Research Article

Lightweight Privacy Preservation for Securing Large-Scale Database-Driven Cognitive Radio Networks with Location Verification

Rui Zhu ¹, Li Xu ¹, Yali Zeng,¹ and Xun Yi²

¹Fujian Provincial Key Laboratory of Network Security and Cryptology, College of Mathematics and Informatics, Fujian Normal University, China

²The School of Science, RMIT University, Australia

Correspondence should be addressed to Li Xu; xuli@fjnu.edu.cn

Received 21 February 2019; Accepted 17 April 2019; Published 6 May 2019

Guest Editor: Mingwu Zhang

Copyright © 2019 Rui Zhu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The database-driven cognitive radio networks (CRNs) are regarded as a promising approach to utilizing limited spectrum resources in large-scale Internet of Things (IoT). However, database-driven CRNs face some security and privacy threats. Firstly, secondary users (SUs) should send identity and location information to the database (DB) to obtain a list of available channels, such that the curious DB might easily misuse and threaten the privacy of SUs. Secondly, malicious SUs might send fake location information to the DB in order to occupy channels with better quantity in advance and so gain benefits. This might also cause serious interference to primary users (PUs). In this paper, we propose a lightweight privacy-preserving location verification protocol to protect the identity and location privacy of each SU and to verify the location of SUs. In the proposed protocol, the SU does not need to provide location information to request an available channel from the DB. Therefore, the DB cannot get the location information of any SU. In the proposed protocol, the base station (BS) selects some SUs as witnesses to generate location proofs for each other in a distributed fashion. This new witness selection mechanism makes the proposed protocol reliable when a malicious SU generates fake location information to cheat the BS and also prevents SU-Witness collusion attacks. The results also show that the proposed protocol can provide strong privacy preservation for SUs and can effectively verify the location of the SUs. The security analysis shows that the proposed protocol can resist various types of attacks. Moreover, compared with previous protocols, the proposed protocol is lightweight because it relies on symmetric cryptography and it is unaffected by the area covered by the DB.

1. Introduction

The explosive growth of wireless devices and services has exacerbated the spectrum scarcity problem in large-scale Internet of Things (IoT) [1]. Database-driven CRNs are regarded as a promising approach that allows the dynamic spectrum sharing in many large-scale IoT applications [2]. In cognitive radio networks (CRNs), primary users (PUs) have exclusive privilege to access the licensed channels, while secondary users (SUs) are allowed to access the licensed channels when the PUs are off-line or at a lower power without causing interference to the PUs. Through spectrum sharing between PUs and SUs, CRNs can effectively improve the spectrum utilization and alleviate the spectrum scarcity crisis.

CRNs have many potential applications including IoT [3], smart cities [4], and vehicular networks [5]. For example, a large number of connected devices will create a major challenge in terms of spectrum scarcity in large-scale IoT. Through dynamic spectrum access capabilities, large-scale IoT not only improves spectrum utilization but also exploits alternate spectrum opportunities. In addition, cognitive IoT is inherently equipped to address the challenges of interference management, energy efficiency, and device heterogeneity.

In CRNs, two main approaches can be used by SUs to obtain channel availability information [6]: the spectrum sensing approach and the geolocation database-driven approach. In the sensing approach [7], several SUs cooperatively sense the idle channels and report these channels to the

fusion center; the latter will then decide whether a channel is available prior to its use so as to avoid interfering with PUs. In the database-driven approach [7, 8], the database (DB) that is administered by some commercial entities (e.g., Microsoft, Google Inc., and Cellular South) stores the spectrum available information (SAI). SUs are required to send their locations to the DB in order to obtain channel availability information. After receiving a query, the DB returns a list of available channels and some transmission parameters to the SU at its location.

In 2012, database-driven CRNs were specified as the primary approach by the Federal Communications Commission (FCC) [8] because the geolocation database-driven approach has more advantages, especially in large-scale CRNs. On the one hand, it pushes the responsibility and complexity spectrum policies conformity from SUs to DBs. On the other hand, it only updates a handful of DB when policy changes, instead of updating large number of devices [9]. However, SUs have to send location information to the DB to obtain the list of available channels. Through this location information, the DB might reveal SUs' critical information such as habits, shopping preferences, behavior, and commuter routes [10]. Therefore, it is urgent to protect the location privacy of the SU during the spectrum query in CRNs. Some protocols based on private information retrieval (PIR) [11, 12] or k-anonymity [13, 14] have been proposed to protect the location privacy of SUs. However, these protocols might cause high computation or communication overheads or cannot provide enough safeguards for location privacy of SUs.

On the other hand, malicious SUs may falsify their locations when querying the DB for the list of available channels [15]. Because there is no location proof and verification when SUs query channels in database-driven CRNs, a malicious SU might send fake location information to the DB in order to occupy channels with better quantity in advance and gain benefits. Besides, if a malicious SU chooses channels that are not in his/her real location, it may cause interference to PUs. Only a few protocols based on private information retrieval [16] or cryptography [2] focus on the location verification: most existing protocols achieve location verification by employing a large number of access points (APs) to be witnesses. It might be too expensive and be inapplicable in some scenarios where no fixed wireless infrastructures are accessible [17]. Moreover, these protocols do not consider collusion between SUs and witnesses.

Noting all these concerns, we propose a lightweight privacy-preserving location verification protocol (LPPLV) for large-scale database-driven CRNs. The proposed protocol aims to guarantee the privacy of SUs to avoid inference by PUs during spectrum queries in database-driven CRNs. To prevent malicious SUs from reporting fake location information, SUs are required to provide location proofs for the DB indicating that they are at the places where they claim to be.

The contributions of this paper are summarized as follows.

- (1) We propose a secure and privacy-aware protocol that can protect the identity and location privacy of SUs. In our protocol, the DB has no knowledge about the

location information of SUs, and this can prevent the DB from obtaining the real identity and location information of SUs.

- (2) Our proposed protocol has a distributed architecture in which SUs generate location proofs for each other instead of employing expensive APs to provide the location proof for SUs in existing protocols. The new witness selection mechanism allows malicious SUs to be discovered by the BS when malicious SUs generate fake location information. Our new witness selection mechanism also reduces the possibility of SU-Witness collusion.
- (3) Security analysis is presented to prove that our protocol resists various types of attacks in database-driven CRNs, such as replaying attack, Man-in-the-Middle attack, and eavesdropping attack.
- (4) Channel allocation is implemented locally in our protocol, and we use the symmetric encryption operation in the communication of database-driven CRNs. Therefore, our protocol improves the efficiency of obtaining available channels for SUs. Experiments' results indicate the efficiency of our privacy-preserving location proof protocol.

The rest of this paper is organized as follows. We first discuss related work in Section 2. Section 3 presents the preliminaries. Then we introduce our proposed protocol in Section 4, which is followed by security and privacy analysis in Section 5. Section 6 analyzes the performance of the proposed protocol. Finally, Section 7 concludes the paper.

2. Related Work

Most of the existing protocols that are related to this work focus on privacy preservation of SUs while some protocols focus on location verification of SUs. In this section, we give a brief overview of these protocols.

