,
Then→compute the confidence level, L1 from AT.
Else→Authentication FAILED, reject the request.
If (L0 = L1),
Then→Authentication SUCCESS, proceeds on the request.
Else→Authentication FAILED, reject the request.

Fig. 5. User verification using confidence level.

2. Querying Process

To enjoy the requested services, users must verify themselves to the service providers. Using our proposed authentication protocol, privacy regarding the user's personal information is guaranteed. Regarding location information, the user's current location is essential for the querying processes of C-LBSs. Using location information, the service provider can provide the user with smart and efficient services depending on the user's location. Using this location, however, can also violate the user's privacy via location tracking.

Let us assume that an adversary Eve gains control of the LBS server to which the user sends his or her request. We also assume that a user Bob sends his request together with his location to the malicious server *SV* controlled by Eve. As a result, when Bob sends his request to *SV*, Eve automatically knows Bob's current location. Knowing Bob's current location allows Eve to threaten Bob's privacy.

SKA is adopted by the proposed framework to deal with the problem of location tracking. In our proposed framework, the user sends his or her context information along with the request packet to the server. The context information can be naively inferred as information used to describe the situation of the user. In this scheme, the anonymizer performs SKA on the user's current location as it appears in the request packet to blur the

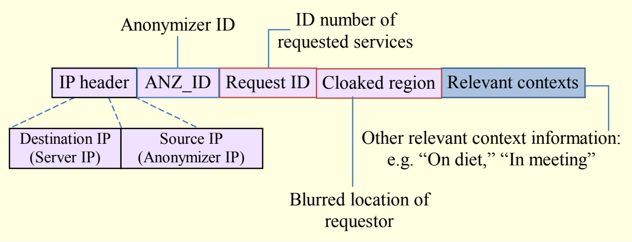
Anonymizer ID
ID number of requested services
IP header
ANZ_ID
Request ID
Cloaked region
Relevant contexts
Destination IP (Server IP)
Source IP (Anonymizer IP)
Other relevant context information: e.g. "On diet," "In meeting"
Blurred location of requestor

Fig. 6. Structure of blurred forward request packet.

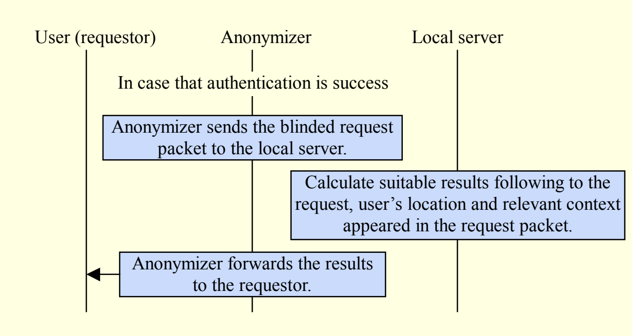
User (requestor)
Anonymizer
Local server
In case that authentication is success
Anonymizer sends the blinded request packet to the local server.
Calculate suitable results following to the request, user's location and relevant context appeared in the request packet.
Anonymizer forwards the results to the requestor.

Fig. 7. Signal flow of querying process.

user's exact location.

In our proposed framework, the anonymizer is used to blur and forward the user's requests to the provider in case the user authentication succeeds. Whenever the user sends his or her request to the anonymizer, it asks for authentication. After completing the authentication, the anonymizer blurs the user's current location information in the request packet via SKA. To perform SKA, the anonymizer finds the k–1 other users within the requestor's surrounding area that are closest to the requestor. The anonymizer then chooses the smallest area that contains k users inside as the cloaked region; note that the cloaked region must contain the requestor's current location. As a result, the cloaked region formed by the anonymizer is used as the location information instead of the requestor's exact location. Figure 6 shows the adjusted structure of the forward packet.

The anonymizer replaces the "`user_id`" that appears in the request packet with the "`anonymizer_id`"; thus, the request packet is blurred. As a result, the server can know only that the anonymizer is requesting a service. With this simple method, the servers have no need to know who is requesting service or how many users are using the system. Using the trust between providers and the anonymizer, the servers only know that whenever an anonymizer requests a service, they must provide the anonymizer with the requested services unconditionally.

