PeDSS: Privacy Enhanced and Database-Driven Dynamic spectrum Sharing

He Li[†], Yaling Yang[†], Yanzhi Dou[†], Jung-Min (Jerry) Park[†], and Kui Ren[‡]
[†]Department of Electrical and Computer Engineering, Virginia Tech, USA
[‡] School of Computer Science and Engineering, Zhejiang University, P. R. China
[†]{heli, yyang8, yzdou, jungmin}@vt.edu; [‡]kuiren@zju.edu.cn

Abstract—**Database driven dynamic spectrum sharing is one of the most promising dynamic spectrum access (DSA) solution to address the spectrum scarcity issue. In such a database-driven DSA system, the centralized spectrum management infrastructure, called spectrum access system (SAS), collects sensitive operational data of both incumbent users (IUs) and secondary users (SUs), which makes privacy protection critical in this paradigm. However, the few existing solutions rely on online trusted third party, which requires extra infrastructure and brings the risk of single point failure. To address the shortcomings of existing solutions, we propose a privacy enhanced and database-driven dynamic spectrum sharing (PeDSS) framework in this paper, which preserves the privacy for both IUs and SUs in database-driven DSA systems without the need for online trusted third party. Privacy for both IUs and SUs are formally defined and analyzed, and experiment results show that SAS under PeDSS is able to handle a single spectrum request in 0.51 ms on average, which is three orders of magnitude faster than prior arts.**

Index Terms—**Privacy-preserving protocols; Dynamic spectrum access; Location privacy**

I. Introduction

Dynamic spectrum access (DSA) technique has been widely accepted as a key technique to address the spectrum scarcity issue. DSA enhances spectrum efficiency by allowing unlicensed secondary users (SUs) to opportunistically utilize the spectrum that is not used by licensed incumbent users (IUs).

It is implied from industrial and governmental acts that database driven dynamic spectrum sharing paradigm is one of the most promising and practical design for DSA. In this paradigm, each SU sends spectrum access system (SAS) their spectrum access request that includes their location, transmission power and antenna height. If the SU will not cause intolerable interference to IUs, it will get a valid spectrum license in return.

User privacy is one of the critical concerns for a successful DSA system design. For national security reasons, operational information of government IUs is often classified data. For example, the IUs in 3.5 GHz DSA band in the U.S. include military and fixed satellite service licensees [1]. These IUs' operational data is highly sensitive, and thus the IUs' operational security is essential. Meanwhile, civil users of SU devices also need protection of their location privacy. According to a 2013 Pew Research project [2], 86% of users surveyed had taken actions to hide their identities to avoid information collection online. Yet the centralized spectrum management entity, SAS, who collects information related to operational status from both IUs and SUs, is usually operated by commercial third parties [3] that are not necessarily trusted by either IUs or SUs. Thus, privacy concern has been hindering the development of DSA systems.

Existing works regarding IU and SU privacy protection can be divided into three types. The first type [4] attempts to protect IU privacy from curious SUs while assuming the SAS is trustworthy. These schemes cannot protect IU operational security from curious SAS. The second type [5][6][7] protects both IU and SU privacy by formulating IU and SU privacy protection problem as a secure multi-party computation problem and solves the problem using partially homomorphic cryptographic primitives. However, these schemes rely on the aid of an **online** trusted third party (OTTP), which brings additional overhead on maintenance. In addition, all users' privacy will be breached if this OTTP service is compromised. Furthermore, these schemes' average processing time per SU spectrum request is several seconds, which is not scalable when the arrival rate of spectrum request is high. In [8] a solution that addresses computational overhead is proposed, but the issue of SU privacy is neglected. The final type [9] applies differential privacy mechanism to protect IU and SU privacy. While they are much faster than the second type of approaches, their level of privacy protection for IU will quickly drop as more SU queries are serviced.

To solve the above problems of existing works, this paper presents PeDSS, a novel framework that protects both IU and SU privacy from untrusted SAS in database-driven DSA system. PeDSS leverages the homomorphic property of a proxy re-encryption scheme, called AFGH scheme, to eliminate the need for an online trusted third party and hence mitigates the single-point-of-failure issue in existing works. In addition, PeDSS successfully integrates differential privacy schemes with homomorphic encryption-based privacy protection, such that it can ensure IU privacy will not be compromised regardless of the number of SU queries while still achieving fast and scalable spectrum request service time.

The main contributions of this paper are summarized below.

1) We proposed a novel privacy enhancing framework for database driven spectrum sharing system, which is computationally efficient, provably secure, and free of online

This work was partially sponsored by NSF through grants 1547366, 1054697, 1265886, 1547241, 1563832, and 1642928, and by the industry affiliates of the Broadband Wireless Access & Applications Center.

trusted third party.

2) We rigorously define and analyze IU operational security for database driven system using standard security experiments.
3) We evaluate the performance of PeDSS based on experiments with real terrain data obtained from USGS and SRTM3, to demonstrate that PeDSS achieves advantage in communication and computation cost compared to prior art. The trade-off between spectrum allocation accuracy and privacy protection is also evaluated.

The rest of this paper is organized as follows. Section II introduces related works. Section III presents an overview of PeDSS and its security properties. Section IV presents features of cryptosystems used in PeDSS, and Section V presents technical details. Security definitions and analysis are presented in Section VI. Evaluations are provided in Section VII. Section VIII concludes the paper.

II. Related Work

Currently, it is identified that addressing the location privacy of IUs is still in its infancy and more still need to be done [10]. In [11], database inference attack is identified, where SUs may try to infer IUs' operational status based on their spectrum access result. In [12] IU data obfuscation techniques are proposed and in [8] a tailored secure MPC protocol is proposed. However, approaches proposed in [11], [12] and [8] only consider IU privacy.

In [4] a k-anonymity based approach is proposed to address both IU privacy and SU privacy. However, in its proposal, SAS are assumed to run the k-anonymity algorithm, so it cannot address IU privacy in the untrusted SAS scenario.

In [9], differential privacy is applied to preserve privacy for both IUs and SUs, and a utility maximization protocol is proposed. However, the system model in [9] implies that a single IU may receive multiple queries from different SUs, and the security level will decrease when multiple queries happen [13]. This issue is not investigated in [9].

In [5][6], efficient MPC protocols dedicated for centralized DSA are proposed to address both IU privacy and SU privacy, under protection zone model and exclusion zone mode respectively. However, as we have discussed in Section I, the design of introducing online trusted third party, the Key Distributor, is not favorable for military IUs, who do not tend to trust other parties and are not willing to participate in spectrum allocation routine of the system. In [7], an MPC protocol called "PISA" is proposed for the DSA system at UHF TV band, yet in PISA there is also an online trusted third party called "Semi-trusted Third Party" (STP). Although STP is claimed to be "semi-trusted", it can decrypt IU private data since it owns the corresponding private key.

III. Overview of PeDSS and Security properties

A. Overview of PeDSS

As shown in Figure. 1, there are four parties in a PeDSS system: (1) IUs, (2) a SAS server for spectrum management, (3) SUs, and (4) a Key Issuer for setting up keys for the system.

