This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2015.2481926, IEEE Internet of Things Journal

Privacy-preserving Location Proof for Securing Large-scale Database-driven Cognitive Radio Networks

Yi Li, Lu Zhou, Haojin Zhu, *Member, IEEE,* Limin Sun, *Member, IEEE*

***Abstract*—The latest FCC ruling has enforced database-driven cognitive radio networks (CRNs), in which all secondary users (SUs) can query a database to obtain Spectrum Availability Information (SAI). Database-driven CRNs are regarded as a promising approach for dynamic and highly efficient spectrum management paradigm for large-scale Internet of the Things (IoTs). However, as a typical location-based service (LBS), before providing services to the user, there is no verification of the queried location, which is very vulnerable to Location Spoofing Attack. A malicious user can report a fake location to the database and access the channels that may not be available for its location. This will introduce serious interference to the PUs. In this study, we identify a new kind of attack coined as location cheating attack, which allows an attacker to spoof other users to another location and make them query the database with wrong location, or allows a malicious user to forge location arbitrarily and query the database for services. To thwart this attack, we propose a novel infrastructure-based approach that relies on the existing WiFi or Cellular network Access Points (or AP) to provide privacy-preserving location proof. With the proposed solution, the database can verify the locations without knowing the user's accurate location. We perform comprehensive experiments to evaluate the performance of the proposed approach. Experimental results show that our approach, besides providing location proofs effectively, can significantly improve the user's location privacy.**

***Index Terms*—Location cheating attack, location proof verification, database-driven CRNs.**

I. Introduction

THE rapid advancement of the emerging wireless technology and the ubiquitous computing applications has significantly increased the demand for the communication media resource, wireless spectrum. According to the conventional static spectrum allocation paradigm, most of the spectrum resources have been assigned to the existing primary users (e.g. such as Military communications and broadcast TV). To address the ever increasing demand for spectrum resources and allow more and more Internet-of-things applications, cognitive radio networks (CRNs) have been proposed to improve the efficiency of spectrum utilization. In CRNs, primary users (PUs) are licensed users who have exclusive privilege to access the licensed channels that have been pre-assigned whenever they need. Secondary users (SUs) are unlicensed users who are only allowed to opportunistically access the channels when the channels are not occupied by the PU.

Database-driven CRNs are regarded as a promising approach to allow the dynamic spectrum sharing in many large-scale IoT applications. In database-driven CRNs, all SUs can query a database to obtain Spectrum Availability Information (SAI). Instead of spectrum sensing, SUs are required to submit a request containing its current location information to the database. Until now, FCC has designed several entities (e.g. Comsearch, Google Inc.) as TV band database administrator. Though database-driven CRNs are considered as a promising approach to improve the efficiency of spectrum utilization, they face serious security challenges. Most of the existing researches [11] [5] focus on the location privacy issue. But as a variant of location-based service (LBS), we focus on another security challenge that the user may cheat about its location when querying the database for services. Since there is no location verification for database-driven CRNs, the user can report a fake location information to the database and access the channels that are not available for its location, which can cause serious interference to the PUs. For instance, the United States has announced the spectrum sharing between federal government including military and non-government systems in 3.5GHz band, which is used by the U.S. Department of Defense (DoD) for critical radar systems. Therefore, location spoofing attack will lead to the unauthorized spectrum access of SUs and thus introduce serious interferences to the PUs, which are not acceptable for CRNs. Therefore, location verification in database-driven CRNs is highly desirable.

On the other hand, privacy issue is another important issue in CRNs. As pointed out by the existing researches [11] [21] [22], the attacker can geo-localize the SUs by tracking the users' spectrum query or spectrum utilization history. The existing researches pointed out that, in an anonymized trace data set, four spatiotemporal points are sufficient to uniquely identify the individuals and little outside or social network information is needed to re-identify a targeted individual or even discover real identities of users. Further, loss of location privacy can expose users to unwanted advertisement and location-based spams/scams, cause social reputation or economic damage, and make them victims of blackmail or even physical violence.

In this study, we study the problem of location proof in Database driven CRNs without leaking the users' accurate location information. A straightforward solution against location spoofing attack is to enforce the users to provide location proof while querying for services. A location proof is a piece of electronic data that certifies someone's presence at a certain

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2015.2481926, IEEE Internet of Things Journal

location for some duration. There are several existing works that study the location verification, which can be classified into two categories.

In the infrastructure-independent approach [8] [12], a user can obtain location claims from neighbors. Zhu et al. [8] propose the APPLAUS system, in which co-located Bluetooth enabled users mutually generate location proofs and report them to a location proof server. However, using short-range communication technology could limit the range of verification. In particular, Zeng et al. [12] firstly propose three solutions to detect and defend against GPS spoofing attack in database-driven CRNs, especially introduce a distributed peer location verification (PLV) scheme, which assumes that a certain number of anchor nodes transmit a r-radius beacon signal containing his position to surrounding SUs to provide location verification. However, the maximum transmission power may be the bottleneck of this scheme.

In the infrastructure-dependent approach [14] [19] [6], a set of WiFi access points (APs) are assumed to be available to produce location proof to the users. Luo et al. [14] propose that a user can obtain location proofs for different precision of her location and choose one to disclose to the server, depending on her privacy situation. Anh Pham et al. [19] propose a secure privacy-preserving system relies on existing WiFi AP networks for reporting location-based activity summaries.

As WiFi APs become increasingly prevalent, using WiFi AP for location proof will be fairly effective, especially in urban areas. Different from the previous researches, we propose a novel hybrid infrastructure-based approach that relies on the existing WiFi AP networks or the cellular networks to provide secure and privacy-preserving location proof. In the case of presence of WiFi APs, the users can prove their locations under the help of WiFi APs. However, in the case of unavailable WiFi APs nearby, the users can turn to the cellular tower to request location proof, since the latter can provide a much larger coverage. To protect their location, we adopt the private proximity testing technology to allow the users to query the database for service without leaking their accurate location. Further, we discuss how to achieve the tradeoff of the user privacy and localization accuracy via various system settings.

