Privacy-Preserving Spectrum Query with Location Proofs in Database-Driven CRNs

Jiajun Xin*, Ming Li*, Changqing Luo†, Pan Li†
* Department of Computer Science and Engineering, University of Nevada, Reno 89577, USA
† Department of Electrical Engineering and Computer Science, Case Western Reserve University,
Cleveland, OH 44106, USA

Abstract—**The database-driven cognitive radio network (CRN) is regarded as a promising way for a better utilization of spectrum resources without introducing the interference to primary users (PUs). However, there are some critical security and privacy issues in database-driven CRNs, which have been rarely discussed before. First of all, in order to retrieve the spectrum available information (SAI) of one's vicinity, an SU's query will inevitably disclose its location information. Second, malicious SUs may query SAI for other locations so as to infer operational patterns of PUs and other SUs. In addition, they can reconstruct the entire SAI of the database and sell it for profit. Therefore, in this paper we aim to guarantee both location privacy of SUs and information security of the database during spectrum query in database-driven CRNs. We first leverage private information retrieval (PIR) techniques to allow the database to find out the SAI regarding a querying SU's location, without learning the query information, i.e., this SU's location. To prevent malicious SUs inferring SAI of other locations, SUs are required to provide location proofs indicating that they are at the places where they claim to be. Theoretical analysis is provided showing that our scheme is privacy-preserving and secure. Experiments are also conducted to evaluate the its efficiency.**

Index Terms—**Spectrum query, database-driven cognitive radio networks, location privacy, information security, location proof**

I. Introduction

The recent exploding growth and popularity of wireless devices and services have exacerbated the spectrum deficiency in wireless networks. Recent studies show that this issue is also largely attributed to inefficient spectrum utilization due to the current static spectrum policies, by which spectrums are exclusively used by their licensed holders, and cannot be accessed by other users even if they are not in use. To address this artificial spectrum scarcity problem, cognitive radio networks (CRNs) have been proposed allowing primary users (PUs) to lease their unused spectrums to secondary users (SUs) who do not have licensed spectrums. Meanwhile, database-driven CRNs have been receiving increasing attention from both academia and industry due to its management efficiency for dynamic spectrum access without introducing interference to PUs. Several commercial entities have been involved in the database-driven CRN administration such as Cellular South, Google Inc. and Wi-Fi Alliance [1].

In database-driven CRNs, the database is managed by some service provider to store the spectrum available information (SAI). When an SU has data to transmit, it first queries the database for the SAI of its vicinity, based on which it further decides which spectrum to use. Apparently, an SU has to report its location to the database in order to retrieve accurate SAI. As a location record can reveal its owner's critical information such as visiting history, commute routes, habits, and preferences, SUs' privacy will be violated during spectrum query if their location is disclosed. Only few existing works discuss protecting SUs' location privacy during spectrum query in CRNs. Gao *et al.* [2] design a private SAI retrieval scheme that applies blinding factors to hide SUs' locations. Zhang *el al.* [3] propose to protect both SUs' and PUs' location privacy while achieving utility maximization. On the other hand, malicious SUs may try to collect SAI of other locations so as to infer operational patterns of PUs and other SUs [4]. In addition, they can reconstruct the entire SAI of the database and sell it for profit. However, this security issue has never been discussed before.

With all these concerns, in this paper we aim to guarantee both location privacy of SUs and information security of the database during spectrum query in database-driven CRNs. To protect SUs' location information from the database, we leverage private information retrieval (PIR) techniques [5], [6] to allow the database to find out the SAI regarding a querying SU's location, without learning its query information, i.e., this SU's location. To prevent malicious SUs inferring SAI of other locations, SUs are required to provide location proofs indicating that they are at the places where they claim to be. The format of location proofs should be carefully designed, such that 1) it will not reveal the SU's location; 2) it ensures that no SU can obtain any useful SAI other than the one for its current location. The contribution of this paper is summarized as follows.

- We jointly discuss the issues of protecting SU location privacy and database information security during spectrum query in database-driven CRNs.
- We develop a novel privacy-preserving spectrum query scheme with location proofs preventing malicious SUs

This work was supported in part by the U.S. National Science Foundation under Grant CNS-1566634. The work of P. Li was supported by the U.S. National Science Foundation under Grant CNS-1602172 and Grant CNS-1566479.

inferring SAI of other locations, without revealing SUs' location information.

