Location Privacy in Cognitive Radios with Multi-Server Private Information Retrieval

Mohamed Grissa, Attila A. Yavuz, and Bechir Hamdaoui
Oregon State University, grissam,hamdaoui@oregonstate.edu
University of South Florida, attilaayavuz@usf.edu

***Abstract*—Spectrum database-based cognitive radio networks (CRNs) have become the de facto approach for enabling unlicensed secondary users (SUs) to identify spectrum vacancies in channels owned by licensed primary users (PUs). Despite its merits, the use of spectrum databases incurs privacy concerns for both SUs and PUs. Single-server private information retrieval (PIR) has been used as the main tool to address this problem. However, such techniques incur extremely large communication and computation overheads while offering only computational privacy. Besides, some of these PIR protocols have been broken.**

In this paper, we show that it is possible to achieve high efficiency and (information-theoretic) privacy for both PUs and SUs in database-driven CRN with multi-server PIR. Our key observation is that, by design, database-driven CRNs comprise multiple databases that are required, by the Federal Communications Commission, to synchronize their records. To the best of our knowledge, we are the first to exploit this observation to harness multi-server PIR technology to guarantee an optimal privacy for both SUs and PUs, thanks to the unique properties of database-driven CRN. We showed, analytically and empirically with deployments on actual cloud systems, that multi-server PIR is an ideal tool to provide efficient location privacy in database-driven CRN.

Keywords*—*Database-driven cognitive radio networks, location privacy, dynamic spectrum access, private information retrieval.

I. INTRODUCTION

The rapid growth of connected wireless devices has dramatically increased the demand for wireless spectrum and led to a serious shortage in spectrum resources. Cognitive radio networks (CRNs) [1] have emerged as a promising technology for solving this shortage problem by enabling dynamic spectrum access (DSA), which improves the spectrum utilization efficiency by allowing unlicensed/secondary users (SUs) to exploit unused spectrum bands (aka spectrum holes or white spaces) of licensed/primary users (PUs).

Currently, two approaches are being adopted to identify these white spaces: spectrum sensing and geolocation spectrum databases. In the spectrum sensing-based approach, SUs need to sense the PU channel to determine whether the channel is available for opportunistic use. The spectrum database-based approach, on the other hand, waives the sensing requirement and instead enables SUs to query a database (DB) to learn about spectrum opportunities in their vicinity. This approach, already promoted and adopted by the Federal Communications Commission (FCC), was introduced as a way to overcome the technical hurdles faced by the spectrum sensing-based approaches, thereby enhancing the efficiency of spectrum utilization, improving the accuracy of available spectrum identification, and reducing the complexity of terminal devices [2]. Moreover, it pushes the responsibility and complexity of complying with spectrum policies to DB and eases the adoption of policy changes by limiting updates to just a handful number of databases, as opposed to updating large numbers of devices [3].

FCC has designated nine entities (e.g. Google [4], iconectiv [5], and Microsoft [6]) as TV bands device database administrators which are required to follow the guidelines provided by PAWS (Protocol to Access White Space) standard [3]. PAWS sets guidelines and operational requirements for both the spectrum database and the SUs querying it. These include: SUs need to be equipped with geo-location capabilities, SUs must query DB with their specific location to check channel availability before starting their transmissions, DB must register SUs and manage their access to the spectrum, DB must respond to SUs' queries with the list of available channels in their vicinity along with the appropriate transmission parameters. As specified by PAWS standard, SUs may be served by several spectrum databases and are required to register to one or more of these databases prior to querying them for spectrum availability. The spectrum databases are reachable via the Internet, and SUs querying these databases are expected to have some form of Internet connectivity [7].

FCC has established a new service in the 3.5 GHz band, known as Citizens Broadband Radio Service (CBRS), in which the spectrum is also managed through a central database-driven CRN, aka spectrum access system (SAS), to enable spectrum sharing between military and federal incumbents and SUs. A separate entity with Environmental Sensing Capability (ESC) is responsible of populating DBs with data regarding PUs that do not wish to reveal their operational information such as their location or transmission characteristics. A similar concept, named licensed shared access (LSA), for the 2.3-3.4 GHz band is also being developed in Europe to enable SUs to opportunistically access spectrum resources in this band owned by incumbent military aircraft services and police wireless communications. A major difference compared to SAS, is that in LSA, PUs are responsible for populating DBs by providing their a priori information; i.e. their activities and, therefore the spectrum availability information, are known upfront [8].

A. Location Privacy Issues in Database-Driven CRNs

Despite their benefits, database-driven CRNs suffer from serious security and privacy threats. Since they could be seen as a variant of of *location based service (LBS)*, the disclosure of location information of SUs represents the main threat to SUs when it comes to obtaining spectrum availability from DBs. The fine-grained location, when combined with publicly available information, can easily reveal other personal information about an individual including his/her behavior, health condition, personal habits or even beliefs. For instance, an adversary can learn some information about the health condition of a user by observing that the user regularly goes to a hospital for example. The frequency and duration of these visits can even reveal the seriousness of a user illness and even the type of illness if the location corresponds to that of a specialty clinic. Matters get worse when SUs are mobile.

As per the PAWS requirements, SUs need to query DBs whenever they change their location by at least 100 meters. This will make SUs constantly share their location as they move which could be exploited by a malicious service provider for tracking purposes.

The location privacy of SUs is not the only privacy concern that database-driven CRNs suffer from. Indeed, the location privacy of PUs may also be critical in CRN systems such as SAS, in the 3.5 GHz CBRS band, and LSA, in the 2.3-2.4 GHz band, where PUs are not commercial but rather military and governmental entities. To achieve efficient spectrum sharing without interference to military and federal incumbents, these systems require PUs, or entities with sensing capabilities such as ESC, to report PUs' operational data (including their location, frequencies time of use, etc.) to be included in the spectrum databases which may present serious privacy risks to these PUs.

Being aware of such potential privacy threats, both SUs and PUs may refuse to share their sensitive information with DBs, which may present a serious barrier to the adoption of database-based CRNs, and to the public acceptance and promotion of the dynamic spectrum sharing paradigm. Therefore, *there is a critical need for developing techniques to protect the location privacy of both PUs and SUs while allowing the latter to harness the benefits of the CRN paradigm without disrupting the functionalities that these techniques are designed for to promote dynamic spectrum sharing*.

B. Research Gap and Objectives

Despite the importance of the location privacy issue in CRNs, only recently has it started to gain interest from the research community [9]. Some works focus on addressing this issue in the context of collaborative spectrum sensing [10]–[14]; others address it in the context of dynamic spectrum auction [15]. Protecting SUs' location privacy in database-driven CRNs is a more challenging task, merely because SUs are required, by protocol design, to provide their physical location to DB to learn about spectrum opportunities in their vicinity. The heterogeneity of wireless devices and the versatility of services relying on the CRN technology [16] could also present some challenges in designing privacy-preserving mechanisms for users in CRNs. In fact, privacy-preserving solutions need to embrace the different resource constraints of each SU device and the various requirements of each service in terms of data rates and delay sensitivities. This makes it hard to leverage general purpose public key encryption-based techniques due to their high cost in terms of computation and communication overheads especially on resource-constrained devices. It is therefore crucial to design cost-effective protocols that offer strong privacy guarantees to users and also adapt to different systems requirements regardless of the constraints of the users.

The existing location privacy preservation techniques for database-driven CRN (e.g., [2], [17]–[21]) generally rely on three main lines of privacy preserving technologies, (i) *k-anonymity* [22], (ii) *differential privacy* [23] and (iii) single-server *Private Information Retrieval (PIR)* [24]. However, the direct adaptation of *k-anonymity* based techniques have been shown to yield either insecure or extremely costly results [25]. The solutions adapting *differential privacy* (e.g., [20]) not only incur a non-negligible overhead, but also introduce a noise over the queries, and therefore they may negatively impact the accuracy of spectrum availability information.

Among these alternatives, single-server PIR seems to be the most popular. PIR technology is a suitable choice for database-driven CRNs, as it permits privacy preserving queries on a public database, and therefore can enable a SU to retrieve spectrum availability information from the database without leaking its location information. However, single-server PIR protocols rely on highly costly partial homomorphic encryption schemes, which need to be executed over the entire database for each query. Indeed, as we also demonstrated with our experiments in Section IV, the execution of a single query even with some of the most efficient single-server PIR schemes [26] takes approximately 20 seconds with a 80 $Mbps/\,30Mbps$ bandwidth on a moderate size database (e.g., 10^6 entries). An end-to-end delay with the orders of 20 seconds might be undesirable for spectrum sensing needs of SUs in real-life applications. Also, some of the state-of-the-art efficient computational PIR schemes [27] that are used in the context of CRNs have been shown to be broken [26]. Thus, there is a significant need for practical location privacy preservation approaches for database-driven CRNs that can meet the efficiency and functionality requirements of SUs.

C. Our Observation and Contribution

The objective of this paper is to develop efficient techniques for database-driven CRNs that preserve the location privacy of SUs during their process of acquiring spectrum availability information. We also try to protect the operational privacy of PUs in systems that require incumbents to provide spectrum availability information to DBs. Specifically, we will aim for the following design objectives: (i) (*location privacy of SUs*) Preserve the location privacy of SUs, whether fixed or mobile, while allowing them to receive spectrum availability information; (ii) *(efficiency and practicality)* Incur minimum computation, communication and storage overhead. The cryptographic delay must be minimum to permit fast spectrum availability decision for the SUs, and storage/processing cost must be low to enable practical deployments. (iii) *(fault-tolerance and robustness)* Mitigate the effects of system failures or misbehaving entities (e.g., colluding databases). (iv) *(location privacy of PUs)* The location information of PUs needs to be protected while still able to provide spectrum availability information to DBs. *It is very challenging to meet all of these seemingly conflicting design goals simultaneously.*

The main idea behind our proposed approaches is to harness special properties and characteristics of the database-driven CRN systems to employ private query techniques that can overcome the significant performance, robustness and privacy limitations of the state-of-the-art techniques. Specifically, our proposed approach is based on the following observation:

Observation: *FCC requires that all of its certified databases synchronize their records obtained through registration procedures with one another [28], [29] and need to be consistent across the other databases by providing exactly the same spectrum availability information, in any region, in response to SUs' queries [30]. That is, the same copy of spectrum database is available and accessible to the SUs via multiple (distinct) spectrum database administrators/providers. Is it possible exploit this observation to achieve efficiency location preservation techniques for database-driven CRN?*

In practice, as stated in PAWS standard [3], SUs have the option to register to multiple spectrum databases belonging to multiple service providers. Currently, many companies (e.g. Google [4], iconectiv [5], etc) have obtained authorization from FCC to operate geo-location spectrum databases upon successfully complying to regulatory requirements. Several other companies are still underway to acquire this authorization [31]. Thus, it is more natural and realistic to take this fact into consideration when designing privacy preserving protocols for database-based CRNs. Based on this observation, our main contribution is as follows:

