Secure and Privacy-Preserving Location Proof in Database-Driven Cognitive Radio Networks

Yi Li[1], Lu Zhou[1], Haojin Zhu[1(✉)], and Limin Sun[2]

[1] Shanghai Jiao Tong University, Shanghai 200240, China
{bluely02,zhoulu,zhu-hj}@sjtu.edu.cn
[2] Chinese Academy of Sciences, Beijing 100093, China
sunlimin@iie.ac.cn

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 available information (SAI). Database-driven CRNs is regarded as a promising approach for dynamic and highly efficient spectrum management paradigm. However, as a typical location-based service (LBS), there is no verification of the queried location, which is very vulnerable to Location Spoofing Attack. This will introduce serious interference to the PUs. In this study, we identify a new kind of attack coined as location cheating attack. To thwart this attack, we propose a novel infrastructure-based approach to provide privacy-preserving location proof. With the proposed solution, the database can verify the locations without knowing the user's accurate location. Experimental results show that our approach, besides providing location proofs effectively, can significantly improve the user's location privacy.

Keywords: Location cheating attack · Location proof verification · Privacy-preserving · Database-driven CRNs

1 Introduction

The rapid advancement of the emerging wireless technology has significantly increased the demand for the wireless spectrum resources. However, most of the spectrum resources have been assigned to the existing systems (e.g. such as Military communications). To address the ever increasing demand for spectrum resources, cognitive radio networks (CRNs) have been proposed to improve the efficiency of spectrum utilization.

In database-driven CRNs, SUs are required to submit a request containing its location to the database to obtain spectrum available information (SAI). As a variant of location-based service (LBS), we focus on the security challenge that the user may cheat about its location when querying for services. Since there is no location verification, this will lead to the unauthorized spectrum access of SUs and introduce serious interferences to PUs. On the other hand, privacy issue

K. Xu and J. Zhu (Eds.): WASA 2015, LNCS 9204, pp. 345–355, 2015.
DOI: 10.1007/978-3-319-21837-3_34

is another important issue in CRNs. Loss of location privacy can expose users to location-based spams, cause social reputation or economic damages. Therefore, location verification in database-driven CRNs is highly desirable.

In this study, we study the problem of location proof in database-driven CRNs without leaking the users' accurate location. A straightforward solution 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 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 [5,9], a user can obtain location claims from neighbors. However, the maximum transmission power may be the bottleneck of this scheme. In the infrastructure-dependent approach [4,12], a set of WiFi access points (APs) exists to produce location proof to the users.

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 location proof. In the cases 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 tune 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.

The contributions of this paper are summarized as below:

- We identify location cheating attack in database-driven CRNs, which allows an attacker to mislead users with a fake location and make them query the database with fake locations, or allows a 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.
- Our experimental results show that our approach, besides providing location proofs effectively, can significantly improve the user's location privacy.

The rest of the paper is organized as follows. Section 2 gives the background of the database-driven CRNs and identifies two kinds of location cheating attacks. Section 3 introduces the proposed system architecture. Section 4 gives a detailed work flow of the approach and analyses the security of the system. Section 5 discusses the experimental evaluation. Section 6 concludes the paper.

2 Background and Attack Model

2.1 Overview of Database-Driven CRNs Service

The Database-driven CRNs contain three components: primary users (PUs), secondary users (SUs), and the database. The SAI is calculated and stored in the database. The database query process [1] 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.

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.

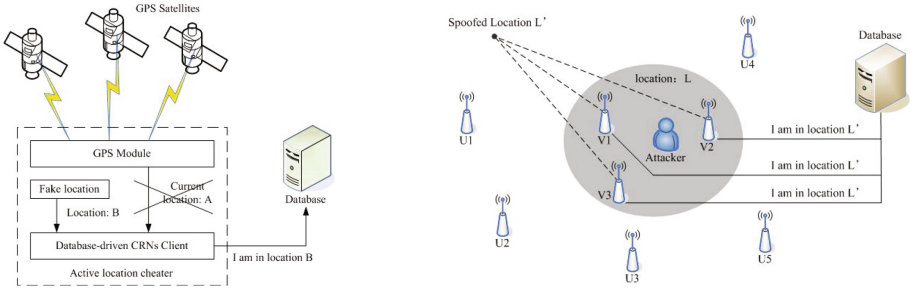
Notify Phase: After receiving the SAI from the database, the SU chooses an available channel from the SAI and registers the chosen channel in database.

2.2 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 hack. In what follows, we define the attack in two cases as summarized below and present more details about the possible damage.

