Preserving Incumbent Users' Privacy in Exclusion-Zone-Based Spectrum Access Systems

Yanzhi Dou*, He Li*, Kexiong (Curtis) Zeng*, Jinshan Liu*, Yaling Yang*, Bo Gao† and Kui Ren‡
* Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA
† Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
‡ Department of Computer Science and Engineering, University at Buffalo, State University of New York, Buffalo, USA
Email: {yzdou, heli, kexiong6, jinshan, yyang8}@vt.edu, gaobo@ict.ac.cn, kuiren@buffalo.edu

Abstract—**Dynamic spectrum access (DSA) technique has emerged as a fundamental approach to mitigate the spectrum scarcity problem. As a key form of DSA, the government is proposing to release more federal spectrum for sharing with commercial wireless users. However, the flourish of federal-commercial sharing hinges upon how the federal privacy is managed. In current DSA proposals, the sensitive exclusion zone (E-Zone) information of federal incumbent users (IUs) needs to be shared with a spectrum access system (SAS) to realize spectrum allocation. However, SAS is not necessarily trust-worthy for holding the sensitive IU E-Zone data, especially considering that FCC allows some industry third parties (e.g., Google) to operate SAS for better efficiency and scalability. Therefore, the current proposals dissatisfy the IUs' privacy requirement. To address the privacy issue, this paper presents an IU-privacy-preserving SAS (IP-SAS) design, which realizes the spectrum allocation process through secure computation over ciphertext based on homomorphic encryption so that none of the IU E-Zone information is exposed to SAS. This paper also proposes mechanisms to prevent malicious parties from compromising IP-SAS. We prove the privacy-preserving properties of IP-SAS and demonstrate the scalability and practicality of IP-SAS using experiments based on real-world data. Evaluation results show that IP-SAS can respond an SU's spectrum request in 1.25 seconds with communication overhead of 17.8 KB.**

Index Terms—**Dynamic spectrum access; exclusion zone; privacy; homomorphic encryption**

I. Introduction

The depletion of wireless spectrum has greatly advanced the development of dynamic spectrum access (DSA). In DSA, unlicensed secondary users can opportunistically access licensed spectrum under the constraint that secondary users (SUs) and incumbent users (IUs) do not impose harmful interference to each other. This can greatly improve spectrum utilization and extend regular services.

Recent proposals recommended to set up a server-driven spectrum access system (SAS) to manage the spectrum sharing between IUs and SUs, which also been reflected in the President's Council of Advisors on Science and Technology (PCAST) report [1]. Two popular types of interference management methods – exclusion zone and protection zone – are being explored for SAS in current literatures [2], [3]. In this paper, we focus on SAS systems that use exclusion zone (E-Zone) enforcement method. E-Zone method is the one adopted by most of the current SAS proposals, such as FCC's recent proposal for DSA in 3.5 GHz [2]. The E-Zone of an IU is a spatial region around the IU where SUs are forbidden to operate to avoid mutual interference with the IU.

One of the critical concerns of the above SAS-driven spectrum sharing between federal government incumbent systems and non-government systems is the IUs' privacy issue. For national security reasons, operation information of government IUs is often classified data. For example, the IUs in 3.5 GHz DSA band include military and fixed satellite service licensees, whose operation data is highly sensitive [2]. Another example is AWS-3 band, where IUs include some government systems [3].

Yet, to realize efficient spectrum access, current database-driven designs require IUs to send their E-Zone data to SAS for spectrum allocation. Such E-Zone data can be analyzed to obtain rich sensitive operation information of IUs, such as approximate location, time duration of operation, operating frequency channel, sensitivity level to interference, etc. It exposes the IUs to potentially severe privacy violation since the SAS system is not necessarily trust-worthy for holding such sensitive E-Zone data. For example, according to FCC report [4], SAS may be operated by some commercial third parties to enhance its efficiency and scalability. To respond this call, Google has already developed the third generation of its 3.5GHz SAS prototype [5]. Even if the operator of the SAS system is trusted, it may be breached by adversaries (e.g. intrusions, malwares, insider attacks, insider theft). In such a case, an adversary would have access to all IU operation information.

In this paper, we aim to fill the loophole of untrusted SAS's threat to IUs' privacy by developing an IU-privacy-preserving SAS (IP-SAS) system. Our contribution can be summarized at the following four aspects:

• To the best of our knowledge, we are the first to identify and address the privacy issues of IUs in E-Zone-based SAS. Through an efficient secure-multiparty-computation design, our IP-SAS realizes E-Zone-based spectrum allocation in SAS without exposing any information that can potentially lead to IU privacy violation.

• To prevent malicious parties from compromising the IP-SAS system, we extend the basic design to make IP-SAS robust against more complex and realistic malicious attacks. The task of countering malicious attacks is challenging due to IP-SAS's privacy-preserving design, since one cannot

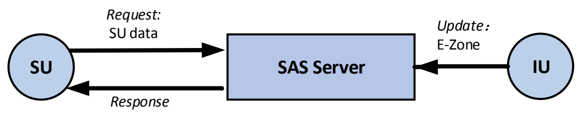

Fig. 1. The architecture of E-Zone-based SAS

easily tell cheating behaviors from information discrepancy. To tackle the challenge, we manipulate the plaintext space to accommodate additive zero-knowledge proof in the secure SAS computation. We also use the deterministic property of decryption algorithm for zero-knowledge proof.

• We leverage the unique properties of IP-SAS computation to develop acceleration mechanisms, which significantly reduce IP-SAS's computation and communication overhead.

• We conduct extensive experiments based on real-world data to demonstrate the scalability and practicality of IP-SAS.

The rest of this paper is organized as follows. Section II introduces the system and adversary model, our design goals, and some preliminaries of homomorphic encryption. In Section III, we present the basic IP-SAS design against semi-honest adversary model. After that, we extend the basic design to defend against malicious adversaries in Section IV. In Section V, we propose acceleration methods to improve IP-SAS's efficiency. We run experiments to evaluate the system and the evaluation results are shown in Section VI. Section VII overviews the related work. Finally, we conclude this paper in Section VIII.

II. Problem Statement

A. System Model

We consider an E-Zone-based SAS involving three parties, as illustrated in Figure 1: IUs, SUs, and SAS Server. SAS Server refers to a cloud-based spectrum management infrastructure that allocates spectrum resources while considering incumbent and secondary operation protection. In the initialization phase, IUs calculate their E-Zones by employing appropriate radio propagation models, and update SAS with the generated E-Zone information. In spectrum computation phase, SAS Server receives SUs' spectrum access request, determines the availability of the spectrum in SUs' location, and returns the corresponding request response to SUs. A spectrum access request contains the identity of the requester, its location information, and operation parameters. A response contains the availability of spectrum channels in SU's location. If for a given spectrum channel, the requesting SU is in the E-Zone of any IU, SAS Server responds the SU with a denial of access to that channel. Otherwise, SAS Server responds with an operation permission.