The rest of this paper is organized as follow. We first discuss related work in section II. Section III describes preliminaries of this paper. The system overview is introduced in Section IV. Then, we elaborate our proposed scheme in section V, which is followed by security and privacy analysis in Section VI. We evaluate the performance our scheme in Section VII. Section VIII concludes the paper.

II. Related Work

There have been only few existing works dealing with location privacy in database-driven CRNs. Gao *el al.* [2], [7], [8] are among the first to discuss this problem. They design a private SAI retrieval scheme that applies blinding factors to hide SUs' locations. Zhang *el al.* [3] propose to protect both SUs' and PUs' location privacy while achieving utility maximization. In the work [4], Bahrak *et al.* develop several schemes based on cloaking technique to protect PUs' operational privacy from malicious SUs. Since we focus on preserving SUs' location privacy from database which is different from the problem discussed in [3], [4], their techniques cannot be applied here. Moreover, malicious SUs may try to collect SAI of other locations so as to infer operational patterns of PUs and other SUs. They can also reconstruct the entire SAI of the database and sell it for profit. Therefore, it is critically important to protect information security for database which, however, has been neglected so far in the above works.

Another line of research that is related to this work is location proof generation, which provides users a way to prove they are at some location where they claim to be. Although most of current mobile devices are equipped with GPS, users can easily fake their locations once they jailbreak their devices. Zeng *el al.* [9] propose a peer location verification (PLV) scheme for location proof. Their scheme does not work when there is no peer around. The APPLAUS system [10] uses bluetooth to deliver location proofs among users. Its efficiency is constrained by the short communication range of bluetooth. In [11], two schemes are proposed to use passive RFID tags for location proofs. However, there is heavy computation load at the central processing unit for the location proof verification process. WiFi APs have also been utilized to generate location proofs for users nearby [12], [13]. Typically, when an AP can directly communicate with a user, it considers that this user is within its transmission range and thus share the same location with it.

III. Preliminaries

An integer ϕ is called a quadratic residue (QR) modulo N_1 if there exists an integer X such that

$$X^2 \equiv \phi \text{ (mod } N_1)$$

where N_1 can be any integer larger than 2. If there is no such an integer X, we call ϕ a quadratic nonresidue (QNR) modulo N_1.

A. Jacobi symbol

Jacobi symbol is a multiplicative function with values 1, -1, and 0

$$\left(\frac{\phi}{N_2}\right) = \begin{cases} 1 & \text{if } \phi \text{ is a QR modulo } N_2 \\ & \text{ and } \phi \not\equiv 0 \pmod{N_2} \\ -1 & \text{if } \phi \text{ is a QNR modulo } N_2 \\ 0 & \text{if } \phi \equiv 0 \pmod{N_2}. \end{cases}$$

Here N_2 must be a positive odd integer. For Jacobi symbols, we have the following properties

$$\left(\frac{\rho\sigma}{N_2}\right) = \left(\frac{\rho}{N_2}\right)\left(\frac{\sigma}{N_2}\right)$$

where ρ and σ can be any positive integers.

B. Quadratic Residue Assumption

This assumption was first used in the cryptographic setting in [14]. Given an integer ϕ and N_3, whether ϕ is a QR modulo N_3 or not, where $N_3 = p'q'$ for two primes p' and q'. It states that if p' and q' are known, one can calculate the Jacobi symbol of ϕ, i.e., $(\frac{\phi}{N_3})$ in $O(|N_3|^3)$ time. If not, there is no polynomial time function to calculate it better than guessing [5]. Obviously, N_3 is also an positive odd integer.

IV. System Overview

A. System Model

As shown in Fig. 1, the system involves three types of entities: the database, Wi-Fi access points and secondary users. In what follows, we describe their functions and interactions.

Database (DB). The DB is the entity where the SAI is stored. When receiving a query from an SU, the DB first verifies its location proof, i.e. this SU is where it claims to be. Then, it calculates and sends back the SAI based on the SU's location.