TABLE I: Performance Comparison

Scheme	Comm.	Delay			Privacy
		DB	***SU***	total	
LP-$Chor$	$753\ KB$	$0.48\ s$	$0.0077\ s$	$0.62\ s$	$(\ell-1)$-*private*
LP-$Goldberg$	$6000\ KB$	$1.21\ s$	$0.32\ s$	$1.78\ s$	*t-private ℓ-comp.-private*
$RAID$-LP-$Chor$	$125\ KB$	$0.022\ s$	$0.00041\ s$	$0.21\ s$	$(\pi-1)$-*private*
$PriSpectrum$ [2]	$512.8\ KB$	$21\ s$	$0.084\ s$	24.2	*underlying PIR broken*
Troja et al [19]	$8.4\ KB$	$11760\ s$	$5.62\ s$	$11766\ s$	*computationally-private*
Troja et al [18]	$12120\ KB$	$11760\ s$	$48\ s$	$11820\ s$	*computationally-private*
XPIR [26]	$4321\ KB$	$17.66\ s$	$0.34\ s$	$20.53\ s$	*computationally-private*
SealPIR [32]	$512\ KB$	$11.03\ s$	$0.008\ s$	$11.35\ s$	*computationally-private*

Parameters: $n = 560\ MB,\ b = 560\ B,\ r = 10^6,\ \ell = 6,\ w = 8,\ k = 6$

Our Contribution: *To the best of our knowledge, we are the first to exploit the fact that multiple copies of spectrum DBs are available by nature in database-driven CRNs, and therefore it is possible to harness multi-server PIR techniques [24], [33] that offer information-theoretic privacy with substantial efficiency advantages over single-server PIR. This is achieved by relying on Shamir secret sharing-based techniques to either divide the content of SUs' queries or the spectrum availability information, or both, among the different DBs to prevent these DBs from inferring SUs' location from their queries or from learning PUs' sensitive operational data from the spectrum availability information.*

We show, analytically and experimentally with deployments on cloud systems, that our adaptation of multi-server PIR techniques significantly outperforms the state-of-the-art location privacy preservation methods as demonstrated in Table I and detailed in Section IV. Moreover, our adaptations achieve information theoretical privacy while existing alternatives offer only computational privacy. This feature provides an assurance against even post-quantum adversaries [34] and can avoid recent attacks on computational PIR [26].

Notice that, multi-server PIR techniques require the availability of multiple (synchronized) replicas of the database. Therefore, despite their high efficiency and security, they received a little attention from the practitioners. For instance, in traditional data outsourcing settings (e.g., private cloud storage), the application requires a client to outsource only a single copy of its database. The distribution and maintenance of multiple copies of the database across different service providers brings additional architectural and deployment costs, which might not be economically attractive for the client.

In this paper, we showcased one of the first natural use-cases of multi-server PIR, in which the multiple copies of synchronized databases are already available by the original design of application (i.e., spectrum availability information in multi-database CRNs), and therefore multi-server PIR does not introduce any extra overhead on top of the application. Exploiting this synergy between multi-database CRN and multi-server PIR permitted us to provide informational theoretical location privacy for SUs with a significantly better efficiency compared to existing single-server PIR approaches.

Desirable Properties: We outline the desirable properties of our approaches below.

- *Computational efficiency:* The adapted approaches are much more efficient than existing location privacy preserving schemes. For instance, as shown in Table I, LP-$Chor$ and LP-$Goldberg$ are more than 3 orders of magnitudes faster than the schemes proposed by Troja et al. [18], [19], and 10 times faster than XPIR [26] and $PriSpectrum$ [2].
- *Information Theoretical Privacy Guarantees:* They can achieve information-theoretic privacy which is the optimal privacy level that could be reached as opposed to computational privacy guarantees offered by existing approaches. In fact some of these approaches are prone to recent attacks on computational-PIR protocols [26] and are not secure against post-quantum adversaries [34].
- *Low communication overhead:* Our approaches incur a reasonable communication overhead that is a middle ground between the fastest computational PIR [26] and the most communication efficient computational PIR [35].
- *Fault-Tolerance and Robustness*: Our proposed approaches are resilient to the issues that are associated with multi-server architectures: failures, byzantine behavior, and collusion. Even though the collusion of all of the service providers is unlikely to happen due to the competing nature of these companies and due to regulatory enforcement from bodies such as FCC to protect users' data, we have however considered collusion in our system and security model. All proposed approaches can handle collusion of multiple DBs up to certain limit that is different for each approach. In addition, some of the proposed approaches can also handle faulty and byzantine DBs. Besides, simply hacking DBs, when the proposed approaches are in place, will not be sufficient to learn users' information since some of these protocols offer hybrid privacy protection by combining both computational and information-theoretic PIR protocols enabling them to offer computational privacy even when all of the DBs are compromised.
- *Experimental evaluation on actual cloud platforms*: We deploy our proposed approaches on a real cloud platform, GENI [36], to show their feasibility. In our experiment, we create multiple geographically distributed VMs each playing the role of a DB. A laptop plays the role of a SU that queries DBs, i.e. VM s. Our experiments confirm the superior computational advantages of the adoption of multi-server PIR over the existing alternatives.

D. Differences Compared to the Preliminary Version

The main differences between this paper and its preliminary versions [37], [38] are as follows: (i) We further consider the location privacy issue of mobile SUs and offer a way to amortize the cost incurred by mobility. (ii) We also leverage multi-server PIR to address the location privacy issue of PUs in database-CRN systems that require PUs to provide spectrum availability to DBs. (iii) We discuss also a way to reduce the cost of LP-$Chor$ by partitioning the spectrum database instead of simply replicating it using the RAID-PIR protocol [39] and we discuss the privacy-performance tradeoff of relying on such approach. (iv) We provide a more detailed performance evaluation that takes into account the latest advances in PIR technology, namely SealPIR [32] which relies on fully homomorphic encryption.

II. Preliminaries and Models

A. Notation and Building Blocks

We summarize our notations in Table II. Our adaptations of multi-server PIR rely on the following building blocks.

TABLE II: Notations

DB	Spectrum database
SU	Secondary user
CRN	Cognitive radio network
ℓ	Number of spectrum databases
$\boldsymbol{D}$	Matrix modeling the content of DB
r	Number of records in $\boldsymbol{D}$
n	Size of the database in bits
b	Size of one record of the database in bits
w	Size of one word of the database in bits
s	Number of words per block
β	Index of the record sought by SU
t	Privacy level (tolerated number of colluding DBs)
k	Number of responding DBs
ϑ	Number of byzantine DBs

Private Information Retrieval ($\boldsymbol{PIR}$): PIR allows a user to retrieve a data item of its choice from a database, while preventing the server owning the database from gaining information on the identity of the item being retrieved [40]. One trivial solution to this problem is to make the server send an entire copy of the database to the querying user. Obviously, this is a very inefficient solution to the PIR problem as its communication complexity may be prohibitively large. However, it is considered as the only protocol that can provide information-theoretic privacy, i.e. perfect privacy, to the user's query in single-server setting. There are two main classes of PIR protocols according to their privacy level: information-theoretic PIR ($itPIR$) and computational PIR ($cPIR$).

- *Information-theoretic or multi-server PIR:* It guarantees information-theoretic privacy to the user, i.e. privacy against computationally unbounded servers. This could be achieved efficiently only if the database is replicated at $k \geq 2$ non-communicating servers [24], [33]. The main idea behind these protocols consists on decomposing each user's query into several sub-queries to prevent leaking any information about the user's intent.
- *Computational or single-server PIR:* It guarantees privacy against computationally bounded server(s). In other words, a server cannot get any information about the identity of the item retrieved by the user unless it solves a certain computationally hard problem (e.g. prime factorization of large numbers), which is common in modern cryptography. Thus, they offer weaker privacy than their $itPIR$ counterparts [27], [41].

Shamir Secret Sharing: This is a concept introduced by Shamir et al. [42] to allow a secret holder to divide its secret $\mathcal{S}$ into ℓ shares $\mathcal{S}_1, \cdots, \mathcal{S}_\ell$ and distribute these shares to ℓ parties. In (t, ℓ)-Shamir secret sharing, where $t < \ell$, if t or fewer combine their shares, they learn no information about $\mathcal{S}$. However, if more than t come together, they can easily recover $\mathcal{S}$. Given a secret $\mathcal{S}$ chosen arbitrarily form a finite field, the (t, ℓ)-Shamir secret sharing scheme works as follows: the secret holder chooses ℓ arbitrary non-zero distinct elements $\alpha_1, \cdots, \alpha_\ell \in \mathbb{F}$. Then, it selects t elements $\sigma_1, \cdots, \sigma_t \in \mathbb{F}$ uniformly at random. Finally, the secret holder constructs the polynomial $f(x) = \sigma_0 + \sigma_1 x + \sigma_2 x^2 + \cdots + \sigma_t x^t$, where $\sigma_0 = \mathcal{S}$. The ℓ shares $\mathcal{S}_1, \cdots, \mathcal{S}_\ell$, that are given to each party, are $(\alpha_1, f(\alpha_1)), \cdots, (\alpha_\ell, f(\alpha_\ell))$. Any $t+1$ or more parties can recover the polynomial f using Lagrange interpolation and thus they can reconstruct the secret $\mathcal{S} = f(0)$. However, t or less parties can learn nothing about $\mathcal{S}$. In other words, if $t+1$ shares of $\mathcal{S}$ are available then $\mathcal{S}$ can be easily recovered.

B. System Model and Security Definitions

We consider a database-driven CRN that contains ℓ DBs, where $\ell \geq 2$, and a SU registered to these DBs to learn spectrum availability information in its vicinity. We assume that these DBs share the same content and that they are synchronized as mandated by PAWS standard [3]. We also assume that DBs may collude in order to infer SU's location. In the following, we present our security definitions.

Definition 1. Byzantine $\boldsymbol{DB}$: *This is a faulty DB that runs but produces incorrect answers, possibly chosen maliciously or computed in error. This might be due to a corrupted or obsolete copy of the database caused by a synchronization problem with the other DBs.*

Definition 2. $\boldsymbol{t}$-private $\boldsymbol{PIR}$: *The privacy of the query is information-theoretically protected, even if up to t of the ℓ DBs collude, where $0 < t < \ell$.*

Definition 3. $\boldsymbol{\vartheta}$-Byzantine-robust $\boldsymbol{PIR}$: *Even if ϑ of the responding DBs are* Byzantine*, SU can reconstruct the correct database item, and determine which of the DBs provided incorrect response.*

Definition 4. $\boldsymbol{k}$-out-of-$\boldsymbol{\ell}$ $\boldsymbol{PIR}$: *SU can reconstruct the correct record if it receives at least k-out-of-ℓ responses, $2 \leq k \leq \ell$.*

Definition 5. Robust $\boldsymbol{PIR}$: *It can deal with DBs that do not respond to SU's queries and allows SU to reconstruct the correct output of the queries in this situation.*

Definition 6. $\boldsymbol{\tau}$-independent $\boldsymbol{PIR}$: *The content of the database itself is information theoretically protected from the coalition of up to τ DBs, where $0 \leq \tau < k - t$.*

III. Proposed Approaches

In the proposed approaches, we tailor multi-server PIR to the context of multi-DB CRNs. We start by illustrating the structure of the spectrum database that we consider. Then, we give several approaches, each adapts a multi-server PIR protocol with different security, performance properties, and use cases. We model the content of each DB as an $r \times s$ matrix $\boldsymbol{D}$ of size n bits, where s is the number of words of size w in each record/block of the database and r is the number of records in the database, i.e. $r = n/b$, where $b = s \times w$ is the block size in bits. The k^{th} row of $\boldsymbol{D}$ is the k^{th} record of the database.

$$\boldsymbol{D} = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1s} \\ w_{21} & w_{22} & \dots & w_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ w_{r1} & w_{r2} & \dots & w_{rs} \end{bmatrix}$$

We further assume that each row of the database corresponds to a unique combination of the tuple (l_x, l_y, C, ts), where l_x and l_y represent one location's latitude and longitude, respectively, C is a channel number, and ts is a time-stamp. We also assume that SUs can associate their location information with the index β of the corresponding record of interest in the database using some inverted index technique that is agreed upon with DBs. An SU that wishes to retrieve record $\boldsymbol{D}_\beta$ without any privacy consideration can simply send to DB a row vector $\boldsymbol{e}_\beta$ consisting of all zeros except at position β where it has the value 1. Upon receiving $\boldsymbol{e}_\beta$, DB multiplies it with $\boldsymbol{D}$ and sends record $\boldsymbol{D}_\beta$ back to SU as we illustrate below:

$$[0 \quad \dots \quad 0 \quad 1 \quad 0 \quad \dots \quad 0] \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1s} \\ w_{21} & w_{22} & \dots & w_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ w_{r1} & w_{r2} & \dots & w_{rs} \end{bmatrix}$$

$$= [w_{\beta 1} \quad w_{\beta 2} \quad \dots \quad w_{\beta s}]$$

This trivial approach makes it easy for DBs to learn SU's location from the vector $\boldsymbol{e}_\beta$ as $\boldsymbol{D}$ is indexed based on location. In the following we present two approaches that try to hide the content of $\boldsymbol{e}_\beta$ from DBs, and thus preserve SU's location privacy. The approaches present a tradeoff between efficiency, and some additional security features.