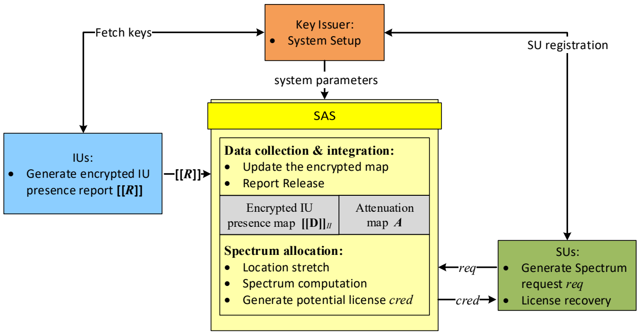


Fig. 1: PeDSS overview

There are two routines of this system: data collection and spectrum allocation. In data collection routine, an IU obtains the IU group public key from Key Issuer and uses the key to encrypt their raw data regarding the presence of IUs. SAS leverages the homomorphic property of underlying cryptosystem to integrate encrypted data reports to form an encrypted map on IU presence, denoted as $[\![\mathbf{D}]\!]_{II}$.

In spectrum allocation routine, an SU sends the spectrum request along with its location, antenna height, and maximum transmission power to SAS. When an SU sends this spectrum request $\texttt{req}$, it will intentionally introduce fuzziness in its location claim so that its location privacy can be protected. Upon receiving a spectrum request along with the fuzzy location, SAS computes the potential spectrum license $\texttt{cred}$, based on the encrypted IU presence map $[\![\mathbf{D}]\!]_{II}$ and the attenuation map A. The SU can manage to recover a valid license $\texttt{cred}^*$ by using its re-encryption key obtained from the Key Issuer. The recovered license is valid if and only if the potential interference does not exceed the interference sensitivity threshold of any IU.

B. Attack model

In this paper, we assume SAS as the adversary. We assume that the SAS is *honest but curious*, which is commonly used to characterize any general service provider. In particular, SAS is trusted to faithfully conduct all algorithms and follow the protocols faithfully, but it is also interested in learning SUs' locations and IUs' operational data.

We also consider outside attackers that focus on compromising a SAS administrator in order to dump all the data stored at SAS. This type of attackers is interested in learning SUs' locations and IUs' operational data.

In database driven DSA systems, there also exists inference attacks that attempts to break IU privacy through forming an adversarial network of SUs to gather spectrum allocation results. It can be thwarted by adding fuzziness to the spectrum allocation process as shown in existing works [11][9][4]. Discussions on such types of countermeasures are beyond the scope of this paper.

C. Security goals

In the following, we provide the security goals of PeDSS. Formal definitions will be presents in section VI.

Correctness: Correctness is the basic design goal of PeDSS, which ensures interference protection for IUs. This property requires that when an SU would cause interference greater than any IU's sensitivity threshold, its spectrum request cannot be approved. i.e. SU cannot receive a valid or useful spectrum license in this case.

IU privacy: IU privacy requires that an honest but curious SAS is not able to identify the IU presence status and operational data. It also requires that a compromised SAS will not leak any information related to IU presence status.

SU location privacy: SU location privacy requires that an honest but curious SAS can only extract limited information on the location of a queried SU. In this paper, we use differential privacy to formally define the concept of "limited information".

IV. Features of Cryptosystems

In this section, we introduce the features of the underlying cryptosystems that are leveraged in PeDSS design.

A. Homomorphic proxy re-encryption scheme

PeDSS uses AFGH scheme, which is a widely used proxy re-encryption scheme proposed by G. Ateniese et al. [14]. The homomorphic multiplication and inverse property of AFGH cryptosystem is critical for the design of PeDSS system, which enables SAS to perform spectrum computation without knowing the actual IU status. Meanwhile, the re-encryption property enables SUs to recover the potential spectrum license without the help of an extra trusted third party.

1) Overview of AFGH: AFGH cryptosystem is defined over a type 1 bilinear groups $(\mathbb{G}_1, \mathbb{G}_T)$, where a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$ exists with the following properties:

1) $\mathbb{G}_1$ and $\mathbb{G}_T$ are multiplicative cyclic groups of prime order p; g is a generator of $\mathbb{G}_1$.
2) e is an efficiently computable bilinear map with the following properties:
   - Bilinear: $e(u^a, v^b) = e(u, v)^{ab}$, $\forall u, v \in \mathbb{G}_1$, $a, b \in \mathbb{Z}_p^*$;
   - Non-degenerate: $e(g, g) \neq 1$.

The following describes the construction of the AFGH scheme, which is "the third attempt" in [14].

- **System parameters**: To setup the system, a Type 1 bilinear pairing system is required. Denote g as the generator of $\mathbb{G}_1$, p as the order of $\mathbb{G}_1$. Set $Z = e(g, g)$.
- **Key Generation**($\mathsf{KG}(a_1, a_2)$): For two inputs $a_1, a_2 \in \mathbb{Z}_p^*$, set secret key $\mathtt{sk} = (a_1, a_2)$, compute public key $\mathtt{pk} := (Z^{a_1}, g^{a_2})$.
- **Re-Encryption Key Generation** ($\mathsf{RG}(\mathtt{sk}_a, \mathtt{pk}_b)$): Taking private key $\mathtt{sk}_a = (a_1, a_2)$ of user A and public key $\mathtt{pk}_b = (Z^{b_1}, g^{b_2})$ of user B, the re-encryption key is computed by $\mathtt{rk}_{A \rightarrow B} := (g^{b_2})^{a_1} = g^{a_1 b_2}$.
- **First-Level Encryption** ($\mathsf{E}_I(M, \mathtt{pk}_a)$): For a message $M \in \mathbb{G}_T$ and public key $\mathtt{pk}_a = (Z^{a_1}, g^{a_2})$, select a random nonce $r \leftarrow_\$ \mathbb{Z}_p^*$, and compute $c_1 = Z^r \cdot M$, $c_2 = Z^{ra_2}$, where $\leftarrow_\$$ means "samples from". The ciphertext is $C := (c_1, c_2)$.
- **Second-Level Encryption** ($\mathsf{E}_{II}(M, \mathtt{pk}_a)$): For a message $M \in \mathbb{G}_T$ and public key $\mathtt{pk}_a = (Z^{a_1}, g^{a_2})$, select $r \leftarrow_\$ \mathbb{Z}_p^*$, and compute $c_1 = Z^{ra_1} \cdot M$, $c_2 = g^r$. The ciphertext is $C := (c_1, c_2)$.
- **First-Level Decryption** ($\mathsf{D}_I(C_r, \mathtt{sk}_b)$): for a first-level ciphertext $C_r = (c_1, c_2)$ and its corresponding private key $\mathtt{sk}_b = (b_1, b_2)$, the plaintext is obtained by computing $M^* := \frac{c_1}{c_2^{1/b_2}}$.
- **Second-Level Decryption** ($\mathsf{D}_{II}(C_r, \mathtt{sk}_a)$): for a second-level ciphertext $C_r = (c_1, c_2)$ and its corresponding private key $\mathtt{sk}_a = (a_1, a_2)$, the plaintext is obtained by computing $M^* := \frac{c_1}{e(g^{a_1}, c_2)}$.
- **Re-Encryption** ($\mathsf{R}(C, \mathtt{rk}_{A \rightarrow B})$): for a message M encrypted by public key (Z^{a_1}, g^{a_2}), its second-level ciphertext $C = (c_1, c_2)$ can be re-encrypted to be a first-level ciphertext encrypted by public key $\mathtt{pk}_b = (Z^{b_1}, g^{b_2})$ by computing $c_2^* := e(c_2, \mathtt{rk}_{A \rightarrow B}) = Z^{(ra_1)b_2}$. The re-encrypted first level ciphertext is $C_r := (c_1, c_2^*)$.