B. Adversary Models and Design Goals

We consider two adversary models, a basic semi-honest model and a more sophisticated malicious model.

1) Semi-honest adversary model: The basic semi-honest adversary model assumes all the parties in the system are semi-honest (a.k.a. honest-but-curious), which means that all the parties act in an "honest" fashion and exactly follow the protocol as prescribed in Section II-A, but they are also "curious" and attempt to infer private operation data of IUs from the information communicated to them. Essentially, a semi-honest SAS Server can only passively monitor the execution of spectrum allocation process to infer IU's operation information, and cannot actively deviate from the process. The goal of IP-SAS is to realize the SAS process described in Section II-A correctly, while preserving the IU/SU data privacy from the semi-honest SAS Server. In the following, we formally define correctness and privacy of a SAS scheme in the semi-honest model using the simulation paradigm [6]. Specifically, we denote the computation of the SAS process in Section II-A as a functionality f, and denote any SAS scheme for computing f as π. Since f is deterministic, correctness and privacy can be defined separately as follows.

Definition 1. (correctness) We say that π correctly computes f if

$$\{\mathsf{output}_{SUs}(x,y,z,n)\} \overset{c}{\equiv} \{f(x,y,z)\}, \tag{1}$$

where x, y, z are the input data from IUs, SUs, and SAS Server respectively, n is the security parameter, and output_{SUs} is the output of SUs during an execution of π. In our scenario, it is the approval/deny information SUs finally obtain. $\overset{c}{\equiv}$ denotes computationally indistinguishability. Intuitively, in this definition, we say a privacy-preserving SAS scheme π is correct if the output of π, which is defined as the approval/deny response sent to an SU, is the same as the original SAS process defined in Section II-A. Basically, this means that π will not alter the spectrum allocation results.

Definition 2. (privacy) We say that π securely computes f in the presence of semi-honest SAS Server if there exist probabilistic polynomial-time algorithms, denoted as S, such that

$$\{S(z, \mathsf{output}_{SAS}(x,y,z,n), n)\} \overset{c}{\equiv} \{\mathsf{view}^{\pi}(x,y,z,n)\}, \tag{2}$$

where view is the view of SAS Server during an execution of π, i.e., the transcript of messages that it receives and its internal states. $\mathsf{output}_{SAS}(x,y,z,n)$ is the output of SAS Server during an execution of π, i.e., the data it finally sends to SUs. The definition essentially says that an SAS scheme is secure in the presence of semi-honest SAS Server if SAS Server cannot obtain any information about IU's operation from the encrypted IU input data and the encrypted spectrum allocation results.

2) Malicious adversary model: Although the assumption of semi-honest adversary model has been widely adopted in related research areas such as cloud security [7], [8], it is often too strong in real world scenarios since it is highly likely that the corrupted parties would deviate from the prescribed protocols by following adversaries' instructions. Therefore, beyond the basic semi-honest model, we also consider sophisticated

TABLE I
PAILLIER CRYPTOSYSTEM

Key generation (KeyGen):
1. Choose two large random prime numbers p and q, ensuring that $\gcd(pq, (p-1)(q-1)) = 1$.
2. Compute $n = pq$, $\lambda = \operatorname{lcm}(p-1, q-1)$.
3. Choose random integer g where $g \in \mathbb{Z}^*_{n^2}$. Compute $\mu = \left(L\left(g^\lambda \mod n^2\right)\right)^{-1} \mod n$ if exists, where $L(x) = \frac{x-1}{n}$.
4. The public key pk is (n, g), and the secret key sk is (λ, μ).

Encryption ($\mathsf{Enc}_{\mathsf{pk}}$):
1. Let m be a message to be encrypted where $m \in \mathbb{Z}_n$.
2. Select random γ in $\mathbb{Z}_n$, i.e. $\gamma \xleftarrow{\$} \mathbb{Z}_n$.
3. Encrypt m and get its ciphertext $\widehat{m}$ by $\widehat{m} = \mathsf{Enc}_{\mathsf{pk}}(m, \gamma) = g^m \cdot \gamma^n \mod n^2$.

Decryption ($\mathsf{Dec}_{\mathsf{sk}}$):
1. Decrypt $\widehat{m}$ and get its plaintext m by $m = \mathsf{Dec}_{\mathsf{sk}}(\widehat{m}) = L\left(\widehat{m}^\lambda \mod n^2\right) \cdot \mu \mod n$.

Homomorphic Addition ($\mathsf{Add}_{\mathsf{pk}}$):
$\mathsf{Dec}_{\mathsf{sk}}\left(\mathsf{Add}_{\mathsf{pk}}(\widehat{m_1}, \widehat{m_2})\right) = \mathsf{Dec}_{\mathsf{sk}}\left(\widehat{m_1} \cdot \widehat{m_2}\right) = m_1 + m_2$.

malicious adversary model, where corrupted parties can behave maliciously by deviating from the protocol. Specifically, we identify some critical malicious attacks from corrupted parties and provide effective countermeasures for these attacks.

C. Preliminaries on Additive-Homomorphic Cryptosystem

An additive-homomorphic encryption scheme is a public-key cryptosystem $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Add})$ that allows addition operation Add on ciphertexts and generates an encrypted result, when decrypted, corresponding to the sum of the plaintexts. The design of IP-SAS system leverages the homomorphic addition operations of additive-homomorphic cryptosystem and can work with any such cryptosystem, including Benaloh [9], Okamoto-Uchiyama cryptosystem [10], Paillier [11], etc. Without loss of generality, we choose Paillier cryptosystem to discuss and implement IP-SAS system due to the wide-availability of its off-the-shelf implementations. Paillier cryptosystem is a probabilistic encryption mechanism by introducing a random value γ at each encryption to ensure that encrypting the same message several times will yield different ciphertexts. Details of Paillier cryptosystem are illustrated in Table I.

III. SEMI-HONEST SYSTEM DESIGN

In this section, we first present the basic design of IP-SAS against the semi-honest adversary model, and then prove the correctness and privacy of the design.