A. Location Privacy with Chor (LP-Chor)

Our first approach, termed *LP-Chor*, harnesses the simple and efficient *itPIR* protocol proposed by Chor et al. [24]. We describe the different steps of *LP-Chor* in Algorithm 1 and highlight these steps in Fig. 1. Elements of $\boldsymbol{D}$ in this scheme belong to $GF(2)$, i.e. $w = 1$ bit and $b = s$.

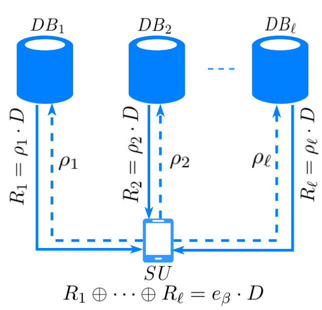


Fig. 1: Main steps of *LP-Chor* Algorithm

In *LP-Chor*, SU starts by invoking the inverted index subroutine $InvIndex(l_x, l_y, C, ts)$ which takes as input the coordinates of the user, its channel of interest, and a time-stamp and returns a value β. This value corresponds to the index of the record $\boldsymbol{D}_\beta$ of $\boldsymbol{D}$ that SU is interested in. SU then constructs $\boldsymbol{e}_\beta$, which is a standard basis vector $\overrightarrow{\mathbf{1}}_\beta \in \mathbb{Z}^r$ having 0 everywhere except at position β which has the value 1 as we discussed previously. SU also picks $\ell - 1$ r-bit binary strings $\boldsymbol{\rho}_1, \cdots, \boldsymbol{\rho}_{\ell-1}$ uniformly at random from $GF(2)^r$, and computes $\boldsymbol{\rho}_\ell = \boldsymbol{\rho}_1 \oplus \cdots \oplus \boldsymbol{e}_\beta$. Finally, SU sends $\boldsymbol{\rho}_i$ to DB_i, for $1 \leq i \leq \ell$. Upon receiving the bit-string $\boldsymbol{\rho}_i = \boldsymbol{\rho}_{i1} \oplus \cdots \boldsymbol{\rho}_{ir}$ of length r, DB_i computes $\boldsymbol{R}_i = \boldsymbol{\rho}_i \cdot \boldsymbol{D}$, which could be seen also as the XOR of those blocks $\boldsymbol{D}_j$ in $\boldsymbol{D}$ for which the j^{th} bit of $\boldsymbol{\rho}_i$ is 1, then sends $\boldsymbol{R}_i$ back to SU. SU receives $\boldsymbol{R}_i$s from DB_is, $1 \leq i \leq \ell$, and computes $\boldsymbol{R}_1 \oplus \cdots \oplus \boldsymbol{R}_\ell = (\boldsymbol{\rho}_1 \oplus \cdots \oplus \boldsymbol{\rho}_\ell) \cdot \boldsymbol{D} = \boldsymbol{e}_\beta \cdot \boldsymbol{D}$, which is the β^{th} block of the database that SU is interested in, from which it can retrieve the spectrum availability information.

Algorithm 1 $\boldsymbol{D}_\beta \leftarrow$*LP-Chor*$(\ell,\ r,\ b)$

SU

$\beta \leftarrow InvIndex(l_x, l_y, C, ts)$
Sets standard basis vector $\boldsymbol{e}_\beta \leftarrow \overrightarrow{1}_\beta \in \mathbb{Z}^r$
Generates $\boldsymbol{\rho}_1, \cdots, \boldsymbol{\rho}_{\ell-1} \in_R GF(2)^r$
$\boldsymbol{\rho}_\ell \leftarrow \boldsymbol{\rho}_1 \oplus \cdots \oplus \boldsymbol{e}_\beta$
Sends $\boldsymbol{\rho}_i$ to DB_i, for $1 \leq i \leq \ell$

Each $\boldsymbol{DB_i}$

Receives $\boldsymbol{\rho}_i = \boldsymbol{\rho}_{i1} \cdots \boldsymbol{\rho}_{ir} \in \{0,1\}^r$
$\boldsymbol{R}_i \leftarrow \bigoplus_{\substack{1 \leq j \leq r \\ \boldsymbol{\rho}_{ij} = 1}} \boldsymbol{D}_j$, $\boldsymbol{D}_j$ is the j^{th} block of $\boldsymbol{D}$
Sends $\boldsymbol{R}_i$ to SU

SU

Receives $\boldsymbol{R}_1, \cdots, \boldsymbol{R}_\ell$
$\boldsymbol{D}_\beta \leftarrow \boldsymbol{R}_1 \oplus \cdots \oplus \boldsymbol{R}_\ell$

LP-Chor is very efficient thanks to its reliance on simple XOR operations only as we discuss in Section IV. It is also $(\ell - 1)$-*private*, by Definition 2, as collusion of up to $\ell - 1$ DBs cannot enable them to learn $\boldsymbol{e}_\beta$, and consequently its location. In fact, only if ℓ DBs collude, then they will be able to learn $\boldsymbol{e}_\beta$ by simply XORing their $\{\boldsymbol{\rho}_i\}_{i=1}^{\ell}$. However this approach suffers from two main drawbacks. First, it is not *robust* since even if one DB fails to respond, SU will not be able to recover $\boldsymbol{D}_\beta$. Second, it is not *byzantine robust*; if one or more DBs return a wrong response, SU will reconstruct a wrong block and also will not be able to recognize which DB misbehaved so as not to rely on it for future queries. In Section III-B we discuss a second approach that improves on these two aspects but with some additional overhead.

B. Location Privacy with Goldberg (LP-Goldberg)

Our second approach, termed *LP-Goldberg*, is based on Goldberg's *itPIR* protocol [33] which uses Shamir secret sharing to hide $\boldsymbol{e}_\beta$, i.e. SU's query. It is a modification of Chor's scheme [24] to achieve both *robustness* and *byzantine robustness*. Rather than working over $GF(2)$ (binary arithmetic), this scheme works over a larger field $\mathbb{F}$, where each element can represent w bits. The database $\boldsymbol{D} = (w_{jk}) \in \mathbb{F}^{r \times s}$ in this scheme, is an $r \times s$ matrix of elements of $\mathbb{F} = GF(2^w)$. Each row represents one block of size b bits, consisting of s words of w bits each. Again, $\boldsymbol{D}$ is replicated among ℓ databases DB_i. We summarize the main steps of *LP-Goldberg* protocol in Algorithm 2 and illustrate them in Fig. 2.

To determine the index β of the record that corresponds to its location, SU starts by invoking the subroutine $InvIndex(l_x, l_y, C, ts)$ then constructs the standard basis vector $\boldsymbol{e}_\beta \in \mathbb{F}^r$ as explained earlier. SU then uses (ℓ, t)-Shamir secret sharing to divide the vector $\boldsymbol{e}_\beta$ into ℓ independent shares $(\alpha_1, , \boldsymbol{\rho}_1) \cdots, (\alpha_\ell, \boldsymbol{\rho}_\ell)$ to ensure a t-*private PIR* protocol as in Definition 2. That is, SU chooses ℓ distinct non-zero elements $\alpha_i \in \mathbb{F}^*$ and creates r random degree-t polynomials $f_1, \cdots, f_r$ satisfying $f_j(0) = \boldsymbol{e}_\beta[j]$. SU then sends to each DB_i its share corresponding to the vector $\boldsymbol{\rho}_i = \langle f_1(\alpha_i), \cdots, f_r(\alpha_i)\rangle$. Each DB_i then computes the product

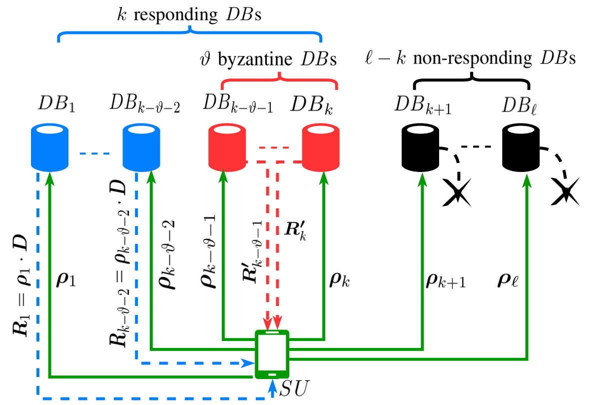


Fig. 2: Illustration of *LP-Goldberg*

$\boldsymbol{R}_i = \boldsymbol{\rho}_i \cdot \boldsymbol{D} = \langle \sum_j f_j(\alpha_i)\boldsymbol{w}_{j1}, \cdots, \sum_j f_j(\alpha_i)\boldsymbol{w}_{js} \rangle \in \mathbb{F}^s$ and sends $\boldsymbol{R}_i$ to SU.

Some DBs may fail to respond to SU's query and only k-out-of-ℓ send their responses to SU. SU collects k responses from the k responding DBs and tries to recover the record at index β from the $\boldsymbol{R}_i$s by using the EASYRECOVER() subroutine from [33] which uses Lagrange interpolation to recover $\boldsymbol{D}_\beta$ from the secret shares $(\alpha_1, \boldsymbol{R}_1), \cdots, (\alpha_k, \boldsymbol{R}_k)$. This is possible thanks to the use of (ℓ, t)-Shamir secret sharing as long as $k > t$ and these k DBs are honest. In fact, by the linearity property of Shamir secret sharing, since $\{(\alpha_i, \boldsymbol{\rho}_i)\}_{i=1}^{\ell}$ is a set of (ℓ, t)-Shamir secret shares of $\boldsymbol{e}_\beta$, then $\{(\alpha_i, \boldsymbol{R}_i)\}_{i=1}^{\ell}$ will be also a set of (ℓ, t)-Shamir secret shares of $\boldsymbol{e}_\beta \cdot \boldsymbol{D}$, which is the β^{th} block of the database. Thus, it is possible for SU to reconstruct $\boldsymbol{D}_\beta$ using Lagrange interpolation as explained in Section II, by relying only on the k responses which makes *LP-Goldberg* robust by Definition 5. Also, the EASYRECOVER can detect the DBs that responded honestly, thus those that are byzantine as well, which should discourage DBs from misbehaving. More details about this subroutine could be found in [33].