2.1. Privacy Preservation Protocols. PIR [18], k-anonymity [19], and cuckoo filter [11] are three techniques that can be used to protect the privacy of SUs. Gao et al. [20] proposed a PIR-based protocol, in which SUs can obtain available spectrum information from the DB without sending geographic location information. They also pointed out that if the SUs switch channels frequently, the DB can locate the SUs according to the channel requesting messages, which can cause user privacy leakage. Troja et al. [12] proposed another PIR-based protocol that allows SUs to share information in a peer-to-peer manner. PIR technology can protect the privacy of SUs' locations well, but it also causes high communication and computational overheads. Grissa et al. [21] proposed a cuckoo filter-based protocol: the DB uses a cuckoo filter to compress the spectrum information and send it to the query server. The SUs then obtain the available channels through the query server. However, in this protocol, SUs query efficiency and spectrum accuracy are affected by cuckoo filters false positive and negative rates. Petrov et al. [13] proposed a privacy protection protocol based on k-anonymity. In this

protocol, SUs select $k-1$ volunteers to form a link forwarding query message, so that the DB cannot distinguish the true identity of the user who sent the query, achieving the purpose of privacy protection. However, this method creates high communication overheads, and when the number of volunteers is too small, SUs can easily expose their true identities and cannot achieve good privacy protection.

2.2. Location Verification Protocols. Another focus of this paper is the problem of location verification of SUs in database-driven CRNs. Numerous protocols have been proposed to achieve the location verification [17, 22–25]. SUs need to provide location proofs before obtaining the available channel in order to secure the communication in database-driven CRNs. Xin et al. [16] provide a PIR-based protocol for the server to verify whether the query SU is located where it claims to be. It uses WiFi access points (APs) to provide location proofs. In particular, if an AP can communicate directly with an SU, it considers this SU to be at the same location as itself. Then it generates a signature as the location proof for this SU. Li et al. [2] proposed a protocol based on the public key cryptographic algorithm. That protocol also relies on APs to provide location proofs. The two protocols have high computation and communication overheads. Moreover, it might be too expensive and is inapplicable to some scenarios in which no fixed wireless infrastructure is accessible.

In our previous work, we mainly focus on the privacy preservation and authentication of SUs. Zeng et al. [9, 26], respectively, utilize elliptic curve cryptography (ECC) and modular square root techniques to achieve privacy preservation and authentication of SUs in database-driven CRN. However, we did not take into account the problem of location verification of SUs.

From the above discussion, we can see that these protocols either incur high communication and computation overheads or rely on existing WiFi AP networks for reporting location-based activity summaries, which means that these existing protocols cannot provide a satisfactory privacy preservation and location proof protocol for SUs in database-driven CRNs.

3. Preliminaries

3.1. System Model. Our network model is shown in Figure 1, which involves four types of entities: the database (DB), multiple based stations (BSs), numerous secondary users (SUs), and the trust authority (TA). Their functions and interactions are described as follows.

DB. The DB is an entity which stores the spectrum available information and makes this information available to SUs.

BSs. In our network model, BSs provide location verification and channel allocation in our system. Each BS requests some channels from the DB. If an SU wants to obtain available channels, it should send a request including its location information to the BS. The BS first verifies its location proof, i.e., this SU is where it claims to be. Then, the BS calculates and allocates a channel based on the SU's location.

SUs. The SUs are those who do not have a licensed spectrum in CRNs. There are three types of SUs: one who wants to obtain an available channel, one who is selected to be a witness to provide location proofs for the SU, and one who is nonselected to be witness. Multiple SUs will be selected as witnesses after an SU sends a request to the BS. Meanwhile, that SU does not want to disclose its location. Besides, SUs and witnesses send their messages to each other through their predefined short-range communication interfaces (e.g., Bluetooth and WiFi).

TA. The TA can be served by federal technical centers (e.g., FCC and NTIA). In this paper, each BS and SU are required to be registered in the TA. The TA stores the reputation of each SU and updates it for a period of time.

A typical process of our protocol can be described as follows. BSs first obtain the list of available channels from the DB. An SU should ask the BS to obtain a channel with its current location. Then the BS chooses some SUs who are using the channels and are near the SU as witnesses. After obtaining the location proofs for the SU from witnesses, the BS verifies the location proof. If it is valid, the BS chooses a channel and returns the channel to the SU.

3.2. Security and Privacy Requirements. The TA in our system is a fully trusted entity which has the knowledge of the mappings between the pseudo-ID and the real identity of each SU. We also assume that BSs are supposed to be trusted entities. Neither the TA nor the BS publishes SUs' identity and data. However, the DB is honest but curious, which means the DB will honestly execute the protocol but is very curious about the identity and location information of the SUs. Hence, our first goal is to prevent the DB from obtaining any SU's identity and location information. Meanwhile, malicious SU might provide witnesses with fake information about his/her location or change contents of a location proof generated for him/her or another SU. They might also collude with witnesses to cheat the BS and obtain the available channels. It is also assumed that both the DB and BSs will not modify the communication data between them and legitimate SUs. No entity ever shares its private keys with another.

An efficient privacy-preserving location verification protocol for the large-scale database-driven CRNs should have the following desirable properties.

Privacy preservation: The privacy we want to preserve is the identity and location information of SUs. That is, except the TA, other entities including the DB, the BSs, and other SUs will never get the knowledge of the SU's real identity. On the other hand, our protocol should prevent the DB from obtaining the location information from the channel query of the SUs, even if the DB colludes with all BSs and other SUs.

Location verification: Malicious SUs might send fake location information to the DB and obtain the channels that are not available in their real locations, so as to infer operational patterns of PUs and other SUs. Therefore, the proposed protocol should verify the SUs' current locations and prevent the SUs from obtaining any available channels at other locations.

FIGURE 1: Network model.

Attack resistance: There are various types of attacks in CRNs, such as replaying attack, Man-in-the-Middle attack, and eavesdropping attack. The security of our protocol will not be compromised under these attacks.

SU-Witness collusion: Malicious SU might collude with malicious witnesses by creating a location proof for him/her even though the SU is not at the location claimed, in order to deceive the BS and the DB. We call this collusion SU-Witness collusion. To the best of our knowledge, there is no protocol to consider SU-Witness collusion in CRNs.

4. The Proposed Protocol

In this section we describe the details of the proposed protocol. It consists of seven phases: system initialization, channel preallocation, channel query, location proof generation, location verification, channel allocation, and reputation update. Notations used in the protocol are defined in Table 1.

4.1. System Initialization Phase. We assume that a network is composed of N_1 BSs and N_2 SUs. Each BS and SU send their real identities to the TA in a secure channel for registration.

For each BS_i ($i \in N_1$) whose real identity is ID_{BS_i} , the TA generates the secret key SK_{BS_i} and public key PK_{BS_i} . The TA also generates a unique session key k_{DB,BS_i} between the DB and the BS_i . Then TA sends SK_{BS_i} and k_{DB,BS_i} to BS_i in a secure channel and publishes PK_{BS_i} . The TA sends k_{DB,BS_i} to the DB. Upon receiving SK_{BS_i} and k_{DB,BS_i} , BS_i stores these keys securely.