A. Design Overview

As shown in Figure 2, IP-SAS involves four parties: (1) a SAS Server $\mathcal{S}$ for spectrum allocation, (2) IUs, (3) SUs, and (4) a Key Distributor $\mathcal{K}$. $\mathcal{K}$ creates a Paillier public/private key pair $(\mathsf{pk}, \mathsf{sk})$ and is trusted for keeping sk a secret only known to itself. In the real world, $\mathcal{K}$ is operated by IUs by default. Nevertheless, if IUs are unwilling to play the role, the duty of $\mathcal{K}$ can be delegated to some trusted authority such as FCC or NTIA. The high-level idea of IP-SAS is that IUs first

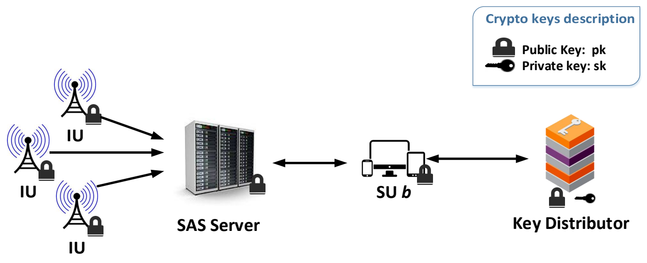

Fig. 2. IP-SAS overview.

encrypt their E-Zone data using pk before sending it to $\mathcal{S}$, and then $\mathcal{S}$ performs the spectrum allocation computation on the encrypted E-Zone data based on the homomorphic properties of Paillier cryptosystem. The protocol of IP-SAS is shown in Table II. In the following, we describe the details of each step.

TABLE II
THE PROTOCOL OF BASIC IP-SAS SYSTEM UNDER SEMI-HONEST MODEL

I. Initialization Phase:
$\mathcal{K}$:
(1) $\mathcal{K}$ runs KeyGen and generates a Paillier key pair $(\mathsf{pk}, \mathsf{sk})$. pk is distributed to $\mathcal{S}$ and IUs, and sk is kept secret.
IUs (numbered as $1, 2, ..., k, ..., K$):
(2) IU k calculates its E-Zone map $\mathbf{T}_k$.
(3) IU k encrypts $\mathbf{T}_k$ with pk and gets $\widehat{\mathbf{T}}_k$.
(4) IU k uploads $\widehat{\mathbf{T}}_k$ to $\mathcal{S}$.
$\mathcal{S}$:
(5) Upon all IUs having uploaded their E-Zone map, $\mathcal{S}$ computes $\widehat{\mathbf{M}} := \oplus_{k \in \{1,2,...,K\}} \widehat{\mathbf{T}}_k$ to aggregate the E-Zone map of all IUs and generates a global E-Zone map $\widehat{\mathbf{M}}$.

II. Spectrum Computation Phase:
SU b:
(6) SU b submits spectrum request containing its operation parameters $(h_s, p_{ts}, g_{rs}, i_s)$ and location l to $\mathcal{S}$.
$\mathcal{S}$:
(7) $\mathcal{S}$ retrieves the corresponding entry in the global E-Zone map $\widehat{\mathbf{M}}$ and obtains $\widehat{\mathbf{X}}_b$.
(8) $\mathcal{S}$ adds random blinding factor $\widehat{\boldsymbol{\beta}}$ to $\widehat{\mathbf{X}}_b$ to generate $\widehat{\mathbf{Y}}_b$.
(9) $\mathcal{S}$ returns $\widehat{\mathbf{Y}}_b$ and $\boldsymbol{\beta}$ to SU b.

III. Recovery Phase:
SU b:
(10) SU b relays $\widehat{\mathbf{Y}}_b$ to $\mathcal{K}$ for decryption.
$\mathcal{K}$:
(11) $\mathcal{K}$ decrypts $\widehat{\mathbf{Y}}_b$ with sk and returns $\mathbf{Y}_b$ to SU b.
SU b:
(12) SU b recovers $\mathbf{X}_b$ by removing the blinding factor $\boldsymbol{\beta}$ from $\mathbf{Y}_b$.

Design concept: The traditional SAS frameworks do not include Key Distributor $\mathcal{K}$, and we import this new component in our design to address the IUs' privacy threat in the traditional SAS frameworks. In security field, it is quite common to import new component to existing frameworks to solve security and privacy issues. For example, to secure the communication over Internet, a new component known as certificate authority (CA) is imported as root of trust for HTTPS connections. Similarly, we use $\mathcal{K}$ as root of trust in our privacy-preserving SAS scheme. The reason why $\mathcal{K}$ is trusted is that it is completely under the control of IUs like the DoD in

3.5 GHz band, or government authorities like FCC and NTIA. By all means, they will guarantee the security of $\mathcal{K}$, such as enforcing the desired security level of encryption scheme, hosting the software on highly protected servers, and even checking the background of software maintainers. Compared with the high security standard for $\mathcal{K}$, SAS Server $\mathcal{S}$, operated by commercial companies like Google, is very unlikely to have the same level of protection. Thus, it is untrusted in our assumption.

B. E-Zone Information Generation & Representation

To reduce unnecessary wastes of spectrum resources, IUs have to compute E-Zone boundaries accurately. Following the recommendations in [12], we assume that the E-Zones computed by an IU can have multiple tiers. Each tier corresponds to the E-Zone for SUs with a specific operation parameter setting. An SU operation parameter setting is a tuple $(f_s, h_s, p_{ts}, g_{rs}, i_s)$. Similarly, an IU operation parameter setting is also a tuple $(f_i, h_i, p_{ti}, g_{ri}, i_i)$. Descriptions of the operation parameters are presented in Table III.

TABLE III
SU/IU OPERATION PARAMETER DESCRIPTIONS

f_s, f_i	Operation frequency
h_s, h_i	Antenna height
p_{ts}, p_{ti}	Transmitter effective radiated power
g_{rs}, g_{ri}	Receiver antenna gain
i_s, i_i	Receiver interference tolerance threshold

Plug SU and IU's operation parameter settings into a sophisticated radio propagation model that incorporates terrain details, IU can accurately compute its E-Zone for SU of the specified setting. Specifically, denoting SU's location by l and assume $f_s = f_i$, the E-Zone with respect to SU setting $(f_s, h_s, p_{ts}, g_{rs}, i_s)$ is defined by:

$$EZ(f_s, h_s, p_{ts}, g_{rs}, i_s) := \{l | p_{ti} a_{is} g_{rs} \geq i_s \text{ or } p_{ts} a_{is} g_{ri} \geq i_i\}$$

where a_{is} is the path attenuation between IU and SU and is a function of SU and IU's locations, SU and IU antenna heights (h_s, h_i), SU/IU shared frequency f_s, and terrain data. The E-Zone information that an IU provides SAS includes the set of $EZ(f_s, h_s, p_{ts}, g_{rs}, i_s)$ for all considered SU settings.