Moreover, ϑ DBs among the k responding ones may even be *byzantine*, as in Definition 1, and produce incorrect response. In that case, it would be impossible for SU to simply rely on Lagrange interpolation to recover the correct responses. Since Shamir secret sharing is based on polynomial interpolation, the problem of recovering the response in the case of *byzantine* failures corresponds to noisy polynomial reconstruction, which is exactly the problem of decoding Reed-Solomon codes [43]. Thus, SU would rather rely on error correction codes and more precisely on the Guruswami-Sudan list decoding [44] algorithm which can correct $\vartheta < k - \lfloor \sqrt{kt} \rfloor$ incorrect responses. In fact, the vector $\langle \boldsymbol{R}_1[q], \boldsymbol{R}_2[q], \cdots, \boldsymbol{R}_\ell[q] \rangle$ is a Reed-Solomon code-word encoding the polynomial $g_q = \sum_j f_j \boldsymbol{w}_{jq}$, and the client wishes to compute $g_q(0)$ for each $1 \leq q \leq s$ to recover all the s words forming the record $\boldsymbol{D}_\beta = \langle g_1(0), \cdots, g_s(0) \rangle$. This is done through the HARDRECOVER() subroutine from [33]. This makes *LP-Goldberg* also *ϑ-Byzantine-robust*, by Definition 3, and solves the robustness issues that *LP-Chor* suffers from, however, this comes at the cost of an additional overhead as we discuss in Section IV.

Corollary 1. *LP-Chor and LP-Goldberg directly inherit the security properties of Chor's [24] PIR and Goldberg's [33] PIR respectively.*

Algorithm 2 $\boldsymbol{D}_\beta \leftarrow$ *LP-Goldberg*(ℓ, r, b, t, w)

SU
1: $\beta \leftarrow InvIndex(l_x, l_y, C, ts)$
2: Sets standard basis vector $\boldsymbol{e}_\beta \leftarrow \overrightarrow{1}_\beta \in \mathbb{Z}^r$
3: Chooses ℓ distinct $\alpha_1, \cdots, \alpha_\ell \in \mathbb{F}^*$
4: Creates r random degree-t polynomials $f_1, \cdots, f_r \in_R \mathbb{F}[x]$ s.t. $f_j(0) = \boldsymbol{e}_\beta[j]$' $\forall j \in [1, \cdots, r]$
5: $\boldsymbol{\rho}_i \leftarrow \langle f_1(\alpha_i), \cdots, f_r(\alpha_i) \rangle, \forall i \in [1, \cdots, \ell]$
6: Sends $\boldsymbol{\rho}_i$ to DB_i, $\forall i \in [1, \cdots, \ell]$

Each honest $\boldsymbol{DB_i}$
7: Receives $\boldsymbol{\rho}_i$
8: $\boldsymbol{R}_i \leftarrow \boldsymbol{\rho}_i \cdot \boldsymbol{D} = \langle \sum_j f_j(\alpha_i)\boldsymbol{w}_{j1}, \cdots, \sum_j f_j(\alpha_i)\boldsymbol{w}_{js} \rangle$
9: Sends $\boldsymbol{R}_i$ to SU

SU
10: Receives $\boldsymbol{R}_1, \cdots, \boldsymbol{R}_k$
11: **if** $k > t$ **then**
12:   **for** c from 1 to s **do**
13:     $\boldsymbol{R}_{ic} \leftarrow \boldsymbol{R}_i[c]\ \forall i \in [1, \cdots, k]$
14:     $S_c \leftarrow \langle \boldsymbol{R}_{1c}, \cdots, \boldsymbol{R}_{kc} \rangle$
15:     $\boldsymbol{D}_{\beta c} \leftarrow$ EASYRECOVER$(t, w, [\alpha_1, \cdots, \alpha_k], S_c)$
16:     **if** Recovery fails **and** $\vartheta < k - \lfloor \sqrt{kt} \rfloor$ **then**
17:       $S_c \leftarrow \langle \boldsymbol{R}_{1c}, \cdots, \boldsymbol{R}_{kc} \rangle$
18:       $\boldsymbol{D}_{\beta c} \leftarrow$ HARDRECOVER$(t, w, [\alpha_1, \cdots, \alpha_k], S_c)$

C. Location Privacy of Mobile SUs Through Batching

Thus far, we concerned only about non-mobile SUs that periodically submit an individual query to DBs to learn spectrum availability in their fixed location. However, things get more interesting with mobility. In fact, a mobile SU will need to query DBs multiple times as its location changes. While the previous two approaches perform well for non-mobile SUs, they will incur a significant overhead on both SU and DBs especially when SU is moving at a relatively high speed, which will require a large number of *PIR* queries.

Our third approach aims to protect the location privacy of mobile SUs while reducing the mobility-associated overhead. The idea is to exploit the fact that a mobile SU usually has an a priori knowledge of its trajectory to make it query DBs for its current and future locations by batching these queries together instead of sending them separately. We achieve this by relying on the $itPIR$ protocol of Lueks et al. [45] that extends the scheme of Goldberg [33] to support batching of the queries using fast matrix multplication mechanisms inspired from batch codes [46]. We refer to this approach as *LP-BatchPIR* and we describe it in the following.

Each DB_i that receives q simultaneous queries $\boldsymbol{\rho}_i^{(1)}, \cdots, \boldsymbol{\rho}_i^{(q)}$ from an SU can process them using *LP-Goldberg* by simply multiplying each query with $\boldsymbol{D}$ as illustrated in Step 8 of Algorithm 2. Alternatively, it can also group these queries into a matrix $\boldsymbol{Q}_i$ of size $q \times r$, where each row j corresponds to a query $\boldsymbol{\rho}_i^{(j)}$, before computing the matrix product $\boldsymbol{Q}_i \cdot \boldsymbol{D}$. The careful reader will notice that this naive multiplication method would cost around $2qrs$ operations (including multiplications and additions) which can be prohibitively expensive especially for a large $\boldsymbol{D}$ or q. This problem boils down to a fast matrix multiplication problem and therefore can benefit from fast matrix multiplication algorithms such as Strassen's [47].

Strassen's algorithm consists on simply dividing both matrices $\boldsymbol{Q}_i$ and $\boldsymbol{D}$ into four equally sized block matrices. Then instead of naively multiplying these submatrices, which will result in 8 submatrix multiplications (fundamentally equivalent to simple matrix multiplication), Strassen's algorithm creates linear combinations of blocks in a way that reduces the number of submatrix multiplications to 7. The exact approach is then applied recursively to the multiplications of the submatrices of the previous step. This simple yet powerful matrix multiplication technique will significantly reduce the overhead for DBs and therefore the delay that SUs experience to learn spectrum availability while moving as illustrated in Section IV.

A row j in the resulting matrix, $\boldsymbol{\mathcal{R}_i} = \boldsymbol{Q}_i \cdot \boldsymbol{D}$, corresponds to DB_i's response to the j^{th} query. SU will then recover the spectrum availability by combining same-index rows of the different $\boldsymbol{\mathcal{R}_i}$s as in $LP\text{-}Goldberg$.

D. Location Privacy of PUs

As we mentioned earlier, in database-driven CRNs, DBs' content comprises operational information of PUs which may be very sensitive in systems such as SAS in the 3.5 GHz CBRS band where PUs are military and governmental entities. The service providers use this operational data to feed their models and populate the spectrum databases with availability information but do not share the PUs' location information in response to SUs' queries. Therefore, SUs do not present a serious threat to PUs privacy as opposed to the service providers which could be malicious, and could misuse PUs' sensitive operational data.

In this subsection, we present another approach to take into account the privacy of these PUs as well. For this we make use of another extension of the Goldberg PIR scheme known as *τ-independence*, to prevent DBs from learning the content of $\boldsymbol{D}$ even if up to τ DBs collude to learn $\boldsymbol{D}$ as defined in Definition 6. This is achieved by making PUs populate the DBs with spectrum availability information pertaining to their respective channels instead of the service providers, by secretly sharing each record they want to add, among the different service providers using Shamir secret sharing techniques, similar to how SUs secretly share their queries. That way, each service provider will not be able to decode this data, and only SUs which have access to the secret can retrieve the record by combining the different shares from the different DBs. This is motivated by the fact that DBs are expected to be populated by PUs themselves as it is the case in LSA systems, or by a highly trusted independent entity, the ESC, as in SAS systems. Therefore, whenever a PU or an ESC submits a PU activity record of index j to DBs it will divide it into s words $W_{j1}, \cdots, W_{js}$ and distributes Shamir secret shares of every word among the ℓ DBs as reflected in Algorithm 3. Each DB_i will now have a different content $\boldsymbol{D}^{(i)}$:

$$\boldsymbol{D}^{(i)} = \begin{bmatrix} w_{11}^{(i)} & w_{12}^{(i)} & \dots & w_{1s}^{(i)} \\ w_{21}^{(i)} & w_{22}^{(i)} & \dots & w_{2s}^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{r1}^{(i)} & w_{r2}^{(i)} & \dots & w_{rs}^{(i)} \end{bmatrix}$$

where $\{w_{jc}^{(i)}\}_{1 \le i \le \ell}$ form a (τ, ℓ)-Shamir secret sharing of word W_{jc}. This requires that the random values α_is, used to create Shamir secret shares as explained in Section II-A, are shared beforehand among SUs and PUs. This could be done by FCC during the registration phase, for instance, and must not be communicated to DBs.

Algorithm 3 $\boldsymbol{D}_\beta \leftarrow \tau\text{-}LP\text{-}Goldberg(\ell, r, b, t, w)$

FCC

Chooses ℓ distinct $\alpha_1, \cdots, \alpha_\ell \in \mathbb{F}^*$.
Shares these α_is only with PUs and SUs.

