Preserving Honest/Dishonest Users' Operational Privacy with Blind Interference Calculation in Spectrum Sharing System

Qingqing Cheng, *Student Member, IEEE,* Diep N. Nguyen, *Senior Member, IEEE,*
Eryk Dutkiewicz, *Senior Member, IEEE,* Markus Mueck, *Member, IEEE*

Abstract—Spectrum sharing has been gaining its popular adoption as a potential solution to improve spectrum utilization in future wireless systems. Both Federal Communications Commission (FCC) and European Telecommunications Standards Institute (ETSI) support dynamic spectrum access (DSA) as an enabling technology for spectrum sharing. To effectively realize DSA in practice, users (from both defense and commercial sectors) are required to share their (radio) operational information, which risks exposing their security, privacy, and business plan to unintended agents. Protecting users' operating information is hence the key to DSA's success. In this paper, taking the FCC's spectrum access system (SAS) as a study case, we investigate the operational privacy issue of Incumbent Users (IUs) and honest/dishonest Secondary Users (SUs). For the case of IUs and honest SUs, we propose a privacy-preserving scheme for DSA by leveraging encryption and obfuscation methods (PSEO). To implement PSEO, we introduce an interference calculation scheme that allows users to calculate an interference budget without revealing operational information (e.g., antenna height, transmit power, location...), referred to as the blind interference calculation scheme (BICS). BICS also reduces the computing overhead of PSEO, compared with FCC's SAS by moving interference budgeting tasks to local users and calculating it in an offline manner. To further save the overhead in calculating the interference map, we introduce a quantization method and optimize the grid sizes of the terrestrial area of interest. Additionally, for the case of IUs and dishonest SUs, we propose a "punishment and forgiveness" (PF) mechanism, which draws support from SUs' reputation scores (RSs) and reputation histories (RHs), to encourage SUs to provide truthful information. Theoretical analysis and extensive simulations show that our proposed PSEO and PF-PSEO schemes can better protect all users' operational privacy under various privacy attacks, yielding higher spectrum utilization with less online overhead, compared with state of the art approaches.

Index Terms—5G security, dynamic spectrum access, SAS architecture, privacy issue, blind interference calculation.

1 Introduction

In recent years, spectrum sharing has been considered as a promising candidate to address the spectrum shortage in future wireless systems. In dynamic spectrum access (D-SA), an enabling technology for spectrum sharing, primary owners of the spectrum (also referred to as primary users, PUs or incumbent users, IUs) are protected from adverse interference effect from the spectrum secondary users (SUs) [1]. To protect PUs, the interference induced by SUs is carefully calculated/budgeted. For that purpose, SUs are required to share their up-to-date operational information (such as users' location and transmitting power,...) with a third party [2]. In the spectrum access system (SAS), proposed by Federal Communications Commission (FCC), a SAS administrator [3] serves as the third party. However, the current mechanism to protect IUs in SAS can be too conservative, leading to poor utilization of spectrum sharing. Sharing IUs' operational data with the SAS administrator, hence, can create new opportunities to improve spectrum utilization and mitigate the problem of spectrum shortage, whereas it also causes the exposure of users' private information, e.g., users' location and transmitting power [3].

Since the user's private data is closely related to its business, the leakage of private information would compromise the user's interest and development [4]. For instance, in SAS which is a database-driven spectrum sharing system operating at the frequency band of $3.5-3.7$GHz in the U.S. [3], the IUs are military facilities, and SUs (a.k.a., Priority Access License (PAL) and General Authorized Access (GAA) users) are commercial institutions. In such a case, the exposure of IUs' and SUs' information would result in serious threats to national security and commercial interests, respectively [5]. For these reasons, protecting user's operational data/information during the process of spectrum sharing is the key to success of any DSA systems, including SAS or the Licensed Spectrum Access (LSA), proposed by European Telecommunications Standards Institute (ETSI) [6].

The efforts in the literature can be categorized into three main types: obfuscation methods [7, 8], anonymity methods [9, 10] and cryptosystems [11, 12]. Specifically, in [8], SU's location is concealed by adding a blind factor to the actual location. In [10], the authors preserve user's operational information by leveraging anonymity, clustering and perturbation algorithms. Although schemes in [8] and [10] have achieved great progress regarding protecting user's precise location or operational information, they inherit weaknesses from anonymity and perturbation algorithms

- *Q. Cheng, D. Nguyen, and E. Dutkiewicz are with the School of Electrical and Data Engineering, University of Technology Sydney, Australia.*
E-mail: {qingqing.cheng, diep.nguyen, eryk.dutkiewicz}@uts.edu.au
- *M. Mueck is with Intel Deutschland GmbH, Germany.*
E-mail: markus.dominik.mueck@intel.com

(e.g., revealing user's rough direction or group information to potential adversaries). To address this problem, authors in [12] recruit the *homomorphic encryption* to protect IUs information. However, the proposed approach in [12] requires an additional entity to the SAS architecture, called the key distributor. The introduction of the key distributor does not honor the original SAS's architecture of FCC. Additionally, a large number of operations in the encrypted domain in [12] (refer to Table 1) with high complexity would prohibit [12] from practical implementation.

Additionally, all the above works assumed that users are always honest, i.e., users report their truthful information and comply with what they report. However, such an assumption is not really realistic. For example, in order to increase the probabilities of getting approval requests from the SAS administrator, some SUs may report lower values than their actual transmitting powers in their requests [13]. These dishonest/greedy behaviors would make the SAS administrator grant requests that can cause adverse interference to corresponding IUs or other SUs sharing the same spectrum source [13]. Therefore, investigating the privacy issue considering dishonest SUs is practically essential to promote effective and accurate spectrum allocations in SAS. However, this issue has not been sufficiently addressed.