To reduce computation complexity, we quantize all the operation parameters into discrete levels. So each IU k in IP-SAS captures its multi-tier E-Zone information using a multidimensional E-Zone map matrix $\mathbf{T}_k := \{T_k(l, f, h_s, p_{ts}, g_{rs}, i_s)\}_{L \times F \times H_s \times P_{ts} \times G_{rs} \times I_s}$, where

$$T_k(l, f, h_s, p_{ts}, g_{rs}, i_s) := \begin{cases} \epsilon, & \text{grid } l \in EZ(f, h_s, p_{ts}, g_{rs}, i_s) \\ 0, & \text{grid } l \notin EZ(f, h_s, p_{ts}, g_{rs}, i_s) \end{cases} \tag{3}$$

where ϵ is a positive random number used to denote that SU is in the E-Zone of IU and should not be allowed to transmit, and 0 means SU is out of the E-Zone of this IU and should be allowed to transmit.

To protect IU privacy against untrusted $\mathcal{S}$, each entry of $\mathbf{T}_k$ is encrypted by pk before sending to $\mathcal{S}$. For notation simplicity, given any plaintext m, we denote $\widehat{m}$ as its ciphertext created using pk. In essence, IU k transmits $\widehat{\mathbf{T}}_k := \{\widehat{T}_k(l, f, h_s, p_{ts}, g_{rs}, i_s)\}_{L \times F \times H_s \times P_{ts} \times G_{rs} \times I_s}$ when updating $\mathcal{S}$ with its E-Zone map.

C. E-Zone Map Aggregation in $\mathcal{S}$

Assume that there are altogether K IUs registered in $\mathcal{S}$ and all of them have sent in their E-Zone map. The first step of $\mathcal{S}$ is to aggregate each IU's E-Zone map to create a global E-Zone map $\widehat{\mathbf{M}} := \{\widehat{M}(l, f, h_s, p_{ts}, g_{rs}, i_s)\}_{L \times F \times H_s \times P_{ts} \times G_{rs} \times I_s}$ by $\widehat{\mathbf{M}} := \oplus_{k \in \{1,2,\ldots,K\}} \widehat{\mathbf{T}}_k$, which means each entry of $\widehat{\mathbf{M}}$ is:

$$\widehat{M}(l, f, h_s, p_{ts}, g_{rs}, i_s) := \oplus_{k \in \{1,2,\ldots,K\}} \widehat{T}_k(l, f, h_s, p_{ts}, g_{rs}, i_s), \tag{4}$$

where $\oplus$ is the homomorphic version of summation symbol $\sum$. From formula (3), it is easy to see that $M(l, f, h_s, p_{ts}, g_{rs}, i_s) = 0$ means that the grid l is out of E-Zones of all IUs, and $M(l, f, h_s, p_{ts}, g_{rs}, i_s) > 0$ means that the grid l is within the E-Zone of at least one IU. Formally, for an SU at location l with operation parameter setting $(h_s, p_{ts}, g_{rs}, i_s)$,

$$M(l, f, h_s, p_{ts}, g_{rs}, i_s) \begin{cases} = 0, & \text{Spectrum } f \text{ is available.} \\ > 0, & \text{Spectrum } f \text{ is unavailable.} \end{cases} \tag{5}$$

D. Spectrum Computation Phase & Recovery Phase

When an SU b needs to access the spectrum, it submits a spectrum request containing its operation parameters $(h_s, p_{ts}, g_{rs}, i_s)$ and location l in plaintext to $\mathcal{S}$. $\mathcal{S}$ retrieves the corresponding entries in the global E-Zone map $\widehat{\mathbf{M}}$ and generates $\widehat{\mathbf{X}}_b := \{\widehat{X}_b(f)\}_F$, where

$$\widehat{X}_b(f) := \widehat{M}(l, f, h_s, p_{ts}, g_{rs}, i_s). \tag{6}$$

Essentially, $\widehat{\mathbf{X}}_b$ holds the information of spectrum availability in SU b's location. $\mathcal{S}$ then homomorphically adds some random blinding factors $\widehat{\boldsymbol{\beta}} := \{\widehat{\beta}(f)\}_F$ to obfuscate the results as follows:

$$\widehat{Y}_b(f) := \mathsf{Add}_{\mathsf{pk}}(\widehat{X}_b(f), \widehat{\beta}(f)), \tag{7}$$

where the plaintext of the blinding factor $\beta(f)$ is a one-time random number. $\widehat{\mathbf{Y}}_b$ and the plaintext $\boldsymbol{\beta}$ are both sent back to SU b. SU b needs to decrypt $\widehat{\mathbf{Y}}_b$ for the channel availability information, so it sends $\widehat{\mathbf{Y}}_b$ to $\mathcal{K}$ for decryption. $\mathcal{K}$ decrypts every entry of $\widehat{\mathbf{Y}}_b$ using sk and gets $\mathbf{Y}_b$. $\mathcal{K}$ cannot infer the channel availability information from $\mathbf{Y}_b$ since it does not have the values of the blinding factors $\boldsymbol{\beta}$ to recover $\mathbf{X}_b$. In this way, SU b's spectrum allocation result is kept secret from $\mathcal{K}$.

$\mathcal{K}$ sends $\mathbf{Y}_b$ to SU b, which uses $\boldsymbol{\beta}$ to recover the correct spectrum computation results $\mathbf{X}_b$ as follows:

$$X_b(f) = Y_b(f) - \beta(f). \tag{8}$$

From formulas (5) and (6), it can be seen that if $X_b(f)$ is a positive number, it means that SU b cannot access channel f. If $X_b(f)$ is 0, SU b is permitted to access channel f.

E. Correctness and Privacy Analysis

1) Correctness: It is straightforward to see that, if the underlying Paillier cryptosystem is correct, the protocol in Table II correctly performs the SAS process described in Section II-A.

2) Privacy: We now analyze the privacy of IP-SAS under the semi-honest adversary model.

Claim 1. *In IP-SAS, it is computationally infeasible for $\mathcal{S}$ to distinguish an IU's operation information as long as Paillier cryptosystem is semantically secure, blinding factors are properly generated.*

We provide a sketch of proof. Claim 1 can be proved using composition theorem (under semi-honest adversary model) [6] by evaluating the security of each step of IP-SAS. Basically, according to composition theorem, if each step of IP-SAS is secure, then the whole IP-SAS system is secure. The operation steps of IP-SAS in Table II are based on the operations $\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Add}$ of Paillier cryptosystem, thus these operations are secure since Paillier cryptosystem is semantically secure [11]. For the blinding factor scheme in step (8)-(10), which are independent of the Paillier operations, the security is guaranteed as long as for all probabilistic polynomial-time adversaries, the probability to distinguish the blinded message is negligible. The detailed proof of blinding factor scheme can be found in [8], [13].