2) Homomorphic property of AFGH scheme:

Proposition 1. ***Homomorphic properties****: Given an AFGH public and private key pair* $(\mathtt{pk}, \mathtt{sk})$*, consider two AFGH second-level encrypted ciphertexts* $C = \mathsf{E}_{II}(M, \mathtt{pk}) = (c_1, c_2)$ *and* $C' = \mathsf{E}_{II}(M', \mathtt{pk}) = (c_1', c_2')$*. The homomorphic multiplication operation* $C \otimes C' := (c_1 c_1', c_2 c_2')$ *produces a ciphertext of* MM'*. In another word,* $\mathsf{D}_{II}(C \otimes C') = MM'$*.*

The homomorphic inverse operation $\mathsf{inv}(C) := (c_1^{-1}, c_2^{-1})$ *produces a ciphertext of* M^{-1}*. In another word,* $\mathsf{D}_{II}(\mathsf{inv}(C)) = M^{-1}$*.*

We omit the proof since the homomorphic feature of AFGH has already been discussed in [15]. Note that this proposition implies that AFGH is homomorphic in terms of division.

B. Cryptographic Assumptions

The security of AFGH scheme is preserved under extended decisional bilinear Diffie-Hellman (EDBDH) assumption and discrete logarithm (DL) assumption [14]. Details of the two assumptions are not provided in this paper due to space limit.

V. System framework

In this section, we present the technical details of each step in PeDSS design.

A. System Setup

To initialize the PeDSS system, the Key Issuer firstly needs to run $\mathsf{Setup}(p)$ algorithm once. The algorithm has three steps:

- Step 1: Set up the AFGH scheme: Let the symmetric bilinear group pair be $\mathbb{G}_1, \mathbb{G}_T$ of prime order p, the corresponding bilinear mapping function be $e : \mathbb{G}_1 \times \mathbb{G}_1 \mapsto \mathbb{G}_T$, and AFGH system parameters be $g \in \mathbb{G}_1$, $Z = e(g, g)$.
- Step 2: Select $a_1, a_2 \leftarrow_\$ \mathbb{Z}_p^*$, and compute an AFGH key pair $(\mathtt{sk}, \mathtt{pk}) = \mathsf{KG}(a_1, a_2)$. Let the master secret key

msk = sk and IU's group public key ipk = pk. Let system parameters params = $(\mathbb{G}_1, \mathbb{G}_T, p, e, g, Z)$.

- Step 3: The Key Issuer publishes ipk and params.

An SU b sets up itself by registering itself at the Key Issuer before sending any spectrum request. During the registration process, it generates a randomized AFGH key pair $(\mathtt{sk}_b, \mathtt{pk}_b)$ and sends $\mathtt{pk}_b$ to the Key Issuer. Upon receiving the SU's public key $\mathtt{pk}_b$, the Key Issuer firstly generates the re-encryption key $\mathtt{rk}_b$ through $\mathtt{rk}_b \leftarrow \mathsf{RG}(\mathtt{msk}, \mathtt{pk}_b)$, and then sends $\mathtt{rk}_b$ back to the SU through a secure channel.

After the initial setup, the Key Issuer, which is the only trusted third party in PeDSS, can go off-line and the PeDSS moves to the normal operation state. Essentially, PeDSS does not require its trusted third party to be engaged during the entire operation of SAS and the Key Issuer is only needed during the system setup stage.

B. Details of data collection and integration routine

In the normal operation state of PeDSS, one function of SAS is to collect and integrate IU input data, whose detail will be introduced in this subsection. The other function of SAS is to process spectrum allocation requests from SUs and we will discuss it in the next subsection.

1) Input data types: We assume that its input data to SAS includes its frequency f_{IU}, location l_{IU}, antenna height h_{IU} and sensitivity level to interference γ. These data have been identified in [5] to be related to spectrum allocation and are also private IU information.

To limit the computation overhead, all input data are discretized into a limited set of possible values. For locations, the discretization process divides PeDSS system's service area into a total of L same size grids and the location of an IU is represented by the grid it belongs to. For antenna height, frequency and interference sensitivity level, each is quantized into several value ranges and the actual value is approximated by the range it belongs to.

2) Generate encrypted IU presence report: An IU runs algorithm $\mathsf{Report\_Gen}(\cdot)$ to generate an encrypted IU presence report. The IU presence information is represented as a matrix $\mathbf{R}$ of dimension $L \times H \times F \times \Gamma$, where L is the total number of grids, H, F and Γ are the total number of discretized antenna height ranges, frequency ranges and interference sensitivity level ranges, respectively.

If an IU is in grid l, has antenna height in range h, frequency in range f, interference sensitivity level in range γ, then $\mathbf{R}$'s entry $\mathbf{R}_{l,h,f,\gamma}$ is a random non-identity element picked from $\mathbb{G}_T$; formally,

$$\mathbf{R}_{l,h,f,\gamma} \leftarrow\!\!\$\ \mathbb{G}_T \backslash \{1_{\mathbb{G}_T}\}\,. \tag{1}$$

For the other entries, we set $\mathbf{R}_{l,h,f,\gamma}$ to be the identity element of $\mathbb{G}_T$, which is denoted as $1_{\mathbb{G}_T}$; formally,

$$\mathbf{R}_{l,h,f,\gamma} \leftarrow 1_{\mathbb{G}_T} \tag{2}$$

Then, IU encrypts every entry of $\mathbf{R}$ using level 2 encryption and public key ipk. We denote the encrypted results as $[\![\mathbf{R}]\!]_{II}$.

3) Update the encrypted map: Upon receiving an encrypted data report $[\![\mathbf{R}]\!]_{II}$, SAS integrates the report into the encrypted map $[\![\mathbf{D}]\!]_{II}$.

The map $[\![\mathbf{D}]\!]_{II}$ is a matrix over level 2 AFGH ciphertext of the same dimension as the incoming report $[\![\mathbf{R}]\!]_{II}$. To initialize this map, SAS sets all elements to be $[\![1_{\mathbb{G}_T}]\!]_{II}$, which is the second level ciphertext of $1_{\mathbb{G}_T}$, encrypted using key ipk.

SAS updates the map by conducting element-wise homomorphic multiplication between $[\![\mathbf{D}]\!]_{II}$ and $[\![\mathbf{R}]\!]_{II}$. That is:

$$[\![\mathbf{D}_{l,h,f,\gamma}]\!]_{II} \leftarrow [\![\mathbf{D}_{l,h,f,\gamma}]\!]_{II} \otimes [\![\mathbf{R}_{l,h,f,\gamma}]\!]_{II} \tag{3}$$

for all entries of $[\![\mathbf{D}]\!]_{II}$ and $[\![\mathbf{R}]\!]_{II}$.

4) Release an expired IU presence data report: When an IU data report $[\![\mathbf{R}]\!]_{II}$ expires, SAS removes the impact of that report by element-wise homomorphically dividing $[\![\mathbf{D}]\!]_{II}$ by $[\![\mathbf{R}]\!]_{II}$. Formally, this means:

$$[\![\mathbf{D}_{l,h,f,\gamma}]\!]_{II} \leftarrow [\![\mathbf{D}_{l,h,f,\gamma}]\!]_{II} \otimes \mathsf{inv}([\![\mathbf{R}_{l,h,f,\gamma}]\!]_{II}) \tag{4}$$

for all entries of $[\![\mathbf{D}]\!]_{II}$ and $[\![\mathbf{R}]\!]_{II}$.