Assume that a user Bob is requesting a C-LBS. Bob sends his request packet R_i together with a geographical coordinating point (m, n) representing his current location. After he verifies himself to the service provider through the user authentication protocol, the anonymizer automatically finds the k–1 users closest to Bob in his surroundings. The anonymizer chooses the smallest region (Z), which contains all k users' locations including Bob's current location. Then, the anonymizer constructs a forward request packet R_i' following the pattern shown in Fig. 5 and forwards the packet to the service provider; the region Z is used as the cloaked region in the forward request packet. An overview of the querying process is shown in Fig. 6.

To summarize, the proposed framework aims to provide users with more privacy in the authentication processes via an anonymizer that uses one-way hash function and timed fuzzy logic to verify the users' identities without revealing any credentials. In the querying processes, users' locations are also kept private by adopting SKA, which is used to blur each user's current location into the cloaked region. By protecting privacy in both the authentication process and the querying process, the user can anonymously enjoy the available services.

IV. Experiment Results and Evaluations

We experimentally verify that the guarantee of stronger protection of privacy comes at a reasonable cost. In all experiments, we iteratively test the proposed framework and related frameworks 10,000 times. The service provider is implemented on an Intel Core i5 2.27-GHz computer running Windows 7 with 2 GB of RAM. The anonymizer is performed on an Intel Core 2 Duo 2.66-GHz computer running Windows XP with 2 GB of RAM. The mobile device is implemented on an 800-MHz smartphone running the Android OS version 2.2 with 256 MB of RAM.

In this section, we evaluate and compare the performances of the proposed framework, a traditional framework in which users communicate with the providers directly and authenticate themselves to the providers using only username and password, and the context-aware authentication framework proposed by Malek and others in [20]. We also test our user authentication method by comparing it with [20].

We divide the experiment into two performance evaluations: one during the authentication process and one during the querying process. The details and results are shown in the following subsections.

1. Performance Evaluation during Authentication

Table 1 presents the comparison of the authentication time. Focusing on the implementation of our proposed protocol, caching, a technique used to store a set of fingerprints F retrieved from the server (see part C of subsection III.1 [step 3]), can be applied to enhance performance. Let us assume that a user U_i verifies herself to the anonymizer, which has

Table 1. Comparison of efficiency during authentication.

Model	Time for authentication (s)
Traditional model	0.033618
Malek and others' model	0.215487
Proposed model (without cache, k = 100)	0.455200
Proposed model (with cache, k = 100)	0.205317

Table 2. Comparison of functionality.

Model	Anonymity	Customized security
Traditional model	No	No
Malek and others' model	No	Yes
Proposed model	Yes	Yes

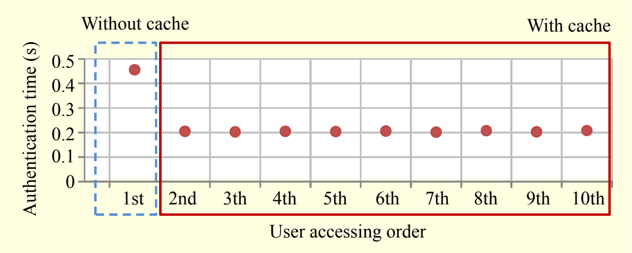

Fig. 8. Time for multiple authentications.

already stored the set of fingerprints F that contains the fingerprint f_x corresponding to the user's fingerprint. As such, there is no need to ask for a set of fingerprints F from the server; the anonymizer can verify the user's fingerprint automatically. Regarding functional properties incurring an additional implementation cost, a comparison is summarized in Table 2.

The fastest authentication model is the traditional one; however, this model cannot guarantee that the user will remain anonymous. Regarding the use of a confidence level, both our proposed model and [20] provide users with customized security allowing users to create their own credentials, which yields different levels of confidence.

A benefit of using caching is that the authentication time of our proposed framework is significantly reduced. Figure 8 shows the time for multiple authentications of a user U_i. The authentication time increases once a new user U_i performs the authentication. After the first authentication, time spent for the following authentication process is significantly decreased and then becomes stable.

2. Performance Evaluation during Querying

In this experiment, we evaluate the performance of our framework during the querying process compared with the performance of the traditional framework. A comparison of time used in the querying process is summarized in Table 3.

Table 3. Comparison of efficiency during querying.

Model	Querying time (s)
Traditional model	0.02571
Proposed model (k = 100)	0.02779

Table 4. Comparison of functionality.

Model	User location privacy	Anonymity
Traditional model	No	No
Proposed model	Yes	Yes

As far as the functionality property is concerned, we also compare our scheme with the traditional framework. As explained earlier, in our proposed framework, SKA is adopted to prevent the user's current location from being tracked. A comparison of functional properties is shown in Table 4.