F. Discussion

If an IU worries that malicious SUs may infer its operation data by analyzing multiple SAS's spectrum responses, the IU can add obfuscation noises to its E-Zone data $\mathbf{T}_k$ as follows:

$$T_k(l, f, h_s, p_{ts}, g_{rs}, i_s) \leftarrow T_k(l, f, h_s, p_{ts}, g_{rs}, i_s) + \phi, \quad (9)$$

where ϕ is the noise. Some preliminary noise generation techniques for E-Zone are proposed in [14] for traditional SAS. Note that these obfuscation techniques for traditional SAS are fully compatible with our IP-SAS design since they only affect $\mathbf{T}_k$ by noise addition, and the following process of IP-SAS stays the same. As pointed out by [14], the potential downside of such obfuscation techniques is the lowered spectrum utilization efficiency due to the added noise. We will investigate the right balance between obfuscation effectiveness and spectrum efficiency in the future.

It is also worth mentioning that, although the basic IP-SAS design does not protect SUs' privacy against untrusted SAS, IP-SAS is ready to apply the similar PIR techniques as [15] to address the issue. Specifically, when an SU submits the spectrum access request, it needs to indicate its location information and operation parameters to retrieve the right E-Zone entry from $\mathcal{S}$. By adopting PIR, the SU can still retrieve the right E-Zone entry without revealing its location information and operation parameters to $\mathcal{S}$. In this way, the SU's privacy is preserved.

IV. Countering Malicious Attacks

In this section, we extend the basic IP-SAS design to defend against the malicious attacks. Defending against the malicious attacks is non-trivial because IP-SAS keeps information known by each party to the minimum to protect IU privacy. Thus, one cannot easily tell cheating behaviors from information discrepancy. In this section, we will show how an attacker can compromise the basic IP-SAS under the malicious adversary model and what are the practical countermeasures. Note that IUs and $\mathcal{K}$ are assumed as trusted parties and are not malicious. Table IV presents the improved protocol of IP-SAS under the malicious adversary model.

TABLE IV
The protocol of IP-SAS system under malicious adversary model

I. Initialization Phase:
$\mathcal{K}$:
(1) $\mathcal{K}$ runs KeyGen and generates a Paillier key pair $(\mathsf{pk}, \mathsf{sk})$. pk is distributed to $\mathcal{S}$ and IUs, and sk is kept secret.
IUs (numbered as $1, 2, ..., k, ..., K$):
(2) IU k calculates its E-Zone map $\mathbf{T}_k$.
(3) For each entry e_k of $\mathbf{T}_k$, IU k computes the commitment c_{ek} to it and records the random factor r_{ek}. e_k and r_{ek} are put into one Paillier plaintext to generate w_k. The commitment c_{ek} is published. When all the entries of $\mathbf{T}_k$ is processed, IU k obtains $\mathbf{W}_k$.
(4) IU k encrypts $\mathbf{W}_k$ with pk and gets $\widehat{\mathbf{W}}_k$.
(5) IU k sends $\widehat{\mathbf{W}}_k$ to $\mathcal{S}$.
$\mathcal{S}$:
(6) Upon all IUs having uploaded their $\widehat{\mathbf{W}}_k$, $\mathcal{S}$ computes $\widehat{\mathbf{M}} := \oplus_{k \in \{1,2,...,K\}} \widehat{\mathbf{W}}_k$.
II. Spectrum Computation Phase:
SU b:
(7) SU b generates spectrum request containing its operation parameters $(h_s, p_{ts}, g_{rs}, i_s)$ and location l and signs it. SU b submits the spectrum request and the signature to $\mathcal{S}$.
$\mathcal{S}$:
(8) $\mathcal{S}$ retrieves the corresponding entries in the global E-Zone map $\widehat{\mathbf{M}}$ and obtains $\widehat{\mathbf{X}}_b$.
(9) $\mathcal{S}$ adds random blinding factor $\widehat{\boldsymbol{\beta}}$ to $\widehat{\mathbf{X}}_b$ and generates $\widehat{\mathbf{Y}}_b$.
(10) $\mathcal{S}$ signs $\widehat{\mathbf{Y}}_b$ and $\boldsymbol{\beta}$, and returns $\widehat{\mathbf{Y}}_b$, $\boldsymbol{\beta}$, and the signature to SU b.
III. Recovery Phase:
SU b:
(11) SU b relays $\widehat{\mathbf{Y}}_b$ to $\mathcal{K}$ for decryption.
$\mathcal{K}$:
(12) $\mathcal{K}$ decrypts $\widehat{\mathbf{Y}}_b$ and gets $\mathbf{Y}_b$.
(13) $\mathcal{K}$ computes the random numbers $\{\gamma\}$ to generate $\{\widehat{Y}_b(f)\}$ from $\{Y_b(f)\}$ as a proof.
(14) $\mathcal{K}$ sends $\mathbf{Y}_b$ and $\{\gamma\}$ to SU b.
SU b:
(15) SU b recovers $\mathbf{X}_b$ by removing the blinding factor $\boldsymbol{\beta}$ from $\mathbf{Y}_b$.
(16) SU b verifies whether $\mathbf{X}_b$ is correctly computed in $\mathcal{S}$ using formula (10).

A. Malicious SUs

Attack: SU b is involved in step (6) and (10) of Table II. By deviating from step (6), a malicious SU b can obtain a fake $\mathbf{X}'_b$ that does not correspond to its true operation parameters and location through submitting faked data in the spectrum request. A malicious SU b can also directly claim an $\mathbf{X}'_b$ that

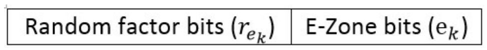

Fig. 3. The format of an entry w_k in $\mathbf{W}_k$

is different from the spectrum computation result calculated by $\mathcal{S}$ by deviating from step (10).

These malicious behaviors can be leveraged by SU b to gain illegal benefits. For example, even if SU b is denied access by $\mathcal{S}$, it can claim the opposite by tempering the spectrum computation result. Without proper extension to IP-SAS, this cheating SU cannot be caught since SU is the only party that knows the true value of $\mathbf{X}_b$. $\mathcal{S}$ and $\mathcal{K}$ are both unable to know $\mathbf{X}_b$'s value due to IP-SAS's IU-privacy-preserving design.

Countermeasure (Step (7), (10), and (13) in Table IV): The countermeasure is based on digital signature system.