The contributions of this paper are summarized as below:

- We identify a new kind of attack coined as location cheating attack in database-driven CRNs, which allows an attacker to mislead a user with a fake location and make them query the database with fake locations, or allows malicious user to claim a location arbitrarily and query the database for service.
- We propose a novel infrastructure-based approach that relies on the existing WiFi AP network or cellular network to provide guarantees for location cheating prevention and location privacy for the users. The users can choose the location privacy level as he needs, and, enable the user to prove his location without leaking his accurate location. We also discuss how to find the user's optimal choice to maximize the location privacy while ensuring the service quality.
- We perform the comprehensive experiments to evaluate the performance of the proposed approach. Our experimental results show that our approach, besides providing location proofs effectively, can significantly improve the user's location privacy and also demonstrate the effectiveness of the optimal strategy.

The rest of the paper is organized as follows. Section II gives the background of the database-driven CRNs and identifies two kinds of location cheating attacks. Section III introduces the proposed system architecture. Section IV gives a detailed work flow of the approach and analyses the security of the system. Section V describe a precision optimization problem. Section VI discusses the experimental evaluation. Section VII concludes the paper.

II. Background And Attack Model

A. Overview of Database-Driven CRNs Service

The Database-driven CRNs are normally comprised of three components: a set of primary users (PUs), s set of secondary users (SUs), and the database. The Spectrum Availability Information (SAI) is calculated and stored in the database based on the knowledge of status of PUs and terrain parameters. In order to obtain the SAI before starting to access the channels, the SUs should query the database. The database query process has three phases:

- **Query Phase**: An SU sends a query that contains his current location obtained from his built-in GPS location readings to the database for services. Note that, an SU can query the database for SAI of multiple locations around, i.e., in the vicinity of his current location.
- **Response Phase**: The database calculates the SAI that contains available channels and corresponding maximum transmission power (MTP) for the SU's locations and sends it back to the SU.
- **Notify Phase**: After receiving the SAI from the database, the SU chooses an available channel from the SAI and registers the chosen channel in database. Note that, the notification message is optional. However, the notification phase is important based on the fact that the database can leverage the notification message to manage the system more efficiently.

B. Location Cheating Attacks in Database-driven CRN

As mentioned above, an SU receives the SAI from the database by sending a query containing its current location. Since this happens completely on the SU side, it is relatively easy to launch the attack. In what follows, we define the attack in two cases as summarized below and present more details about the possible damage.

*1) **Active Location Cheating Attack**:* A malicious SU can simply launch an active location cheating attack by reporting a fake location to the database accordance with his own wish. His goal is to obtain the SAI for the reported fake location to gain more advantages.

From the system implementation point of view, there are several ways for a malicious SU to forge a location and make the device believe that it is really in the fake location [18]. In [10], a LocationFaker is developed as a system device to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2015.2481926, IEEE Internet of Things Journal

conduct a fake location arbitrarily which can be accepted as a real location by Android device. Figure 1 shows the concept of such location cheating. Thus, a malicious SU can implement this kind of component to forge a location as they wish, and then report it to the database to obtain the SAI for the fake location.

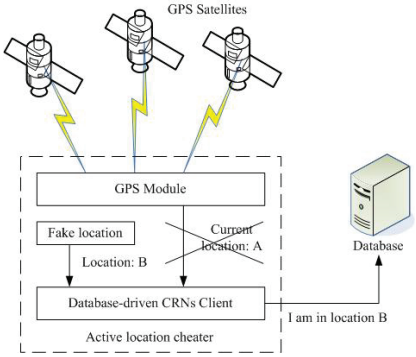


Fig. 1. Illustration of active location cheating. Location Faker generates location B and makes the device believe it is really in location B.

In database-driven CRNs, a mobile SU prefers to choosing a channel with better quality and stable available time to achieve larger communication throughput [20] when it is moving. According to FCC ruling, the system allows an SU to load SAI for multiple locations around, i.e., in the vicinity of its current location and use such information to obtain one or multiple available channels within that area [2]. If the location is a little far away from his current location and also on its moving route, malicious SU can obviously launch an active location cheating attack to occupy the channels with better quality in advance and gain more benefits. For example, he obtains the SAI for location B while actually is located at location A (see Figure 1). Then, he chooses a channel with better quality and sends a notification message to the database, thus making the database believe that he is accessing this channel while he is actually not. If the attacker chooses several channels, this introduces Denial of service (DoS) to other SUs in location B, and also causes loss of the quality of service.

*2) **Passive Location Cheating Attack**:* The attacker is another malicious attacker that is located in the same cell with the victim who is launching a query towards the database for SAI. The attacker's goal is to mislead the victim that he is located in a wrong location and obtain the wrong SAI, which will introduce the interference to the PU.

As pointed out in [12], an attacker can use GPS spoofing device (like a GPS signal simulator) to generate and broadcast fake GPS signals synchronized with the real GPS signals to the target receiver. Then, the fake GPS signals gradually overpower the real GPS signals and replace it. Finally, the target receive locks on the fake GPS signals. After replacing the real GPS, the attacker can fool the target receivers to an arbitrary location. If all victims receive the fake signals from the same attacker, they are all spoofed to the same location $L^{'}$ as shown in Figure 2. Thus, a malicious SU can launch such an attack to spoof SUs that are located in the same cell and make them query the database for services by reporting the spoofed location.

Then, the attacker can occupy the available channel with better quality for location L as his exclusive channel to achieve better transmission throughput. The SUs who query the database for services with spoofed location $L^{'}$ may also cause interference to the primary users (PUs), since they access the channels that may not be available for location L.

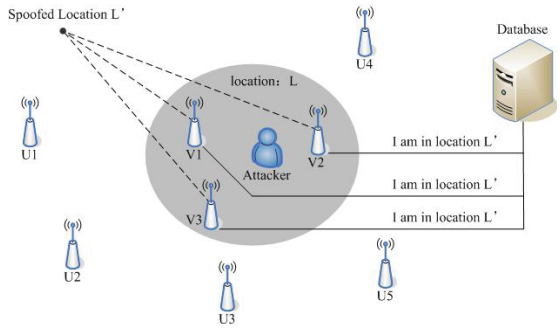


Fig. 2. Illustration of passive location cheating. All victims in location L that query the database for services are spoofed to location L'.