Wi-Fi access points (APs). Following that in [12], [13], we have APs to provide location proofs in our system. In particular, if an AP can communicate directly with an SU, it considers this SU at the same location as itself. Then, it generates a signature as the location proof for this SU. Because the traditional public-key cryptography based signature will reveal the ID of an AP[1], we propose to use ring signature instead.

Secondary users (SUs). SUs are the ones who do not have licensed spectrum in CRNs. Whenever an SU has data to transmit, it connects to the DB and queries for the SAI of its current location. Meanwhile, it does not want to disclose its location information to the DB during the query.

[1] One can link the unique public decryption key of an AP's signature to this AP's ID.

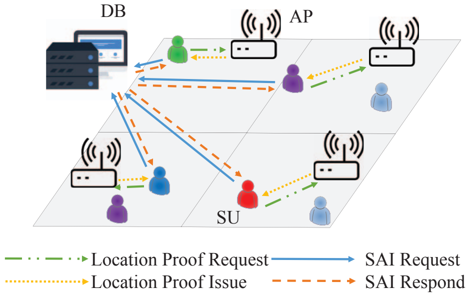

Fig. 1. System model.

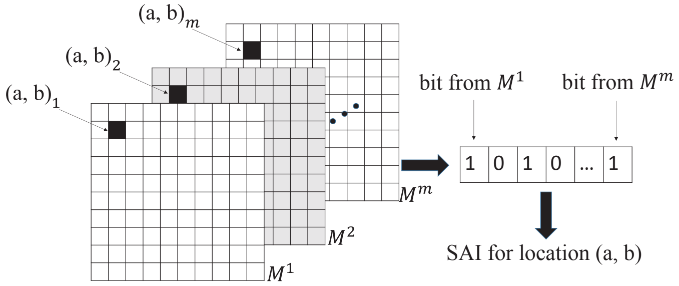

Fig. 2. Database structure for SAI.

A typically workflow can be described as follow. Before querying the DB for SAI, an SU first obtains the location proof from an AP which it can directly connect to. Then, the SU generates its spectrum query and sends it to the DB together with the location proof. The DB first verifies the location proof. If it is valid, the DB calculates and returns the SAI to the SU.

B. Security and Privacy

Apart from service accuracy, we aim to achieve the following security and privacy objectives in our proposed scheme.

Security. Malicious SUs may try to collect SAI of other locations so as to infer operational patterns of PUs and other SUs [4]. In addition, they can reconstruct the entire SAI of the database and sell it for profit. Therefore, SUs should not obtain any useful SAI of other locations expect the one for its current location.

Privacy. We assume that the DB works in a semi-honest mode, i.e., the DB correctly follows the protocol but is curious about SUs' location by analyzing their queries. Therefore, query messages cannot reveal their owners' location. Besides, the DB cannot learn SUs' location from the SAI to be sent back[2]. Moreover, since an AP can only generate location proof for the SUs nearby, neither the AP's location nor its ID should be known to the DB. Otherwise, SUs' location is equivalently disclosed. Besides, to prevent APs colluding with the DB by directly sending SUs' locations and their IDs to the DB, SUs request for the location proof anonymously.

C. Block Retrieval

Suppose the SAI of a specific location is composed with m bits as shown in Fig. 2. We construct the DB as a stack of m matrices M^μ $(1 \leq \mu \leq m)$, each with the size of $s \times t$, which stands for the entire geographic area. Assume an SU locates at (a, b) $(1 \leq a \leq s,\ 1 \leq b \leq t)$[3]. It first retrieves the content (bit) of the a-th row, b-th column from all M^μ's, then it reconstructs them into an m-bit SAI. In the following, we describe our scheme for retrieving 1 bit from M^μ. The processes to retrieve the rest $m-1$ bits directly follow.

V. Our Proposed Scheme

Our proposed scheme consists of five processes, location proof request, location proof issue, SAI request, SAI respond and SAI recover. Considering an SU is located at (a, b), we now explain how it privately obtain $M^\mu_{a,b}$, i.e., the element in a-th row and b-th column of M^μ.