To verify whether SU b submits faked data in the spectrum request, a verifier requests SU b to sign its spectrum request before submitting it to $\mathcal{S}$. The verifier can verify the spectrum request by measuring the operation parameters and location of SU b in the field and comparing them with those contained in the spectrum request. The non-repudiation property of digital signature ensures that SU b cannot deny its cheating behavior after been caught.

To verify whether SU b claims a different $\mathbf{X}'_b$ in the final spectrum allocation results, $\mathcal{S}$ is requested to sign $\widehat{\mathbf{Y}}_b$ and $\boldsymbol{\beta}$ and send the signature to SU b as well. The signature ensures that SU b cannot modify $\widehat{\mathbf{Y}}_b$ and $\boldsymbol{\beta}$. Then, based on formula (8), the verifier can compute a $\mathbf{Y}'_b$ by adding $\mathbf{X}'_b$ to $\boldsymbol{\beta}$. Finally, if the verifier is able to prove that $\mathbf{Y}'_b$ is not the decryption of $\widehat{\mathbf{Y}}_b$, then it can expose SU b's cheating behavior. Note that the verifier cannot simply decrypt $\widehat{Y}_b(f)$ and compare the decryption with $Y'_b(f)$, since it does not hold the secret key sk. Instead, we propose a zero-knowledge proof for $Y'_b(f) \neq \mathsf{Dec}_{\mathsf{sk}}(\widehat{Y}_b(f))$. Firstly, the verifier requests $\mathcal{K}$ to provide the random number γ for generating $\widehat{Y}_b(f)$ (refer to the Enc operation in Table I). Then, the verifier verifies whether $Y'_b(f)$ is the decryption of $\widehat{Y}_b(f)$ by re-encrypting $Y'_b(f)$ using pk and the random number γ provided by $\mathcal{K}$.

B. Malicious $\mathcal{S}$

Attack: $\mathcal{S}$ is involved in step (5), (7), (8), (9) of the protocol described in Table II. By deviating from these steps, a malicious $\mathcal{S}$ can create a wrong output $\mathbf{Y}'_b$ and/or $\boldsymbol{\beta}'$, which will cause SU b to recover a wrong spectrum computation result $\mathbf{X}'_b$. For example, the malicious $\mathcal{S}$ may alter IU k' E-Zone map by modifying the entries of $\widehat{\mathbf{T}}_k$. Alternatively, the malicious $\mathcal{S}$ may incorrectly execute the aggregation operation in formula (4), such as intentionally omitting the E-Zone map of some IU, or involving the E-Zone map of some IU in the aggregation computation for multiple times. The malicious $\mathcal{S}$ may also retrieve wrong entries in $\widehat{\mathbf{M}}$ for SU, i.e. entries that are not associated with SU's operation parameters or location.

Countermeasure (step (3) and (16) in Table IV): To guard against malicious $\mathcal{S}$, SUs should be able to verify whether the computation is correctly performed in $\mathcal{S}$ based on the true SU and IU input data. To make the computation in $\mathcal{S}$ verifiable, we propose a novel method based on Pedersen commitment scheme [16]. Pedersen commitment scheme consists of three phases, commitment setup phase (**Setup**), commit phase (**Commit**), and open phase (**Open**). **Setup** generates the parameters par of the commitment scheme. **Commit**(par, r_x, x) outputs a commitment c_x to x, where r_x is a random input factor. **Open**(par, c_x, x, r_x) outputs $accept$ if c_x is the commitment to x, otherwise outputs $deny$. Pedersen commitment scheme is additive-homomorphic, which means that given two commitments c_{x1} and c_{x2} to x_1 and x_2 with the corresponding random factors r_{x1} and r_{x2} respectively, **Open**$(par, c_{x1} * c_{x2}, x_1 + x_2, r_{x1} + r_{x2})$ outputs $accept$. In the following, we employ Pedersen commitment scheme to verify the computation in $\mathcal{S}$.

The commitment computation is performed by an IU k after it finishes calculating its E-Zone map $\mathbf{T}_k$ (step (2) in Table II) and before encrypts it using pk (step (3) in Table II). For each entry e_k in $\mathbf{T}_k$, IU k computes a commitment c_{ek} to e_k, records the corresponding random factor r_{ek}, and publishes c_{ek}. Then, as shown in Figure 3, IU k partitions the space of a Paillier plaintext message w_k into two segments and puts the random factor r_{ek} in the first segment and $\mathbf{T}_k$ entry e_k in the second segment, essentially making $w_k = \langle r_{ek} || e_k \rangle$. These Paillier plaintext messages created in this way form a new matrix $\mathbf{W}_k$ that carries both $\mathbf{T}_k$ and the random factors of $\mathbf{T}_k$ entries' commitments. Note that to ensure 112-bit security level, Paillier cryptosystem has to use a large 2048-bit plaintext size, which is more than enough to accommodate both r_{ek} and e_k without overlapping. All the IUs use the same way to partition the plaintext message. Then, IU k encrypts the plaintext entries of $\mathbf{W}_k$ and sends $\widehat{\mathbf{W}}_k$ to $\mathcal{S}$. When all IUs finish uploading their $\widehat{\mathbf{W}}_k$, $\mathcal{S}$ homomorphically aggregates $\widehat{\mathbf{W}}_k$ from all IUs to generate the global E-Zone map $\widehat{\mathbf{M}}$, where the random factors of different IUs are also aggregated and stored in $\widehat{\mathbf{M}}$. Due to the very large size of Paillier plaintext message, the bit length reserved of both the r_{ek} and e_k segments in the Paillier plaintext is long enough so that the aggregation in each segment will not cause overflow.

At the end of the recovery phase, an SU b obtains the final spectrum computation results $\mathbf{X}_b$. Consider the entry $X_b(f)$ in $\mathbf{X}_b$ for a given frequency channel f. The first segment of $X_b(f)$ is the aggregation of random factor bits from IU's input matrix $\widehat{\mathbf{W}}_k$, denoted as $R := \sum_{k \in \{1,2,\ldots,K\}} r_{ek}$, and the second part stores the aggregation of E-Zone bits, denoted as $E := \sum_{k \in \{1,2,\ldots,K\}} e_k$. We use a subscript k to denote that r_{ek} and e_k are from IU k. SU b then finds out all the commitments published by IUs that are corresponding to the index $(l, f, h_s, p_{ts}, g_{rs}, i_s)$, where $(l, h_s, p_{ts}, g_{rs}, i_s)$ are the operation parameters provided by SU b in its spectrum request. These commitments are denoted as $c_{e1}, c_{e2}, ..., c_{eK}$. According to additive-homomorphic property of Pedersen commitment scheme,

$$\textbf{Open}(par, c_{e1} * c_{e2} * ... * c_{eK}, E, R) \qquad (10)$$

should output $accept$ if all the computation in $\mathcal{S}$ is correct.