Given the above, this work aims to provide practical methods to protect the operational information of IUs and honest/dishonest SUs in SAS architectures. It is important to note that, in this work, we don't study the resource allocation but focus on realizing the results of any resource allocation/auctioning frameworks (e.g., [14–16]) while protecting users' operational data (e.g., location, antenna heights, etc) under SAS architectures. For IUs and honest SUs, we first propose a privacy-preserving scheme by leveraging encryption and obfuscation methods (PSEO). The core of PSEO is our novel *blind interference calculation scheme* (BICS) that allows users to calculate the interference budget without revealing their operational information (e.g., antenna height, transmitting power, location...). Moreover, in comparison with the FCC's SAS, BICS moves interference budgeting tasks to local users and calculates it in an offline manner (i.e., before the spectrum requesting stage). That calculated result is later reused in all future spectrum requests, which dramatically reduces the online computing overhead of PSEO. To further mitigate the overhead in calculating the interference map, we introduce a quantization method and optimize the grid sizes of a terrestrial area of interest. To deal with dishonest SUs, we augment a "punishment and forgiveness" mechanism into PSEO (namely PF-PSEO). PF-PSEO punishes dishonest/greedy SUs by rejecting their requests but also forgives certain punished SUs by providing spectrum opportunities in later spectrum requesting periods. Main contributions of this paper are summarized as below:

- We propose PSEO to protect the privacy of both IUs and honest SUs, in which all operations are conducted with users' encrypted or obfuscated data. As a result, the user's actual operating information is only accessible to itself, whereas other users or adversaries are not able to attain or infer any relevant information. Moreover, PSEO is capable of successfully guaranteeing efficient spectrum utilization while preserving users' privacy.
- Besides protecting users' information, we propose BICS to minimize the online overhead of PSEO, which calculates the interference from the SU to all potential IUs before spectrum request stages. That significantly reduces the number of online operations of PSEO. In addition, the interference is calculated at the SU's side instead of the SAS administrator's side, using SU's operational information only. Therefore, neither the IU nor SU is required to share their operational information with the SAS administrator.
- For dishonest SUs, we introduce an authentication scheme to prevent dishonest SUs from using other users' information. Moreover, we account for SUs' reputation scores (RSs) and reputation history (RHs) [17] to evaluate SUs' behaviors, which encourage SUs to engage in honest activities.
- Unlike the traditional obfuscation methods, the obfuscation in our proposed schemes does not degrade the spectrum utilization.
- We consider four kinds of threats scenarios and tackle them using our proposed schemes. We provide the theoretical analysis to analyze the performance of PESO and PF-PSEO from various aspects. Moreover, we carry out extensive simulation experiments to evaluate the performance of PSEO and PF-PSEO. The experiments show that our proposed PSEO and PS-PSEO are capable of protecting all users' privacy as well as successfully realizing efficient spectrum allocation, with less online overhead.

The rest of the paper is organized as follows. Section 2 discusses the system model and problem formulation. Section 3 and Section 4 present the details of PSEO and PF-PSEO, respectively. Section 5 describes the quantization of the attenuation map and discussion. Section 6 analyzes the performance of the proposed PSEO and PF-PSEO. Simulation experiments are in Section 7. The conclusion and future work are drawn in Section 8.

2 System Model and Problem Formulation

2.1 Spectrum Sharing in SAS Architecture

We study the spectrum sharing in the protection zone of SAS, in which the spectrum management consists of four entities: the SAS administrator and three kinds of users (IUs, PALs, and GAAs) [3], as shown in Fig. 1. The SAS administrator is the database/server which is used to collect all relevant information and provide reliable spectrum access for SUs. IUs, which are the agencies of defense (e.g., military radars), hold the highest priorities regarding the spectrum access and interference protection, while GAAs has the lowest rights. PALs and GAAs, which are categorized as SUs, are usually from the industry. According to FCC proposals (refer to Subpart A-96.3 in [3]), SUs are fixed stations or networks of such stations that transmit and receive radio communication signals at a fixed location. They may be moved from time to time but they must turn off and re-send report to the SAS administrator prior to transmitting from a new location [3]. As such, we assume

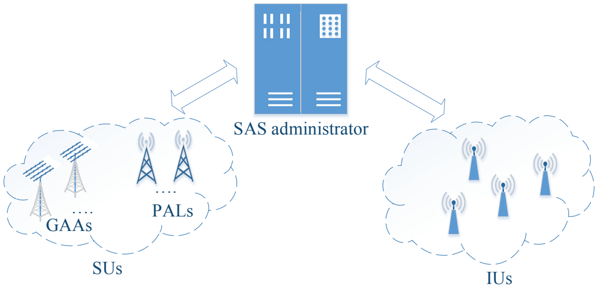


Fig. 1. Spectrum sharing in a SAS architecture

SUs are semi-stationary, i.e., they are stationary during the spectrum request and spectrum leasing terms. The SAS administrator allocates a spectrum chunk to a user based on the operational information provided by users. For that, the SAS administrator processes multiple SUs' requests (with grant or reject decisions) in a sequential manner.

The process of spectrum sharing between IU i and SU g in a SAS Architecture is summarized as follows [7].

1) In order to get the access of IU i's spectrum band, SU g is required to send its operational information to the SAS administrator, which includes its location (x_g, y_g), operating frequency f_g, antenna height h_g, etc.

2) IU i, for its interference protection, sends its operational data (including its location (x_i', y_i'), operating frequency f_i', antenna height h_i', and interference threshold X_i) to the SAS administrator to update its spectrum usage.

3) The SAS administrator first calculates the attenuation map of SAS architecture $\mathbf{A}'$ using terrestrial propagation models [18]. $\mathbf{A}'$, which specifies the attenuation between any two points in the area. The attenuation map is a function of location, antenna height, and operating frequency, etc.

$$\mathbf{A}' := \{A'(x_1, x_2, y_1, y_2, h_1, h_2, f_1, f_2)\}, \tag{1}$$

where (x_1, y_1), h_1, and f_1 denote the first point's location, antenna height and operating frequency, respectively. (x_2, y_2), h_2, and f_2 are the second point's location, antenna height and operating frequency, respectively.

Based on the received operational information from IU i and SU g, the SAS administrator calculates the interference from SU g to IU i:

$$G_g := P_g - \mathbf{A}'(x_g, x_i', y_g, y_i', h_g, h_i', f_g, f_i'), \tag{2}$$

where P_g is the transmitting power of SU g. Next, the SAS administrator compares G_g with IU's interference threshold X_i and decides whether approve SU g's request or not.

4) The SAS administrator sends a reply to SU g, which includes a response to SU g's request. If SU g gets an approval response, it can use IU i's spectrum band for data communication. Otherwise, SU g is not allowed to access IU i's spectrum.

As the SAS administrator receives and stores operational information from both the IU and SU, serious threats to user's privacy and interest are of concern. Notably, some IUs (e.g., military radars) in FCC's SAS are even forbidden to share any operational information with the SAS administrator [3]. For that case, the SAS administrator receives IUs' operational information or activity from the Environmental Sensing Capability (ESC). However, ESC is not protected at the level of IUs, hence it is still vulnerable to attacks, which would leak the IU's operational information. Therefore, the privacy threat always exists whether IUs communicate directly with the SAS administrator or not.