III. System Architecture

To prevent SUs from cheating their reported locations, we propose a novel infrastructure-based approach which is based on an infrastructure of WiFi APs or cellular towers to provide secure and privacy location proofs, such that the database can verify the reported location before providing spectrum services. In this section, we describe the different entities involved in our system: SUs, a WiFi AP network operator or a cellular network, and the database that contains SAI provider database, location proof server, and certificate authority (CA). Figure 3 depicts the overview of the system we consider.

A. The Users

We assume that some users are going to obtain the Spectrum Availability Information (SAI) from the database when they are moving. According to the latest ITEF paws-protocol, a user is allowed to query the database for the SAI by submitting a region that contains his location [1]. To protect the location privacy, we assume that the location submitted to the database by the users specifies a region. These users are equipped with GPS-, WiFi-, and Cellular-enabled devices, and are capable of connecting to the Internet through WiFi or Cellular networks [16]. We also assume a unit-disc model for WiFi APs and cellular towers, that means a user can communicate with a WiFi AP or a cellular tower only if the distance between them is lower than a given radius R, which is equal for all users, WiFi APs and cellular towers. Before querying the database for services, the user should obtain the location proof from a WiFi AP or a cellular tower firstly.

To protect the user's privacy, the users will register to the *Certification Authority* (CA) with some randomly generated pseudonyms and they can use such pseudonyms to protect their privacy while gaining location proof. A pseudonym contains a public/private key pair (K_{pri}, K_{pub}), generated with a public-key encryption scheme. The public key K_{pub} serves as the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2015.2481926, IEEE Internet of Things Journal

pseudonym of the user, while the private key K_{pri} enables the user to digitally sign the message. We assume that users do not give their pseudonyms to other users, and the pseudonyms also should not be easily spoofed and cloned. While registering, we also assume that the CA can generate some other public/private key pairs (PK_{pri}, PK_{pub}), in which PK_{pub} is given to the user and PK_{pri} is kept by the CA.

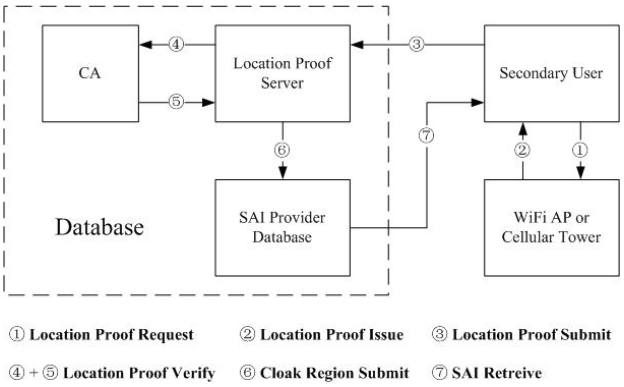


Fig. 3. Overview of the system. First, the user obtains location proof from the nearby WiFi AP or cellular tower, then submits it to the location proof server. Second, CA verifies whether the location proof is legitimate. Only if the verification is pass, then SAI provider database provides the SAI to the user.

B. WiFi AP network and Cellular network

We assume that there are one or multiple WiFi AP networks or cellular networks and each network contains a set of fixed WiFi APs or cellular towers deployed in the area. Each WiFi AP or cellular tower knows its geographic position and its transmission range and can embed its location information into the location proof. All WiFi APs or cellular towers have synchronized clocks within a few hundreds of milliseconds (this can be achieved by using the NTP [3]). Each WiFi AP or cellular tower from the same network shares a public-key group key pairs (GK_{pub}, GK_{pri}), in which GK_{pub} is known to the users and the database, whereas GK_{pri} is only known to the WiFi APs or the cellular towers.

We assume that the WiFi AP network and cellular network are honest but curious, which means that they will obey the rules that we proposed and also may be interest in tracking the users' locations based on the collected information. We also assume that the WiFi AP network and cellular network do not collude with the database.

C. Database

To prevent users from cheating about their location, we need to add the location verification functionality in the database's side, thus in our system we make a little change to the database and divide it into three parts: *Location Proof Server*, *Certification Authority (CA)* and *SAI Provider Database*.

1) Location Proof Server: *Location proof Server* directly communicate with the users who submit their location proofs. The goal of the *Location proof Server* is to collect location proofs. As the identities of the location proofs are stored as pseudonyms, even though the *Location proof Server* is compromised by the attacker, it is impossible for the attacker to know the real identity of the location proof.

2) CA: As commonly assumed in many networks, we consider an online CA run by a trusted party. CA is the only party who knows the mapping between real identity and pseudonym. CA also knows the secret key PK_{pri} corresponding to the user, since the location proof is encrypted with PK_{pub}, thus it can use PK_{pri} to verify the location proof. We assume the CA is trusted and does not collude with the WiFi AP network.

3) SAI Provider Database: The *SAI Provider Database* is more similar to the traditional database described in the previous database-driven CRNs system. After the verification of location proof is pass, the *Location Proof Server* will submit the region in spectrum request to the *SAI Provider Database*. Then, the *SAI Provider Database* will calculate the SAI for the region and send it back to the user.

IV. The Proposed Privacy Preserving Location Verification Scheme

In this section, we present our approach for privacy-preserving location verification (PPLV) scheme. First, we give an overview of the proposed approach and define the main processes it involves. Subsequently, we present the detailed work flow. Finally, we analysis the security and privacy. Figure 3 shows an overview of the approach and main processes involved.

A. Overview of PPLV

As WiFi APs become increasingly prevalent and can provide more accurate location proof, in our scheme, the users prefer to requesting location proof with WiFi AP; while there are no WiFi APs nearby, then the users choose the nearby cellular tower to request for location proof. To protect the location privacy, we adopt a grid reference system with different levels to represent locations, and users can choose appropriate level to qwuery for location proof.