PU

Divides its activity record j into s words $W_{j1}, \cdots, W_{js}$
Creates s random degree-τ polynomials $g_{j1}, \cdots, g_{js} \in_R \mathbb{F}[x]$ s.t. $g_{jc}(0) = W_{jc}$ $\forall c \in [1, \cdots, s]$
Sends $w_{jc}^{(i)} \leftarrow g_{jc}(\alpha_i)$ to DB_i, $\forall\, i \in [1, \cdots, \ell], \forall c \in [1, \cdots, s]$
DB_i adds j^{th} record formed by $w_{j1}^{(i)}, \cdots, w_{js}^{(i)}$ to $\boldsymbol{D}^{(i)}$

SU

$\beta \leftarrow InvIndex(l_x, l_y, C, ts)$
Sets standard basis vector $\boldsymbol{e}_\beta \leftarrow \overrightarrow{1}_\beta \in \mathbb{Z}^r$
Creates r random degree-t polynomials $f_1, \cdots, f_r \in_R \mathbb{F}[x]$ s.t. $f_j(0) = \boldsymbol{e}_\beta[j]$ $\forall j \in [1, \cdots, r]$
$\boldsymbol{\rho}_i \leftarrow \langle f_1(\alpha_i), \cdots, f_r(\alpha_i) \rangle$, $\forall i \in [1, \cdots, \ell]$
Sends $\boldsymbol{\rho}_i$ to DB_i, $\forall i \in [1, \cdots, \ell]$

Each honest $\boldsymbol{DB}_i$

Receives $\boldsymbol{\rho}_i$
$\boldsymbol{R}_i \leftarrow \boldsymbol{\rho}_i \cdot \boldsymbol{D}^{(i)} = \langle \sum_j f_j(\alpha_i) w_{j1}^{(i)}, \cdots, \sum_j f_j(\alpha_i) w_{js}^{(i)} \rangle$
Sends $\boldsymbol{R}_i$ to SU

SU

Receives $\boldsymbol{R}_1, \cdots, \boldsymbol{R}_k$
if $k > t + \tau$ **then**
  for c from 1 to s **do**
    $\boldsymbol{R}_{ic} \leftarrow \boldsymbol{R}_i[c]$ $\forall i \in [1, \cdots, k]$
    $S_c \leftarrow \langle \boldsymbol{R}_{1c}, \cdots, \boldsymbol{R}_{kc} \rangle$
    $\boldsymbol{D}_{\beta c} \leftarrow \textsc{EasyRecover}(t, w, [\alpha_1, \cdots, \alpha_k], S_c)$
    if Recovery fails **and** $\vartheta < k - \lfloor \sqrt{k(t+\tau)} \rfloor$ **then**
      $S_c \leftarrow \langle \boldsymbol{R}_{1c}, \cdots, \boldsymbol{R}_{kc} \rangle$
      $\boldsymbol{D}_{\beta c} \leftarrow \textsc{HardRecover}(t, w, [\alpha_1, \cdots, \alpha_k], S_c)$

This way, records revealing operational data of PUs, which could be used by DBs to build knowledge of the activity of these PUs and track them, are information-theoretically protected from DBs as long as no more than τ of these DBs collude. However, for this protocol to work, this condition must hold: $0 < t \le t + \tau < k \le \ell$. While this extension of $LP\text{-}Goldberg$ should have no impact on the performance from SUs and DBs side as we show in Section IV, it has, however, an impact on the *t-privacy* of the protocol. In fact as the *τ-independence* level, controlling how many DBs can collude to learn the record submitted by PU, sought by PU increases, the maximum achievable *t-privacy level* will decrease since $t + \tau < k$ must always hold.

E. Location Privacy of SUs in Partitioned-database CRNs

In this section, we present another location privacy-preserving approach for SUs in the case where the spectrum database content is distributed among the different DBs instead of simply replicating it as in the previous approaches. This could be motivated by the fact that some database-driven CRNs may have multiple DBs covering different or slightly overlapping regions. It could also be a way to reduce cost by making each DB manage a portion of the database.

For that we rely on the RAID-PIR protocol due to Demmler et al. [39] which builds on Chor's scheme to reduce the communication overhead and the computation required at the server side. The idea here is very similar to that of Chor's but here the vector $\boldsymbol{e}_\beta$ is divided into ℓ chunks. Each query q_i sent to DB_i is divided into π chunks as illustrated in Figure 3, where π is a redundancy parameter that controls the minimum number of DBs that need to collude to recover the record $\boldsymbol{D}_\beta$ with $2 \leq \pi \leq \ell$. This parameter also controls the number of chunks in every query and how often the chunks overlap throughout these queries [39].

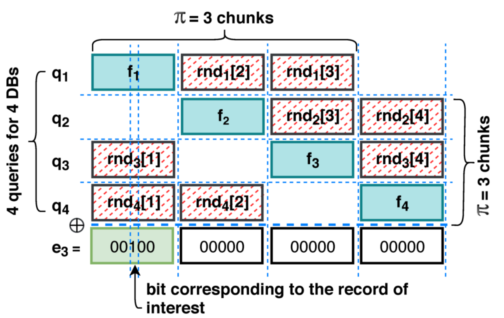


Fig. 3: RAID-PIR [39]

The details of this approach are described in Algorithm 4. To optimize the cost, SU can use a pseudo random generator, PRG, to generate the $\pi - 1$ chunks of q_i as illustrated in Algorithm 4. For that, SU randomly generates ℓ seeds $s_1, \cdots, s_\ell$ of size κ bits each, where κ is the symmetric security parameter, and expands each seed s_i into $\pi - 1$ random chunks $rnd_i[j]$, using PRG, each of size $\frac{r}{\ell}$ as depicted in step 4 of Algorithm 4. The first chunk of query q_i, denoted as f_i, is computed to cancel out the $\pi - 1$ other i^{th} chunks $rnd_i[j]$ of each of the other DBs, if applicable, and is obtained by xoring those $\pi - 1$ chunks with the i^{th} chunk of $\boldsymbol{e}_\beta$. Thanks to the use of the PRG, SU does not need to send the whole query and needs only to send a compacted version of q_i, denoted as q'_i, composed of f_i and the seed s_i, used to generate the other chunks of the full query q_i, to DB_i. Then, DB_i will use the same pseudo-random generator, PRG, with the seed that it received to generate the full query q_i. Once q_i recovered, DB_i will construct its answer $\boldsymbol{R}_i$ by xoring the records in $\boldsymbol{D}$ whose indices match those of the set bits in q_i. Finally, SU needs only to xor the results from the different DBs to recover the β^{th} record.

As the size of the query q_i is just $\pi/\ell \cdot r$, each DB now needs to store and process only $\pi/\ell \cdot r$ records of $\boldsymbol{D}$ which will be beneficial to DBs especially if the number of these databases increases.

IV. Evaluation and Analysis

A. Analytical Comparison

We start by studying the proposed approaches' performance analytically and we compare them to existing approaches. For LP-$Goldberg$, we choose $w = 8$ to simplify the

Algorithm 4 $\boldsymbol{D}_\beta \leftarrow RAID\text{-}LP\text{-}Chor(\ell,\ r,\ b)$

SU
1: $\beta \leftarrow InvIndex(l_x, l_y, C, ts)$
2: Sets standard basis vector $\boldsymbol{e}_\beta \leftarrow \overrightarrow{1}_\beta \in \mathbb{Z}^r$
3: Picks ℓ seeds $s_i \in_R \{0,1\}^\kappa$
4: Expands s_i to $\pi - 1$ chunks $rnd_i[j] \leftarrow PRG(s_i, j)\ \forall j \in [(i \bmod \ell) + 1, (i + \pi - 2 \bmod \ell) + 1]$, $\forall i \in [1, \ell]$
5: $f_i \leftarrow \bigoplus_j rnd_j[i]$, $j = (i - 1 \bmod \ell) + 1, (i - 2 \bmod \ell + 1), \cdots$
6: $f_i \leftarrow \boldsymbol{e}_\beta \oplus f_i\ \forall i \in [1, \ell]$
7: Sends q'_i consisting of chunk f_i and seed s_i to DB_i

Each DB_i
8: Expands its received s_i as in Step 4 to get full query q_i
9: $\boldsymbol{R}_i \leftarrow \bigoplus_{\substack{1 \leq j \leq r \\ q_{ij}=1}} \boldsymbol{D}_j$, $\boldsymbol{D}_j$ is the j^{th} record of $\boldsymbol{D}$
10: Sends $\boldsymbol{R}_i$ to SU

SU
11: Receives $\boldsymbol{R}_1, \cdots, \boldsymbol{R}_\ell$
12: $\boldsymbol{D}_\beta \leftarrow \boldsymbol{R}_1 \oplus \cdots \oplus \boldsymbol{R}_\ell$

cost of computations as in [43]; since in $GF(2^8)$, additions are XOR operations on bytes and multiplications are lookup operations into a 64 KB table [43]. We summarize the system communication complexity and the computation incurred by both DB and SU and we illustrate the difference in architecture and privacy level of the different approaches in Table III. As we mentioned earlier, existing research focuses on the single DB setting. We compare the proposed approaches to existent techniques despite the difference of architecture to show the great benefits that multi-server PIR brings in terms of performance and privacy as we discuss next. We briefly discuss these approaches in the following.

Gao et al. [2] propose a PIR-based approach, termed $PriSpectrum$, that relies on the PIR scheme of Trostle et al. [27] to defend against the new attack that they identify. This new attack exploits spectrum utilization pattern to localize SUs. Troja et al. [18], [19] propose two other PIR-based approaches that try to minimize the number of PIR queries by either allowing SUs to share their availability information with other SUs [18] or by exploiting trajectory information to make SUs retrieve information for their current and future positions in the same query [19].

Despite their merit in providing location privacy to SUs these PIR-based approaches incur high overhead especially in terms of computation. This is due to the fact that they rely on $cPIR$ protocols to provide location privacy to SUs, which are known to suffer from expensive computational cost. In fact, answering an SU's query through a $cPIR$ protocol, requires DB to process all of its records, otherwise DB would learn that SU is not interested in them and would then learn partial information about the record $\boldsymbol{D}_\beta$, and consequently SU's location. This makes the computational cost of most $cPIR$ based location preserving schemes linear on the database size from DB side as we illustrate in Table III. Now this is not exclusive to $cPIR$ protocols as even $itPIR$ protocols may require processing all the records to guarantee privacy, however, the main difference with $cPIR$ protocols is that the latter have a very large cost per bit in the database, usually involving expensive group operations like multiplication modulo a large modulus [26] as opposed to multi-server $itPIR$ protocols. This could be seen clearly in Table III as both $LP\text{-}Chor$ and $LP\text{-}Goldberg$ require DB to perform a very efficient XOR operation per bit of the database. The same applies to the overhead incurred by SU which only performs XOR operations in both $LP\text{-}Chor$ and $LP\text{-}Goldberg$, while performing expensive modular multiplications and even exponentiations over large primes in the $cPIR$-based approaches.

In terms of communication overhead, the proposed approaches incur a cost that is linear in the number of records r and their size b. As an optimal choice of these parameters is usually $r = b = \sqrt{n}$ [24], [26], [33], [43] then this cost could be seen as $\mathcal{O}(\sqrt{nw})$ to retrieve a record of size $\sqrt{nw}$ bits, which is a reasonable cost for an information theoretic privacy.

Moreover, as illustrated in Table III, existent approaches fail to provide information theoretic privacy as the underlying security relies on computational PIR schemes. The only approaches that provide information theoretic location privacy are $LP\text{-}Chor$, $LP\text{-}Goldberg$, and $RAID\text{-}LP\text{-}Chor$ which are *$(\ell-1)$-private*, *t-private*, and *$(\pi-1)$-private* respectively, by Definition 2. It is worth mentioning that $PriSpectrum$ [2] relies on the well-known $cPIR$ of Trostle et al. [27] representing the state-of-the-art in efficient $cPIR$. However, this $cPIR$ scheme has been broken [26], [48]. Since the security of $PriSpectrum$ follows that of Trostle et al. [27] broken $cPIR$, then $PriSpectrum$ fails to provide the privacy objective that it was designed for. However, we include it in our performance analysis for completeness.

B. Experimental Evaluation

We further evaluate the performance of the proposed schemes experimentally to confirm the analytical observations.