As shown in Tables 3 and 4, the querying process using the traditional framework, in which a query is sent to the provider directly (without the anonymizer), can reduce time consumption. With a small implementation cost (time) increase, our model guarantees a higher level of location privacy by using SKA.

To summarize, with a small extra implementation cost, the proposed framework achieves three outstanding properties: anonymity, customized security, and user location privacy.

V. Privacy Concerns

In this paper, we present a framework that protects the anonymity of the user. The user's privacy is guaranteed to be protected by the proposed authentication protocol, which uses timed fuzzy logic to compute the confidence level and a one-way hash function to create fingerprints from users' credentials. In the proposed protocol, a fingerprint of the user's credential, instead of the real credential, is used to verify the user. By choosing k users' ID numbers within the range of all subscribers, the anonymizer disguises the user's fingerprint using k-anonymity, which asks the service provider for fingerprints belonging to k users; the fingerprint of the requestor is included. Using this technique, users can authenticate themselves without revealing any of their credentials to the anonymizer or to the untrustworthy provider.

The user's location is essential to perform C-LBSs. It is very important to keep the user's exact or current location secret from adversaries and the provider to avoid location tracking. SKA is adopted to blur the user's current location into a small region that contains his or her location and the locations of other k–1 users, referred to as the cloaked region. Using the cloaked region, the server and adversaries are unable to specify the user's exact location. As a result, the goal of protecting the privacy of the user regarding location is achieved.

VI. Effect of Anonymity over Quality of Service

In both CASs and LBSs, there is a tradeoff between privacy and QoS. In our developed framework, two major goals must be achieved: privacy in the authentication process and privacy in user context (including user's location) during the querying process. In the authentication process, the goal of anonymity is achieved by applying k-anonymity to the requestor's fingerprint (see part C of subsection III.1 [step 3]). A major goal of the authentication process is to indicate the legitimacy of the user before he or she can enjoy the requested service; thus, anonymity deployed during the authentication phase does not affect the QoS.

On the other hand, regarding the querying process, anonymity is also adopted to prevent the user's location from being tracked. Focusing on the use of SKA, it is clear that increasing k can enhance the privacy of the user. For example, if the user requests a service by performing SKA with k=2, there is a 50% chance that one of the k-users is the corresponding requestor. We can form the generic equation of the disclosure probability (DP) as follows:

$$DP = 1/k. \tag{6}$$

DP represents the chance that one within a set of k-users is the requestor. Changing the value k can affect the privacy and anonymity of the user. Increasing k, however, also reduces the accuracy of LBSs. As a result, in the querying process, the user can choose his or her level of privacy by adjusting the value of k.

VII. Framework Analysis

We compare the proposed framework with other important privacy-aware frameworks using the following criteria: application domain, overlay network distribution, storage model, access mechanism, security, privacy, and anonymity. In this analysis, two important frameworks are compared with our proposed framework. The main criteria that should be considered are privacy and anonymity. User privacy can be classified into three subcategories: privacy during authentication, privacy in user's context information (for example, user's location), and privacy in user's credential used in the authentication processes. Anonymity is classified into two subcategories: anonymity of the user during the authentication process and anonymity of the user during the querying process.

There are a number of frameworks attempting to achieve privacy and anonymity. The proposed framework aims to guarantee that the user's privacy is protected both during the querying process and in respect to the user's context information. Table 5 shows the big picture of our proposed framework, including a comparison with other works. In this table, we use "+" to denote a feature that a system supports; "–" is used to denote a feature that a system does not support; "N/A" means that it is unclear whether the feature is supported or not.

Our framework allows users to enjoy available services anonymously without disclosing any of their personal information, context information, or identities. The outstanding

Table 5. Privacy-aware framework for context-aware systems.

System	Application domain		Overlay network distribution		Access mechanism		Privacy				Security			Anonymity	
	Domain specific	Generic	Centralized	P2P	Directly	With TTP	During authentication	In context information (for example, location)	In personal identities	Customizability	Encryption	Authentication	Customizability	In authentication	In querying process
Proposed model	–	+	+	–	N/A	+	+	+	+	+	–	+	+	+	+
Malek and others' model	+	N/A	–	+	+	N/A	+	+	+	+	+	+	+	–	–
Traditional model	–	+	+	–	+	–	–	–	–	–	–	–	–	–	–

feature of the proposed framework is anonymity. Note that communication through a trusted third party would slow down the system; the speed of requesting services directly from the provider might be faster than using the anonymizer.