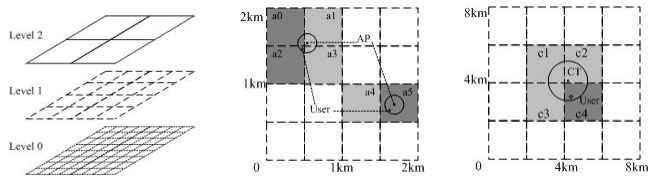
In the case of cellular tower, since the cellular tower can provide a larger coverage, the user does not need to specify the region. He specifies a granularity of level to protect his location privacy, and requests location proof with the cellular tower. Then the cellular tower embeds its coverage to the location proof and sends back to the user. Then the user can query the database for services by submitting the location proof containing the cellular tower's coverage. Finally, the database calculates the SAI for the coverage and sends back to the user.

In the case of WiFi AP, since the WiFi AP's coverage is much smaller than the cell size, the user not only specifies the granularity of level, but also specifies the region. To further protect the location privacy (i.e. enable the user to prove his location without leaking the accurate cell to the database), we adopt private equality testing [4] to determine if two cells match without revealing the exact cell number. The basic idea is that if the user is located at cell a and WiFi AP is located at cell b, CA learns if $a = b$ and nothing else. We will give a detailed work flow in section IV-C.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2015.2481926, IEEE Internet of Things Journal

B. System Initialization

1) ***Global setup****:* The location of a user can be defined with different granularities. The user may want to define their location in appropriate granularity under different situations. For example, the user may be willing to use fine-grained location information in urban area while using coarse-gained location information in countryside. As show in Figure 4(a), the system adopts a grid reference [9] to represent locations, where grid indices represent areas covered by grid cells. All users, all WiFi APs, all cellular towers and the *SAI provider Database* share a list of coordinate-axis aligned grid system denoted by $\Gamma(l)(l = 0, 1, 2, \cdots)$ of different levels. For each level l, the grid cell size, i.e. width and height, is fixed and equal. The grid cell size at level 0 is equal to $250m$, and the size at level $l-1$ is always lower than that at level l. Every grid cell $c \in \Gamma(l)$ is identifiable by an index $id(c) \in \mathbb{N}$ and is fully contained by several grid cells $c \in \Gamma(l-1)$.



(a) Grid system (b) WiFi AP (c) Cellular Tower

Fig. 4. Grid reference system. We assume the grid cell with side length of 250 meters for level 0, the unit-disc communication model with a radius of 25 meters for WiFi APs and of 2 kilometers for cellular towers.

2) ***User setup****:* Let G be a cyclic group of prime order p and g a generator of G. We assume that the Diffie-Hellman problem is hard in G. All users, WiFi APs, and the CA in the system are pre-configured with the same G and g. We will use $\mathbb{Z}_p$ to denote the set $\{0, \cdots, p-1\}$. When the user firstly registers to the CA, the CA generates several public/private key pairs (e.g. CA chooses a random x in $\mathbb{Z}_p$ and computes $h = g^x$), in which h given to the user served as PK_{pub}, and x is kept by the CA served as PK_{pri}. Thus, the user has h, and h will be used as his ElGamal public key. We assume that the WiFi APs has the user's public key h. And also the user can obtain different pseudonyms (provided by the CA) while registering.

C. System Process

1) ***Location Proof Request****:* The user periodically uses its WiFi module to scan the channels, hearing beacons from the nearby WiFi APs. A user does not have to transmit any data to receive a beacon. He merely needs to listen, thus he can choose a beacon with stronger power to request location proof. Upon receiving a beacon, the user extracts the beacon's sequence number to use it in the request for location proof. Sending back to the WiFi AP guarantees the freshness of the request. To protect its location privacy, the user can choose the appropriate granularity (the level of the grid system) suitable for him. Thus, the location proof request can be denoted as:

$$Request = (P_{user}, n, l, t, R_{user}, C_{loc_{user}}) \quad (1)$$

Here, P_{user} denotes the user's pseudonym; n denotes the beacon's sequence number or preamble's random number; l denotes the granularity of level. t denotes the request time. R_{user} is a set of cell ids that denote the region that the user queries for. $C_{loc_{user}}$ encrypted with the public key PK_{pub} contains the user's location information, which can be denoted as

$$C_{loc_{user}} = (g^r, h^{a+r}) \quad (2)$$

Here, r is a random number in $\mathbb{Z}_p$, a is the user's grid cell id under the level of the grid system l.

Assume that a user and a WiFi AP use granularity of level 1 in Figure 4(b). The user specifies the region R_{user}, containing grid cells $\{a_0, a_2\}$, in which grid cell a_2 is the user's cell, then he computes an encryption of his location a_2 encoded as h^{a_2} and sends the ciphertext to the WiFi AP. In particular, the user computes

$$C_{loc_{user}} = (g^r, h^{a_2+r}) \quad (3)$$

and embeds it into the request.

2) ***Location Proof Issue****:* Upon receiving the location proof request from the user, the WiFi AP firstly checks whether the number is a current one. We assume that the WiFi AP can accept requests whose sequence number were broadcasted within last 100 milliseconds. Since 802.11 sequence numbers repeat after 4096 frames, the $100ms$ time interval is small enough to prevent security attacks [15]. Then, the WiFi AP should verify that the region R_{user} is reasonable (i.e. since the user's cell must be in coverage area of the WiFi AP R_{AP}, R_{user} should have intersection with R_{AP} [17]). If the intersection is denoted as $\{b_1, \cdots\}$, then the WiFi AP uses the element of $\{b_1, \cdots\}$ and the ciphertext $C_{loc_{user}} = (g_1, g_2)$ from the user to construct a new encryption message, which can be denoted as

$$C_{loc_{AP}} = (g_1^s g^t, g_2^s h^{(t-s\cdot b_1)}, \cdots) \quad (4)$$

Here, s is a random none-zero number in $\mathbb{Z}_p$, t is a random number in $\mathbb{Z}_p$. Note that setting $w = s \cdot r + t$, we get

$$C_{loc_{AP}} = (u_0, u_1, \cdots) = (g^w, h^{s\cdot(a-b_1)+w}, \cdots) \quad (5)$$