C. Details of spectrum allocation routine

In order to protect its privacy, when an SU sends SAS its spectrum request, it will not tell SAS its exact location. Instead, it sends a fuzzy location that is different from its true location and guarantees differential privacy. Upon receiving the SU request with the fuzzy location, SAS needs to conduct proper spectrum allocation based on the fuzzy SU location and encrypted IU information. Finally, SAS needs to ensure that from the information that it sends to the SU, the SU can recover a proper license if and only if the spectrum computation result indicates that no harmful interference will be generated by the SU.

In the following, we describe the details of spectrum access request generation, spectrum allocation, and license recovery.

1) Generating spectrum access request: Let $l_{SU} := (x, y)$ be the true location of the SU. To protect SU location privacy, the SU firstly picks a privacy parameter r_{SU}, which indicates the mean value of the distance between fuzzy location and true location. Then, the SU sets the parameter ϵ by

$$\epsilon \leftarrow \frac{2}{r_{SU}}, \tag{5}$$

and generates a fuzzy location l^*_{SU} by sampling from polar Laplacian distribution with parameter ϵ, which proceeds as follows[16]:

- Step 1: Select $\hat{\theta} \leftarrow\!\!\$\ [0, 2\pi)$;
- Step 2: Select $z \leftarrow\!\!\$\ [0, 1)$. Define function $C_\epsilon(r) = 1 - (1 + \epsilon r)e^{-\epsilon r}$, and find the root of equation $z = C_\epsilon(r)$ through bisection. Let the root be $\hat{r}$;
- Step 3: Set $x^* \leftarrow x + \hat{r}\cos\hat{\theta}$ and $y^* \leftarrow y + \hat{r}\sin\hat{\theta}$. Output $l^*_{SU} = (x^*, y^*)$.

Afterwards, the SU sends its spectrum request message using the fuzzy location l^*_{SU} instead of its true location, i.e., req $= (l^*_{SU}, r_{SU}, h_{SU}, f_{SU}, P_{\max})$, where h_{SU} is the discretized antenna height of SU and $P_{\max}$ is the maximum transmission power (in dBm).

In the following, we claim and proof that the above algorithm correctly generates fuzzy-locations following Laplacian distribution with mean value r_{SU}:

Proposition 2. *The above fuzzy-location generating algorithm creates locations following polar Laplacian distribution centered in the origin l_{SU} with parameter $\epsilon = \frac{2}{r_{SU}}$. That is,*

$$\Pr[l^*_{SU} = (r, \theta)] = \frac{\epsilon^2}{2\pi} r e^{-\epsilon r}, \tag{6}$$

where (r, θ) is the polar coordinate of fuzzy location l^*_{SU} centered at l_{SU}. Moreover, the mean value of distance between fuzzy location and true location is r_{SU}; that is,

$$E[d(l^*_{SU}, l_{SU})] = r_{SU}, \tag{7}$$

where $d(\cdot, \cdot)$ denotes Euclidean distance.

Proof. From the algorithm description, the fuzzy location l^*_{SU} has a polar coordinate of $(\hat{r}, \hat{\theta})$, centered at l_{SU}. Thus, $d(l^*_{SU}, l_{SU})$ has a cumulative distribution function (CDF) as $C_\epsilon(r)$, and its probability distribution function (PDF) $D_\epsilon(r)$ can be obtained as:

$$D_\epsilon(r) = \frac{dC_\epsilon(r)}{dr} = \epsilon^2 r e^{-\epsilon r}. \tag{8}$$

Since $\hat{\theta}$ is uniformly selected and independent of $\hat{r}$, we have:

$$\begin{aligned} \Pr[l^*_{SU} = (r, \theta)] &= \Pr[d(l^*_{SU}, l_{SU}) = r] \cdot \frac{1}{2\pi} = D_\epsilon(r) \frac{1}{2\pi} \\ &= \epsilon^2 r e^{-\epsilon r} \cdot \frac{1}{2\pi} = \frac{\epsilon^2}{2\pi} r e^{-\epsilon r}. \end{aligned} \tag{9}$$

Note that $D_\epsilon(r)$ is the pdf of the gamma distribution with shape 2 and scale $1/\epsilon$. So its mean value is:

$$E[d(l^*_{SU}, l_{SU})] = 2 \cdot \frac{1}{\epsilon} = r_{SU}. \tag{10}$$

□

2) Location stretch: Upon receiving the SU request, the first step that SAS does is to perform a location stretch, which creates a location set $\mathcal{L}$, where the true location of the SU is expected to belong to $\mathcal{L}$.

To stretch the fuzzy location, SAS selects a stretch radius r_0, draws a circle centered at the fuzzy location with radius r_0, and puts all location grids that are located in the circle in the stretched set. That is,

$$\mathcal{L} \leftarrow \{l_{SU} : d(l_{SU}, l^*_{SU}) < r_0\}, \tag{11}$$

where $d(\cdot, \cdot)$ denotes Euclidean distance.

3) Spectrum computation on ciphertext domain: Note the IU presence map $\llbracket \mathbf{D} \rrbracket_{II}$ is encrypted and SAS cannot directly compute the interference an SU would cause to IUs. Instead, SAS firstly enumerates all feasible operational parameters of an IU (i.e., all possible combinations of discretized IU location l, antenna height h, frequency f and sensitivity level γ). If for any IU parameter combination, the SU located in some location in $\mathcal{L}$ can cause interference larger than IU sensitivity level, the combination of IU parameters are put into a set $\mathcal{U}$.

Afterwards, the SAS homomorphically multiplies all items in the encrypted map $\llbracket \mathbf{D} \rrbracket_{II}$ with index in set $\mathcal{U}$. According to equation (1)(2)(3)(4), the result, denoted as I, will be

$$I = \llbracket 1_{\mathbb{G}_T} \rrbracket_{II} \otimes \llbracket 1_{\mathbb{G}_T} \rrbracket_{II} \otimes \cdots \otimes \llbracket 1_{\mathbb{G}_T} \rrbracket_{II} = \llbracket 1_{\mathbb{G}_T} \rrbracket_{II}, \tag{12}$$

if the SU will not generate harmful interference assuming it is located at *any* locations in $\mathcal{L}$; otherwise, I will be almost surely[1] a level 2 ciphertext of a random element in $\mathbb{G}_T$ other than $1_{\mathbb{G}_T}$.

Figure. 2 presents the pseudocode of the above spectrum computation on ciphertext domain, where $A(\cdots)$ is the attenuation map function. $A(l_{SU}, h_{SU}, f_{SU}, l, h, f)$ is assigned to be the **minimum** path loss between a transmitter located at grid l_{SU} with antenna height range h_{SU}, frequency range f_{SU} and a receiver located at grid l with antenna height range h, frequency range f. This map function can be pre-computed to reduce the online computation overhead of handling a spectrum request.

```
$\mathcal{U} \leftarrow \emptyset,\ I \leftarrow \mathsf{E}_{II}(1_{\mathbb{G}_T}, \mathtt{ipk})$
for all possible $(l, h, f, \gamma)$ combinations and all $l_{SU} \in \mathcal{L}$
  if $P_{\max} - A(l_{SU}, h_{SU}, f_{SU}, l, h, f) \geq \gamma$
    $\mathcal{U} \leftarrow \mathcal{U} \cup \{(l, h, f, \gamma)\}$
  fi
endfor
for all $(l, h, f, \gamma) \in \mathcal{U}$
  $I \leftarrow I \otimes \llbracket \mathbf{D}_{l,h,f,\gamma} \rrbracket_{II}$
endfor
return $I$
```

Fig. 2: Spectrum computation on ciphertext domain

4) Generate potential spectrum license: A valid spectrum license consists of the PKI certificate of SAS $\mathtt{cert}_{SAS}$, license related message $\mathtt{msg}$ and the signature σ of $\mathtt{msg}$ signed by $\mathtt{sk}_{SAS}$. Moreover, $\mathtt{msg}$ is composed of SAS-stretched location set $\mathcal{L}$, SU's antenna height h, allowed maximum power $P_{\max}$ and extra information. The expiration date and SU's credential are put in the extra information field.