- **Location Proof Request.** Before an SU queries for SAI from the DB, it needs to obtain a location proof from an AP that is also located at (a, b). In the case that there are multiple APs, the SU chooses the one with the strongest signal strength. The SU picks two large primes p and q, and calculates $N = pq$. The SU sends N to the AP as a location proof request. Note that the SU communicates with the AP anonymously.
- **Location Proof Issue.** When receiving a location proof request, the AP first checks if it is directly sent by an SU within its coverage area by analyzing the received signal strength. If so, it generates an s-element vector $\overrightarrow{\delta}$ where the a-th element is the square of a random number and other elements are random numbers from Z_N^*. In this way, only the a-th number is sure to be a QR while the rest are either QRs or QNRs. Similarly, the AP generates a t-element vector $\overrightarrow{\delta}'$ where only the b-th element is the square of a random number while the rest are either QRs or QNRs. The AP further signs $\overrightarrow{\delta}$ and $\overrightarrow{\delta}'$. As we discussed above, to prevent the DB from inferring SU's location by linking the AP's ID with its location, the AP uses ring signature [15] to hide its ID here. The AP further encrypts $\overrightarrow{\delta}$, $\overrightarrow{\delta}'$, their signature and a time stamp[4] with some symmetric encryption scheme, say AES, and sends it to the SU as the location proof.
- **SAI Request.** The SU prepares an s-element query vector $\overrightarrow{\gamma} = [\gamma_1, \ldots, \gamma_r, \ldots, \gamma_s]$ $(\gamma_r \in Z_N^+)$ such that γ_a is a QNR and the rest elements are QRs. Similarly, the SU prepares another t-element query vector

[2] For example, if the returned SAI contains information such as spectrum A is available, it can help the DB to shrink the SU's location to the region where spectrum A is available.

[3] Instead of a specific coordinate, (a, b) stands for a geographic area where spectrums demonstrate consistent property, i.e., either available or occupied.

[4] The purpose of embedding the current time stamp is to prevent a malicious SU double-using this location proof for SAI query after it travels to somewhere else.

$\overrightarrow{\gamma}= [\gamma'_1, \ldots, \gamma'_c, \ldots, \gamma'_t]$ ($\gamma'_c \in Z_N^+$) such that γ_b is a QNR with the rest QRs. Then, the SU sends N, $\overrightarrow{\gamma}$, $\overrightarrow{\gamma}'$ and the location proof to the DB.

- **SAI Respond.** The DB first decrypts the message and checks the time stamp as well as the signature to make sure the location proof is valid. It then calculates every row r $(1 \leq r \leq s)$ a value $\eta_r \in Z_N^*$ as follows. Denote by $M_{r,c}$[5] the element in the r-th row and c-th column. The DB calculates

$$m_{r,c} = \begin{cases} \gamma_c^2 \mod N \text{ if } M_{r,c} = 0 \\ \gamma_c \mod N \text{ if } M_{r,c} = 1. \end{cases}$$

The observation here is that if $c \neq b$, then $m_{r,c}$ is always a QR. If $c = b$, then we have

$$\left(\frac{m_{r,c}}{N}\right) = \begin{cases} 1 & \text{if } M_{r,c} = 0 \\ -1 & \text{if } M_{r,c} = 1. \end{cases} \tag{1}$$

Furthermore, the DB randomly sets $\xi_{r,c}$ to be one of the two values $\{m_{r,c}, m_{r,c}^2\}$ and stores the selection. Then, the DB calculates η_r as

$$\eta_r = (\prod_{c=1}^{t} \xi_{r,c}) \mod N$$

and gets an s-element vector $\overrightarrow{\eta}= [\eta_1, \ldots, \eta_a, \ldots, \eta_s]$. The DB further prepares the other vector in a similar way and gets

$$m'_{r,c} = \begin{cases} \gamma_r'^2 \mod N \text{ if } M_{r,c} = 0 \\ \gamma'_r \mod N \text{ if } M_{r,c} = 1. \end{cases}$$

The observation here is that if $r \neq a$, then $m'_{r,c}$ is always a QR, and if $r = a$, we have

$$\left(\frac{m'_{r,c}}{N}\right) = \begin{cases} 1 & \text{if } M_{r,c} = 0 \\ -1 & \text{if } M_{r,c} = 1. \end{cases}$$