As show in Figure 4(b), the WiFi AP finds the coverage area $\{a_0, a_1, a_2, a_3\}$, and compares with R_{user}. The intersection grid cells are $\{a_0, a_2\}$. Then, the WiFi AP computes

$$C_{loc_{AP}} = (u_0, u_1, u_2) = (g^w, h^{s\cdot(a_2-a_0)+w}, h^{s\cdot(a_2-a_2)+w}) \quad (6)$$

Then, the WiFi AP embeds its location information into the location proof response that is signed with private group key GK_{pri}, and sends back to the user. The location proof response can be denoted as

$$Response = sig_{GK_{pri}}(P_{user}, l, t, R_{user}, C_{loc_{AP}}) \quad (7)$$

3) ***Location Proof Verify****:* To submit a location proof, a user must sign it before transmission. Upon receiving the location proof, the *Location Proof Server* performs four steps. First it checks the user's signature to make sure that the location proof has not been tampered with while submitting. Second, it checks the WiFi AP's signature that is embedded in the location proof. This step makes sure that the location proof

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2015.2481926, IEEE Internet of Things Journal

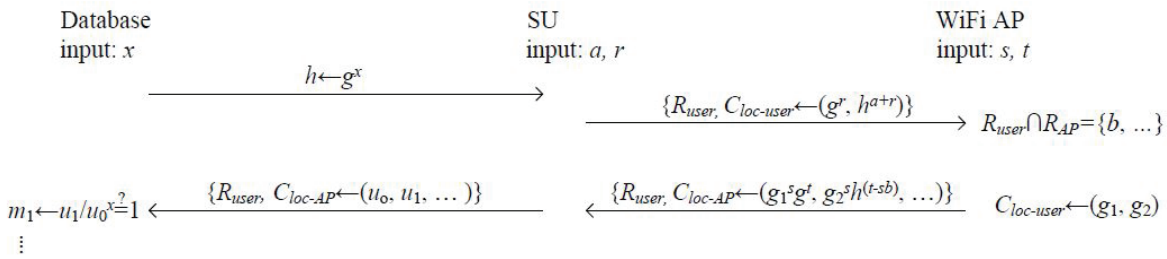


Fig. 5. location proof verification with WiFi AP.

has not been modified by the user. Third, it checks that the user is indeed the recipient of the location proof. Fourth, if these three steps are successful, it forwards the P_{user} and $C_{loc_{AP}}$ to the CA for verification. CA searches the corresponding secret key PK_{pri} for P_{user}, and decrypts $C_{loc_{AP}}$, or computes $\{m_1 \leftarrow u_1/u_0^x, m_2 \leftarrow u_2/u_0^x, \cdots\}$. If one of elements is equal to 1, the location proof is considered as legitimate, then the *Location Proof Server* submits R_{user} and l to the *SAI Provider Database*. Otherwise, it is rejected.

4) ***SAI Retrieval****:* When *SAI Provider Database* receives l from *Location Proof Server*, it applies the granularity of level l and calculates the SAI for grid cells in R_{user} based on the granularity of level l (e.g. in Figure 4(b), it will calculate the SAI for grid cell $\{a_0, a_2\}$). Note that, a channel in the SAI for grid cell a_2 means that the channel is available for all subcells in cell a_2, thus when a user specifies a higher granularity of level, the database may respond with the SAI contains less available channels.

D. Security and Privacy Analysis

1) ***Malicious User****:* First, we prevent users from forging the location proofs by using the digital signature GK_{pri}. Moreover, the users can only obtain the valid location proof if they are in transmission range with the WiFi APs or cellular towers. Second, a fake region R_{user} can be verified by the WiFi AP. Third, a fake location a can be verified by the CA based on our location proof verification scheme. Thus, a malicious user can be detected immediately when he is cheating about his location.

2) ***Curious Database****:* In our scheme, the *Location Proof Server* can only access to location proofs and pseudonyms of the users. It can not know the real identities of the location proofs. Moreover, the location proof verification do not reveal the user's accurate location. By using *Spectrum Utilization based Location Inferring attack* [11], the user can be geo-located to an accurate estimated location. However in our scheme, we can also use different granularity level to protect the user's location privacy.

3) ***Curious WiFi AP Network****:* Several WiFi APs could collude and track the location of a user based on the collected location proof requests. To address this issue, our scheme employs randomized pseudonyms as well as randomized encryption keys. Since WiFi APs only know pseudonym P_{user} and encryption key PK_{pub}, and the user uses a group key containing a pseudonym and a public key PK_{pub} each time while requesting location proof, it could not link different location proof requests to a same user, thus it can not track the user's trajectory.

V. Optimal Location Proof Request

As mentioned in IV-C4, a higher granularity of level introduces a less number of available channels. A user may want to specify a higher granularity of level while he can tolerate the service quality loss. We describe this as a precision optimization problem that formalizes the objectives of the user. In this section, we first define the metrics of service quality and location privacy, and then conduct the simulations to find the user's optimal choice.