For each SU_j ($j \in N_2$) whose real identity is ID_{SU_j} , the TA chooses X distinct pseudo-IDs (PIDs) $PID_{SU_{j,1}}, PID_{SU_{j,2}}, \dots, PID_{SU_{j,X}}$ and sends them to SU_j . The TA initializes the reputation value $Rep_{SU_j} = 0$. Upon receiving PIDs, SU_j stores the PIDs securely and uses one PID for each request. After all PIDs are allocated, SU_j asks for another X PIDs from the TA.

At the outset of a new system, there are not enough SUs in the network that can be selected as witnesses. At this time, the BS presets some trusted fixed devices (e.g., the roadside units and cellular tower) distributed in its coverage area. However, it might be too expensive to employ the trusted fixed devices all the time. We will reduce the employed fixed devices when the number of SUs in the network increases to a more practical number.

4.2. Channel Preallocation Phase. After registration, BS_i sends a request to the DB for enabling channel preallocation. This phase includes the following steps.

(1) BS_i sends the following message m_1 to the DB for enabling channel preallocation.

$$m_1 = ID_{BS_i} \parallel t_1 \parallel E_{k_{DB,BS_i}} (ID_{BS_i} \parallel Loc_{BS_i} \parallel t_1) \parallel MAC_{k_{DB,BS_i}} (m'_1), \quad (1)$$

where $m'_1 = (ID_{BS_i} \parallel Loc_{BS_i} \parallel t_1)$, Loc_{BS_i} is the location of BS_i , and $MAC_{k_{DB,BS_i}} (m'_1)$ is message authentication code of m'_1 using the key k_{DB,BS_i} .

(2) After receiving the message from BS_i , the DB decrypts the message with the key k_{DB,BS_i} that matches BS_i and checks whether t_1 is a valid time stamp. Then the DB computes $MAC_{k_{DB,BS_i}} (m'_1)$ and verifies that it is the same as $MAC_{k_{DB,BS_i}} (m'_1)$ received through m_1 . If one or both of the verifications do not hold, the DB drops this message. Otherwise, BS_i is authenticated by the DB. The DB chooses a channel list Ch_{BS_i} based on Loc_{BS_i} and sends the message m_2 to the BS_i using a secure channel.

$$m_2 = ID_{BS_i} \parallel t_2 \parallel E_{k_{DB,BS_i}} (Ch_{BS_i} \parallel t_2) \parallel MAC_{k_{DB,BS_i}} (m'_2), \quad (2)$$

where $m'_2 = (Ch_{BS_i} \parallel t_2)$.

TABLE 1: Notations in the protocol.

Notation	Description
$\parallel$	Concatenation symbol
ID_U	Identity of entity U
PID_U	The pseudo-ID of entity U
N_1	The number of BSs
N_2	The number of SUs
PK_U	The public key of entity U
SK_U	The secret key of entity U
$E_{K_U}(m)$	Encryption of message m using key of entity U
$MAC_{K_U}(m)$	Message authentication code of message m using key of entity U
Rep_U	The reputation value of entity U
Loc_U	The location of entity U
t_i	The time stamp
CL_i	A channel list
Ch_i	An available channel
k_{U_1,U_2}	The session key between entity U_1 and U_2
r_i	A random number
$H()$	A secure hash function

(3) Upon receiving the message m_2 from the DB, BS_i checks whether the ID_{BS_i} is its identity. Then BS_i uses its secret key SK_{BS_i} to decrypt the message m_2 and verifies $MAC_{k_{DB,BS_i}}(m'_2)$. If these verifications do not hold, BS_i drops this message. Otherwise, BS_i obtains the channel list CL_{BS_i} . Then the BS_i stores the channel list securely.

4.3. Channel Query Phase. Before an SU accesses the channel, the SU sends a request to the BS and informs the BS that he/she wants to get the location verification. This phase includes the following steps.

(1) SU_j randomly selects a PID from $\{PID_{SU_{j,1}}, PID_{SU_{j,2}}, \dots, PID_{SU_{j,x}}\}$, generates a unique session key k_{SU_j,BS_i} between himself/herself and the BS_i , and sends a location verification request m_3 to BS_i .

$$m_3 = E_{PK_{BS_i}}(PID_{SU_j} \parallel Loc_{SU_j} \parallel k_{SU_j,BS_i} \parallel t_3) \parallel MAC_{k_{SU_j,BS_i}}(m'_3), \quad (3)$$

where $m'_3 = (PID_{SU_j} \parallel Loc_{SU_j} \parallel k_{SU_j,BS_i} \parallel t_3)$.

(2) Upon receiving the message m_3 from SU_j , BS_i uses its secret key SK_{BS_i} to decrypt the message m_3 . BS_i first checks whether the location of SU_j has the available channels. If not, BS_i responds with the following message m_4 to reject the request. Otherwise, BS_i executes the location proof generation steps described in Section 4.4.

$$m_4 = E_{k_{SU_j,BS_i}}(PID_{SU_j} \parallel Loc_{SU_j} \parallel t_4 \parallel null) \parallel MAC_{k_{SU_j,BS_i}}(m'_4), \quad (4)$$

where $m'_4 = (PID_{SU_j} \parallel Loc_{SU_j} \parallel t_4 \parallel null)$.

4.4. Location Proof Generation Phase. Since each SU has to request location verification and available channels from BS_i ,

BS_i stores PIDs, locations of the SUs who have requested the channels, and the unique session keys k_{W_j,BS_i} between these SUs and BS_i . k_{W_j,BS_i} is the key generated by these SUs in their channel query phase, which is similar to k_{SU_j,BS_i} . This phase includes the following steps.

(1) Upon receiving the message m_3 , BS_i searches for K SUs who are using the channel and are near SU_j . Then, BS_i sends a list of the PIDs of these SUs and PID_{SU_j} to the TA. After receiving the list and PID_{SU_j} from BS_i , the TA first searches for the real identity of SU_j and verifies whether SU_j has registered with it. Then the TA responds with the reputation values of these SUs to BS_i according to the PIDs. If SU_j has not registered in the TA, the TA rejects BS_i .

(2) We assume Rep_T is the threshold of reputation value. If $Rep_{SU_j} < Rep_T$, that means SU_j has a high probability of being dishonest. BS_i sorts K SUs according to their reputation values and checks the top N ($N \leq K$) SUs' reputation values. We assume N_H is the threshold number of location proofs that BS_i will receive. If the number of dishonest SUs is more than threshold N_H , BS_i will remove some SUs who have lower reputation values. BS_i then chooses some preset trusted fixed devices as a supplement to ensure that the number of dishonest selected witnesses is less than N_H . At the beginning of the system, there are not enough SUs in the network that can be selected as witnesses. At this time, BS_i selects some preset trusted fixed devices as witnesses.