Based on the stored selection before, the DB sets $\xi'_{r,c}$ as $m_{r,c}'^2$ if $\xi_{r,c}$ was set as $m_{r,c}$, and it sets $\xi'_{r,c}$ as $m'_{r,c}$ if $\xi_{r,c}$ was set as $m_{r,c}^2$. After that, the DB calculates η'_c as

$$\eta'_c = (\prod_{r=1}^{s} \xi'_{r,c}) \mod N$$

and gets a t-element vector $\overrightarrow{\eta}'= [\eta'_1, \ldots, \eta'_b, \ldots, \eta'_t]$. For vectors $\overrightarrow{\eta}$ and $\overrightarrow{\eta}'$, only the a-th element in $\overrightarrow{\eta}$ and the b-th element in $\overrightarrow{\eta}'$ contain the SAI that the SU needs. In order to prevent the SU obtaining SAI of location (a', b') where $(a, b) \neq (a', b')$, the DB masks the SAI as

$$A_k = \delta_k \times \eta_k \mod N \ (1 \leq k \leq s)$$
$$A'_k = \delta'_k \times \eta'_k \mod N \ (1 \leq k \leq t).$$

Recall that $\delta_k(\delta'_k)$ is the k-th element of vector $\overrightarrow{\delta}$ $(\overrightarrow{\delta}')$. Since the database does not know the factorization of N, it cannot tell if $\delta_k(\delta'_k)$ is a QR or QNR according to quadratic residue assumption. Thus it cannot infer this SU's location from $\overrightarrow{\delta}$ and $\overrightarrow{\delta}'$. The DB finally sends $\overrightarrow{A}$ and $\overrightarrow{A}'$ to the SU as the answer.

- **SAI Recover.** When the SU receives the answer, it extracts the a-th element in $\overrightarrow{A}$ and the b-th element in $\overrightarrow{A}'$, and calculates the result as

$$R = A_a \times A'_b \mod N.$$

Since the SU knows the factorization of N, it can easily check if the result R is a QR or a QNR. With this information, the SU derives the value of $M_{a,b}$ following

$$M_{a,b} = \begin{cases} 1 \text{ if } \left(\frac{R}{N}\right) = -1 \\ 0 \text{ if } \left(\frac{R}{N}\right) = 1. \end{cases}$$

Correctness: To verify the correctness of our scheme, it is equivalent to prove

$$\left(\frac{R}{N}\right) = \begin{cases} 1 & \text{if } M_{a,b} = 0 \\ -1 & \text{if } M_{a,b} = 1. \end{cases} \tag{2}$$

Without loss of generality, we assume that the DB sets $\xi_{a,b}$ as $m_{a,b}$ and sets $\xi'_{a,b}$ as $m_{a,b}'^2$. Thus we have

$$\begin{aligned} \left(\frac{R}{N}\right) &= \left(\frac{A_a}{N}\right) \times \left(\frac{A'_b}{N}\right) \\ &= \left(\frac{\eta_a}{N}\right) \times \left(\frac{\delta_a}{N}\right) \times \left(\frac{\eta'_b}{N}\right) \times \left(\frac{\delta'_b}{N}\right) \\ &= \left(\frac{\prod_{c=1}^{t} \xi_{a,c}}{N}\right) \times \left(\frac{\prod_{r=1}^{s} \xi'_{r,b}}{N}\right) \\ &= \left(\frac{m_{a,b}}{N}\right) \times \left(\frac{m_{a,b}'^2}{N}\right) = \left(\frac{m_{a,b}}{N}\right). \end{aligned}$$

According to (1), we can conclude that (2) holds.

VI. Security and Privacy Analysis

A. Security

We first show that the SU can get no useful SAI of other locations from $\overrightarrow{A}$ and $\overrightarrow{A}'$. Assume that SU is located at (a, b). The analysis is conducted over the following three cases, i.e., Case I, $a \neq a'$, $b = b'$, Case II, $a = a'$, $b \neq b'$, and Case III, $a \neq a'$, $b \neq b'$. In Case I, we can directly infer that $\overrightarrow{A}'$ contains no SAI for the location (a', b). We further show that the SU cannot get any useful SAI from $\overrightarrow{A}$. Specifically, when $M_{a',b} = 1$, it is easy to derive that $P((\frac{\eta_{a'}}{N}) = 1) = P((\frac{\eta_{a'}}{N}) = -1) = 1/2$. According to [14], for any random number in Z_N^*, its Jacobi symbol takes value 1 or -1 with equal probabilities. Therefore, $P((\frac{A_{a'}}{N}) = 1) = P((\frac{A_{a'}}{N}) = -1) = 1/2$. Similarly, when $M_{a',b} = 0$, we have $P((\frac{A_{a'}}{N}) = 1) = P((\frac{A_{a'}}{N}) = -1) = 1/2$. Thus, the SU cannot derive any useful information from $\overrightarrow{A}$. The analysis is similar for Case II and III. To sum up, the SU cannot get SAI of other locations.