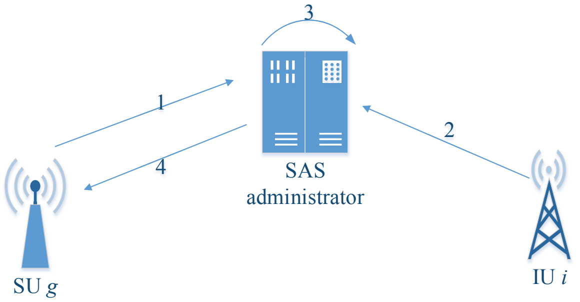


Fig. 2. Process of spectrum sharing between IU i and SU g in SAS architecture

2.2 Privacy Threats

We consider four types of threat that would compromise the user's operational privacy.

1) Privacy threats occurring at IU i's side

IU i's information is eavesdropped through the communication link between the SAS administrator and IU i, which is the 2nd step shown in Fig. 2.

2) Privacy threats occurring at SU g's side

SU g's privacy is compromised by eavesdroppers from communication links between the SAS administrator and SU g (e.g., the 1st and the 4th steps in Fig.2).

3) Privacy threats occurring at SAS administrator's side

The SAS administrator can be attacked by adversaries and hence expose IU's and/or SU's operational information (e.g., the 3rd step in Fig.2), which brings risks on the operational privacy of IU i and/or SU g. It is also possible for the SAS administrator to steal/reveal IU i's and/or SU g's operational data. For example, the SAS administrator could be operated by commercial parties (e.g., Google) [19]. For this kind of SAS administrator, it can faithfully execute the proposed spectrum allocation, however, it may attempt to access users' operational data for its own interest.

4) Collusion attack between SAS administrator and rogue SUs

As the SAS administrator can be owned and operated by a third party, there is a chance for them to be compromised (as high-value targets for attackers). In such a case, the SAS administrator can collude with a rogue SU to reveal the IU's actual operational data. Another possibility is that an adversary attacks/eavesdrops both the SAS administrator and the SU to obtain information from both sides to infer the actual operational data of IUs. These two threats are referred to as a Collusion attack in this paper.

In the sequel, we present our proposed privacy-preserving schemes for DSA, which leverage properties of public key cryptosystem and obfuscation methods. According to the public key cryptosystem, any one that holds additive homomorphic properties (details in Section 2.3) can be applicable to our proposed schemes. In this work, we adopt the Paillier cryptosystem [20].

2.3 Overview of Paillier Cryptosystem

The Paillier encryption cryptosystem consists of three main parts: key generation, encryption, and decryption [20].

2.3.1 Key Generation

p and q are two large prime numbers which are randomly selected, as long as they follow the condition that

$$\gcd\big(pq\,(p-1)(q-1)\big) = 1. \tag{3}$$

Compute n and λ by

$$n = pq, \tag{4}$$

$$\lambda = \mathrm{lcm}(p-1, q-1). \tag{5}$$

Then compute u using n and λ,

$$u = \big(L(z^{\lambda} \bmod n^2)\big)^{-1} \bmod n, \tag{6}$$

where z is a integer number that is randomly selected, $z \in \mathbf{Z}^*_{n^2}$, $L(u)$=$(u-1)/u$.

The public key (for encryption) K_P is (n, z).

The secret key (for decryption) K_S is (λ, u).

2.3.2 Encryption

Let m be the plaintext that needs to be encrypted, and $E_P(m)$ denote the ciphertext (encrypted from m) with the public key K_P. $E_P(m)$, is calculated by

$$E_P(m) = z^m \cdot r^n \bmod n^2, \tag{7}$$

where r is a one-time random number. With the help of r, the same m with the same public key $K_P$would result in different ciphertexts for each encryption process.

2.3.3 Decryption

The plaintext m can be obtained from $E_P(m)$ by

$$m = L\big([E_P(m)]^{\lambda} \bmod n^2\big) \cdot u \bmod n. \tag{8}$$

From the above steps, obviously, it is impossible to decrypt the ciphertext $E_P(m)$ without the corresponding secret key K_S. Moreover, r can make a difference in the process of Encryption, whereas it does not affect the Decryption process (refer to equation (7) and (8)). In other words, for each encryption process, one plaintext can be encrypted to different ciphertexts using the same public key as a result of r. However, one ciphertext can only be decrypted to one plaintext with the same secret key.

2.3.4 Additive Homomorphic Properties

The additive homomorphic properties of the Paillier cryptosystem allow one to execute additive operations (e.g., addition, subtraction and scalar multiplication operations) on the ciphertexts, i.e., without decrypting them. In our design, this homomorphic feature enables the interference budgeting calculation on the ciphertext. The additive homomorphic properties are as follows.

Let $E_P(m_1)$ and $E_P(m_2)$ are ciphertexts of m_1 and m_2, respectively. The ciphertext of $(m_1 + m_2)$ then can be calculated as:

$$E_P(m_1 + m_2) = E_P(m_1) \cdot E_P(m_2). \tag{9}$$

For a constant c_1, the ciphertext of $c_1 \cdot m_1$ is:

$$E_P(c_1 \cdot m_1) = \big[E_P(m_1)\big]^{c_1}. \tag{10}$$

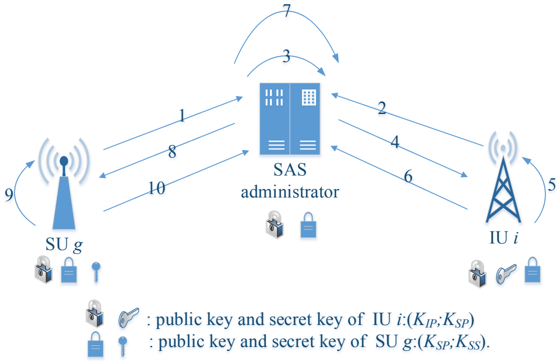

7
3
1
8
10
9
2
4
6
5
SAS
administrator
SU g
IU i
: public key and secret key of IU i:(K_{IP};K_{SP})
: public key and secret key of SU g:(K_{SP};K_{SS}).