After selecting N witnesses, BS_i generates a unique index (IND_P) for this location proof, numbers selected witnesses to n_i , and generates the session keys k_{W_k,SU_j} ($k \in N$) for his/her communication process. Then BS_i sends the following message m_5 to SU_j and sends the message m_6 to each witness.

$$m_5 = E_{k_{SU_j,BS_i}}(IND_P \parallel k_{W_1,SU_j} \parallel k_{W_2,SU_j} \parallel \dots \parallel k_{W_N,SU_j} \parallel t_5) \parallel MAC_{k_{SU_j,BS_i}}(m'_5), \quad (5)$$

$$m_6 = E_{k_{W_k,BS_i}}(IND_P \parallel n_i \parallel k_{W_k,SU_j} \parallel t_6) \parallel MAC_{k_{W_k,BS_i}}(m'_6), \quad (6)$$

where $m'_5 = (IND_P \parallel k_{W_1,SU_j} \parallel k_{W_2,SU_j} \parallel \dots \parallel k_{W_N,SU_j} \parallel t_5)$ and $m'_6 = (IND_P \parallel n_i \parallel k_{W_k,SU_j} \parallel t_6)$. BS_i generates a table for this location proof and verification, as shown in Table 2.

(3) After receiving the message m_6 , a selected witness generates a random number r_i and broadcasts the following message m_7 through its predefined short-range communication interface for a period of time T .

$$m_7 = n_i \parallel IND_P \parallel t_7 \parallel E_{k_{W_k,SU_j}}(n_i \parallel IND_P \parallel r_i \parallel t_7) \parallel MAC_{k_{W_k,SU_j}}(m'_7), \quad (7)$$

where $m'_7 = (n_i \parallel IND_P \parallel r_i \parallel t_7)$. Another r_i is generated and broadcast in a similar way. This process is repeated until the witness receives a response from SU_j .

(4) SU_j first ensures that the IND_P is the same as the one already received from the BS when the SU receives the message m_7 . Then, SU_j checks whether t_7 is fresh and verifies $MAC_{k_{W_k,SU_j}}(m'_7)$. If these verifications are not sustained, SU_j drops this message and waits for the next message. Otherwise, SU_j decrypts the message with the key k_{W_k,SU_j} that matches n_i , and verifies whether or not the message has been tampered with. If the message cannot be decrypted by the corresponding key, SU_j discards this message. If all the verifications are successful, SU_j must immediately send message m_8 to W_k :

$$m_8 = IND_P \parallel t_8 \parallel E_{k_{W_k,SU_j}}(IND_P \parallel r_i \parallel t_8 \parallel H(PID_{SU_i} \parallel r_i)) \parallel MAC_{k_{W_k,SU_j}}(m'_8), \quad (8)$$

where $m'_8 = (IND_P \parallel r_i \parallel t_8 \parallel H(PID_{SU_i} \parallel r_i))$.

(5) Upon receiving m_8 , W_k first checks whether IND_P in m_8 is the same as the IND in m_6 . Then, W_k verifies $MAC_{k_{W_k,SU_j}}(m'_8)$ and checks whether r_i in m_8 is the last random number that had been broadcast by itself. If so, W_k generates the following location proof LP and sends it to BS_i .

$$LP = n_i \parallel IND_P \parallel t_9 \parallel E_{k_{W_k,BS_i}}(IND_P \parallel PID_{W_k} \parallel Loc_{W_k} \parallel r_i \parallel t_9 \parallel H(PID_{SU_j} \parallel r_i)) \parallel MAC_{k_{W_k,BS_i}}(LP'), \quad (9)$$

where $LP' = (IND_P \parallel PID_{W_k} \parallel Loc_{W_k} \parallel r_i \parallel t_9 \parallel H(PID_{SU_j} \parallel r_i))$. Otherwise, the following *null* LP is sent to BS_i :

$$nLP = n_i \parallel IND_P \parallel t_{10} \parallel E_{k_{W_k,BS_i}}(IND_P \parallel PID_{W_k} \parallel null \parallel t_{10}) \parallel MAC_{k_{W_k,BS_i}}(nLP'), \quad (10)$$

where $nLP' = (IND_P \parallel PID_{W_k} \parallel null \parallel t_{10})$.

4.5. Location Verification Phase. For each *non – null* LP that has been received before the time threshold, BS_i checks *non – null* LP s as follows.

(1) Checking whether IND_P is the unique IND for this location proof, and finding out PID_{SU_j} in the message m_3 according to IND_P .

(2) Checking whether the time stamp t_9 is fresh.

(3) According to n_i , trying to find the unique session key k_{W_k,BS_i} , and using k_{W_k,BS_i} to decrypt LP . Then, BS_i checks whether PID_{W_k} is the selected witness.

(4) Computing $MAC_{k_{W_k,BS_i}}(LP')$ and verifying it is the same as $MAC_{k_{W_k,BS_i}}(LP')$ received through LP .

(5) Verifying whether Loc_{SU_j} is in an acceptable range of Loc_{W_k} .

(6) Computing $H(PID_{SU_j} \parallel r_i)$ and verifying it is the same as $H(PID_{SU_j} \parallel r_i)$ received through LP .

(7) Checking whether the number of *non – null* LP s is greater than a predefined threshold N_H .

If all the above checks are successful, BS_i accepts SU_j , which means that SU_j is legitimate. Otherwise, SU_j 's request is rejected.

4.6. Channel Allocation Phase. If the above location verification is successful, which means SU_j is where he/she claims to be, BS_i chooses a channel from the channel list, which is preallocated from the DB. One crucial step for an SU to protect location privacy and to get a better service quality, is to choose the most stable channel. The available channels in the channel list can be divided into four cases and are analyzed below.

(1) All PUs operating on these channels are on-line but close to SU_j . The BS should choose the channel with the higher power for the SU, so that the SU can obtain the best quality of service.

(2) Some PUs in this case might be off-line. BS_i chooses the channel with the highest power, but not the maximum power, and long usage time for SU_j . Although in this case, SU_j can obtain the channels where the PUs are off-line and get the best service quality, once the PU goes on-line, the DB will force SU_j to go off-line and switch channels. Therefore, SU_j may be caught in frequent channel switching.

(3) All PUs operating on these channels are on-line but far away from SU_j . This kind of channel is preferred to be used, so BS_i chooses this channel for SU_j . When SU_j accesses such channel, SU_j can work at the maximum power regardless of whether the PU is on-line or off-line, so that SU_j can use the channel with the best quality of service and avoid the privacy leaking that can be caused by channel switching.

(4) Some PUs might be off-line while other PUs are on-line but might be far away from SU_j . BS_i should choose the channel with the highest (but not the maximum) power and long usage time for SU_j , ensuring that SU_j can get the best quality channel.

As analyzed above, the power and the channel usage time are determined by the channel quality of service. BS_i should choose the channel with the best quality of service for SU_j .

TABLE 2: The information of this location proof and verification.

IND	PID _{SU}	k _{SU,BS}	PID _W	k _{W,BS}	k _{W,SU}	t
IND _P	PID _{SU_j}	k _{SU_j,BS_i}	PID _{W₁} ... PID _{W_N}	k _{W₁,BS_i} .. k _{W_N,BS_i}	k _{W₁,SU_j} ... k _{W_N,SU_j}	t ₅

according to the power and the channel usage time. After choosing the best channel, BS_i sends the message *Quc* to SU_j.

$$Quc = E_{k_{SU_j,BS_i}}(Ch_i || t_{11}) || MAC_{k_{SU_j,BS_i}}(Quc'), \quad (11)$$

where $Quc' = (Ch_i || t_{11})$. When multiple SUs need the same channel at the same time, the BS preferentially allocates this channel to the SU with a high reputation value.