Active Location Cheating Attack. A malicious SU can simply launch an active location cheating attack by reporting a fake location to the database. His goal is to obtain the SAI for the 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 [11]. In [7], a LocationFaker is developed as a system device to conduct a fake location arbitrarily which can be accepted as a real location



(a) active location cheating (b) passive location cheating

Fig. 1. (a) Illustration of active location cheating: Location Faker generates location B and makes the device believe it is really in location B. (b) Illustration of passive location cheating: all victims in location L that query the database for services are spoofed to location L'.

by Android device. Figure 1(a) shows the concept of such location cheating. In Fig. 1(a), the attacker obtains the SAI for location B while actually locates at location A. Then, he chooses several channels with better quality and sends a notification message to the database, making the database believe that he is accessing these channels while actually not. This introduces Denial of service (DoS) to other SUs in location B, and also causes service quality loss.

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 [9], an attacker can use GPS spoofing device 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 make the target receiver lock on them. 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 Fig. 1(b). 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.

3 System Architecture

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 2 depicts the overview of the system we consider.

3.1 The Users

We assume that some users are going to obtain the SAI from the database when they are moving. These users are equipped with GPS-, WiFi-, and cellular-enabled devices. We also assume a unit-disc model for WiFi APs and cellular towers, that means an user can communicate with a WiFi AP or a cellular tower only if the distance between them is lower than a given radius R. 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 register to the CA with some random 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}). We assume that users do not give their pseudonyms to other users, and pseudonyms should not be easy to spoof and clone. While registering, we also assume that the CA can generate 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.

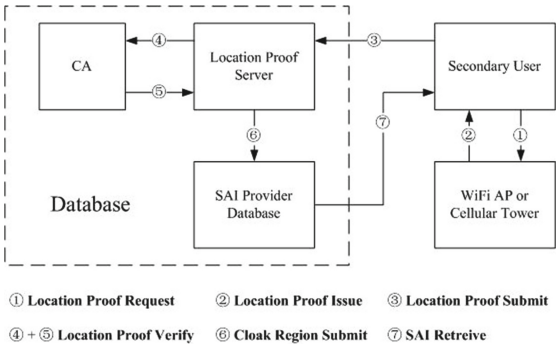


Fig. 2. Overview of the system. First, the user obtains location proof from the nearby WiFi AP or cellular tower, then submit it to the location proof server. Second, CA verifies whether the location proof is legitimate. Only if the verification is pass, then SAI provider database provides the SAI to the user.

3.2 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. Each WiFi AP or cellular tower from the same network shares a public-key group key pairs (GK_{pub}, GK_{pri}). We assume that the WiFi AP network and cellular network are honest but curious, and do not collude with the database.

3.3 Database

We make a little change to the database and divide it into three parts: *Location Proof Server*, *Certification Authority (CA)* and *SAI Provider Database*.

Location Proof Server. *Location Proof Server* directly communicate with the users to collect location proofs.

CA. CA is the only party who knows the mapping between real identity and pseudonym. CA also knows the corresponding secret key PK_{pri}.

SAI Provider Database. The *SAI Provider Database* calculates the SAI and sends it back to the users.

4 The Proposed Privacy Preserving Location Verification Scheme

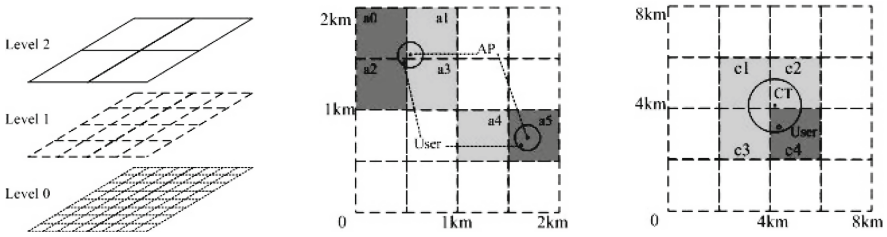
In this section, we present our approach for privacy-preserving location verification (PPLV) scheme. First, we give an overview of the proposed approach. Subsequently, we present the detailed work flow. Finally, we analysis the security and privacy. Figure 2 shows an overview of the approach and main processes.

4.1 Overview of PPLV

In our scheme, the users prefer to request location proof with WiFi AP; while there are no WiFi APs nearby, 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 query 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 [2] 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 Sect. 4.3.



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

Fig. 3. 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 km for cellular towers.

4.2 System Initialization

Global setup. The location of a user can be defined with different granularities. 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 Fig. 3(a), the system adopts a grid reference system [6] denoted by $\Gamma(l)(l = 0, 1, 2, \cdots)$ to represent locations. For each level l, the grid cell size (i.e. width and height) is fixed and equal. 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 $c \in \Gamma(l - 1)$.

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 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}). We assume the WiFi APs have the user's public key h.

4.3 System Process

Location Proof Request. The user periodically uses its WiFi module to scan the channels, hearing beacons from the nearby WiFi APs. 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. The location proof request can be denoted as:

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

Here, P_{user} denotes the user's pseudonym; n denotes the beacon's sequence number; l denotes the granularity of level; t denotes the request time. R_{user} is a set of cell ids that denotes 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

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

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

Assume that a user and a WiFi AP use granularity of level 1 in Fig. 3(b). The user specifies the region R_{user}, containing cells $\{a_0, a_2\}$, in which 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

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

and embeds it into the request.