Fig. 3. Key steps of proposed PSEO

The ciphertext of $m_1 - m_2$ can be also directly computed from $E_P(m_1)$ and $E_P(m_2)$

$$E_P(m_1 - m_2) = E_P(m_1) \cdot [E_P(m_2)]^{-1}. \tag{11}$$

Note that the Paillier cryptosystem is only suitable for the non-negative integer numbers. As such, we map all non-integer numbers by rounding them up to three digits after the decimal and multiply all data with 10^3 before the encryption. Additionally, under our setting and interference calculation below, $m_1 - m_2$ is the only possible operation which may result in a negative number. To deal with this, we adopt the additive property of Paillier cryptosystem, which is

$$\begin{aligned} E_P(m_3) &= E_P(c_2 + m_1 - m_2) \\ &= E_P(c_2) \cdot E_P(m_1) \cdot [E_P(m_2)]^{-1}, \end{aligned} \tag{12}$$

where c_2 is a positive integer and $0 \le m_2 < c_2$. $E_P(c_2)$ is the cipertext of c_2 with the public key K_P. When m_3 is obtained from the decryption operation, $m_1 - m_2$ can be easily obtained by $m_3 - c_2$.

3 Privacy-preserving Scheme based on Encryption and Obfuscation Methods

In this section, we first present the overview of PSEO. Then we discuss the details of BICS that is the core of PSEO.

3.1 Overview of PSEO

PSEO is made up of 10 steps, as shown in Fig. 3, in which (K_{IP}, K_{IS}) and (K_{SP}, K_{SS}) are key pairs of IU i and SU g, respectively. K_{IP} and K_{SP} are public keys, which are used to encrypt original data into the encrypted domain. Moreover, these public keys are distributed to each part of the system. K_{IS}, the secret key of IU i, can decrypt the data encrypted with the public key K_{IP}. K_{SS}, the secret key of SU g, is used to decrypt the data encrypted with the public key K_{SP}. These secret keys K_{IS} and K_{SS} are confidential information and only held by IU i and SU g, respectively.

Notably, in PSEO, IU i and SU g keep their own secret keys by themselves and don't require a third party to manage the key distribution. That makes PSEO much more practical than the privacy-preserving schemes using public cryptosystems which require the third party for key management (such as [12]). Moreover, drawing support from obfuscation methods [10], our scheme is able to prevent IUs' information from being inferred, even if there is a collusion between the SAS administrator and SUs (refer to 6th step in section 3.3). In addition, the obfuscation methods can also protect SU g's privacy from being revealed, even if IU i can decrypt the SU g's ciphertext that is encrypted with IU i's public key (refer to 3rd and 4th steps in section 3.3).

BICS is the core of PSEO that allows the interference budgeting to be calculated locally at SUs while no specific information of IUs is required. Additionally, it reduces the online overhead and facilitates the homomorphic operations at the SAS administrator without requiring IUs or SUs to share/send their operational information (i.e., referred to as the blind interference calculation method).

3.2 Blind Interference Calculation Scheme

In the SAS framework, the interference from SU g to IU i is calculated by the SAS administrator with operational information from both IUs and SUs. That increases the online overhead (referring to the computational complexity during the spectrum requesting stage) and risks leaking users' actual operational information. By contrast, in BICS, the interference from SU g to IU i is calculated at SU g's side instead of SAS administrator's side, bringing two major associated advantages. First, the online overhead of proposed PSEO is significantly reduced, as the interference calculation is now conducted offline before the spectrum requesting stage, which does not occupy the resource or time in the spectrum requesting stage. Second, SU g calculates the interference with its operational data only, without any information from IU i. That prevents IU i and SU g from leaking their operational information (through sharing it with each other or with the third party).

In BICS, the SAS administrator sends the attenuation map of SAS $\mathbf{A}'$ to all users before the process of spectrum request. To reduce the complexity, we introduce a quantization method to quantize $\mathbf{A}'$. We consider a service area $X \times Y$. H is the maximum value of antenna height. F_{max} and F_{min} are the maximum and minimum value of operating spectrum, respectively. $F = F_{max} - F_{min}$ is the spectrum bandwidth in SAS. Then we quantize X, Y, F, and H into equal-size X_q, Y_q, F_q, and H_q grids, respectively, obtaining a quantized attenuation map $\mathbf{A}$ (refer to Section 5.1 for more details).

The quantized attenuation map $\mathbf{A}$ which is an eight-dimensional matrix can be represented by

$$\begin{aligned}\mathbf{A} :=& \ \{\mathbf{A}'\}_{X_q^2 \times Y_q^2 \times H_q^2 \times F_q^2} \\ =& \left\{A(x_1, x_2, y_1, y_2, h_1, h_2, f_1, f_2)\right\}_{X_q^2 \times Y_q^2 \times H_q^2 \times F_q^2}. \end{aligned} \tag{13}$$

Upon receiving $\mathbf{A}$ from the SAS administrator, SU g determines its location in $\mathbf{A}$ and obtains its eight-dimensional attenuation map $\mathbf{A}_g$, which is

$$\begin{aligned}\mathbf{A}_g :=& \ \mathbf{A}(x_g, :, y_g, :, h_g, :, f_g, :) \\ =& \left\{A(x_g, x_2, y_g, y_2, h_g, h_2, f_g, f_2)\right\}_{X_q^2 \times Y_q^2 \times H_q^2 \times F_q^2}, \end{aligned} \tag{14}$$

where x_g, y_g, h_g, and f_g are operational parameters of SU g. (x_2, y_2, h_2, f_2) is the operational data of a *potential* IU that happens to be at location (x_2, y_2) with antenna height h_2 and operating frequency f_2. Note that the potential IU does not necessary mean the existence of a real/physical IU. For that, the operational information of the IU is protected (not being shared).

SU g then calculates the interference caused by itself to the potential IU i. Since SU g is prohibited from accessing any IU's operational information, it assumes IU i would be located in any grid in $\mathbf{A}_g$. The interference map of SU g is calculated by

$$\begin{aligned}\mathbf{G}_g \ :& = \left\{G\left(x_g, :, y_g, :, h_g, :, f_g, :\right)\right\}_{X_q^2 \times Y_q^2 \times H_q^2 \times F_q^2} \\ &= \ P_g - \mathbf{A}_g \\ &= \ P_g - \mathbf{A}(x_g, :, y_g, :, h_g, :, f_g, :), \end{aligned} \tag{15}$$

where P_g is the transmitting power of SU g.

BICS is capable of preserving the operational privacy of all users. This is because under BICS, each SU does not need to know the operational information of any IU (e.g., IU's physical locations). Instead, the SU assumes that an IU can locate in any grid. Then the interference from the SU to the IU (any grid) is completed at the SU's side before the spectrum requesting stage, only using the SU's operational information. In other words, only the SU is involved in the process of the interference calculation using its own operational information, without any participation from other parties, e.g., IUs or the SAS administrator. Therefore, neither the IU nor SU have to share their operational information. The BICS can thus protect users' operational privacy.