When SU_j plans to go off-line, SU_j sends the message *Bye* to BS_i.

$$Bye = E_{k_{SU_j,BS_i}}(0 || t_{12} || PID_{SU_j}) || MAC_{k_{SU_j,BS_i}}(Bye'), \quad (12)$$

where $Bye' = (0 || t_{12} || PID_{SU_j})$. After receiving the message *Bye*, the BS reports to the TA, and the TA deletes PID_{SU_j} that SU_j has used.

4.7. Reputation Update Phase. As each SU sends his/her real identity to the TA for registration, the TA stores the mappings between the pseudo-IDs and the real identity of each SU. After receiving messages from each BS, the TA constructs a table to record the location verification results of each SU, as shown in Table 3. In Table 3, R_{SU_j} indicates whether BS_i accepts SU_j; 0/1 means BS_i rejects/accepts SU_j. In addition, WR_{SU_j} indicates whether SU_j provides an effective location proof as a selected witness; 0/1 means SU_j has provided an effective/invalid location proof as a selected witness.

From this table, the TA can trace each SU and check the honesty of each SU, for instance, whether an SU is honest in claiming his/her location or has provided effective location proof for the surrounding SUs as witness.

After tracing each SU, the TA updates the reputation value Rep_{SU_j} of SU_j in Table 4 as

$$Rep_{SU_j} = \frac{C_{R_{SU_j}=1} + WR_{SU_j=1}}{C_{SU_j}}, \quad (13)$$

where $R_{SU_j=1}$ means R_{SU_j} 's value that is equal to 1 and $WR_{SU_j=1}$ means WR_{SU_j} 's value that is equal to 1 in Table 3. And $C_{R_{SU_j}=1} + WR_{SU_j=1}$ is the sum of the numbers of $R_{SU_j=1}$ and $WR_{SU_j=1}$, and C_{SU_j} is the sum of the numbers of R_{SU_j} and WR_{SU_j} . There are three cases.

(1) $Rep_{SU_j} = (C_{R_{SU_j}=1} + WR_{SU_j=1} + 1) / (C_{SU_j} + 1)$: SU_j has honestly claimed his/her location or SU_j has provided the effective location proof for the surrounding SUs as witness.

(2) $Rep_{SU_j} = (C_{R_{SU_j}=1} + WR_{SU_j=1} - 1) / (C_{SU_j} + 1)$: SU_j has been rejected by the BS or SU_j has not provided the effective location proof for the surrounding SUs as a witness.

(3) $Rep_{SU_j} = Rep_{SU_j}$: During this update period, SU_j has not sent the channel request to the BS or SU_j has not been selected as a witness.

5. Security Analysis

In this section, we analyze the security of the proposed protocol with respect to the security requirements given in Section 3.2.

Privacy preservation: In our protocol, the SUs obtain pseudo-IDs in the system initialization phase. The SUs use these pseudo-IDs instead of real identities at the channel query phase. Except for the TA, anyone (including the DB, the BSs, and other SUs) cannot obtain the SUs' real identities. However, in our protocol, the SUs do not provide their location information for the DB for a channel request, so that the DB has no knowledge about the location information of SUs. Further, since there are only pseudo-IDs stored in each BS, the BS can only obtain the current location of SUs. The pseudo-IDs will change at the next channel request. Therefore, the BS cannot obtain the trajectory of SUs. Even if all BSs have been attacked, the attacker cannot map between the real identity and the location information of SUs. That is, our protocol can prevent anyone from discovering the real identity and location information of SUs.

Location verification: Each SU who wants to obtain the available channels must be verified by the BS. The BS selects some surrounding SUs to be the witnesses to verify the SU's location. The location proofs generated by the witness are unforgeable. We consider several scenarios. If a malicious SU wants to generate a location proof by himself, the BS will detect this: the BS receives location proofs from selected witnesses. The BS checks the received LPs and decrypts the LPs by the unique session key $k_{W_k,BS}$. Since any entity does not share his/her private key, this SU cannot generate a LP even if he knows the identity of each selected witness. Malicious SUs who try to send fake location information to the BS or to generate a location proof by himself will be rejected by BS.

Attack resistance: Since all messages have been encrypted and protected by secret keys, even though an attacker can capture all messages transmitted between each entity, it cannot acquire the content of messages. We use a message authentication code to prevent messages from being tampered with by attackers. Anyone who wants to change the information in these messages will first need to be verified, and fake messages will be removed. Meanwhile, we use time stamps to prevent the attack from being replayed, so such an attack is infeasible in our protocol.

Resistance to SU-Witness collusion: At the location verification phase, the BS checks all LPs and rejects any LP which is not generated by selected witnesses. Thus, a malicious

TABLE 3: The information of each SU in the TA.

ID_{SU}	PID_{SU}	R_{SU_i}	WR_{SU_i}	Record time
ID_{SU_1}	$PID_{SU_{1,1}}, \dots, PID_{SU_{1,x}}$	$1 \parallel 1 \parallel \dots$	$1 \parallel 1 \parallel \dots$	t_1
ID_{SU_2}	$PID_{SU_{2,1}}, \dots, PID_{SU_{2,x}}$	$1 \parallel 0 \parallel \dots$	$1 \parallel 1 \parallel \dots$	t_2
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$

TABLE 4: The reputation value of each SU.

ID_{SU}	PID_{SU}	Reputation	Update time
ID_{SU_1}	$PID_{SU_{1,1}}, \dots, PID_{SU_{1,x}}$	Rep_{SU_1}	t_1
ID_{SU_2}	$PID_{SU_{2,1}}, \dots, PID_{SU_{2,x}}$	Rep_{SU_2}	t_1
$\dots$	$\dots$	$\dots$	$\dots$

SU cannot generate a LP for himself or any other users. This makes it very difficult for a malicious SU to set up successful SU-Witness collusion. Each selected witness has a high reputation value, which means they have a high probability of being an honest SU and will not collude with a malicious SU. Reputation values motivate them to generate efficient LP s since reputation values will affect the service quality of the channel they receive from the BS. However, we cannot altogether exclude the possibility of SU-Witness collusion. Any malicious SU needs to increase the size of his/her collusion group to improve his/her chances of success. In other words, the number of non-null LP s should be greater than a predefined threshold N_T . It means unless an SU colludes with at least N_T selected witnesses, it cannot succeed. However, in our protocol the BS selects witnesses: this selection mechanism ensures that the number of dishonest witnesses is less than N_T . Therefore, the possibility of SU-Witness collusion is minuscule in the proposed protocol.

Comparison with other protocols: We compare the security of the proposed protocol with protocols discussed in Section 2. We also give an overview of the related literature and summarize the comparison between these protocols and our protocol in Table 5. Table 5 shows that our protocol is more secure than other protocols.