SAS firstly generates the valid license $\mathtt{cred}$ based on previously computed $\mathcal{L}$ and SU information, yet $\mathtt{cred}$ should only be sent to the SU if the SU causes no potential harmful interference. To achieve this, SAS firstly picks random element

[1] The probability with which I being the ciphertext of $1_{\mathbb{G}_T}$ is $\mathcal{O}(1/p)$ where $p = 2^{80}$ if we are considering 80 bit level security.

α in $\mathbb{G}_T$ and hashes it to random bit string $\mathtt{k}$. Then it encrypts the valid license using any symmetric key cryptosystem (say, AES) assuming $\mathtt{k}$ as the secret key, which is denoted as $\mathsf{E}_{AES}(\mathtt{cred},\mathtt{k})$. Meanwhile, SAS encrypts α to level 2 ciphertext using key $\mathtt{ipk}$ and homomorphically multiplies it with the I returned by the algorithm in Fig. 2. That is, set $C \leftarrow \mathsf{E}_{II}(\alpha,\mathtt{ipk}) \otimes I$. Note that if the SU cannot create harmful interference to IU, then $I = [\![1_{\mathbb{G}_T}]\!]_{II}$ and hence C can be decrypted to α. Otherwise, I is the ciphertext of some random number and decryption of C just results in a random number.

Finally, SAS sets the potential license $\mathtt{cred}_P = (C, \mathsf{E}_{AES}(\mathtt{cred},\mathtt{k})))$ and sends it to the SU.

5) License recovery: Upon receiving a potential spectrum license $\mathtt{cred}_P = (C, \mathtt{str})$, an SU b recovers a license by running algorithm Rec, which proceeds as follows:

- Using the re-encryption key $\mathtt{rk}_b$ obtained during its registration process with the Key Distributor (See section III.B), SU b re-encrypts C to be a first level cipher text encrypted by $\mathtt{pk}_b$: $C^* \leftarrow \mathsf{R}(C, \mathtt{rk}_b)$.
- SU b decrypts C^* and recovers an AES key, and then obtains the recovered spectrum license using that key:

$$k^* \leftarrow H(\mathsf{D}_I(C^*, \mathtt{sk}_b)),$$
$$\mathtt{cred}^* \leftarrow \mathsf{D}_{AES}(\mathtt{cred}, k^*).$$

Afterwards, the SU uses the signature part of $\mathtt{cred}^*$ to check if $\mathtt{cred}^*$ is a valid license. If the validation fails, it means that $I \neq 1_{\mathbb{G}_T}$ (i.e. SU can create harmful interference). Only when $\mathtt{cred}^*$ has been successfully validated, can the SU access the spectrum from a location inside the set $\mathcal{L}$ indicated in the license.

VI. Security Definitions and Analysis

In this section, we provide formal definitions and proof of PeDSS's security properties.

A. Correctness

We denote the whole PeDSS functionality as a function f:

$$\mathtt{cred}^* := f(\mathcal{R}, \mathcal{E}, A, \mathtt{req}), \tag{13}$$

where the $\mathcal{R}$ is the set of all received IU presence reports and $\mathcal{E}$ is the set of all expired IU presence reports. The formal definition of correctness is given as follows:

Definition 1. *Denote SU's true location as l_{SU} and its spectrum request as $\mathtt{req} = (l^*_{SU}, \epsilon, h_{SU}, f_{SU}, P_{\max})$. PeDSS is correct if for any input $(\mathcal{R}, \mathcal{E}, A, \mathtt{req})$ to PeDSS functionality and an IU parameter combination (l, h, f, γ) such that*

$$P_{\max} - A(l_{SU}, h_{SU}, f_{SU}, l, h, f) \geq \gamma$$
$$\exists \mathbf{R} \in \mathcal{R}, \mathbf{R} \notin \mathcal{E} \textit{ s.t. } \mathbf{R}_{l,h,f,\gamma} \neq 1_{\mathbb{G}_T},$$

either the signature of $\mathtt{cred}^* := f(\mathcal{R}, \mathcal{E}, A, \mathtt{req})$ *is invalid or* $l_{SU} \notin \mathcal{L}$*. ($\mathcal{L}$ is the location set in* $\mathtt{cred}^*$*)*

Theorem 1. *The probability with which PeDSS is NOT correct is negligible.*

Proof. If $l_{SU} \notin \mathcal{L}$, then PeDSS is correct.

Now assume $l_{SU} \in \mathcal{L}$. Define $\mathbf{R}_{l,h,f,\gamma} = Q \neq 1_{\mathbb{G}_T}$ for some $Q \leftarrow_{\$} \mathbb{G}_T \backslash \{1_{\mathbb{G}_T}\}$. Note that

$$[\![\mathbf{D}_{l,h,f,\gamma}]\!]_{II} = \bigotimes_{\mathbf{R}\in\mathcal{R},\mathbf{R}\notin\mathcal{E}} [\![\mathbf{R}_{l,h,f,\gamma}]\!]_{II}, \tag{14}$$

and $I = \otimes_{(l,h,f,\gamma)\in\mathcal{U}} [\![\mathbf{D}_{l,h,f,\gamma}]\!]_{II}$. Hence, I is essentially homomorphic multiplication of $Q \neq 1_{G_T}$ with several items which are either $[\![1_{\mathbb{G}_T}]\!]_{II}$ or some other random elements. Hence, the probability with which $I = [\![1_{\mathbb{G}_T}]\!]$, is negligible. For example, for 80 bit security level, the probability is 2^{-80}. Therefore, the probability with which the recovered key is correct (i.e. $k^* = k$) and the signature of $\mathtt{cred}^*$ can be verified as valid is negligible. This demonstrate that the chance that PeDSS is incorrect is negligible. □

B. Privacy for IU

To formally define IU privacy property, we setup a security experiment, which is shown in Figure. 3. In the privacy experiment, the adversary $\mathcal{A}$ is required to respond a challenge by showing its ability to distinguish two different IU presence data patterns. The adversary $\mathcal{A}$ submits two actually IUs presence data sequences arbitrarily on his own choice, and then the challenger will pick a random one and generate the all corresponding data reports, based on which the adversary will output a single bit indicating the guess on which sequence is picked by the challenger. In addition, we also allow the adversary $\mathcal{A}$ to query a CrptReg oracle, which implies the adversary can corrupt SU registration channel.

The formal definition of IU privacy is shown as follows:

Definition 2. *PeDSS is private for IUs if for all $\lambda \in \mathbb{N}$, the advantage $\mathsf{Adv}^{\mathrm{Priv}}_{\mathcal{A}}(\lambda)$ is negligible in λ for all Probabilistic Polynomial-Time (PPT) Adversaries $\mathcal{A}$, where*

$$\mathsf{Adv}^{\mathrm{Priv}}_{\mathcal{A}}(\lambda) = \left| \Pr\left[\mathbf{Exp}^{\mathrm{Priv}}_{\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2} \right|$$

Theorem 2. *If AFGH scheme is semantically secure, PeDSS is private for IUs.*

Proof. To prove PeDSS is private for IUs, we assume that there exists an adversary $\mathcal{A}$ that can break IU privacy with non-negligible probability. Then, we construct a simulator $\mathcal{S}$ that aims at breaking the semantic security of AFGH scheme by taking advantage of the adversary $\mathcal{A}$. $\mathcal{S}$ plays as the adversary in a given AFGH semantic security experiment, yet meanwhile it sets up a simulated PeDSS privacy experiment to interact with $\mathcal{A}$. Finally, it can leverage the response from $\mathcal{A}$ to gain a non-negligible advantage in AFGH semantic security experiment.