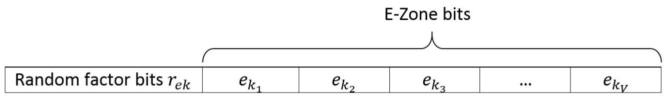

Fig. 4. Packing V entries of E-Zone map into a plaintext message w_k

Otherwise, the computation in $\mathcal{S}$ is incorrect.

V. Improving SAS Efficiency

In this section, we adopt two techniques, ciphertext packing and parallel computing, to imporve IP-SAS's efficiency.

A. Ciphertext Packing

Instead of encrypting every entry of E-Zone map individually, the packing technique packs multiple entries into one ciphertext message, so that we not only reduce the amount of bits that need to be exchanged between different parties, but also achieve homomorphic addition and encryption of multiple entries with only one addition/encryption operation. Ciphertext packing technique [17] is feasible for Paillier crytosystem since the plaintext space of Paillier cryptosystem is significantly larger than the space that is needed for E-Zone entries. To adopt ciphertext packing, we can modify step (3) in Table IV as follows. As shown in Figure 4, IU k packs V entries of $\mathbf{T}_k$ into one Paillier plaintext message w_k. Then, IU k computes the Pedersen commitment to the entire packed entries $e_{k1}||e_{k2}||e_{k3}||...||e_{kV}$ and puts the random factor of the commitment in the leftmost segment of w_k. Using this packing technique, we can reduce the dimension of $\widehat{\mathbf{W}}_k$ and the number of homomorphic addition/encryption operations by a factor of V.

One undesirable side effect of adopting the packing technique is that $\mathbf{X}_b$ contains E-Zone entries that are not related to SU's spectrum request. This may cause unnecessary leakage of IU information when SU obtains $\mathbf{X}_b$ in step (15) of Table IV. To solve this problem, $\mathcal{S}$ needs to perform an extra masking step before step (9) in Table IV. In the masking step, $\mathcal{S}$ homomorphically adds a random mask to hide irrelevant entries in $\mathbf{X}_b$.

B. Parallel Computing

In the initialization phase of IP-SAS, all the computation tasks of IUs and $\mathcal{S}$ are readily to be parallelly computed to further reduce the computation overhead. Specifically, in Table IV, E-Zone map generation (step (2)), commitments computation (step (3)), encryption (step (4)), and aggregation (step (6)) can be parallelly computed by splitting the matrices into subsets. The computation tasks of the subsets are distributed to multiple threads and servers. Moreover, for the spectrum computation phase and recovery phase, $\mathcal{S}$ and $\mathcal{K}$ can handle multiple SUs' request concurrently.

VI. Experiments and Evaluation

A. Implementation and Experiment Settings

We set the service area of SAS to be a large 154.82 km^2 area in Washington DC, and all the IUs need to generate the E-Zone maps on this area. We employ the sophisticated Longley-Rice (L-R) radio propagation model provided by SPLAT! [18] to calculate the point-to-point attenuation values for E-Zone computation. In order to obtain accurate attenuation values, we feed high resolution terrain elevation data SRTM3 [19] to SPLAT!.

We implement a Paillier cryptosystem of 112-bit security level by setting the security parameter n as 2048 bits. We build IP-SAS on the Paillier implementation and execute IP-SAS over two Dell desktops, both with Intel Core i7-3770 CPU @3.40GHz and 12GB RAM. The plaintext size the Paillier cryptosystem is 2048-bit long, and the first 1024-bit space is used to store random factor bits of commitment scheme. In the rest 1024-bit space, we pack 20 50-bit E-Zone entries. We distribute the computation tasks to 16 concurrent threads, and each of the two desktops runs 8 threads. Table V shows the experiment parameter settings.

TABLE V
Experiment parameter settings

Number of IUs (K)	500
Number of grids (L)	15482
Number of frequency channels (F)	10
Number of SU antenna heights (H_s)	5
Number of SU effective radiated power values (P_{ts})	3
Number of SU receiver antenna gain values (G_{rs})	3
Number of SU interference tolerance thresholds (I_s)	3

B. Computation and Communication Overhead

Table VI shows the computation overhead of each step in Table IV before/after applying the acceleration methods presented in Section V. After adopting the acceleration methods, an IU can generate an encrypted E-Zone map in 2 hours, compared with 101 hours without acceleration. Note that more than 80% of the 2-hour time is actually spent on the plaintext computation (e.g. E-Zone map calculation using L-R model), and the extra overhead brought by IP-SAS's privacy-preserving operations takes less than 20% of the total time. Also note that E-Zone map calculation does not need to be repeated frequently since IUs' operation parameters are often static. After $\mathcal{S}$ receives the E-Zone data from all IUs, it needs about 5.2 minutes for aggregation with acceleration. In the spectrum computation phase and recovery phase, it takes a short response time of around 1.25 second for an SU to get the final spectrum allocation result after submitting its request.

TABLE VI
Computation Overhead

	Before Acceleration	After Acceleration
(2) E-Zone map calculation	21.2 hours	1.65 hours
(3) Commitment	11.7 hours	3.21 minutes
(4) Encryption	68.5 hours	17.9 minutes
(6) Aggregation	29.0 hours	5.2 minutes
(8)-(10) $\mathcal{S}$ Response	1.12 seconds	1.11 seconds
(12)(13) Decryption	0.134 seconds	0.134 seconds
(15) Recovery	-	-
(16) Verification	0.118 seconds	0.118 seconds

Table VII shows the communication overhead of IP-SAS before/after applying the ciphertext packing technique. In the initialization step, the ciphertext packing technique reduces the amount of encrypted E-Zone data transmitted from an IU to $\mathcal{S}$ by 95% from nearly 10 GB to 510 MB. Note that IUs do not need to provide its E-Zone data frequently since their E-Zones are often static. Moreover, IUs can send the E-Zone data through wired backbone with high data transmission speed, thus the E-Zone data transmission can be finished in short time. In the spectrum computation phase and recovery phase, the communication overhead is 17.8 KB for an SU, which is small enough to satisfy the requirement of both static and mobile SUs in real world.