6. Performance Evaluation

In this section, we evaluate the performance of our proposed protocol, by showing its computation and communication costs, and compare them with the protocols proposed by Grissa et al. [21] and Gao et al. [20]. All experiments have been conducted on a 64-bit computer with an Intel Core i7 CPU of 2.5 GHz and 16G memory. At the channel preallocation phase, a BS encrypted m_1 with AES with 256-bit key size (AES-256) and the TA also encrypts m_2 with AEC-256. At the channel query phase, the channel request message from an SU to a BS is also encrypted with ECC-256. In other phases, the messages transmitted between a BS and an SU, an SU and witnesses, and witnesses and a BS are encrypted with AES with 256-bit key size.

It is assumed that the network is divided into $m \times m$ cells; the output of hash function is 160 bits; ID_{BS_i}, PID_{SU_j} ,

PID_{W_k} , $timestamp$, and channel list are 32 bits, respectively; a random number r_i is 128 bits; and Loc_{BS_i} , Loc_{SU_j} , and Loc_{W_k} are 40 bits, respectively. Further, let T_{EC} , T_{DC} , T_{EA} , T_{DA} , T_H , and T_{MAC} denote the running time for one ECC-256 encryption operation, one ECC-256 decryption operation, one AES-256 encryption operation, one AES-256 decryption operation, one hash function operation (SHA1 on 512-bit block), and one message authentication code operation (SHA1 on 512-bit block), respectively. The running time of T_{EC} , T_{DC} , T_{EA} , T_{DA} , T_H , and T_{MAC} is 3.124ms, 2.926ms, 0.006ms, 0.006ms, 0.001ms, and 0.003ms, respectively.

6.1. The Performance Analysis of Our Protocol. In this subsection, we analyze the computation and communication cost of channel preallocation phase (CPAP), channel query phase (CQP), location proof generation phase (LPGP), location verification phase (LVP), and channel allocation phase (CAP) in our protocol in Table 6.

6.1.1. Computation Cost. We analyze the computation cost of each phase.

(1) *CPAP:* At the channel preallocation phase, each BS encrypts m'_1 and sends m_1 to the DB. Each BS executes one AES encryption operation and one message authentication code operation. To allocate the channel list to the BS, the DB needs to execute one AES encryption operation to encrypt m'_2 . After receiving m_2 , each BS also needs to execute one AES decryption operation to decrypt m_2 and obtains the channel list. In this phase, the computation cost of the BS is $T_{EA} + T_{DA} + 2T_{MAC}$ and the computation cost of the DB is $T_{EA} + T_{DA} + 2T_{MAC}$.

(2) *CQP:* At the channel query phase, the computation cost of the SU is one elliptic curve encryption operation and one message authentication code operation, which is $T_{EC} + T_{MAC}$. The computation cost of the BS is one elliptic curve decryption operation and one message authentication code operation, which is $T_{DC} + T_{MAC}$.

(3) *LPGP:* At the location proof generation phase, the computation cost of each BS is $N + 1$ AES encryption operations and $N + 1$ message authentication code operations, which is $(N + 1) \cdot (T_{EA} + T_{MAC})$. The computation cost of

TABLE 5: Comparison of existing privacy-preserving and location verification protocols in CRNs.

Protocol	Technique	Privacy protection	Location verification	Collusion resistant
Gao et al. [20]	PIR	√	×	×
Troja et al. [12]	PIR	√	×	×
Grissa et al. [21]	Cuckoo Filter	√	×	×
Petrov et al. [13]	k-anonymity	√	×	×
Xin et al. [16]	PIR and APs-based	√	√	×
Li et al. [4]	Cryptography and APs-based	√	√	×
Our protocol	Cryptography and distributed	√	√	√

TABLE 6: Computation and communication costs of our protocol.

Phase	Entity	Computation cost	Communication cost (bits)
CPAP	DB	$T_{EA} + T_{DA} + 2T_{MAC}$	288
	BS	$T_{EA} + T_{DA} + 2T_{MAC}$	328
CQP	BS	$T_{DC} + T_{MAC}$	0
	SU	$T_{EC} + T_{MAC}$	1440
LPGP	TA	0	$(K + 1) \cdot 34$
	BS	$(N + 1) \cdot (T_{EA} + T_{MAC})$	$(K + 1) \cdot 32 + 738N + 224$
	SU	$N \cdot T_{EA} + (N + 1) \cdot T_{DA}$ $+ (2N + 1) \cdot T_{MAC} + N \cdot T_H$	608N
	Witnesses	$(n + 1) \cdot N \cdot T_{EA} + 2N \cdot T_{DA} + (n + 3) \cdot N \cdot T_{MAC}$	$452 \cdot n \cdot N + 682Nbits$
LVP	BS	$N \cdot (T_{DA} + T_{MAC} + T_H)$	0
CAP	BS	$T_{EA} + T_{MAC}$	224
	SU	$T_{DA} + T_{MAC}$	0

each SU consists of N AES encryption operations, $N + 1$ AES decryption operations, $2N + 1$ message authentication code operations, and N hash function operations, which is $N \cdot T_{EA} + (N + 1) \cdot T_{DA} + (2N + 1) \cdot T_{MAC} + N \cdot T_H$ (N denotes the number of selected witnesses). Similarly, the computation cost of witnesses consists of $(n + 1) \cdot N$ AES encryption operations (n denotes the number of broadcasted rounds), $2N$ AES decryption operations, and $(n + 3) \cdot N$ message authentication code operations, which is $(n + 1) \cdot N \cdot T_{EA} + 2N \cdot T_{DA} + (n + 3) \cdot N \cdot T_{MAC}$.

(4) *LVP and CAP*: In the location verification phase and the channel allocation phase, the computation cost of each BS is N AES decryption operations, $N + 1$ message authentication code operations, N hash function operations, and one AES encryption operation, which is $N \cdot (T_{DA} + T_H) + (N + 1) \cdot T_{MAC} + T_{EA}$. The SU needs one AES decryption operation and one message authentication code operation, which is $T_{DA} + T_{MAC}$.

6.1.2. Communication Cost. We analyze the communication cost of each phase.

(1) *CPAP*: At the channel preallocation phase, each BS sends the message $m_1 = ID_{BS_i} \parallel t_1 \parallel E_{k_{DB,BS_i}}(ID_{BS_i} \parallel Loc_{BS_i} \parallel t_1) \parallel MAC_{k_{DB,BS_i}}(m'_1)$ to the DB, and the DB chooses a channel list and sends the message $m_2 = ID_{BS_i} \parallel t_2 \parallel E_{k_{DB,BS_i}}(Ch_{BS_i} \parallel t_2) \parallel MAC_{k_{DB,BS_i}}(m'_2)$ to the BS. Hence, the communication cost of the BS is $(32 + 32 + 32 + 40 + 32 + 160) = 328$ bits, and the communication cost of the DB is $(32 + 32 + 32 + 32 + 160) = 288$ bits.

(2) *CQP*: At the channel query phase, each SU sends the channel request message $m_3 = E_{PK_{BS_i}}(PID_{SU_j} \parallel Loc_{SU_j} \parallel k_{SU_j,BS_i} \parallel t_3)$ to the BS. Therefore, the communication cost is $(32 + 40 + 256 + 32) \times 4 = 1440$ bits.