$\mathcal{S}$ set up the the simulated PeDSS privacy experiment as follows. Firstly $\mathcal{S}$ sets $\mathtt{msk} \leftarrow \mathtt{sk}_B$ and $\mathtt{ipk} \leftarrow \mathtt{pk}_B$. Here $\mathtt{msk}$ is actually unknown by $\mathcal{S}$. Let $\mathtt{pk}_h = (Z^{h_1}, g^{h_2})$ and $\mathtt{sk} = (B_1, B_2)$. Then to simulate the corruption oracle CrptReg, $\mathcal{S}$ selects $r_1, r_2 \leftarrow_{\$} \mathbb{Z}^*_p$ and compute $\mathtt{pk}_x \leftarrow ((Z^{h_1})^{r_1}, (g^{h_2})^{r_2})$ and $\mathtt{rk}_C \leftarrow (\mathtt{rk}_{B\rightarrow h})^{r_2}$. In this case $(\mathtt{sk}_x, \mathtt{pk}_x)$ is a randomly

```
$\mathbf{Exp}_{\mathcal{A}}^{\mathrm{Priv}}(\lambda)$
$(\mathtt{msk}, \mathtt{ipk}, \mathtt{params}) \leftarrow \mathsf{Setup}(2^{\lambda})$.
$(\mathbf{T}^{(0)}, \mathbf{T}^{(1)}) \leftarrow \mathcal{A}^{\mathsf{CrptReg}}(\mathtt{ipk}, \mathtt{params})$,  where
    $\mathbf{T}^{(0)} := \left\{\mathbf{R}^{(0)}(k)\right\}_{k=1}^{M}, S^{(0)}, lt^{(0)}$;
    $\mathbf{T}^{(1)} := \left\{\mathbf{R}^{(1)}(k)\right\}_{k=1}^{M}, S^{(1)}, lt^{(1)}$
$\mathbf{return} \perp \mathbf{if} \left\{\mathbf{R}^{(0)}(k)\right\}_{k=1}^{M} = \left\{\mathbf{R}^{(1)}(k)\right\}_{k=1}^{M}$.
$b \leftarrow_{\$} \{0, 1\}$.
$\mathbf{for}\ k = 1, \cdots, M$
    $[\![\mathbf{R}^*(k)]\!]_{II} \leftarrow \mathsf{Report\_Gen}(\mathbf{R}^{(b)}(k))$.
$\mathbf{endfor}$
$b' \leftarrow \mathcal{A}^{\mathsf{CrptReg}}(\mathtt{ipk}, \mathtt{params}, \{[\![\mathbf{R}^*(k)]\!]_{II}\}_{k=1}^{M})$.
$\mathbf{return}\ 1$ if $b = b'$; otherwise $\mathbf{return}\ 0$.

CrptReg
$x_1, x_2 \leftarrow_{\$} \mathbb{Z}_p^*,\ (\mathtt{sk}_x, \mathtt{pk}_x) \leftarrow \mathsf{KG}(x_1, x_2)$.
$\mathtt{rk}_C = \mathsf{RG}(\mathtt{msk}, \mathtt{pk}_x),\ \mathbf{return}\ (\mathtt{pk}_x, \mathtt{rk}_C)$.
```

Fig. 3: Definition of privacy experiment

valid key pair where $\mathtt{sk}_x = (h_1 r_1, h_2 r_2)$, and $\mathtt{rk}_C = g^{B_1 h_2 r_2} = g^{B_1 (h_2 r_2)}$ is generated correctly. Hence CrptReg is perfectly simulated. Afterwards, $\mathcal{S}$ submits $m_0 = 1_{\mathbb{G}_T}$ and $m_1 \leftarrow_{\$} \mathbb{G}_T \backslash \{1_{\mathbb{G}_T}\}$ to AFGH semantic security experiment defined in [14]. It obtains $C := \mathsf{E}_{II}(\mathtt{pk}_B, m_b) \in \mathbb{G}_T \times \mathbb{G}_1$. Let $C = (c_1, c_2)$, where $c_1 = Z^{sB_1} m_b$ and $c_2 = g^s$. Upon receiving $\mathbf{T}^{(0)}$ and $\mathbf{T}^{(1)}$ submitted by $\mathcal{A}$, $\mathcal{S}$ skips the procedure of selecting b and generates simulated $\{[\![\mathbf{R}^*(k)]\!]_{II}\}_{k=1}^{M}$ as follows. For any $k = 1, \cdots, M$, if $\mathbf{R}^{(0)}(k)_{l,h,f,\gamma} = \mathbf{R}^{(1)}(k)_{l,h,f,\gamma}$, then $\mathcal{S}$ computes the value of $[\![\mathbf{R}^*(k)_{l,h,f,\gamma}]\!]_{II}$ faithfully according to algorithm Report_Gen(·). Otherwise, $\mathcal{S}$ selects $r \leftarrow_{\$} \mathbb{Z}_p^*$ and sets $[\![\mathbf{R}^*(k)_{l,h,f,\gamma}]\!]_{II} \leftarrow \otimes_{i=1}^{r} C$ if $\mathbf{R}^{(0)}(k)_{l,h,f,\gamma} = 0$; $\mathcal{S}$ sets $[\![\mathbf{R}^*(k)_{l,h,f,\gamma}]\!]_{II} \leftarrow ((c_1/m_1)^r, c_2^r)$ if $\mathbf{R}^{(0)}(k)_{l,h,f,\gamma} = 1$. Finally, $\mathcal{S}$ outputs 0 in AFGH semantic security experiment if $\mathcal{A}$ outputs 0; otherwise $\mathcal{S}$ outputs 1.

In the AFGH semantic security experiment, if b is selected as 0, then $[\![\mathbf{R}^*(k)_{l,h,f,\gamma}]\!]_{II} = ((Z^{sB_1} m_b)^r, (g^s)^r) = (Z^{(sr)B_1}, g^{sr})$ when $\mathbf{R}^{(0)}(k)_{l,h,f,\gamma} = 0$; $[\![\mathbf{R}^*(k)_{l,h,f,\gamma}]\!]_{II} = ((Z^{sB_1} m_b/m_1)^r, (g^s)^r)) = (Z^{(sr)B_1} m_1^{-r}, g^{sr})$ when $\mathbf{R}^{(0)}(k)_{l,h,f,\gamma} = 1$. Therefore in this case $\mathcal{S}$ simulates $\mathbf{R}^*(k)$ perfectly in the case of setting b as 0 in the PeDSS privacy experiment.

On the other hand, if b is selected as 1, then we have $[\![\mathbf{R}^*(k)_{l,h,f,\gamma}]\!]_{II} = ((Z^{sB_1} m_b)^r, (g^s)^r) = (Z^{(sr)B_1} m_1^r, g^{sr})$ when $\mathbf{R}^{(0)}(k)_{l,h,f,\gamma} = 0$; when $\mathbf{R}^{(0)}(k)_{l,h,f,\gamma} = 1$, we have $[\![\mathbf{R}^*(k)_{l,h,f,\gamma}]\!]_{II} = ((Z^{sB_1} m_b/m_1)^r, (g^s)^r)) = (Z^{(sr)B_1}, g^{sr})$. Therefore in this case $\mathcal{S}$ also simulates $\mathbf{R}^*(k)$ perfectly in the case of setting b as 1 in the PeDSS privacy experiment. Hence we conclude that the advantage for breaking AFGH semantic security is the same as the advantage for breaking PeDSS privacy.