TABLE VII
COMMUNICATION OVERHEAD

	Before Packing	After Packing
(4) IU$\rightarrow \mathcal{S}$	9.97 GB	510 MB
(6) SU$\rightarrow \mathcal{S}$	25 B	25 B
(9) $\mathcal{S} \rightarrow$SU	7.75 KB	7.75 KB
(10) SU$\rightarrow \mathcal{K}$	5 KB	5 KB
(13) $\mathcal{K} \rightarrow$SU	5 KB	5 KB

VII. RELATED WORK

There are some existing works related to privacy in SAS. But none of them can solve IUs' privacy issues against untrusted E-Zone-based SAS. In [14], to counter the inference attack from malicious SUs, obfuscation techniques were applied to introduce noises in the response to SU's spectrum queries. In [15], private information retrieval (PIR) techniques were adopted to address the location privacy issues of SUs against untrusted SAS. The closest related works to ours are [20], [21], [22], which built protection-zone-based privacy-preserving SAS. The protection-zone method allowed interference from multiple IUs and SUs to be aggregated. Other related work about security of SAS-driven DSA systems aimed at mitigating attacks including malware infection [23], location spoofing [24], and secondary user faking [25]. The preliminary results of this work were presented in [26], [27].

VIII. CONCLUSIONS

In this paper, we build IP-SAS to perform efficient SAS process while preserving IUs' privacy. This system leverages additive-homomorphic encryption to allow secure spectrum allocation computation. Moreover, we design mechanisms to prevent malicious parties from compromising IP-SAS. Finally, we implement acceleration methods to increase IP-SAS's efficiency. Experiments based on real-world data demonstrate the scalability and practicality of IP-SAS in real-world deployment. Evaluation results show that IP-SAS can respond an SU's spectrum request in only 1.25 seconds with communication overhead of 17.8 KB.

ACKNOWLEDGMENTS

This work was supported by the U.S. National Science Foundation under Grant CNS-1228903, Grant CNS-1054697, Grant CNS-1547366, and Grant CNS-1547223.

REFERENCES

[1] PCAST, "Report to the president realizing the full potential of government-held spectrum to spur economic growth," 2012.
[2] FCC, "Amendment of the commission's rules with regard to commercial operations in the 3550-3650 MHz band," *Notice of Proposed Rulemaking and Order, FCC*, 2012.
[3] ——, "Amendment of the commission's rules with regard to commercial operations in the 1695-1710 MHz, 1755-1780 MHz, and 2155-2180 MHz bands," *Notice of Proposed Rulemaking and Order, FCC*, 2014.
[4] ——, "Shared Commercial Operations in the 3550–3650 MHz Band," *Federal Register*, vol. 80, no. 120, June 2015.
[5] " Google's Spectrum Access System Allows Spectrum Sharing," http://www.androidheadlines.com/2015/05/googles-spectrum-access-system-allows-spectrum-sharing.html.
[6] O. Goldreich, *Foundations of cryptography: volume 2, basic applications*. Cambridge university press, 2004.
[7] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *INFOCOM, 2010 Proceedings IEEE*. Ieee, 2010, pp. 1–9.
[8] B. Wang, M. Li, S. S. Chow, and H. Li, "Computing encrypted cloud data efficiently under multiple keys," in *Communications and Network Security (CNS), 2013 IEEE Conference on*. IEEE, 2013, pp. 504–513.
[9] J. Benaloh, "Dense probabilistic encryption," in *Proceedings of the workshop on selected areas of cryptography*, 1994, pp. 120–128.
[10] T. Okamoto and S. Uchiyama, "A new public-key cryptosystem as secure as factoring," in *Advances in Cryptology—EUROCRYPT'98*. Springer, 1998, pp. 308–318.
[11] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *EUROCRYPT'99*. Springer, 1999, pp. 223–238.
[12] A. Ullah, S. Bhattarai, J.-M. Park, J. Reed, D. Gurney, and B. Bahrak, "Multi-Tier Exclusion Zones for Dynamic Spectrum Sharing," in *IEEE ICC*, 2015.
[13] S. S. Chow, J.-H. Lee, and L. Subramanian, "Two-Party Computation Model for Privacy-Preserving Queries over Distributed Databases." in *NDSS*, 2009.
[14] B. Bahrak, S. Bhattarai, A. Ullah, J.-M. Park, J. Reed, and D. Gurney, "Protecting the primary users' operational privacy in spectrum sharing," in *IEEE DySpan*, 2014, pp. 236–247.
[15] Z. Gao, H. Zhu, Y. Liu, M. Li, and Z. Cao, "Location privacy in database-driven cognitive radio networks: Attacks and countermeasures," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 2751–2759.
[16] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Advances in Cryptology—CRYPTO'91*. Springer, 1992, pp. 129–140.
[17] T. Ge and S. Zdonik, "Answering aggregation queries in a secure system model," in *Proceedings of the 33rd international conference on Very large data bases*. VLDB Endowment, 2007, pp. 519–530.
[18] http://www.qsl.net/kd2bd/splat.html.
[19] http://dds.cr.usgs.gov/srtm/version2_1/SRTM3/.
[20] Y. Dou, K. C. Zeng, H. Li, Y. Yang, B. Gao, C. Guan, and S. Li, "P2-SAS: Preserving Users' Privacy in Centralized Dynamic Spectrum Access Systems," *ACM MobiHoc*, 2016.
[21] Y. Dou, K. C. Zeng, H. Li, Y. Yang, B. Gao, K. Ren, and S. Li, "P2-sas: Privacy-preserving centralized dynamic spectrum access system," *IEEE Journal on Selected Areas in Communications*, 2016.
[22] Y. Dou, K. C. Zeng, and Y. Yang, "Poster: Privacy-preserving server-driven dynamic spectrum access system," *ACM MobiCom*, pp. 218–220, 2015.
[23] Y. Dou, K. C. Zeng, Y. Yang, and D. D. Yao, "Madecr: Correlation-based malware detection for cognitive radio," in *IEEE INFOCOM*, 2015, pp. 639–647.
[24] K. Zeng, S. K. Ramesh, and Y. Yang, "Location spoofing attack and its countermeasures in database-driven cognitive radio networks," in *IEEE CNS*, 2014, pp. 202–210.
[25] X. Jin, J. Sun, R. Zhang, and Y. Zhang, "Safedsa: Safeguard dynamic spectrum access against fake secondary users," in *ACM CCS*, 2015, pp. 304–315.
[26] Y. Dou, K. C. Zeng, Y. Yang, and K. Ren, "Preserving incumbent users' privacy in exclusion-zone-based spectrum access systems: poster," *ACM MobiCom*, pp. 473–474, 2016.
[27] Y. Dou, H. Li, K. Zeng, J. Liu, Y. Yang, B. Gao, and K. Ren, "Preserving incumbent users' privacy in server-driven dynamic spectrum access systems," *IEEE ICDCS*, pp. 729–730, 2016.