A. Service Quality Metric

Intuitively, the service quality can be represented by channel quality. As mentioned in II-A, the SAI produced by *SAI Provide Database* contains the available channels and the corresponding allowed MTP. We assume that the SAI can be denoted as $\{(CH_1, MTP_1), (CH_2, MTP_2), \cdots\}$, in which CH_i denotes the i-th available channel, and MTP_i denotes the corresponding allowed MTP level. Each channel has five different MTP levels, denoted as level 1 to 5, and level 5 denotes the maximum transmission power, which means the user is out of the area that CH_i covers or the PU is off. Each level has a different weight w_i denoting the quality of the channel. Thus, the service quality Q_a for grid cell a can be denoted as

$$Q_a = \sum_{i \in SAI} MTP_i \cdot w_i \tag{8}$$

Thus, when a user located at location a queries the database with region R, the service quality loss Q_{loss} between location a and region R can be denoted as

$$\begin{aligned} Q_{loss}(a, R) &= Q_a - Q_R \\ &= \sum_{i \in SAI_a} MTP_i \cdot w_i - \sum_{i \in SAI_R} MTP_i w_i \end{aligned} \tag{9}$$

We assume that users impose a maximum tolerable service quality loss, Q_{loss}^{max}, caused by reporting region instead of their actual locations. Formally,

$$Q_{loss} \leq Q_{loss}^{max} \tag{10}$$

This constraints the specified granularity of level l, which must not result in lower quality for users, thus the maximum tolerable service loss Q_{loss}^{max} can be used as a threshold when a user queries the databases for services.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2015.2481926, IEEE Internet of Things Journal

B. Location Privacy Metric

The attacker's goal is to infer user's actual location a given the reported region R. As mentioned in IV-D2, by using spectrum utilization based Location attack, the user can be geo-located to an accurate location. We assume that the user can be located to a grid cell under granularity of level l, and we describe the attack result as a probability density function $Pr(a'|R)$. We follow the definition in [13] and quantify the user's location privacy as expected error distance in the attack. Thus, the user's location privacy can be denoted as

$$Pri(a, R) = \sum_{a' \in R} \psi(l) Pr(a'|R) d(a, a') \tag{11}$$

Here, $\psi(l)$ is a linear function about l, denotes the user's diverse profile, and the higher level l specifies, the better location privacy achieves, a and a' are the actual location and the estimated location respectively, $d(a, a')$ denotes the Euclidean distance between them.

We assume that users consider a minimum location privacy level Pri^{min} when querying for services.

C. Optimal Strategy for the User

The user sets a minimum accepted threshold of location privacy Pri^{min} and a maximum accepted threshold of service quality loss Q_{loss}^{max}, then do the following maximization program to choose an optimal granularity of level l to obtain maximum privacy while ensuring the service quality:

$$\textbf{Maximize} \quad \sum_{a' \in R} \psi(l) Pr(a'|R) d(a, a') \tag{12}$$

suject to

$$Q_{loss}(a, R) \leq Q_{loss}^{max} \tag{13}$$

$$\sum_{a' \in R} \psi(l) Pr(a'|R) d(a, a') \geq Pri^{min} \tag{14}$$

$$Pr(a'|R) = 1, \forall a' \in R \tag{15}$$

VI. Evaluation

In this section, we evaluate the effectiveness and efficiency of the proposed infrastructure-based approach from following aspects: 1) cost of involved three entities; 2) effectiveness of the proposed approach; 3) tradeoff between location privacy and service quality.

A. Cost of Involved Entities

We conduct experiments on both of mobile device and computer. The implementation platform includes a 64-bit computer with Intel i5 CPU of 2.5GHz and 4G memory and an android smart phone with Exynos 4412 1.6GHz CPU and 2G RAM, 16G ROM. We evaluate the cost at user side on smart phone, and evaluate the cost at WiFi AP and CA side on computer. In the experiment, we evaluate the efficiency of three involved entities under different sizes of prime p as shown in table I.

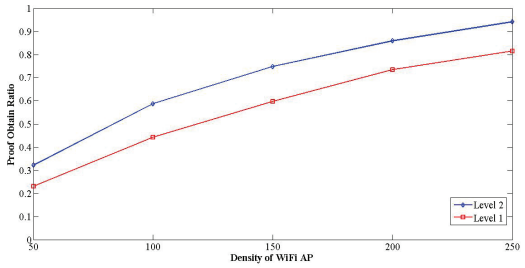


Fig. 6. Location proof obtain ratio under different density of WiFi AP.

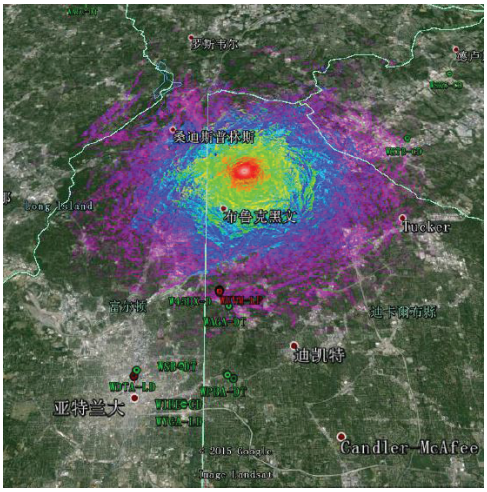


Fig. 7. The coverage of WAGA-DT located at $33.941997°, 84.410411°$ with a TV tower, whose channel is $ch27$ and ERP is $1000kW$.

*1) **Cost of Location Proof Request on User Side**:* The first metric is the cost of location proof request, which is generated at the user side. We evaluate the cost of location proof request by evaluating the computation latency. Since the user needs to perform two exponentiations when generating location proof request, it is observed that the computation latency is approximate to a constant value. Note that, this process could be sped up considerably by using pre-computations, which could further reduce the computation latency of location proof request.

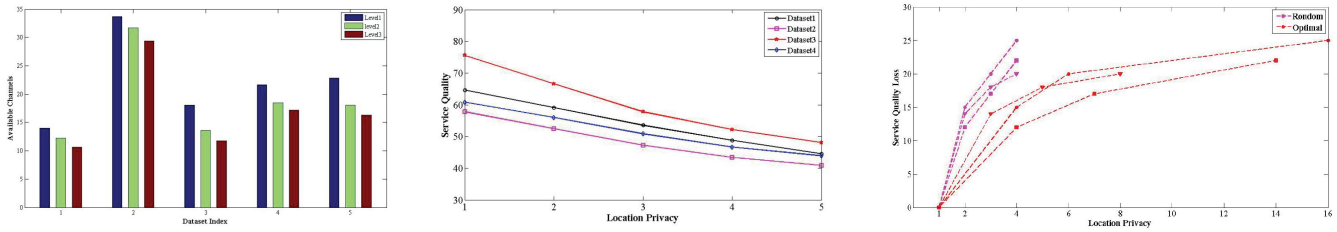
*2) **Cost of Location Proof Issue on WiFi AP Side**:* The second metric is the cost of location proof issue, which is generated at WiFi AP side. The cost of location proof issue depends on the number of the intersection grid cells. Since computing a product of exponents such as $g_1^s g^t$ is only slightly expensive than computing a single exponent, we count these as a single exponentiation. The best case is to compute two exponentiations while the worst case is to compute five exponentiations.