3.3 Details of PSEO

The main steps of PSEO are presented as follows.

1) In order to apply IU i's spectrum bands, SU g first encrypts its interference map $\mathbf{G}_g$ (which is calculated through the above BICS) with the IU i's public key K_{IP} and obtains the ciphertext $E_{K_{IP}}[\mathbf{G}_g]$, which is

$$\mathbf{G}_g \xrightarrow[encryption]{K_{IP}} E_{K_{IP}}[\mathbf{G}_g], \tag{16}$$

where *encryption* denotes the encryption process discussed in Section 2.3. SU g then sends $E_{K_{IP}}[\mathbf{G}_g]$ to the SAS administrator.

2) IU i determines the interference threshold map based on its operational information. If the operational information of IU i is (x_i', y_i', h_i', f_i'), the corresponding quantized interference threshold map is

$$\begin{aligned}\mathbf{X}_i :=& \left\{X\left(:, x_i', :, y_i', :, h_i', :, f_i'\right)\right\}_{X_q^2 \times Y_q^2 \times H_q^2 \times F_q^2} \\ =& \left\{X\left(x_1, x_i', y_1, y_i', h_1, h_i', f_1, f_i'\right)\right\}_{X_q^2 \times Y_q^2 \times H_q^2 \times F_q^2}, \end{aligned} \tag{17}$$

where (x_1, y_1, h_1, f_1) is the operational data of a *potential* SU. Notably, if there are more than one IU's threshold values in one grid, the smallest value should be selected as the new threshold for that gird. IU i encrypts $\mathbf{X}_i$ using its public key K_{IP} and obtains $E_{K_{IP}}[\mathbf{X}_i]$ by

$$\mathbf{X}_i \xrightarrow[encryption]{K_{IP}} E_{K_{IP}}[\mathbf{X}_i]. \tag{18}$$

Upon obtaining $E_{K_{IP}}[\mathbf{X}_i]$, IU i sends $E_{K_{IP}}[\mathbf{X}_i]$ to the SAS administrator to update its spectrum usage.

3) The SAS administrator compares $E_{K_{IP}}[\mathbf{X}_i]$ and $E_{K_{IP}}[\mathbf{G}_g]$ without decrypting them, by

$$\begin{aligned} E_{K_{IP}}[\mathbf{B}_g] &= E_{K_{IP}}[\mathbf{C}+\mathbf{X}_i-\mathbf{G}_g] \\ &= E_{K_{IP}}[\mathbf{C}]\cdot E_{K_{IP}}[\mathbf{X}_i]\cdot\left[E_{K_{IP}}[\mathbf{G}_g]\right]^{-1}, \end{aligned} \tag{19}$$

where $E_{K_{IP}}[\mathbf{B}_g]$ is the interference budget in the ciphertext domain. $E_{K_{IP}}[\mathbf{C}]$ is the ciphertext of $\mathbf{C}$ with the public key K_{IP}. $\mathbf{C}$ is a positive integer matrix, which has the same dimensions with $\mathbf{X}_i$. As mentioned above, $\mathbf{C}$ is used to ensure that $\mathbf{B}_g$ is a positive integer matrix, which guarantees $E_{K_{IP}}[\mathbf{B}_g]$ is valid. Notably, $\mathbf{C}$ is randomly generated and kept by the SAS administrator only.

4) Since IU i is the only part that holds the secret key K_{IS}, the SAS administrator sends $E_{K_{IP}}[\mathbf{B}_g]$ to IU i to obtain the decrypted value of $E_{K_{IP}}[\mathbf{B}_g]$. However, this process would risk the operational information of SU g being exposed to IU i, which is contrary to the goal of PSEO. In order to solve this problem, we recruit the following obfuscation methods before sending $E_{K_{IP}}[\mathbf{B}_g]$ to IU i:

$$\begin{aligned} E_{K_{IP}}[\mathbf{B}_g'] &= E_{K_{IP}}[c_3\cdot\mathbf{B}_g] \\ &= \left[E_{K_{IP}}[\mathbf{B}_g]\right]^{c_3}, \end{aligned} \tag{20}$$

and

$$\begin{aligned} E_{K_{IP}}[\mathbf{C}'] &= E_{K_{IP}}[c_3\cdot\mathbf{C}] \\ &= \left[E_{K_{IP}}[\mathbf{C}]\right]^{c_3}, \end{aligned} \tag{21}$$

where c_3 is a positive integer and generated by the SAS administrator randomly. Notably, c_3 does not degrade the spectrum utilization, which will be discussed in Section 5.3. After obtaining $E_{K_{IP}}[\mathbf{C}']$ and $E_{K_{IP}}[\mathbf{B}_g']$, the SAS administrator sends them to IU i to be decrypted.

5) IU i decrypts $E_{K_{IP}}[\mathbf{B}_g']$ and $E_{K_{IP}}[\mathbf{C}']$ with the secret key K_{IS}, by

$$E_{K_{IP}}[\mathbf{B}_g'] \xrightarrow[decryption]{K_{IS}} \mathbf{B}_g', \tag{22}$$

$$E_{K_{IP}}[\mathbf{C}'] \xrightarrow[decryption]{K_{IS}} \mathbf{C}', \tag{23}$$

where $decryption$ denotes the decryption process discussed in Section 2.3. Then IU i obtains the interference calculation result $\mathbf{R}_g'$ by $\mathbf{R}_g' = \mathbf{B}_g' - \mathbf{C}'$, which is a eight-dimensional matrix. IU i compares the value of each grid in $\mathbf{R}_g'$ with 0. If all numbers in $\mathbf{R}_g'$ are positive, which means the interference caused by SU g is lower than IU i's interference threshold, IU i generates a decision 'YES'. Otherwise, IU i generates a decision 'NO'.

6) In order to preserve its operational privacy, IU i obfuscates $\mathbf{R}_g'$ by

$$\mathbf{R}_g = c_4\cdot\mathbf{R}_g', \tag{24}$$

where c_4 is a positive integer which is randomly generated by IU i. Note that, like c_3, thanks to our design, c_4 doesn't impact the spectrum utilization (refer to Section 5.3). Then IU i encrypts $\mathbf{R}_g$ with SU g's public key K_{SP} by

$$\mathbf{R}_g \xrightarrow[encryption]{K_{SP}} E_{K_{SP}}[\mathbf{R}_g]. \tag{25}$$

IU i sends $E_{K_{SP}}[\mathbf{R}_g]$ and decision 'YES/NO' to the SAS administrator.