□

Claim 1. *Under EDBDH assumption, PeDSS is private for IUs.*

Proof. According to theorem 3.1 in [14], AFGH is semantically secure under EDBDH assumption, so PeDSS is private for IUs under EDBDH assumption. □

C. Location Privacy for SU

In PeDSS, the concept of geo-indistinguishability [16] can be adopted to formally define location privacy for SUs. Intuitively, for a curious SAS, if an SU at two locations l_{SU}, l'_{SU} generates the same fuzzy location l^*_{SU} with similar probabilities, then the SAS holds little information about whether the true location is l_{SU} or l'_{SU}. The formal definition of SU geo-indistinguishability is given as follows[2]:

Definition 3. *PeDSS satisfies ϵ-geo-indistinguishability if and only if for any received fuzzy location l^*_{SU}:*

$$\frac{P(l^*_{SU}|l_{SU})}{P(l^*_{SU}|l'_{SU})} \leq e^{\epsilon r} \qquad \forall r > 0\ \forall l_{SU}, l'_{SU} : d(l_{SU}, l'_{SU}) \leq r. \tag{15}$$

Theorem 3. *PeDSS preserves ϵ'-geo-indistinguishability for SUs in the service area with diameter D, with*

$$\epsilon' = \frac{2}{r_{SU}} + \frac{1}{u} \ln \frac{q - 2 + 3e^{2v\sqrt{2}/r_{SU}}}{q - 5}, \tag{16}$$

where u, v is the smaller and larger value of reported location precision in Cartesian coordinates; q is a parameter obtained $q = \frac{u}{D\delta_\theta}$, where δ_θ is the precision of angle.

Proof. From the analysis in [16] a mechanism drawing fuzzy locations from polar Laplacian distribution with parameter ϵ preserves ϵ-geo-indistiguishabiltiy. However, locations reported to SAS have precisions and thus are actually discrete values. From Theorem 4.1 in [16], within a given diameter $r_{\max} := \frac{u}{q\delta_\theta}$, PeDSS provides ϵ'-geo-indistinguishability for SUs in the service area with diameter D, with

$$\begin{aligned} \epsilon' &= \epsilon + \frac{1}{u} \ln \frac{q - 2 + 3e^{\epsilon v \sqrt{2}}}{q - 5} \\ &= \frac{2}{r_{SU}} + \frac{1}{u} \ln \frac{q - 2 + 3e^{2v\sqrt{2}/r_{SU}}}{q - 5}. \end{aligned} \tag{17}$$

Note that $r_{\max} = \frac{u}{q\delta_\theta} = D$, so ϵ'-geo-indistinguishability for SUs is preserved in the diameter D. □

VII. Evaluation

A. Implementation details

To instantiate a secure Type 1 pairing, we utilized the pairing with "A-internal" described in [17]. The security level is symmetric 80 bits, which provides approximately the same level of security as an RSA signature with a modulus size of 1024 bits. By using the pairing-based cryptography (PBC) library available at [17], we implemented the AFGH scheme and the homomorphic operation algorithms.

[2] There are three equivalent definitions proposed in [16], and in this paper we adopt the third one.

To evaluate PeDSS, we set the service area to be a 405 km^2 area in Washington D.C.. We employed L-R model using SPLAT![18] to calculate the attenuation map in this area, where real world terrain data obtained from USGS and SRTM3 was used. The IU interference sensitivity level was set as -80 dBm, which is the sensitivity for a general military radar. We splitted this urban area of Washington D.C. into 36×45 grids with a side length of 500m. All the experiments were conducted on a laptop with Intel i7-4770 CPU @ 3.4GHz.

We compared the performance of PeDSS with two MPC-based privacy preserving system: P^2-SAS proposed in [5], and PISA proposed in [7]. We selected these two works as the benchmark because of the similarity that both PeDSS and them address privacy for both IUs and SUs under protection zone model with the assumption that SAS is not fully trusted.

B. Computational overhead

Table I compares the computational overhead of PeDSS with P^2-SAS and PISA. We simulated 5000 random spectrum requests to evaluate PeDSS and utilized the benchmark of Paillier cryptosystem to evaluate P^2-SAS and PISA under *the same scale of network and the same resolution for discretization.*

TABLE I: Computational overhead comparison between PeDSS, P^2-SAS and PISA

	PeDSS	P^2-SAS	PISA
IU: report generation	2.221 s	197.5 s	2.16 s
SAS: report update	34.36 ms	26.03 ms	70.20 ms
SU: request generation	0.26 ms	197.5 s	5.97 s
SAS: spectrum allocation	0.51 ms	3.48 s	5.91 s

From Table I, we observe that in terms of spectrum allocation, PeDSS is three orders of magnitude faster than P^2-SAS and PISA. PeDSS achieves advantage in computational overhead on spectrum allocation as a result of the novel construction that leverages homomorphic proxy re-encryption, where SAS does not have to aggregate all items in the encrypted database.

Compared to P^2-SAS, PeDSS is more than one order of magnitude faster in terms of generating IU report and performs similarly in integrating one IU report into the database. This is due to the difference between AFGH and Paillier cryptosystem. PeDSS does not achieve significant advantage over PISA in terms of generating IU report, since PISA is tailored for DSA system in UHF TV band, where some IU operational data are assumed to be known by the SAS, including the location, antenna height, and transmission power. Hence in PISA IUs are sending a smaller matrix to SAS under the same scale of network.

The computational overhead of generating an SU request in PeDSS is about two orders of magnitude faster than P^2-SAS. This is achieved by the incorporation of differential privacy in PeDSS, so that SU does not generate a whole encrypted matrix to hide its actual location.

C. Communication overhead

Table II compares the communication overhead of PeDSS with P^2-SAS and PISA. For the spectrum license $\texttt{cred}$ in PeDSS, each field of the message is assumed to be 32 bits long. The SAS certificate $\texttt{cert}_{SAS}$ is assumed to be a X.509 certificate. The length of $\texttt{cert}_{SAS}$ is set to be 1888 bytes, which is the same as the site certificate of Google. The signature σ in $\texttt{cred}$ is a 512 bit ECDSA signature.

TABLE II: Communication overhead comparison between PeDSS, P^2-SAS and PISA

	PeDSS	P^2-SAS	PISA
IU report (single IU)	297.9 KB	3.174 MB	50.2 KB
SU request	3.65 KB	1.19 MB	783.0 KB
SAS response	3.95 KB	4.01 KB	4.01 KB

In PeDSS, the size of a single IU report is about one order of magnitude smaller than P^2-SAS. PeDSS achieves the advantage since the bilinear pairing cryptosystem of AFGH is constructed on elliptic curves, while the Paillier cryptosystem is based on number theory and consumes larger spaces under the same level of security. One can also observe that in PISA, the size of IU report is smaller than the one in PeDSS. This is because in PISA smaller matrix are sent to SAS under the same scale of network; as we mentioned above, PISA is customized for UHF TV band, where the privacy of IU locations, antenna height, and transmission power are not protected.