Hardware setting and configuration. We have deployed the proposed approaches on GENI [36] cloud platform using the percy++ library [49]. We have created 6 virtual machines (VMs), each playing the role of a DB and they all share the same copy of $\boldsymbol{D}$. We deploy these GENI VMs in different locations in the US to count for the network delay and make our experiment closer to the real case scenario where spectrum service providers are located in different locations. These VMs are running Ubuntu 14.04, each having 8 GB of RAM, 15 GB SSD, and 4 vCPUs, Intel Xeon X5650 2.67 GHz or Intel Xeon E5-2450 2.10 GHz. To assess the SU overhead we use a Lenovo Yoga 3 Pro laptop with 8 GB RAM running Ubuntu 16.10 with an Intel Core m Processor 5Y70 CPU 1.10 GHz. The client laptop communicates with the remote VMs through ssh tunnels. We are also aware of the advances in $cPIR$ technology, and more precisely the fastest $cPIR$ protocols in the literature: XPIR which is proposed by Aguilar et al. [26] and SealPIR due to Angel et al. [32]. We include these protocols in our experiment to illustrate how multi-server PIR performs against the best known $cPIR$ schemes if they are to be deployed in CRNs. We use the available implementation of these protocols provided in [50] and [51] and we deploy their server components on a remote GENI VM while the client component is deployed on the Lenovo Yoga 3 Pro laptop.

Dataset. Spectrum service providers (e.g. Google, Microsoft, etc) offer graphical web interfaces and APIs to interact with their databases allowing to retrieve basic spectrum availability information for a user-specified location. Access to full data from real spectrum databases was not possible, thus, we generated random data for our experiment. The generated data consists of a matrix that models the content of the database, $\boldsymbol{D}$, with a fixed block size $b = 560$ B while varying the number of records r. The value of b is estimated based on the public raw data provided by FCC [52] on a daily basis and which service providers use to populate their spectrum databases.

Results and Comparison. We first measure the query end-to-end delay of the proposed approaches and plot the results in Fig. 4. We also include the delay introduced by the existing schemes based on our estimation of the operations included in Table III. The end-to-end delay that we measure takes into consideration the time needed by SU to generate the query, the network delay, the time needed by DB to process the query, and finally the time needed by SU to extract the β^{th} record of the database. We consider two different internet speed configurations in our experiment. We first rely on a high-speed internet connection of $80Mbps$ on the download and $30Mbps$ on the upload for all compared approaches. Then we use a low-speed internet connection of $1Mbps$ on the upload and download to assess the impact of the bandwidth on $LP\text{-}Chor$ and $LP\text{-}Goldberg$, and also on XPIR as well.

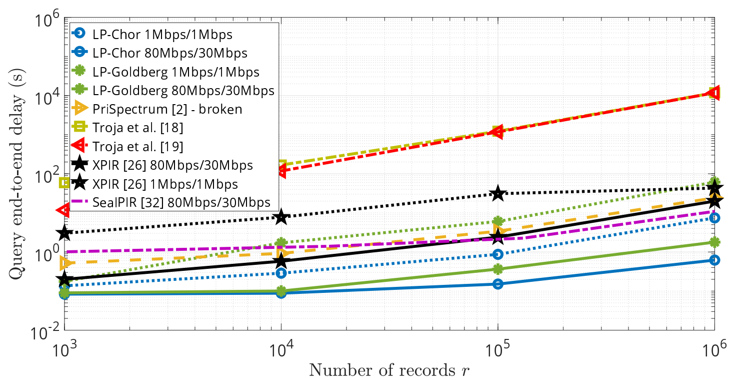


Fig. 4: Query RTT of the different PIR-based approaches

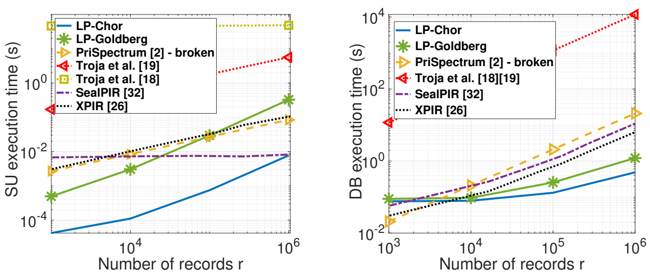
Fig. 4 shows that the proposed schemes perform much better than the existing approaches in terms of delay even with low-speed internet connection. They also perform better than the fastest existing $cPIR$ protocols XPIR and SealPIR. This shows the benefit of relying on multi-server $itPIR$ in multi-DB CRNs. Also, and as expected, $LP\text{-}Chor$ scheme performs better than $LP\text{-}Goldberg$ thanks to its simplicity. As we will see later, $LP\text{-}Goldberg$ also incurs larger communication overhead than $LP\text{-}Chor$ as well. This could be acceptable knowing that $LP\text{-}Goldberg$ can handle collusion of up-to ℓ DBs, and is robust in the case of $(\ell - k)$ non-responding DBs, and ϑ byzantine DBs, as opposed to $LP\text{-}Chor$. This means that $LP\text{-}Goldberg$ could be more suitable to real world scenario as failures and byzantine behaviors are common in reality. Fig. 4 also shows that the network bandwidth has a significant impact on the end-to-end latency. This is due to the relatively large amount of data that needs to be exchanged during the execution of these protocols which requires higher internet speeds.

We also compare the computational complexity experienced by each SU and DB separately in the different approaches as shown in Table III. We further illustrate this

TABLE III: Comparison with existent schemes

Scheme	Communication	Computation		Setting	Privacy
		DB	***SU***		
LP-Chor	$(r+b)\cdot\ell$	$nt_\oplus$	$(r+b)\cdot((\ell-1)\cdot t_\oplus)$	ℓ *DBs*	$(\ell-1)$*-private*
LP-Goldberg	$r\cdot w\cdot\ell+k\cdot b$	$(n/w)\cdot t_\oplus$	$\ell\cdot(\ell-1)\cdot rt_\oplus+3\ell\cdot(\ell+1)t_\oplus$	ℓ *DBs*	*t-private* ℓ*-comp.-private*
RAID-LP-Chor	$r+\ell\cdot\kappa+\ell\cdot b$	$(\pi/\ell)\cdot nt_\oplus$	$(r\cdot(\pi-1)+b\cdot(\ell-1))t_\oplus$	ℓ *DB*	$(\pi-1)$*-private*
PriSpectrum [2]	$(2\sqrt{r}+3)\cdot\lceil\log p\rceil$	$\mathcal{O}(r)\cdot Mulp$	$4\sqrt{r}\cdot Mulp$	1 *DB*	*underlying PIR broken*
Troja et al [19]	$12\delta\cdot b$	$\mathcal{O}(n)\cdot Mulp$	$4\sqrt{n}\cdot Mulp$	1 *DB*	*computationally-private*
Troja et al [18]	$n_g\cdot\psi\cdot\log_2 q+(2\sqrt{n}+3)\cdot\lceil\log p\rceil$	$\mathcal{O}(n)\cdot Mulp$	$n_g\cdot\psi\cdot(2Expp+Mulp)+4\sqrt{n}\cdot Mulp$	1 *DB*	*computationally-private*
XPIR [26]	$\mathcal{O}(Nd\sqrt[d]{n})$	$2d\cdot(r/\alpha)\cdot(b/\ell_0)\cdot Mulp$	$d\cdot(r/\alpha)^{1/d}\cdot Enc+d\cdot\alpha\cdot b/\ell_0\cdot Dec$	1 *DB*	*computationally-private*
SealPIR [32]	$\mathcal{O}(Nd\lceil\sqrt[d]{n}/N\rceil)$	$\mathcal{O}(d\sqrt[d]{n})$	$d\cdot\mathcal{E}+(F^{d-1}+1)\cdot\mathcal{D}$	1 *DB*	*computationally-private*

Variables: $t_\oplus$ is the execution time of one XOR operation. p is a large prime, and $Mulp$ and $Expp$ are the execution time of performing one modular multiplication, and one modular exponentiation respectively. ψ denotes the number of bits that an SU shares with other SUs in [18], n_g is the number of SUs within a same group in [18]. δ is the number of DB segments in [19]. d is the recursion level, α is the aggregation level, $\mathcal{C}$ is the Ring-LWE ciphertext size, λ is the number of elements returned by DB, F is the expansion factor of the underlying cryptosystem, ℓ_0 is the number of bits absorbed in a cyphertext, all are used in [26]. (Enc, Dec) are respectively the encryption and decryption cost for Ring-LWE cryptosystem used in [26]. $(\mathcal{E}, \mathcal{D})$ are respectively the encryption and decryption cost for Fan-Vercauteren [53] cryptosystem used in [32]. N is the query size bound in XPIR and SealPIR and is typically is typically 2048 or 4096 based on recommended security parameters.



(a) SU Computation Overhead. (b) DB Computation Overhead.

Fig. 5: Computation Comparison

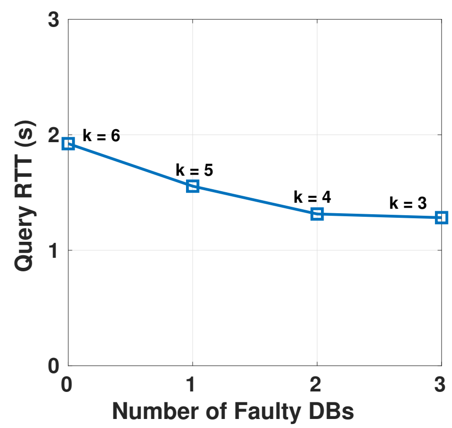
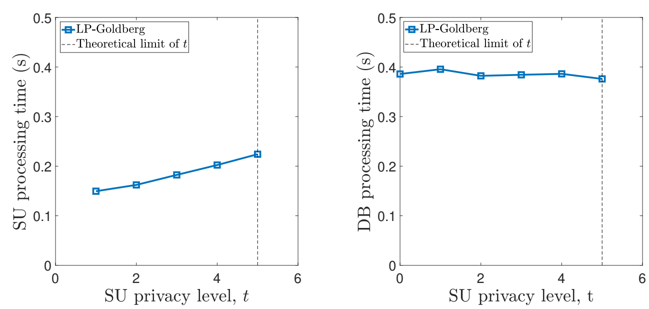


Fig. 6: Impact of the number of faulty DBs on the query RTT.



(a) SU Computation Overhead. (b) DB Computation Overhead.

Fig. 7: Impact of increasing query privacy level, t

through experimentation and we plot the results in Fig. 5a, which shows that the proposed schemes incur lower overhead on the SU than the existing approaches. The same observation applies to the computation experienced by each DB which again involves only efficient XOR operations in the proposed schemes. We illustrate this in Fig. 5b.

We also study the impact of non-responding DBs on the end-to-end delay experienced by the SU in *LP-Goldberg* as illustrated in Fig. 6. This Figure shows that as the number of faulty DBs increases, the end-to-end delay decreases since SU needs to process fewer shares to recover the record $\boldsymbol{D}_\beta$. As opposed to *LP-Chor*, in *LP-Goldberg*, SU is still able to recover the record β even if only k out-of-ℓ DBs respond. Please recall also that our experiment was performed on resource constrained VMs to emulate DBs, however in reality, DBs should have much more powerful computational resources than those of the used VMs which will have a tremendous impact on further reducing the overhead of the proposed approaches.