Summary 1: Key Steps of PSEO

```
1 SU g:          sends $E_{K_{IP}}[\mathbf{G}_g]$ to SAS admin.
                 $\mathbf{G}_g \xrightarrow[encry]{K_{IP}} E_{K_{IP}}[\mathbf{G}_g]$
2 IU i:          sends $E_{K_{IP}}[\mathbf{X}_i]$ to SAS admin.
                 $\mathbf{X}_i \xrightarrow[encry]{K_{IP}} E_{K_{IP}}[\mathbf{X}_i]$
3 SAS admin:     calculates $E_{K_{IP}}[\mathbf{B}_g]$.
                 $E_{K_{IP}}[\mathbf{B}_g] = E_{K_{IP}}[\mathbf{C}+\mathbf{X}_i-\mathbf{G}_g]$
4 SAS admin:     sends $E_{K_{IP}}[\mathbf{B}_g']$ and $E_{K_{IP}}[\mathbf{C}']$ to IU i
                 $E_{K_{IP}}[\mathbf{B}_g'] = \left[E_{K_{IP}}[\mathbf{B}_g]\right]^{c_3}$, $E_{K_{IP}}[\mathbf{C}'] = \left[E_{K_{IP}}[\mathbf{C}]\right]^{c_3}$
5 IU i:          calculates $\mathbf{R}_g'$.
                 $E_{K_{IP}}[\mathbf{B}_g'] \xrightarrow[decry]{K_{IS}} \mathbf{B}_g'$, $E_{K_{IP}}[\mathbf{C}_g'] \xrightarrow[decry]{K_{IS}} \mathbf{C}_g'$
                 $\mathbf{R}_g' = \mathbf{B}_g' - \mathbf{C}_g'$
                 determines the property of all numbers in $\mathbf{R}_g'$
                 if all numbers are positive
                   generates "YES"
                 else
                   generates "NO"
                 end
6 IU i:          sends "YES/NO" and $E_{K_{SP}}[\mathbf{R}_g]$ to SAS admin
                 $\mathbf{R}_g = c_4\cdot\mathbf{R}_g'$, $\mathbf{R}_g \xrightarrow[encry]{K_{SP}} E_{K_{SP}}[\mathbf{R}_g]$
7 SAS admin:     updates $E_{K_{IP}}[\mathbf{X}_i]$
                 if receives "YES"
                   approve SU g's request and
                   update $E_{K_{IP}}[\mathbf{X}_i]$ by $E_{K_{IP}}[\mathbf{B}_g]$
                 else
                   reject SU g's request and
                   do not update $E_{K_{IP}}[\mathbf{X}_i]$
                 end
8 SAS admin:     sends $E_{K_{SP}}[\mathbf{R}_g]$ and $E_{K_{SP}}[D_g]$ to SU g
                 $D_g \xrightarrow[encry]{K_{SP}} E_{K_{SP}}[D_g]$
9 SU g:          verifies $\mathbf{R}_g$ and $D_g$
                 $E_{K_{SP}}[\mathbf{R}_g] \xrightarrow[decry]{K_{SS}} \mathbf{R}_g$, $E_{K_{SP}}[D_g] \xrightarrow[decry]{K_{SS}} D_g$
10 SU g:         sends a confirmation message to SAS admin.
```

7) The SAS administrator updates interference threshold budget $E_{K_{IP}}[\mathbf{X}_i]$ based on 'YES/NO'. If the SAS administrator receives 'NO', it will reject SU g's request and not update $E_{K_{IP}}[\mathbf{X}_i]$. Otherwise, it will approve SU g's request and update $E_{K_{IP}}[\mathbf{X}_i]$ by

$$E_{K_{IP}}[\mathbf{X}_i] \longleftarrow E_{K_{IP}}[\mathbf{B}_g]. \tag{26}$$

8) The SAS administrator generates a signature D_g that includes the request decision and encrypts it with public key of SU g, which denotes as $E_{K_{SP}}[D_g]$. The SAS administrator sends $E_{K_{SP}}[\mathbf{R}_g]$ and $E_{K_{SP}}[D_g]$ to SU g.

9) SU g decrypts $E_{K_{SP}}[\mathbf{R}_g]$ and $E_{K_{SP}}[D_g]$ using its K_{SS} by

$$E_{K_{SP}}[D_g] \xrightarrow[decryption]{K_{SS}} D_g, \tag{27}$$

$$E_{K_{SP}}[\mathbf{R}_g] \xrightarrow[decryption]{K_{SS}} \mathbf{R}_g. \tag{28}$$

D_g is used to inform the request decision to SU g. Moreover, since it is generated using some digital signature techniques, it can identify SU g's identity and prevent the spoofing attack. $\mathbf{R}_g$ is used to guarantee the validity of the decision D_g (to SU g's request). This can be seen as a second layer of protection (e.g., Cyclic Redundancy Check, CRC, and checksum) for the message D_g from any kind of attack that may corrupt/flip the "YES" or "NO" decision in D_g. Specifically, D_g is regarded as a valid decision if all numbers in $\mathbf{R}_g$ are positive and D_g is 'YES' OR at least one number in $\mathbf{R}_g$ is non-positive and D_g is 'NO'. For all the other cases, the request decision is marked as invalid, and SU g will report it to the SAS administrator. Moreover, $\mathbf{R}_g$ (achieved from equation (24)) is the obfuscated interference calculation result that does not reveal user's actual operational data, so SU g cannot infer the IU i's actual operational information with $\mathbf{R}_g$.

10) SU g sends a confirmation message to the SAS administrator to confirm its received response. If the value of each grid in $\mathbf{R}_g$ is positive and the request is approved by the SAS administrator, SU g will generate a positive confirmation message V_{gp}. Otherwise, SU g will generate a negative confirmation message V_{gn}. Then SU g sends back V_{gp} or V_{gn} to the SAS administrator to verify its received decision. If the SAS administrator receives V_{gp}, it will complete the request stage of SU g and turn to another SU's request. Otherwise, the SAS administrator will re-process the request from this SU.

Based on all the steps described above, users' actual information is only kept by themselves through the whole process of PSEO, hence, their operational privacy can be protected. The summary of key steps for PSEO is listed in **Summary 1**, in which "SAS admin" denotes the SAS administrator.