(3) *LPGP*: At the location proof generation phase, the TA responds to the reputation value of the K SUs surrounding the SU, so that the communication cost is $(K + 1) \cdot 34$. The BS sends the reputation value request message to the TA, $m_5 = E_{k_{SU_j,BS_i}}(IND_P \parallel k_{W_1,SU_j} \parallel k_{W_2,SU_j} \parallel \dots \parallel k_{W_N,SU_j} \parallel t_5) \parallel MAC_{k_{SU_j,BS_i}}(m'_5)$ to the SU, and N $m_6 = E_{k_{W_k,BS_i}}(IND_P \parallel n_i \parallel k_{W_k,SU_j} \parallel t_6) \parallel MAC_{k_{W_k,BS_i}}(m'_6)$ to each witness, respectively. The communication cost of the BS is $(K + 1) \times 32 + 32 + 256N + 32 + 160 + (32 + 32 + 2 + 256 + 160) \times N = (K + 1) \cdot 32 + 738N + 224$ bits. The SU responds to the message $m_8 = IND_P \parallel t_8 \parallel E_{k_{W_k,SU_j}}(IND_P \parallel r_i \parallel t_8 \parallel H(PID_{SU_i} \parallel r_i)) \parallel MAC_{k_{W_k,SU_j}}(m'_8)$ to each witness, so that the communication cost is $N \times (32 + 32 + 32 + 160 + 32 + 160 + 160) = 608Nbits$. Each witness broadcasts n rounds of the message $m_7 = n_i \parallel IND_P \parallel t_7 \parallel E_{k_{W_k,SU_j}}(n_i \parallel IND_P \parallel r_i \parallel t_7) \parallel MAC_{k_{W_k,SU_j}}(m'_7)$ and sends $LP = n_i \parallel IND_P \parallel t_9 \parallel E_{k_{W_k,BS_i}}(IND_P \parallel PID_{W_k} \parallel Loc_{W_k} \parallel r_i \parallel t_9 \parallel H(PID_{SU_j} \parallel r_i)) \parallel MAC_{k_{W_k,BS_i}}(LP')$ to the BS, respectively. Hence, the communication cost of witnesses is $(n \times N \times (2 + 32 + 32 + 32 + 32 + 2 + 160 + 160)) + N \times (2 + 32 + 32 + 32 + 40 + 160 + 32 + 160 + 160) = 452 \cdot n \cdot N + 682N$ bits.

(4) *LVP and CAP*: At the location verification phase and the channel allocation phase, the BS chooses a channel

FIGURE 2: Computation cost and communication cost of our protocol.

and sends the message $Quc = E_{k_{SU_j, BS_i}}(Ch_i \parallel t_{11}) \parallel MAC_{k_{SU_j, BS_i}}(Quc')$ to the SU, so that the communication cost is $(32 + 32 + 160) = 224$ bits.

Moreover, the computation and communication cost of the different number of witnesses is shown in Figure 2 using the expressions established in Table 6, wherein K is set to be 20 and n is set to be 10. In order to show our communication and computational costs clearly, we use a logarithmic way of drawing. As shown in Figure 2(a), when the number of witnesses is 3, the running time of the BS and that of the DB at the channel preallocation phase are less than 1 ms; the running time of an SU obtaining the location proof at the location proof generation phase is less than 0.4 ms, and the running time of a BS to verify the location and to allocate a channel to an SU at the location verification and channel allocation phases is less than 0.05 ms. Even though the number of witnesses is 12, the running time of an SU obtaining the location proof at the location proof generation phase and obtaining a channel in location verification phase and channel allocation phase is less than 1.4 ms. As shown in Figure 2(b), when the number of witnesses is 3, the communication cost of the BS and that of the DB in channel preallocation phase are about 77 bytes, and the communication cost of an SU obtaining the location proof at the location proof generation phase is about 228 bytes; the communication cost of a BS to verify the location and to allocate a channel to an SU at the location verification and channel allocation phases is about 8 bytes. Although the number of witnesses is 12, the communication cost of an SU obtaining the location proof in that and the subsequent phases and obtaining a channel is about 912 bytes.

6.2. Comparison with Other Protocols. In this subsection, we compare the performance of the proposed protocol with other protocols proposed by Gao et al. [20] and Grissa et al. [21], and we provide analytical expressions of the communication of these protocols in Table 7 and computation cost of these protocols in Table 8. We plot the comparison of computation and communication costs of these protocols in Figure 3 using the expressions established in Tables 7 and 8. Figure 3(a) shows that our protocol is superior to the protocols proposed by both Gao et al. [20] and Grissa et al. [21] in communication costs, because we used symmetric encryption operations in our protocol, while Figure 3(b) shows the comparison of the computation cost of these protocols. As shown in Figure 3(b), when the number of cells increases, our computation cost remains 1.4 ms, which proves our protocol is lightweight in computation cost and unaffected by the number of cells.

In summary, Figure 3 shows the performance of our protocol is superior to that of the protocols proposed by Gao et al. [20] and Grissa et al. [21] and is unaffected by the number of cells. Meanwhile, the performance analysis also shows our protocol is lightweight in both computation and communication costs, which achieves a much better performance in comparison with other protocols.

7. Conclusion

In this paper, we propose a lightweight privacy-preserving location verification protocol (LPPLV) for large-scale database-driven CRNs. In LPPLV, the SU does not need to provide his/her real identity and location information

FIGURE 3: Comparison of computation cost and communication cost.

TABLE 7: Communication cost of different protocols.

	Communication cost
Gao et al. [20]	$(2m + 3) \cdot [\log p]$
Grissa et al. [21]	$\frac{\sigma_{query} + \varphi \cdot s \cdot m^2 \cdot (\log_2(1/\epsilon) + \log_2(2\delta))}{\delta} + s \cdot \sigma_{HMAC}$
Our protocol	$2306 + 66K + 1548N + 292 \cdot n \cdot N$

TABLE 8: Computation costs of different protocols.

	SU	BS	DB
Gao et al. [20]	$4m \cdot T_M$	-	$O(m^2) \cdot T_M$
Grissa et al. [21]	$s \cdot T_H$	$3s \cdot T_H$	$3\varphi \cdot s \cdot m^2 \cdot T_H$
Our protocol	$T_{EC} + (N + 2) \cdot T_{DA} + N \cdot T_{EA} + N \cdot T_H$ $(2N + 3) \cdot T_{MAC}$	$(N + 3) \cdot T_{EA} + N \cdot T_H + (N + 1) \cdot T_{DA} + T_{DC} + (2N + 5) \cdot T_{MAC}$	$T_{EA} + T_{DA} + 2T_{MAC}$

for the DB, which prevents the DB from obtaining the real identity and location information of SUs. LPPLV has a distributed architecture in which SUs generate location proofs for each other. The new witness selection mechanism allows malicious SUs to be discovered by the BS when malicious SUs generate fake location information. This new witness selection mechanism also provides a solution for possible SU-Witness collusion, which has not been well solved in existing works. Further, by channel preallocation, LPPLV improves the efficiency of obtaining available channels for SUs. The results of security analysis prove that LPPLV resists various types of attacks in database-driven CRNs. Experimental results indicate the efficiency of LPPLV.