[5] In the rest of this paper, we use $M_{r,c}$ instead of $M_{r,c}^{\mu}$ for simplicity.

We now show the SU cannot gain additional information when it queries for the SAI of other locations (a', b'). First, the result $(\frac{R}{N})$ can be written as $(\frac{m_{a',b'}}{N}) \times (\frac{\delta_{a'}}{N}) \times (\frac{\delta'_{b'}}{N})$. According to [14], $P((\frac{\delta_{a'}}{N}) \times (\frac{\delta'_{b'}}{N}) = 1) = P((\frac{\delta_{a'}}{N}) \times (\frac{\delta'_{b'}}{N}) = -1) = 1/2$. Suppose in the DB, $P(M_{a',b'} = 0) = P(M_{a',b'} = 1) = 1/2$, then the result $P((\frac{R}{N}) = 1) = P((\frac{R}{N}) = -1) = 1/2$. Therefore, the claim holds.

B. Privacy

In this part, we show that the DB cannot learn SUs' location information from their SAI requests and location proofs. The available parameters at the DB regarding an SU's query include N, $\overrightarrow{\gamma}$, $\overrightarrow{\gamma}'$, $\overrightarrow{\delta}$ and $\overrightarrow{\delta}'$. For $\overrightarrow{\gamma}$, it is an s-element vector $\overrightarrow{\gamma} = [\gamma_1, \ldots, \gamma_r, \ldots, \gamma_s]$ prepared by SU, where γ_a is a QNR and the rest elements are QRs. Since the DB does not know the factorization of N, it has no idea whether an element γ_r is a QR and QNR according to the quadratic residue assumption [5]. Therefore, it cannot identify γ_a from $\overrightarrow{\gamma}$ and thus the x-coordinate of the SU. Similarly, the DB cannot identify γ'_b from $\overrightarrow{\gamma}'$ and thus the y-coordinate of the SU. The DB cannot learn SU's location by analyzing $\overrightarrow{\delta}$ and $\overrightarrow{\delta}'$ due to the similar reason. Besides, since APs adopt ring signature to sign $\overrightarrow{\delta}$ and $\overrightarrow{\delta}'$. According to the property of the ring signature, the DB can only tell if the signature comes from an authorized AP or not, without knowing this AP's ID and thus its location.

VII. Performance Evaluation

In this section, we evaluate the performance of our proposed scheme by showing its computation cost and communication cost. All experiments are conducted on a 64-bit computer with i7 CPU of 3.6GHz and 8G memory. We implement our scheme based on the GNU Multiple Precision Arithmetic Library and *PBC* [16] in c language. In particular, in *PBC* we use the Type A elliptic curve, which has the form of $y^2 = x^3 + x$. The base field of ring signature is 512 bits and the public key size for ring signature is 1024 bits. The data transmitted from an AP to an SU is encrypted with AES-256. We consider that the SAI consists of one matrix with equal size of rows and columns ($s = t$). We set by default $s = t = 400$ and the bit numbers l_N, l_p and l_q of N, p and q as 512, 256 and 256, respectively. All experiment results are the average of 100 trials.