*3) **Cost of Location Proof Verification on CA Side**:* The last metric is the cost of location proof verification, which is performed at the CA side. The cost of location proof verification is to compute $\{u_1/u_0^x, \cdots\}$. Since computing division is much faster than computing exponentiation, the cost of location proof verification for each user is to compute an exponentiation.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2015.2481926, IEEE Internet of Things Journal

bit number of p	location proof request cost(user)	location proof issue cost(WiFi AP)	location proof verification cost(CA)
128	48 $\sim$ 52ms	20 $\sim$ 60ms	10 $\sim$ 12ms
256	85 $\sim$ 90ms	42 $\sim$ 150ms	21 $\sim$ 30ms
512	185 $\sim$ 190ms	82 $\sim$ 250ms	41 $\sim$ 50ms

TABLE I
EVALUATION OF THE COST OF INVOLVED THREE ENTITIES ON SMART PHONE WITH EXYNOS 4412 1.6GHZ CPU AND COMPUTER WITH INTEL I5 CPU OF 2.5GHZ



(a) number of available channels (b) service quality vs location privacy (c) optimal strategy under various service quality loss

Fig. 8. tradeoff between location privacy and service quality.

B. Effectiveness of the Proposed Approach

We evaluate the effectiveness of the proposed approach by setting up an simulation environment with several WiFi APs uniformly distributed in a region of $10km \times 10km$ which is divided into 100×100 cells. We use the Levy walk mobility model to generate trajectory for mobile user. For each simulation, we generate a Levy trace for the user, and assume the user should update a location proof with certain time interval. A successful location proof obtained when the user is within the transmission range of a WiFi AP.

Figure 6 shows the location proof obtain ratio under different granularity of level with different densities of WiFi AP. We can see that the location proof obtaining ratio reaches 90% when the density of WiFi is $200/km^2$. The higher granularity of level the user specifies, the more location proof obtains.

C. Tradeoff between Location Privacy and Service Quality

We conduct the experiment by using the the SAI of Atlanta in white space database release on TVFool [7]. In Atlanta area, there are 48 channels totally, one of which is shown in Figure 7. Then we choose 5 regions of $50km \times 50km$, and each region is divided into 100×100 cells. We assume 10000 users uniformly distributed in 10000 cells for each region. We randomly choose 100 users in different cells and perform two kinds of experiments.

As mentioned in IV-C4, a higher granularity of level introduces a less number of available channels. In the first experiment, for each channel (see Figure 7), we clarify the cells into two kinds. The cells in the chromatic coverage represent "0", which means the channel is not available while the rest cells represent "1", which means the channel is available. Then we randomly choose 100 users from the above 5 regions and calculate the SAI for them. Figure 8(a) shows the results. We measure the average number of available channels in SAI under granularity of level 1, 2 and 3 for 100 users. It is observed that the number of available channels between two levels can be different from 2 $\sim$ 5, and the average number of available channels has an obvious reduction with the increasing of level for all 5 data sets, which is right on target with our expectations.

In the second experiment, for each channel, we clarify the cells into six kinds according to their different colors. "0" represents the cell which is not allowed to use the channel. "1 $\sim$ 5" represent the different allowed MTP levels for different cells, from weak to strong, and we assign different weights to different levels. Then, we calculate the average service quality for 100 users in 4 data sets according to the Equation 8. Figure 8(b) show the results. Unsurprisingly, increasing the granularity of level also degrades the service quality, which is in line with the first experimental results.

We then demonstrate the efficiency of the optimal strategy, comparing with random selections. Figure 8(c) illustrates the maximum location privacy achieved given various service quality loss. We randomly choose serval samples of users, and compare the location privacy with random strategy and optimal strategy. It is obvious that the better location privacy can be achieved with optimal strategy. The experimental result shows that our optimal strategy can significantly improve the user's location privacy with a given service quality loss.

VII. CONCLUSION

In this paper, we identify a new kind of attack coined as location cheating attack in database-driven CRNs, in which users can cheat their locations to gain more advantages, and this can cause interference to PUs. To thwart this attack, we propose a novel infrastructure-based approach that relies on the existing WiFi AP network or cellular network to provide secure and privacy location proof. On the one hand, we use a grid reference system with different granularities to represent locations, on the other hand, we adopt the private proximity testing technology to further improve the user's location privacy. We conduct the program to find the optimal strategy to maximum the user's location privacy. Simulations well demonstrate the effectiveness and efficiency of the proposed approach. Experiments by using the SAI of Atlanta in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2015.2481926, IEEE Internet of Things Journal

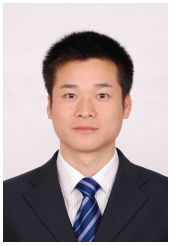
white space database release on TVFool show the tradeoff between location privacy and service quality and demonstrate the effectiveness of the optimal strategy. Our future work includes other security issues in database-driven CRNs.

ACKNOWLEDGMENT

This work is supported by National High-Tech R&D (863) Program (no. 2015AA01A707) and National Science Foundation of China (no. 61272444, U1401253, U1405251, 61411146001, 61472418).