Figure 7 illustrates the impact of SU's desired privacy level in *LP-Goldberg* on the processing time incurred by both SU and DBs. As expected, increasing the value of t, which controls the number of DBs that can collude without inferring the content of the query, should not have any impact on each DB as they will always perform the same operations regardless of the privacy level. However, since the results sent by DBs could also be considered as a (t, ℓ)-Shamir secret sharing of the retrieved record, when t increases, then the number of secret shares required to recover the record increases which will result in more computation for the SU when performing Lagrange interpolation over higher degree-t polynomials.

We further study the impact of the number of byzantine DBs on the processing time on SU side in *LP-Goldberg* as depicted in Figure 8. As expected, having more byzantine DBs will increase the complexity of decoding the different shares, that SU receives from DBs, using the relatively expensive HARDRECOVER subroutine from [33].

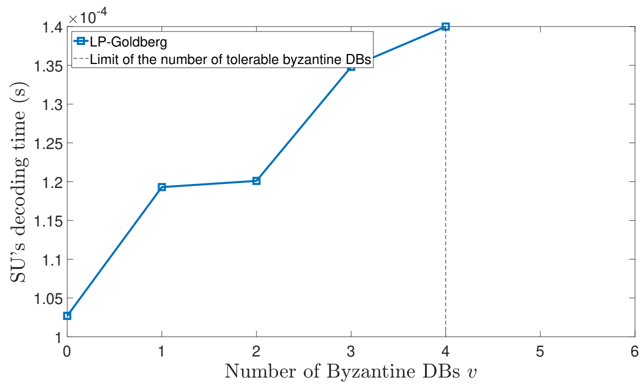


Fig. 8: Performance of $LP\text{-}Goldberg$ in the presence of byzantine DBs

As for τ-$LP\text{-}Goldberg$, the τ-*independence* extension will have no impact on the processing time of DBs and should also have no impact on SUs as long as $t+\tau$ is constant. This means that both PUs and SUs will always seek the maximum privacy levels for their data and queries such that $t+\tau<k$. This is reflected in Figure 9. However the processing time will be linear in $t+\tau$ similar to Figure 7a.

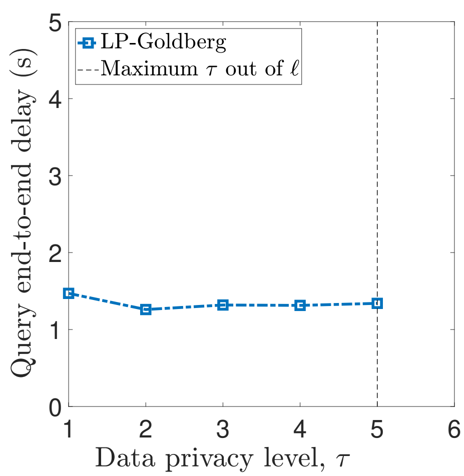


Fig. 9: Performance of τ-independent $LP\text{-}Goldberg$, with $k=\ell=6$ and $t+\tau<k$

As for the case of mobile SUs, we compare the performance of batching multiple queries for the future locations of a SU to that of sending separate consecutive queries using $LP\text{-}Goldberg$, SealPIRand,and XPIR as depicted in Figure 10. Using batching mainly reduces the computation on DBs side and will reduce the end-to-end delay for answering the queries of the moving SU.

We also demonstrate the benefit of relying on $RAID\text{-}LP\text{-}Chor$ and partitioning the database content among DBs, instead of simply replicating it, on the DBs' side for several values of the redundancy parameter π. As expected, $\pi=2$ yields the best performance however it also offers the lowest level of resistance to collusion. Setting π to be equal to ℓ will is equivalent to the original scheme $LP\text{-}Chor$ and will have the best performance. Therefore,

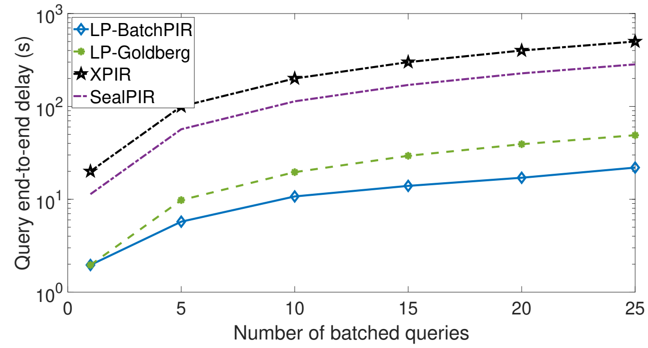


Fig. 10: Query RTT for a moving SU

$RAID\text{-}LP\text{-}Chor$ offers a performance-privacy tradeoff that is controlled by the redundancy parameter π.

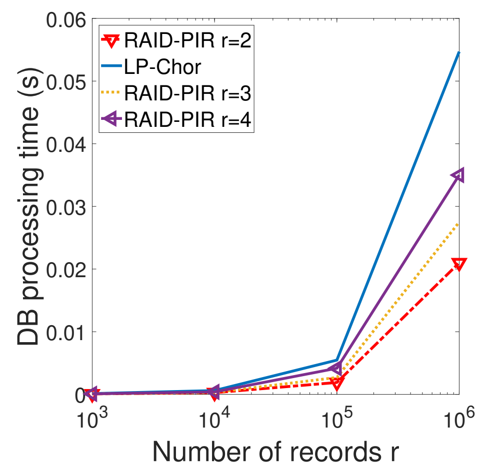


Fig. 11: DB's processing time under $RAID\text{-}LP\text{-}Chor$ compared to $LP\text{-}Chor$

In terms of communication overhead, most of the approaches, including ours, have linear cost in the number of records in the database as shown in Table III. What really makes a difference between these schemes' communication overheads is the associated constant factor which could be very large for some protocols. Based on our experiment and the expressions displayed in Table III, we plot in Fig. 12, the communication overhead that the CRN experiences for each private spectrum availability query issued by SU for the different schemes. The scheme with the lowest communication overhead is that of Troja et al. [19] especially for a large number of records thanks to the use of Gentry et al. PIR [35] which is the most communication efficient single-server protocol in the literature having a constant communication overhead. However this scheme is computationally expensive just like most of the existing $cPIR$-based approaches as we show in Fig. 4. $RAID\text{-}LP\text{-}Chor$ is the second best scheme in terms of communication overhead followed by$LP\text{-}Chor$, but they also provide information theoretic privacy. As shown in Figure 12, $RAID\text{-}LP\text{-}Chor$ is significantly more efficient than $LP\text{-}Chor$, which again shows the benefit, in terms of overhead, of distributing the spectrum availability information among multiple DBs. As shown in Fig. 12, $LP\text{-}Chor$ incurs much lower communication overhead than $LP\text{-}Goldberg$ thanks to the simplicity of the underlying Chor PIR protocol. However, as we discussed earlier, $LP\text{-}Goldberg$ provides additional security features compared to $LP\text{-}Chor$. SealPIR has a relatively high communication overhead especially for smaller database size but its overhead becomes comparable to that of $LP\text{-}Chor$ when the database's size gets larger as shown in Fig. 12. This could be a good alternative to the $cPIR$ schemes used in the context of CRNs especially that it introduces much lower latency which is critical in the context of CRNs. Still, the proposed approaches have better performance and also provide information-theoretic privacy to SUs, which shows their practicality in real world.

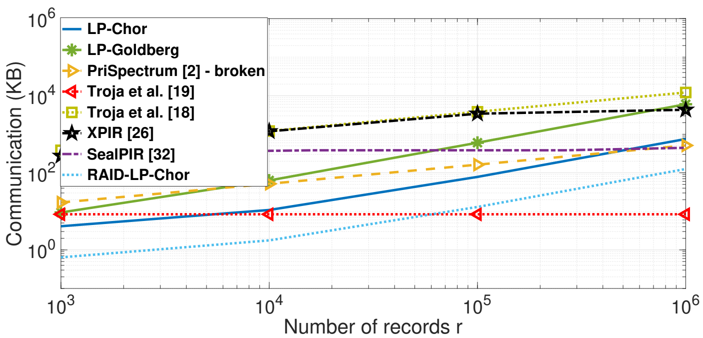


Fig. 12: Comparison of the communication overhead of the different approaches: $b = 560$ B, $k = \ell$, $\vartheta = 0$.

V. Related Work

There are other approaches that address the location privacy issue in database-driven CRNs. However, for the below mentioned reasons we decided not to consider them in our performance analysis. For instance, Zhang et al. [17] rely on the concept of *k-anonymity* to make each SU queries DB by sending a square cloak region that includes its actual location. *k-anonymity* guarantees that SU's location is indistinguishable among a set of k points. This could be achieved through the use of dummy locations by generating $k-1$ properly selected dummy points, and performing k queries to DB, using the real and dummy locations. Their approach relies on a tradeoff between providing high location privacy level and maximizing some utility. This makes it suffer from the fact that achieving a high location privacy level results in a decrease in spectrum utility. However, *k-anonymity*-based approaches cannot achieve high location privacy without incurring substantial communication/computation overhead. Furthermore, it has been shown in a recent study led by Sprint and Technicolor [25] that anonymization based techniques are not efficient in providing location privacy guarantees, and may even leak some location information. Grissa et al [21], [54] propose an information theoretic approach which could be considered as a variant of the trivial PIR solution. They achieve this by using set-membership probabilistic data structures/filters to compress the content of the database and send it to SU which then needs to try several combinations of channels and transmission parameters to check their existence in the data structure. However, LPDB is only suitable for situations where the structure of the database is known to SUs which is not always realistic. Also, LPDB relies on probabilistic data structures which makes it prone to false positives that can lead to erroneous spectrum availability decision and cause interference to PU's transmission. Zhang et al. [20] rely on the ϵ*-geo-indistinguishability* mechanism [55], derived from *differential privacy* to protect bilateral location privacy of both PUs and SUs, which is different from what we try to achieve in this paper. This mechanism helps SUs obfuscate their location, however, it introduces noise to SU's location which may impact the accuracy of the spectrum availability information retrieved.

VI. Conclusion

In this paper, with the key observation that database-driven CRNs contain multiple synchronized DBs having the same content, we harnessed multi-server PIR techniques to achieve an optimal location privacy for both SUs and PUs and for different use cases with high efficiency. Our analytical and experimental analysis indicates that our adaptation of multi-server PIR for database-driven CRNs achieve magnitudes of time faster end-to-end delay compared to the fastest state-of-the-art single-server PIR adaptation with an information theoretical privacy guarantee. Given the demonstrated benefits of multi-server PIR approaches without incurring any extra architectural overhead on database-driven CRNs, we hope this work will provide an incentive for the research community to consider this direction when designing location privacy preservation protocols for CRNs.

Acknowledgment

This work was supported in part by the US National Science Foundation under NSF awards CNS-1162296 and CNS-1652389

References

[1] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE personal comm.*, vol. 6, no. 4, pp. 13–18, 1999.

[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*, 2013, pp. 2751–2759.

[3] V. Chen, S. Das, L. Zhu, J. Malyar, and P. McCann, "Protocol to access white-space (paws) databases," Tech. Rep., 2015.

[4] "Google spectrum database," https://www.google.com/get/spectrumdatabase/, accessed: 2017-04-14.

[5] "iconectiv white spaces database," https://spectrum.iconectiv.com/main/home/, accessed: 2017-04-14.

[6] "Microsoft white spaces database," http://whitespaces.microsoftspectrum.com/, accessed: 2017-04-14.