Regarding customizability, the proposed framework allows the service provider to customize the required level of security by changing the confidence level; moreover, the location privacy of users can be adjusted by changing the value of *k*. With customizability of security, our proposed framework can be applied with a commercial product for which various kinds of permissions are needed.

VIII. Conclusion

In this paper, we presented an anonymity-preserving framework for C-LBSs. A new authentication protocol was proposed to support the protection of users' privacy and the security of their identities during the authentication process. By adopting a one-way hash function, the proposed authentication protocol does not disclose any of the users' credentials. The outstanding features of our proposed framework are as follows:

- provides users with anonymity, which allows them to request and enjoy the available services anonymously;
- allows both users and providers to manage and customize their level of security and privacy;
- ensures security and privacy using the proposed authentication protocol.

In addition, spatial *k*-anonymity is adopted to blur the user's current location into a cloaked region, which also provides users with more privacy. Regarding the privacy and QoS tradeoff, as discussed in section VI, anonymity deployed during user authentication does not affect the QoS. However, in the querying process, increasing the level of protection of the user's privacy reduces the QoS.

Regarding further research, we plan to study the impact of our proposed scheme on real devices, for example, the battery life of the mobile device. Moreover, we plan to extend our proposed framework to location-based social networking, which is a developing trend of C-LBSs.

References

[1] C.-Y. Chow and M.F. Mokbel, "Privacy in Location-Based Services: A System Architecture Perspective," *SIGSPATIAL*, Special Issue, vol. 1, no. 2, 2009, pp. 23-27.

[2] T.H.N. Vu, J.W. Lee, and K.H. Ryu, "Spatiotemporal Pattern Mining Technique for Location-Based Service System," *ETRI J.*, vol. 30, no. 3, June 2008, pp. 421-431.

[3] C. O'Driscoll, "Privacy in Context: Privacy Issues in Ubiquitous Computing Applications," *Proc. ICDM*, 2008, pp. 827-837.

[4] M.F. Mokbel, "Privacy in Location-Based Services: State-of-the-Art and Research Directions," *Proc. MDM*, 2007, p. 228.

[5] J.L. Wang and M.C. Loui, "Privacy and Ethical Issues in Location-Based Tracking Systems," *Proc. ISTAS*, 2009.

[6] H. Truong and S. Dustdar, "A Survey on Context-Aware Web Service Systems," *IJWIS J.*, vol. 5, 2009.

[7] A. Moon et al., "Context-Aware Active Services in Ubiquitous Computing Environments," *ETRI J*, vol. 29, no. 2, 2007, pp. 169-178.

[8] A.M. Bernardos, P. Tarrio, and J.R. Casar, "A Data Fusion Framework for Context-Aware Mobile Services," *Proc. MFI*, 2008.

[9] T.C. Wang, G. Jia, and J. Huang, "Toward Context-Aware Location Based Services," *Proc. ICEIE*, 2010.

[10] S. Martin et al., "A Context-Aware Application Based on Ubiquitous Location," *Proc. UBICOMM*, 2008, pp. 83-88.

[11] M. Nonjur, S.I. Ahamed, and C.S. Hasan, "ELALPS: A Framework to Eliminate Location Anonymizer from Location Privacy Systems," *Proc. COMPSAC*, 2009.

[12] D. Sambandaraksa, "Privacy a Double-Edged Sword," *Bangkok Post.*, Oct. 13, 2010. http://www.bangkokpost.com/tech/techscoop/201103/privacy-a-double-edged-sword

[13] W. Zhang et al., "The Location Privacy Protection Research in Location-Based Service," *Proc. GEOINFORMATICS*, 2010, pp. 1-4.

[14] T. Hashem and L. Kulik, "Safeguarding Location Privacy in Wireless Ad-Hoc Networks," *Proc. UbiComp*, 2007, pp. 372-390.

[15] S. Wang and X.S. Wang, "In-Device Spatial Cloaking for Mobile User Privacy Assisted by the Cloud," *Proc. MDM*, 2010, pp. 381-386.

[16] Z. Gong, G.-Z. Sun, and X. Xie, "Protecting Privacy in Location-Based Services Using *k*-Anonymity without Cloaked Region," *Proc. MDM*, 2010, pp. 366-371.