A. Computation Cost

Fig. 3 depicts the computation time for each process under different row (column) sizes of the SAI matrix. In the location proof request process, an SU finds two random prime numbers p and q and calculates N by multiplying these two numbers. Therefore, the time consumption of location proof request is negligible. In the location proof issue process, an AP creates vectors $\overrightarrow{\delta}$ and $\overrightarrow{\delta}'$ and their ring signatures, and further encrypts them with AES-256. When $s = 400$, its total computation time is 0.2691 s. In the SAI request process, the SU prepares two vectors $\overrightarrow{\gamma} = [\gamma_1, \ldots, \gamma_a, \ldots, \gamma_s]$ and $\overrightarrow{\gamma} = [\gamma'_1, \ldots, \gamma'_b, \ldots, \gamma'_t]$, where γ_a and γ'_b are QNRs while the rest are QRs. For this purpose, in our experiment we first randomly choose from Z_N^+ an integer. If it is a QNR, the iteration ends; otherwise a new random number is picked until it is QNR. When $s = 400$, the corresponding time is 0.2228 s. Note that the SAI request can be accomplished offline at the SU. In the SAI respond process, the DB calculates the square of every elements in the SAI matrix, resulting in a computation time of 0.1427 s when $s = 400$. That explains why its computation time grows fast as the matrix size increases. In the SAI recover process, the SU only needs to compute one modulo N multiplication and one Jacobi symbol with low time consumption.

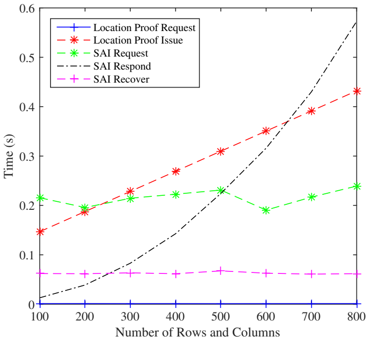

Fig. 3. Computation time under different row (column) sizes of SAI matrix.

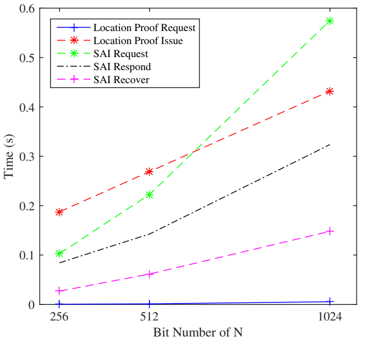

Fig. 4. Computation time under different bit number of modulus N.

We further show in Fig. 4 the relation between computation time and the bit number of modulus N, i.e., l_N. We find that the time consumed in all processes, except SAI recovery, grows as l_N increases. For the location proof request, location proof issue and SAI request process, it becomes more time-consuming to find a QNR or QR as l_N becomes larger. In SAI respond process, as the DB calculates the square of every element in the SAI matrix, the computation time is directly influenced by the size of each element, i.e., l_N.

B. Communication Cost

Fig. 5 illustrates the communication cost for each process under different row (column) sizes of the SAI matrix. In

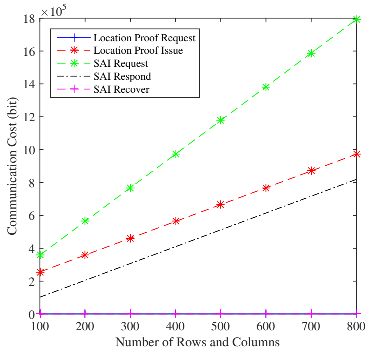

Fig. 5. Communication cost under different row (column) sizes of SAI matrix.

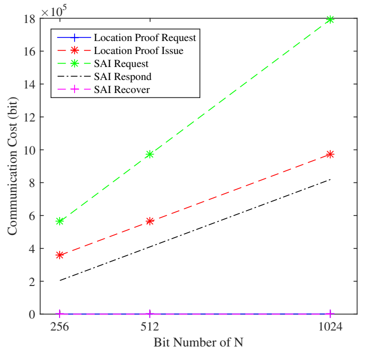

Fig. 6. Communication cost under different bit number of modulus N.

the location proof request process, an SU sends N to the AP, resulting in the communication cost as 512 bits. In the location proof issue process, the AP sends the ciphertext of $\overrightarrow{\delta}$, $\overrightarrow{\delta}'$ and their signatures to the SU. When $s = 400$, the communication cost is 563200 bits. We find that as s increases the communication cost grows, since the size of $\overrightarrow{\delta}$ and $\overrightarrow{\delta}'$ depends on the matrix size s. In the SAI request process, the SU sends N, $\overrightarrow{\gamma}$, $\overrightarrow{\gamma}'$ and the location proof to the DB. When $s = 400$, the corresponding communication cost is 972800 bits. In the SAI response process, the DB sends $\overrightarrow{A}$ and $\overrightarrow{A}'$ to the SU as the answer, resulting in the communication cost of 409600 bits when $s = 400$. Similarly, the communication cost in the SAI request and SAI response grows as s increases also due to the reason that the size of data $\overrightarrow{\gamma}$, $\overrightarrow{\gamma}'$, $\overrightarrow{A}$ and $\overrightarrow{A}'$ depends on the matrix size s. Since no data is transmitted during the SAI recover process, its communication cost is 0.