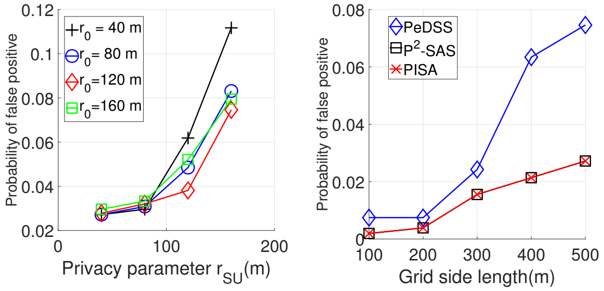
Note that in PeDSS, the size of a spectrum request is more than two orders of magnitude smaller than ones in P^2-SAS and PISA. This is also achieved by the incorporation of differential privacy, where SU does not need to send a whole matrix to SAS to hide its actual location.

D. Accuracy

There are two types of errors in a general DSA system: false positive and false negative. A false positive error means that an SU does not get access to the spectrum although it will not generate harmful interference to IUs, and a false negative error means that an SU gets a a valid spectrum license although it will generate harmful interference to IUs. False positives result in lost spectrum access opportunity, which is undesirable but tolerable. False negatives create interference to IUs, which violates the FCC DSA rules and should be avoided whenever possible. Because of the correctness feature of PeDSS, there is no false negative errors *assuming the measurements and attenuation calculation is accurate*. Recall that the attenuation map records the minimum attenuation values for those discretized parameters, so the discretization can only overestimate the interference and never underestimate interference (if we neglect the errors of L-R model itself). Therefore, the only possible false negatives are introduced by errors of L-R model. Yet, performing a fine-grained measurements over a 405 km^2 area to evaluate the accuracy of L-R model is beyond the scope of this paper. Compared to other privacy-preserving techniques that also introduce location fuzziness (e.g [12][9][4]), PeDSS achieves the advantage of negligible false negatives.

In PeDSS, there are two factors contributing to the false positive error: stretching error and discretization error. This is because both stretching and discretization can lead to overestimation of SU interference. Figure. 4a shows the false positive error of PeDSS, where r_{SU} is the mean distance between the fuzzy location submitted by an SU and the true location of the SU. From the figure we can observe the trade-off between false positive and SU privacy. When privacy parameter r_{SU} is set as 160m, parameter $\epsilon = 2/r_{SU} = 0.0125$ and from the CDF function $C_\epsilon(r)$, 97% fuzzy locations are in an radius of 428.4m. In this case PeDSS achieves false positive rate of 7.46% by setting stretch radius $r_0 = 120$m. Therefore, PeDSS incurs small false positive error by setting a proper stretch radius r_0.

By comparing different curves in Figure. 4a, we also observe that when stretching radius r_0 is set to be 120m, PeDSS achieves lower false positive than the case where stretching radius r_0 is set to be 80m or 160m. Intuitively stretching radius r_0 should not be too large since larger r_0 leads to more interference overestimation. Meanwhile, if the stretching radius r_0 is too small, then SAS will possibly not cover the true location of SU; as a result, the license will not be valid and a false positive error happens.



(a) False positive vs. privacy parameter r_{SU}.

(b) False positive vs. grid side length.

Fig. 4: False positive evaluation.

Figure. 4b compares the false positive error between PeDSS, P^2-SAS and PISA. For PeDSS, privacy parameter r_{SU} is set as 160m, and stretch radius r_0 is set as 120m. The false positive probability of PISA and P^2-SAS shows the errors introduced by discretization, and the gap of false positive probability between PeDSS and P^2-SAS/PISA shows the errors introduced by SAS location stretching of PeDSS.

VIII. Conclusion

In this paper, we proposed a novel DSA framework called PeDSS, which preserves privacy for both IUs and SUs under the scenario of untrusted SAS. By leveraging the homomorphic property of AFGH scheme and incorporation of proxy re-encryption, PeDSS manages to achieve the privacy protection goal while eliminating the online trusted third party. The location privacy of SUs is preserved by differentially private mechanisms and the potential false negative error from this mechanism is mitigated through location stretch algorithm. PeDSS also achieves significantly lower computational overhead compared to prior art, as a result of faster cryptographic primitive and the usage of differential privacy technique. Future work may focus on further eliminating the offline trusted third party by leveraging secure hardware such as Trusted Platform Module (TPM) and deeper study on the impact of location stretch algorithm on false positives.

References

[1] F. C. Commission, “Amendment of the commissions rules with regard to commercial operations in the 3550-3650 mhz band,” *Report and Order and Second Further Notice of Proposed Rulemaking in GN Docket*, no. 12-354, 2015.

[2] L. Rainie, S. Kiesler, R. Kang, M. Madden, M. Duggan, S. Brown, and L. Dabbish, “Anonymity, privacy, and security online,” *Pew Research Center*, vol. 5, 2013.

[3] F. C. Commission, “Wireless telecommunications bureau and office of engineering and technology conditionally approve seven spectrum access system administrators for the 3.5 ghz band,” no. DA/FCC: DA-16-1426, 2016.

[4] 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*. IEEE, 2015, pp. 7640–7645.

[5] Y. Dou, K. C. Zeng, H. Li, Y. Yang, B. Gao, C. Guan, K. Ren, and S. Li, “P 2-sas: preserving users’ privacy in centralized dynamic spectrum access systems,” in *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 2016, pp. 321–330.

[6] Y. Dou, H. Li, K. C. Zeng, J. Liu, Y. Yang, B. Gao, and K. Ren, “Preserving incumbent users privacy in exclusion-zone-based spectrum access systems,” in *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on*. IEEE, 2017, pp. 2486–2493.

[7] C. Guan, A. Mohaisen, Z. Sun, L. Su, K. Ren, and Y. Yang, “When smart tv meets crn: Privacy-preserving fine-grained spectrum access,” in *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on*. IEEE, 2017, pp. 1105–1115.

[8] H. Li, Y. Dou, C. Lu, D. Zabransky, Y. Yang, and J.-M. Park, “Preserving incumbent users’ location privacy in the 3.5 ghz band,” in *Dynamic Spectrum Access Networks (DYSPAN), 2018 IEEE International Symposium on*. IEEE, 2018.

[9] 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.

[10] M. Grissa, B. Hamdaoui, and A. A. Yavuz, “Location privacy in cognitive radio networks: A survey,” *IEEE Communications Surveys & Tutorials*, 2017.

[11] 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.

[12] M. Clark and K. Psounis, “Can the privacy of primary networks in shared spectrum be protected?” in *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*. IEEE, 2016, pp. 1–9.

[13] A. Blum, K. Ligett, and A. Roth, “A learning theory approach to noninteractive database privacy,” *Journal of the ACM (JACM)*, vol. 60, no. 2, p. 12, 2013.

[14] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, “Improved proxy re-encryption schemes with applications to secure distributed storage,” *ACM Trans. Inf. Syst. Secur.*, vol. 9, no. 1, pp. 1–30, Feb. 2006. [Online]. Available: http://doi.acm.org/10.1145/1127345.1127346

[15] H. Shafagh, A. Hithnawi, L. Burkhalter, P. Fischli, and S. Duquennoy, “Secure sharing of partially homomorphic encrypted iot data,” in *Proceedings of the 15th ACM Conference on Embedded Network Sensor System*, 2017.

[16] 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 &#38; Communications Security*, ser. CCS ’13. New York, NY, USA: ACM, 2013, pp. 901–914. [Online]. Available: http://doi.acm.org/10.1145/2508859.2516735

[17] “PBC: Pairing-based cryptography,” https://crypto.stanford.edu/pbc/, accessed: May 1, 2015.

[18] [Online]. Available: http://www.qsl.net/kd2bd/splat.html