REFERENCES

[1] Chen V, Das S, Zhu L, et al. *Protocol to Access White-Space (PAWS) Databases*. draft-ietf-paws-protocol-10 (work in progress), 2014.
[2] Band, Broadcast. "FEDERAL COMMUNICATIONS COMMISSION 47 CFR Part 15."
[3] Mills D, Martin J, Burbank J, et al. *Network time protocol version 4: Protocol and algorithms specification*. IETF RFC5905, June, 2010.
[4] Narayanan, Arvind, et al. "Location Privacy via Private Proximity Testing." *NDSS*. 2011.
[5] Zhang L, Fang C, Li Y, et al. "Optimal Strategies for Defending Location Inference Attack in Database-driven CRNs," *International Conference on Communications(ICC), 2015 IEEE Conference on*. IEEE, 2015.
[6] Capkun, Srdjan, Levente Buttyan, and Jean-Pierre Hubaux. "SECTOR: secure tracking of node encounters in multi-hop wireless networks." *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*. ACM, 2003.
[7] "TV Fool," March, 2012. [Online]. Available: http://www.tvfool.com/
[8] Zhu, Zhichao, and Guohong Cao. "Applaus: A privacy-preserving location proof updating system for location-based services." *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2011.
[9] Zheng, Yao, et al. "Sharp: Private proximity test and secure handshake with cheat-proof location tags." *Computer Security-ESORICS 2012*. Springer Berlin Heidelberg, 2012. 361-378.
[10] Li, Muyuan, et al. "All your location are belong to us: Breaking mobile social networks for automated user location tracking." *ACM MobiHoc*. ACM, 2014.
[11] Gao, Zhaoyu, et al. "Location privacy in database-driven cognitive radio networks: Attacks and countermeasures." *INFOCOM*, IEEE, 2013.
[12] Zeng, Kexiong, et al. "Location spoofing attack and its countermeasures in database-driven cognitive radio networks." *Communications and Network Security (CNS)*, IEEE, 2014.
[13] Shokri, Reza, et al. "Quantifying location privacy." *Security and Privacy (SP), 2011 IEEE Symposium on*. IEEE, 2011.
[14] Luo, Wanying, and Urs Hengartner. "Veriplace: a privacy-aware location proof architecture." *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2010.
[15] Saroiu, Stefan, and Alec Wolman. "Enabling new mobile applications with location proofs." *Proceedings of the 10th workshop on Mobile Computing Systems and Applications*. ACM, 2009.
[16] Jiajia Liu, Shangwei Zhang, Nei Kato, et al. "Device-to-device communications for enhancing quality of experience in software defined multi-tier LTE-A networks." *IEEE Network*, 2015, 29(4):46-52.
[17] Siksnys, Laurynas, et al. "Private and flexible proximity detection in mobile social networks." *Mobile Data Management (MDM), 2010 Eleventh International Conference on*. IEEE, 2010.
[18] He, Wenbo, Xue Liu, and Mai Ren. "Location cheating: A security challenge to location-based social network services." *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*. IEEE, 2011.
[19] Pham, Anh, et al. "Secure and private proofs for location-based activity summaries in urban areas." *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2014.
[20] Jiajia Liu, Nei Kato, et al. "Throughput and Delay Tradeoffs for Mobile Ad Hoc Networks With Reference Point Group Mobility." *IEEE Transactions on Wireless Communications*, 2015, 14(3): 1266-1279.
[21] Gao Z, Zhu H, Liu Y, et al. "Location privacy leaking from spectrum utilization information in database-driven cognitive radio network." *ACM CCS*. ACM, 2012: 1025-1027.
[22] Gao Z, Zhu H, Li S, et al. "Security and privacy of collaborative spectrum sensing in cognitive radio networks." *Wireless Communications*, IEEE, 2012, 19(6): 106-112.



Yi Li is a second year master candidate in computer science department, Shanghai Jiaotong University. He received the B.Sc. degree in Department of Electronic Science and Engineering from Nanjing University, China, in 2006. He started working at China Satellite Maritime Tracking and Control Department since 2006. His research interests include wireless network security issues.

Lu Zhou received the B.S. degree in Department of computer science from SiChuan University, Chengdu, China, and is currently working toward the Ph.D. degree in computer science at Shanghai Jiao Tong University, China. His research interests include wireless network security issues.

Haojin Zhu is currently an Associate Professor with Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. He received his B.Sc. degree (2002) from Wuhan University (China), his M.Sc.(2005) degree from Shanghai Jiao Tong University (China), both in computer science and the Ph.D. in Electrical and Computer Engineering from the University of Waterloo (Canada), in 2009. His current research interests include network security and data privacy.

He published 29 international journal papers, including IEEE Trans. On Parallel and Distributed Systems, IEEE Trans. on Wireless Communication, IEEE Trans. on Vehicular Technology, IEEE Wireless Communications, IEEE Communications, and 50 international conference papers, including ACM MOBICOM, ACM MOBIHOC, IEEE INFOCOM, IEEE ICDCS, IEEE GLOBECOM, IEEE ICC, IEEE WCNC. He received the IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award (2014) due to his contribution to wireless network security and privacy, Distinguished Member of the IEEE INFOCOM 2015 Technical Program Committee, Outstanding Youth Post Expert Award for Shanghai Jiao Tong University, SMC-Young Research Award of Shanghai Jiao Tong University. He was a co-recipient of best paper awards of IEEE ICC 2007 and Chinacom 2008. He serves as the Associate/Guest Editor of IEEE Internet of Things Journal, IEEE Wireless Communications, IEEE Network, and Peer-to-Peer Networking and Applications.

Limin Sun received his B.S., M.S., and D.Sc. degree in College of Computer, National University of Defense Technology in China in 1988, 1995, and 1998, respectively. He is currently a Professor in Institute of Information Engineering at Chinese Academy of Sciences, chair of Beijing Key Lab of Internet of Things Security, deputy chair of China Computer Federation Technical Committee on Sensor Network. He is also a member of IEEE and a senior member of China Computer Federation (CCF). His research interests include Wireless Sensor Networks, Internet of Things and its security, and intelligent transportation systems. He led dozens of important projects, including National 973 projects, key projects of Natural Science Foundation of Chinese, the National Major Projects and the Pilot projects of Chinese Academy of Sciences. He published 5 academic books and 130 papers on journals such as IEEE TPDS, TOSN, TMC, and on international conferences such as SENSYSMOBISYSMOBICOMICDCSINFOCOM. He was granted more than 30 patents and software copyrights.