In future work, we will study how to improve the security and privacy of PUs. Because SUs can learn the state of PUs from the channel, this might be easily misused and threaten the privacy of PUs. Therefore, we will take the PUs' privacy into account in our future work.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by National Natural Science Foundation of China [Grant No. 61771140, No.61841701], Major Program of Fuzhou Municipal Bureau of Science and Technology [Grant No. (2017)325], Major Science and Technology Project in Fujian Province [Grant No. 2017H6005], and Natural Science Foundation of Fujian Province (Grant No.2018J01560).

References

- [1] M. Zhang, Y. Zhang, Y. Jiang, and J. Shen, "Obfuscating EVES algorithm and its application in fair electronic transactions in public cloud systems," *IEEE Systems Journal*, pp. 1–9.
- [2] Y. Li, L. Zhou, H. Zhu, and L. Sun, "Privacy-preserving location proof for securing large-scale database-driven cognitive radio networks," *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 563–571, 2016.
- [3] A. Aijaz and A. H. Aghvami, "Cognitive machine-to-machine communications for internet-of-things: a protocol stack perspective," *IEEE Internet of Things Journal*, vol. 2, no. 2, pp. 103–112, 2015.
- [4] W. Ejaz, M. Naeem, A. Shahid, A. Anpalagan, and M. Jo, "Efficient energy management for the internet of things in smart cities," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 84–91, 2017.
- [5] Y. Sun and K. Chowdhury, "Enabling emergency communication through a cognitive radio vehicular network," *IEEE Communications Magazine*, vol. 52, no. 10, pp. 68–75, 2014.
- [6] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: a survey," *Physical Communication*, vol. 4, no. 1, pp. 40–62, 2011.
- [7] M. Grissa, B. Hamdaoui, and A. A. Yavuz, "Location privacy in cognitive radio networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1726–1760, 2017.
- [8] "Federal Communications Commission," <http://transition.fcc.gov/DailyReleases/DailyBusiness/2012/db0405/FCC-12-36A1.pdf>, [Last accessed May 2012].
- [9] Y. Zeng, L. Xu, X. Yang, and X. Yi, "An efficient privacy-preserving protocol for database-driven cognitive radio networks," *Ad Hoc Networks*, pp. 1–11, 2018.
- [10] M. Grissa, A. A. Yavuz, and B. Hamdaoui, "Cuckoo filter-based location-privacy preservation in database-driven cognitive radio networks," in *Proceedings of the IEEE Computer Networks and Information Security on Proceedings, 2015 World Symposium on*, pp. 1–7, Hammamet, Tunisia, 2015.
- [11] S. Pontarelli, P. Reviriego, and J. Maestro, "Parallel d-pipeline: a cuckoo hashing implementation for increased throughput," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 65, no. 1, pp. 326–331, 2016.
- [12] E. Troja and S. Bakiras, "Leveraging P2P interactions for efficient location privacy in database-driven dynamic spectrum access," *International Journal of Network Security*, vol. 17, no. 5, pp. 569–579, 2015.
- [13] D. Petrov and T. Znati, "Location privacy preserving protocols in database-enabled cognitive radio networks," in *Proceedings of the Wireless Communications and Mobile Computing Conference (IWCMC) on Proceedings, 13th International*, pp. 147–152, IEEE, Spain, 2017.
- [14] L. Zhang, C. Fang, Y. Li, H. Zhu, and M. Dong, "Optimal strategies for defending location inference attack in database-driven CRNs," in *Proceedings of the IEEE International Conference on Communications, ICC on Proceedings*, pp. 7640–7645, IEEE, London, UK, 2015.
- [15] B. Bahrak, S. Bhattarai, A. Ullah, J.-M. J. Park, J. Reed, and D. Gurney, "Protecting the primary users' operational privacy in spectrum sharing," in *Proceedings of the Dynamic Spectrum Access Networks (DYSPAN) on Proceedings, 2014 IEEE International Symposium on*, pp. 236–247, IEEE, McLean, VA, USA, 2014.
- [16] J. Xin, M. Li, C. Luo, and P. Li, "Privacy-preserving spectrum query with location proofs in database-driven CRNs," in *Proceedings of the IEEE Global Communications Conference on Proceedings*, pp. 1–6, IEEE, Washington, DC, USA, 2016.
- [17] M. Reza Nosouhi, S. Yu, M. Grobler, Y. Xiang, and Z. Zhu, "SPARSE: privacy-aware and collusion resistant location proof generation and verification," in *Proceedings of the GLOBECOM 2018 - 2018 IEEE Global Communications Conference*, pp. 9–13, IEEE, Abu Dhabi, UAE, 2018.
- [18] C. Aguilar-Melchor, J. Barrier, L. Fousse, and M. Killijian, "Xpir: private information retrieval for everyone," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 2, pp. 155–174, 2016.
- [19] Y. Zhang, W. Tong, and S. Zhong, "On designing satisfaction-ratio-aware truthful incentive mechanisms for k-anonymity location privacy," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2528–2541, 2016.
- [20] Z. Gao, H. Zhu, Y. Liu, M. Li, and Z. Cao, "Location privacy in database-driven cognitive radio networks: attacks and countermeasures," in *Proceedings of the IEEE International Conference on Computer Communications on Proceedings*, pp. 2751–2759, IEEE, Turin, Italy, 2013.
- [21] M. Grissa, A. A. Yavuz, and B. Hamdaoui, "Location privacy preservation in database-driven wireless cognitive networks through encrypted probabilistic data structures," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 2, pp. 255–266, 2017.
- [22] X. Wang, A. Pande, J. Zhu, and P. Mohapatra, "STAMP: enabling privacy-preserving location proofs for mobile users," *IEEE/ACM Transactions on Networking*, vol. 24, no. 6, pp. 3276–3289, 2016.
- [23] Z. Zhu and G. Cao, "APPLAUS: a privacy-preserving location proof updating system for location-based services," in *Proceedings of the IEEE International Conference on Computer Communications on Proceedings*, pp. 1889–1897, Shanghai, China, 2011.
- [24] C. Javali, G. Revadigar, K. B. Rasmussen, W. Hu, and S. Jha, "I am alice, i was in wonderland: secure location proof generation and verification protocol," in *Proceedings of the IEEE 41st conference on local computer networks (LCN) on Proceedings*, pp. 477–485, IEEE, Dubai, UAE, 2016.
- [25] Y. Zhang, C. C. Tan, F. Xu, H. Han, and Q. Li, "VProof: Lightweight privacy-preserving vehicle location proofs," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 1, pp. 378–385, 2015.
- [26] Y. Zeng, X. Li, X. Yang, Q. Xu, and D. Wang, "A practical privacy preserving protocol in database-driven cognitive radio networks," in *Proceedings of the 23rd Australasian Conference on Information Security and Privacy (ACISP) on Proceedings*, pp. 634–648, Wollongong, Australia.