4 Dealing with dishonest SUs: punishment and forgiveness Mechanism

In this section, we investigate the operational privacy issue of IUs and dishonest SUs. Since IUs are usually the agents of defense (e.g., military radars), it is quite reasonable to assume that they are trustworthy. By contrast, SUs, which are commercial organizations may behave dishonestly/greedily for their own benefit. As noted in Section 1, existing privacy-preserving schemes are proposed mainly relying on the assumption of honest SUs, hence, their performance would be dramatically degraded, suffering from dishonest SUs.

To deal with greedy/dishonest SUs, we propose the "punishment and forgiveness" cooperating with PSEO, namely PF-PSEO, which is able to effectively deal with two types of dishonest SUs: the SUs who provide invalid identity number (IDN) and the SUs who provide a valid IDN but proceed to report lower transmit power values than they actually plan to use, referred to as greedy SUs. For the first type of SUs, PF-PSEO can detect their dishonest behaviors utilizing our proposed "Authentication of SU" and ask them to send valid IDNs. For the second type of SUs, i.e., the greedy SUs, PF-PSEO can punish them (when they are found dishonest) using our "Punishment and Forgiveness (PF)" scheme and hence consequently encourage them to comply with what they report.

The objective of PF in PF-PSEO is to punish the greedy SUs by rejecting their requests, which reduces their spectrum access opportunities. However, PF also forgives greedy SUs who received the punishment (i.e., they were rejected previously) by giving them spectrum access opportunities during the future requests. It is worth noting that after a finite times of being dishonest (i.e., a threshold), the greedy SUs' requests will never be granted even if they behave honestly. In other words, our proposed framework only tolerates a given number of dishonest times. All greedy SUs that exceed this threshold will be banned forever from accessing the spectrum from SAS. That helps eliminate greedy/dishonest SUs/behaviors and "softly" enforce SUs to report the true information and behave in accordance with their reported/request information (to be able to access shared spectrum). In the following section, we will specifically illustrate how PF-PSEO can effectively deal with the two types of dishonest SUs, as shown in Fig. 4. .

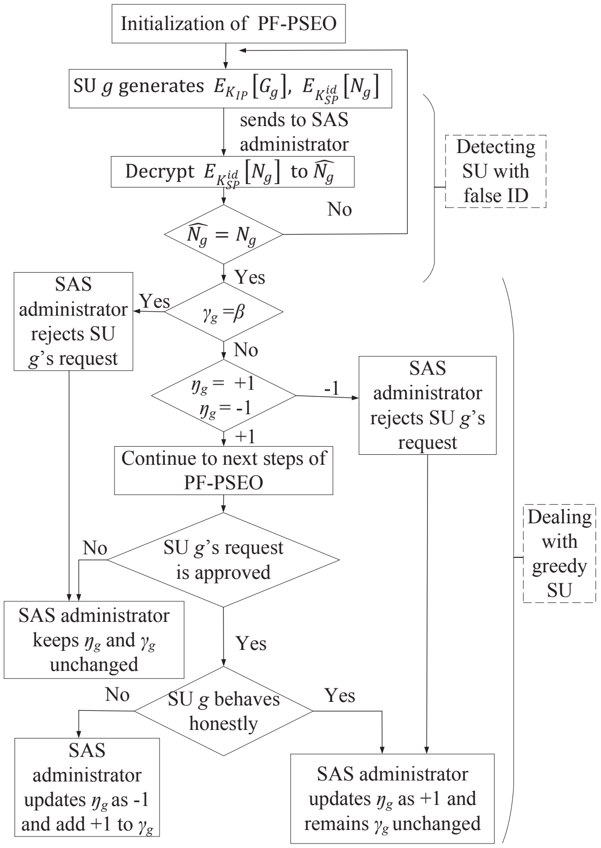

Initialization of PF-PSEO
SU g generates $E_{K_{IP}}[G_g]$, $E_{K_{SP}^{id}}[N_g]$
sends to SAS administrator
Decrypt $E_{K_{SP}^{id}}[N_g]$ to $\widehat{N_g}$
No
$\widehat{N_g} = N_g$
Yes
Detecting SU with false ID
SAS administrator rejects SU g's request
Yes
$\gamma_g = \beta$
No
$\eta_g = +1$
$\eta_g = -1$
-1
SAS administrator rejects SU g's request
+1
Continue to next steps of PF-PSEO
No
SU g's request is approved
Dealing with greedy SU
SAS administrator keeps η_g and γ_g unchanged
Yes
No
SU g behaves honestly
Yes
SAS administrator updates η_g as -1 and add +1 to γ_g
SAS administrator updates η_g as +1 and remains γ_g unchanged

Fig. 4. Dealing with greedy/dishonest SUs in PF-PSEO

4.1 Initialization of PF-PSEO

In order to implement PF-PSEO, the SAS administrator first generates each SU's identity number (IDN) and the IDN key pair (i.e., the IDN public key and the IDN secret key). The IDN key pair is used to verify the SU's IDN to prevent the SU from using other users' IDNs. The SAS administrator distributes the IDN and the IDN public key to the corresponding SU, and keeps the IDN secret key by itself. Note that the IDN is a positive integer for the purpose of encryption. For each SU, it can only access its own IDN and IDN public key. Let N_g denote the SU g's IDN, and $(K_{SP}^{id}, K_{SS}^{id})$ denote its IDN key pair, where K_{SP}^{id} is the IDN public key of SU g, K_{SS}^{id} is the corresponding IDN secret key (kept by the SAS administrator only).

4.2 Authentication of SU: Detecting SUs with False ID

In this subsection, we develop an authentication scheme to prevent SUs from sending other users' IDNs during spectrum requesting periods. The SU g's spectrum request sent to the SAS administrator consists of three parts: its own IDN N_g, $E_{K_{SP}^{id}}[N_g]$ and $E_{K_{IP}}[\mathbf{G}_g]$. N_g and $E_{K_{SP}^{id}}[N_g]$ are used to verify SU g's authenticity. $E_{K_{SP}^{id}}[N_g]$ is the ciphertext of N_g using the IDN public key K_{SP}^{id}, which is

$$N_g \xrightarrow[encryption]{K_{SP}^{id}} E_{K_{SP}^{id}}[N_g]. \tag{29}$$

Upon receiving N_g and $E_{K_{SP}^{id}}[N_g]$, the SAS administrator verifies SU g's authenticity. Specifically, the SAS administrator searches for the IDN secret key stored in its memory with the IDN N_g, and uses that IDN secret key to decrypt $E_{K_{SP}^{id}}[N_g]$ by

$$E_{K_{SP}^{id}}[N_g] \xrightarrow[decryption]{K_{SS}^{id}} \hat{N}_g. \tag{30}$$

The SAS administrator then verifies SU g's authenticity by comparing $\hat{N}_g$ and N_g, which has two results:

$\hat{N}_g \neq N_g$: this means SU g does not send its actual IDN to the SAS administrator, then the SAS administrator will ask SU g to re-sends its IDN.