We further show in Fig. 6 the relation between communication cost and l_N. We find that the communication cost in location proof request, location proof issue, SAI request and SAI respond process grows as l_N increases, as the size of data transmitted in these processes depends on the size of modulus N.

VIII. Conclusion

In this paper we aim to guarantee both location privacy of SUs and information security of the database during spectrum query in database-driven CRNs. For this purpose, we develop a scheme based on PIR techniques, allowing the database to find out the SAI regarding a querying SU's location without learning its detail. To further prevent malicious SUs inferring SAI of other locations, SUs are required to provide location proofs indicating that they are at the places where they claim to be. We prove that our scheme is privacy-preserving and secure. We also show its computation and communication cost.

References

[1] "Third memorandum opinion and order," http://transition.fcc.gov/Daily_Releases/Daily_Business/2012/db0405/FCC-12-36A1.pdf, Federal Communications Commission, May 2012.

[2] Z. Gao, H. Zhu, Y. Liu, M. Li, and Z. Cao, "Location privacy in database-driven cognitive radio networks: Attacks and countermeasures," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 2751–2759.

[3] Z. Zhang, H. Zhang, S. He, and P. Cheng, "Achieving bilateral utility maximization and location privacy preservation in database-driven cognitive radio networks," in *Mobile Ad Hoc and Sensor Systems (MASS), 2015 IEEE 12th International Conference on*. IEEE, 2015, pp. 181–189.

[4] B. Bahrak, S. Bhattarai, A. Ullah, J.-M. Park, J. Reed, and D. Gurney, "Protecting the primary users' operational privacy in spectrum sharing," in *Dynamic Spectrum Access Networks (DYSPAN), 2014 IEEE International Symposium on*. IEEE, 2014, pp. 236–247.

[5] E. Kushilevitz and R. Ostrovsky, "Replication is not needed: Single database, computationally-private information retrieval," in *focs*. IEEE, 1997, p. 364.

[6] K. Narayanam and C. Rangan, "A novel scheme for single database symmetric private information retrieval," in *Proceedings of Annual Inter Research Institute Student Seminar in Computer Science (IRISS)*. Citeseer, 2006, pp. 803–815.

[7] Z. Gao, H. Zhu, S. Li, S. Du, and X. Li, "Security and privacy of collaborative spectrum sensing in cognitive radio networks," *IEEE Wireless Communications*, vol. 19, no. 6, pp. 106–112, 2012.

[8] Z. Gao, H. Zhu, Y. Liu, M. Li, and Z. Cao, "Location privacy leaking from spectrum utilization information in database-driven cognitive radio network," in *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM, 2012, pp. 1025–1027.

[9] K. Zeng, S. K. Ramesh, and Y. Yang, "Location spoofing attack and its countermeasures in database-driven cognitive radio networks," in *Communications and Network Security (CNS), 2014 IEEE Conference on*. IEEE, 2014, pp. 202–210.

[10] Z. Zhu and G. Cao, "Applaus: A privacy-preserving location proof updating system for location-based services," in *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2011, pp. 1889–1897.

[11] H. Gao, R. M. Lewis, and Q. Li, "Location proof via passive rfid tags," in *Wireless Algorithms, Systems, and Applications*. Springer, 2012, pp. 500–511.

[12] W. Luo and U. Hengartner, "Veriplace: a privacy-aware location proof architecture," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2010, pp. 23–32.

[13] Y. Li, L. Zhou, H. Zhu, and L. Sun, "Privacy-preserving location proof for securing large-scale database-driven cognitive radio networks," 2012.

[14] S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of computer and system sciences*, vol. 28, no. 2, pp. 270–299, 1984.

[15] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *Advances in CryptologyASIACRYPT 2001*. Springer, 2001, pp. 552–565.

[16] L. Ben, "The pairing-based cryptography," http://crypto.standford.edu/pbc.