[17] C. Bettini et al., "Anonymity in Location-Based Services: Towards a General Framework," *Proc. MDM*, 2007, pp. 69-76.

[18] Z.-J. Zhang, Z.-J. Wang, and B.-H. Zhang, "Studies on Median Value of Fuzzy Numbers Based on Confidence Level," *Proc. ICMLC*, 2008, pp. 588-593.

[19] H. Li, S.H. Dick, and W. Pedrycz, "Similarity Confidence Level for Fuzzy Rulebases," *Proc. NAFIPS*, 2004, pp. 882-887.

[20] B. Malek, A. Miri, and A. Karmouch, "A Framework for Context-Aware Authentication," *Proc. IET*, 2008.

[21] A.H.M. Ragab, N.A. Ismail, and O.S.F. Allah, "An Efficient Message Digest Algorithm (MD) for Data Security," *Proc. TENCON*, 2001, pp. 191-197.

[22] X.-H. Cheng and J.-Z. Deng, "Design of SHA-1 Algorithm Based on FPGA," *Proc. NSWCTC*, 2010, pp. 532-534.

[23] P. Kitsos and O. Koufopavlou, "Whirlpool Hash Function: Architecture and VLSI Implementation," *Proc. ISCAS*, 2004.

Songpon Teerakanok is a 4th-year student in the Computer Engineering Program, Faculty of Engineering, Prince of Songkla University (PSU), Hat Yai, Songkhla, Thailand. His scholarship was awarded by JSTP (Junior Science Talent Project), NSTDA (National of Science and Technology Development Agency). His security project work won the 2nd, 3rd, and 1st prize of the Young Scientist Competition (YSC) in 2006, 2007, and 2008, respectively. For more than five years, his research has focused on cryptography, network security, ubiquitous computing, and location-based services.

Chalee Vorakulpipat received his BEng. in electronics engineering from King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, in 1997 and MS in information technology from Kasetsart University, Bangkok, Thailand, in 2000. He was awarded a scholarship from the Royal Thai Government to pursue his doctorate. He earned his PhD in information systems from the University of Salford, Salford, Greater Manchester, UK, in 2008. He has worked as an information security researcher at the National Electronics and Computer Technology Center of Thailand. He has been involved in several projects regarding information security, social networking sites, ubiquitous computing, context-aware computing, e-health, and information systems development. He has authored over 20 refereed papers in these areas, published in conference proceedings and in international journals, including *Computers & Security*, *Advanced Engineering Informatics*, *Automation in Construction*, and *Knowledge Engineering Review*. He also serves on the subcommittee for the Electronic Transactions Commission regarding national information security of Thailand. In an academic role, he works as a lecturer for information systems courses at several universities in Thailand.

Sinchai Kamolphiwong received his PhD from the University of New South Wales, Kensington, New South Wales, Australia. He is now an associate professor in the Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University (PSU), Hat Yai, Songkhla, Thailand. He is a founder and a director of both the Centre for Network Research (CNR) and the Center for NGI. He has authored 80 published technical papers. His main research interests include NGN/NGI, multimedia communications, tele-medicine, network mobility, P2P, and performance evaluation. He is a co-founder of the IPv6 forum in Thailand, the chair of IPv6 UniNet, IPv6-APAN-TH, and a member of the Asia-Pacific IPv6 Task Force. He serves as the chair, co-chair, advisory board member, or technical committee member for over 60 conferences.

Siwaruk Siwamogsatham received his BSEE from Chulalongkorn University, Bangkok, Thailand, in 1994 and was awarded a scholarship from the Thai Government to pursue graduate studies in the USA. He earned his MS and PhD in the field of wireless communications and networking from Ohio State University (OSU), Columbus, OH, USA, in 1997 and 2002, respectively. After his graduation, he joined the National Electronics and Computer Technology Center (NECTEC), Thailand, and is currently the director of the Wireless Information Security and Eco-Electronics Research Unit (WISRU). The highlighted R&D contributions of his research unit include development of a zero-watt standby power consumption system for electronic devices, development of a wireless energy harvesting device, development of mobile applications for energy management, development of ubiquitous and context-aware systems, development of WiMAX broadband access services in a rural province of Thailand, development of strategic award-winning wireless communication devices for national security applications, and design and implementation of secure wireless LAN systems for large enterprises.