[7] A. Mancuso, S. Probasco, and B. Patil, "Protocol to access white-space (paws) databases: Use cases and requirements," Tech. Rep., 2013.

[8] M. Massaro, "Next generation of radio spectrum management: Licensed shared access for 5g," *Telecommunications Policy*, vol. 41, no. 5-6, pp. 422–433, 2017.

[9] M. Grissa, B. Hamdaoui, and A. A. Yavuz, "Location privacy in cognitive radio networks: A survey," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, 2017.

[10] S. Li, H. Zhu, Z. Gao, X. Guan, K. Xing, and X. Shen, "Location privacy preservation in collaborative spectrum sensing," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 729–737.

[11] M. Grissa, A. A. Yavuz, and B. Hamdaoui, "Lpos: Location privacy for optimal sensing in cognitive radio networks," in *Global Communications Conference (GLOBECOM), 2015 IEEE*. IEEE, 2015.

[12] W. Wang and Q. Zhang, "Privacy-preserving collaborative spectrum sensing with multipleservice providers," *Wireless Communications, IEEE Transactions on*, 2015.

[13] M. Grissa, A. A. Yavuz, and B. Hamdaoui, "An efficient technique for protecting location privacy of cooperative spectrum sensing users," in *INFOCOM WKSHPS*. IEEE, 2016.

[14] ——, "Preserving the location privacy of secondary users in cooperative spectrum sensing," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 2, pp. 418–431, 2017.

[15] S. Liu, H. Zhu, R. Du, C. Chen, and X. Guan, "Location privacy preserving dynamic spectrum auction in cognitive radio network," in *ICDCS*. IEEE, 2013, pp. 256–265.

[16] W. Wang, Y. Chen, Q. Zhang, and T. Jiang, "A software-defined wireless networking enabled spectrum management architecture," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 33–39, 2016.

[17] L. Zhang, C. Fang, Y. Li, H. Zhu, and M. Dong, "Optimal strategies for defending location inference attack in database-driven crns," in *Communications (ICC), 2015 IEEE International Conference on*.

[18] E. Troja and S. Bakiras, "Leveraging p2p interactions for efficient location privacy in database-driven dynamic spectrum access," in *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2014.

[19] ——, "Efficient location privacy for moving clients in database-driven dynamic spectrum access," in *ICCCN*. IEEE, 2015.

[20] Z. Zhang, H. Zhang, S. He, and P. Cheng, "Achieving bilateral utility maximization and location privacy preservation in database-driven cognitive radio networks," in *MASS*. IEEE, 2015.

[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] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st international conference on Mobile systems, applications and services*. ACM, 2003, pp. 31–42.

[23] C. Dwork, "Differential privacy: A survey of results," in *International Conference on Theory and Applications of Models of Computation*. Springer, 2008, pp. 1–19.

[24] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," *J. ACM*, vol. 45, no. 6, pp. 965–981, Nov. 1998.

[25] H. Zang and J. Bolot, "Anonymization of location data does not work: A large-scale measurement study," in *Proc. of the 17th annual int'l conf. on Mobile computing and networking*. ACM, 2011, pp. 145–156.

[26] C. Aguilar-Melchor, J. Barrier, L. Fousse, and M.-O. Killijian, "Xpir: Private information retrieval for everyone," *Proceedings on Privacy Enhancing Technologies*, vol. 2, pp. 155–174, 2016.

[27] J. Trostle and A. Parrish, "Efficient computationally private information retrieval from anonymity or trapdoor groups," in *International Conference on Information Security*. Springer, 2010, pp. 114–128.

[28] "White space database administrator group database-to-database synchronization interoperability specification," FCC, Tech. Rep., 2012.

[29] F. (2012), "TVWS database system requirements and tests," https://transition.fcc.gov/oet/whitespace/guides/TVWS_Database_Tests4.doc.

[30] R. Ramjee, S. Roy, and K. Chintalapudi, "A critique of fcc's tv white space regulations," *GetMobile: Mobile Computing and Communications*, vol. 20, no. 1, pp. 20–25, 2016.

[31] "White space database administrators guide," https://www.fcc.gov/general/white-space-database-administrators-guide, FCC, accessed: 2017-04-14.

[32] S. Angel, H. Chen, K. Laine, and S. Setty, "Pir with compressed queries and amortized query processing," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 962–979.

[33] I. Goldberg, "Improving the robustness of private information retrieval," in *Security and Privacy, 2007. IEEE Symp. on*, pp. 131–148.

[34] L. Chen, S. Jordan, Y. Liu, D. Moody, R. Peralta, R. Perlner, and D. Smith-Tone, "Report on post-quantum cryptography. nistir 8105," 2016.

[35] C. Gentry and Z. Ramzan, "Single-database private information retrieval with constant communication rate," *Automata, Languages and Programming*, pp. 103–103, 2005.

[36] M. Berman, J. S. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, and I. Seskar, "Geni: A federated testbed for innovative network experiments," *Computer Networks*, vol. 61, no. 0, pp. 5 – 23, 2014, special issue on Future Internet Testbeds – Part I.

[37] M. Grissa, A. A. Yavuz, and B. Hamdaoui, "When the hammer meets the nail: Multi-server pir for database-driven crn with location privacy assurance," in *2017 IEEE Conference on Communications and Network Security (CNS)*, Oct 2017, pp. 1–9.

[38] M. Grissa, B. Hamdaoui, and A. A. Yavuz, "Unleashing the power of multi-server pir for enabling private access to spectrum databases," *IEEE Communications Magazine*, vol. 56, no. 12, pp. 171–177, 2018.

[39] D. Demmler, A. Herzberg, and T. Schneider, "Raid-pir: Practical multi-server pir," in *Proceedings of the 6th edition of the ACM Workshop on Cloud Computing Security*. ACM, 2014, pp. 45–56.

[40] A. Beimel and Y. Ishai, "Information-theoretic private information retrieval: A unified construction," in *International Colloquium on Automata, Languages, and Programming*. Springer, 2001, pp. 912–926.

[41] C. A. Melchor and P. Gaborit, "A fast private information retrieval protocol," in *ISIT 2008*. IEEE, pp. 1848–1852.

[42] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[43] C. Devet, I. Goldberg, and N. Heninger, "Optimally robust private information retrieval." in *USENIX Security Symp.*, 2012, pp. 269–283.

[44] V. Guruswami and M. Sudan, "Improved decoding of reed-solomon and algebraic-geometric codes," in *Foundations of Computer Science, 1998. Proceedings. 39th Annual Symposium on*. IEEE, 1998, pp. 28–37.

[45] W. Lueks and I. Goldberg, "Sublinear scaling for multi-client private information retrieval," in *International Conference on Financial Cryptography and Data Security*. Springer, 2015, pp. 168–186.

[46] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai, "Batch codes and their applications," in *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*. ACM, 2004, pp. 262–271.

[47] V. Strassen, "Gaussian elimination is not optimal," *Numerische mathematik*, vol. 13, no. 4, pp. 354–356, 1969.

[48] T. Lepoint and M. Tibouchi, "Cryptanalysis of a (somewhat) additively homomorphic encryption scheme used in pir," in *Int'l Conf. on Financial Cryptography and Data Security*. Springer, 2015, pp. 184–193.

[49] "Percy++ library," http://percy.sourceforge.net, accessed: 2017-04-14.

[50] "Xpir implementation," https://github.com/XPIR-team/XPIR, accessed: 2017-04-14.

[51] "Sealpir implementation," https://github.com/sga001/SealPIR, accessed: 2018-08-14.

[52] "Cdbs data," https://transition.fcc.gov/Bureaus/MB/Databases/cdbs/, accessed: 2017-04-20.

[53] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption." *IACR Cryptology ePrint Archive*, vol. 2012, p. 144, 2012.

[54] M. Grissa, A. A. Yavuz, and B. Hamdaoui, "Cuckoo filter-based location-privacy preservation in database-driven cognitive radio networks," in *Computer Networks and Information Security (WSCNIS), 2015 World Symposium on*. IEEE, 2015, pp. 1–7.

[55] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013, pp. 901–914.

Mohamed Grissa (S'15) received the Diploma of Engineering (with highest distinction) in telecommunication engineering from Ecole Superieure des Communications de Tunis (Sup'Com), Tunis, Tunisia, in 2011. He also received the M.S. degree (June 2015) and the Ph.D. degree (September 2018) both in electrical and computer engineering (ECE) from Oregon State University, Corvallis, OR, USA.

Before joining Oregon State University, he worked as a Value Added Services Engineer at Orange France Telecom Group from 2012 to 2013. His research interests include privacy and security in computer networks, cognitive radio networks, spectrum access systems, IoT, Blockchain, and eHealth systems.

Attila Altay Yavuz (M'11) is an Assistant Professor in the Department of Computer Science and Engineering, University of South Florida (2018). He was an Assistant Professor in the School of Electrical Engineering and Computer Science, Oregon State University (2014-2018). He was a member of the security and privacy research group at the Robert Bosch Research and Technology Center North America (2011- 2014). He received his PhD degree in Computer Science from North Carolina State University in August 2011. He received his MS degree in Computer Science from Bogazici University (2006) in Istanbul, Turkey. He is broadly interested in design, analysis and application of cryptographic tools and protocols to enhance the security of computer networks and systems. Attila Altay Yavuz is a recipient of NSF CAREER Award (2017). His research on privacy enhancing technologies (searchable encryption) and intra- vehicular network security are in the process of technology transfer with potential world-wide deployments. He has authored more than 40 research articles in top conferences and journals along with several patents. He is a member of IEEE and ACM.

Bechir Hamdaoui (S'02–M'05–SM'12) is a Professor in the School of Electrical Engineering and Computer Science at Oregon State University. He received the Diploma of Graduate Engineer (1997) from the National School of Engineers at Tunis, Tunisia. He also received M.S. degrees in both ECE (2002) and CS (2004), and the Ph.D. degree in ECE (2005) all from the University of Wisconsin-Madison. His research interests are in the general fields of computer networking, mobile computing, and wireless communication, with a current focus on cloud computing, data analytics, distributed optimization and control, internet of things, cognitive radio and dynamic spectrum access, and security and privacy. He has won several awards, including the ICC 2017 Best Paper Award, the IWCMC 2017 Best Paper Award, the 2016 EECS Outstanding Research Award, and the 2009 NSF CAREER Award. He currently serves as Associate Editor for IEEE Transactions on Mobile Computing and for IEEE Network. He also served as Associate Editor for IEEE Transactions on Wireless Communications (2013-2018), IEEE Transactions on Vehicular Technology (2009-2014), Wireless Communications and Mobile Computing Journal (2009-2016), and Journal of Computer Systems, Networks, and Communications (2007-2009). He served as the chair for the 2017 INFOCOM Demo/Posters program, the 2016 IEEE GLOBECOM Mobile and Wireless Networks symposium, the 2014 IEEE GLOBECOM Communications Theory symposium, the 2011 ACM MOBICOM's SRC program, and many other IEEE symposia and workshops, including ICC 2014, IWCMC 2009-2018, CTS 2012, and PERCOM 2009. He also served on technical program committees of many IEEE/ACM conferences, including INFOCOM, ICC, and GLOBECOM. He was selected and served as a Distinguished Lecturer for the IEEE Communication Society for 2016 and 2017. He is a Senior Member of IEEE, IEEE Computer Society, IEEE Communications Society, and IEEE Vehicular Technology Society.