Location Proof Issue. Upon receiving the location proof request, the WiFi AP firstly checks whether the number is a current one. We assume that the WiFi AP can accept requests whose sequence number was broadcasted within last 100 milliseconds. 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}[10]). 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

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

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

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

As show in Fig. 3(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

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

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

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

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 in the location proof. This step makes sure that the location proof 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 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.

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}. 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.

4.4 Security and Privacy Analysis

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. Thus, a malicious user can be detected immediately when he is cheating about his location.

Curious Database. In our scheme, the *Location Proof Server* has access only 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*[8], the user can be geo-located to an accurate estimated location. However in our scheme, we can also use different granularity levels to protect the user's location privacy.

Curious WiFi AP Network. Several WiFi APs could collude and track the location of a user based on the collected location proof requests. To thwart 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 pair of 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.

5 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.

5.1 Cost of Involved Entities

We conduct experiments on 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. In the experiment, we evaluate the efficiency of three involved entities under different sizes of prime p as shown in Table 1.

Table 1. Evaluation of the cost of involved three entities on smart phone with Exynos 4412 1.6 GHz CPU and computer with Intel i5 CPU of 2.5 GHz

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

Cost on User Side. The first metric is the cost of location proof request. The user needs to perform two exponentiations when generating location proof request. Note that, this process could be sped up considerably using pre-computations, which could further reduce the computation latency of location proof request.

Cost on WiFi AP Side. The second metric is the cost of location proof issue. 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.

Cost on CA Side. The last metric is the cost of location proof verification. The cost of location proof verification is to computer $\{u_1/u_0^x, \cdots\}$. Since computing division is much faster than computing exponentiation, the cost of location proof verification for each user is to compute a exponentiation.

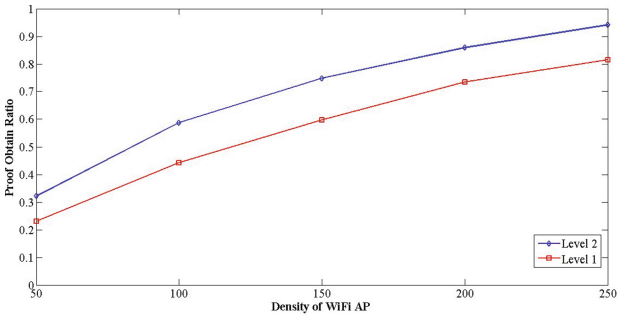


Fig. 4. Location proof obtain ratio under different density of WiFi AP.

5.2 Effectiveness of the Proposed Approach

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

Figure 4 shows the location proof obtain ratio under different granularities of level with different densities of WiFi AP. We can see that the location proof obtain 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.

6 Conclusion

In this paper, we identify a new kind of attack coined as location cheating attack in database-driven CRNs, which 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. We use a grid reference system and adopt the private proximity testing technology to further improve the user's location privacy. Simulations well demonstrate the effectiveness and efficiency of the proposed approach.

Acknowledgments. This work is supported by National Science Foundation of China (no. 61272444, U1401253, U1405251, 61411146001).

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. Narayanan, Arvind, et al. Location Privacy via Private Proximity Testing. NDSS. (2011)
3. Zhang, L., Fang, C., Li, Y., et al.: Optimal Strategies for Defending Location Inference Attack in Database-driven CRNs. ICC. IEEE (2015)
4. Capkun, S., Buttyan, L., Hubaux, J.-P.: SECTOR: secure tracking of node encounters in multi-hop wireless networks. In: Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks. ACM (2003)
5. Zhu, Z., Cao, G.: Applaus: a privacy-preserving location proof updating system for location-based services. INFOCOM. IEEE (2011)
6. Zheng, Y., Li, M., Lou, W., Hou, Y.T.: SHARP: private proximity test and secure handshake with cheat-proof location tags. In: Foresti, S., Yung, M., Martinelli, F. (eds.) ESORICS 2012. LNCS, vol. 7459, pp. 361–378. Springer, Heidelberg (2012)
7. Li, M., et al.: All your location are belong to us: breaking mobile social networks for automated user location tracking. MOBIHOC. ACM (2014)
8. Gao, Z., et al.: Location privacy in database-driven cognitive radio networks: attacks and countermeasures. INFOCOM. IEEE (2013)
9. Zeng, K., Ramesh, S.K., Yang, Y.: Location spoofing attack and its countermeasures in database-driven cognitive radio networks. In: 2014 IEEE Conference on Communications and Network Security (CNS). IEEE (2014)
10. Siksnys, L., et al.: Private and flexible proximity detection in mobile social networks. MDM. IEEE (2010)
11. He, W., Liu, X., Ren, M.: Location cheating: a security challenge to location-based social network services. ICDCS. IEEE (2011)
12. Pham, A., et al.: Secure and private proofs for location-based activity summaries in urban areas. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing. ACM (2014)