$\hat{N}_g = N_g$: this means SU g sends its own IDN, then the SAS administrator will continue to process SU g's request. Note that the SU g cannot be approved to proceed to next steps of PF-PSEO (e.g., the steps in PF scheme) until providing its true IDN.

4.3 Punishing, Forgiving, and Banning Greedy SUs

In our PF scheme, we use γ_g to denote the number of times that SU g has been dishonest, referred to as the reputation history (RH). η_g, referred to as the reputation score (RS), reflects the honesty of SU g's behavior during the last data transmission: $\eta_g = +1$ ($\eta_g = -1$) if SU g behaved honestly (dishonestly). Both γ_g and η_g are first generated (then kept track) by the SAS administrator with initial values of 0 and $+1$, respectively. If a SU g is found honest, the SAS administrator will not change the values of γ_g and η_g. If not, the SAS administrator will update η_g as -1, which means SU g's next request will be rejected. The value of γ_g is then added by $+1$. If γ_g reaches to the honesty threshold, β, SU g will be classified as a *regularly dishonest user*.

Upon receiving a request from the SU g, the SAS administrator first checks the value of γ_g to verify whether SU g is a regularly dishonest user, which has two potential cases.

4.3.1 Case A

$\gamma_g = \beta$, meaning that SU g is regarded as a regularly dishonest user. In that case, SU g's request will be rejected. Moreover, as a regularly dishonest user, SU g's future requests will never be granted even if it behaves honestly in the future (i.e., no forgiveness). The values of γ_g and η_g will not be updated, and the user is forever banned from obtaining the shared spectrum.

4.3.2 Case B

$\gamma_g < \beta$, which means SU g is not a regularly dishonest user. Then the SAS administrator checks the value of η_g to verify the honesty of SU g's behavior during the last data transmission.

$\eta_g = -1$: i.e., SU g's behavior did not abide by its report during the last transmission experience (e.g., reporting a lower transmitting power than its actual one), the SAS administrator will jump to the punishment stage that denies SU g's request directly. After that, the SAS administrator will update η_g from -1 to $+1$, which is called forgiveness. Moreover, the SAS administrator does not update γ_g.

$\eta_g = +1$: i.e., SU g's behavior complied with its report during the last data transmission, then the SAS administrator will continue to process SU g's request, which are the $3-10$ steps of PSEO in Section 3.3. The values of γ_g and η_g will be updated depending on the outcome of SU g's request and its behavior. Specifically, if SU g's request is denied by the SAS administrator, the SAS administrator will keep the values of γ_g and η_g unchanged. If SU g's request is approved, γ_g and η_g will be updated as aforementioned (depending on if its transmission complies with its reported information or not).

Note that although the SAS administrator does not receive the unencrypted data from SUs, it can still find out if a granted SU complies with its requested values or not. Specifically, the government agency (e.g., FCC or National Telecommunications and Information Administration, NTI-A [21]) monitors SUs behavior during the data transmission period (to find out greedy behaviors) using the spectrum monitoring and signal analysis techniques [22]. Interested readers are also referred to [23] and [24] for similar spectrum monitoring approaches. Then the agency generates an interference map of this SU using the observed data and attenuation map (public knowledge). Next, the agency encrypts this map using the IU's public key before sending it to the SAS administrator. Upon receiving the ciphertext from the agency, the SAS administrator calculates a subtraction in the ciphertext domain, by subtracting it from the ciphertext data reported by the SU in the spectrum requesting period. After that, the SAS administrator sends the calculated result to the IU to verify the honesty of the SU. The IU decrypts the received result and checks its value: If all values are non-negative which means the actual operating value is no more than the SU's requested value, the IU will report to the SAS administrator that the SU is honest during its data transmission. Otherwise, the IU will report to the SAS administrator that the SU is dishonest.

5 Quantization of Attenuation Map and Discussion

In this section, we first focus on the quantization of the attenuation map. We then discuss the effect of the quantization selection on the attenuation map, followed by the effect of obfuscating factors on the spectrum utilization.

5.1 Quantization of Attenuation Map

The quantization process aims to partition the attenuation map into N grids to reduce the online complexity. For PSEO and PF-PSEO, the size of each grid in the quantized attenuation map $\mathbf{A}$ dramatically affects the precision of attenuation value in each grid and the computational complexity. The smaller grid size, the higher precision of the attenuation map, hence the fewer decision errors made by the SAS administrator. However, that comes at the extra cost of the overhead computation. Similar to that in [12], to optimize the balance between the precision of attenuation value and the computational complexity, we formulate the following optimization problem that aims to find the optimal quantization tuple:

$$\begin{aligned}\underset{X_q,Y_q,H_q,F_q}{\text{minimize}} \quad & E_r\left(X_q, Y_q, H_q, F_q\right)\\ \text{subject to} \quad & cost(X_q, Y_q, H_q, F_q) \leq C_n\\ & X_q \in \{2, 3, ..., X_N\},\\ & Y_q \in \{2, 3, ..., Y_N\},\\ & H_q \in \{2, 3, ..., H_N\},\\ & F_q \in \{2, 3, ..., F_N\},\end{aligned} \tag{31}$$

where $E_r(\cdot)$ indicates the decision error made by the SAS administrator. We adopt three types of metrics to measure $E_r(\cdot)$: the positive decision error (PDE), the negative decision error (NDE), and the total decision error (TDE). PDE occurs when the SU's request should have been denied by the SAS administrator, but gets an approval response. NDE denotes the case that the SAS administrator makes a denial decision to the SU's request that should have been approved. TDE is the total number of PDE and NDE, which is the number of all incorrect requests decisions. $cost(\cdot)$ is the cost function which denotes the computational complexity of each request period. That refers to the total number of online operations (as Table 1) or online processing time for each SU's request, which is proportional to the total number of grids. X_N, Y_N, F_N, and H_N are maximum quantization numbers for X, Y, F, and H, respectively. C_n is the computational complexity (cost) limit that depends on specific requirements in different scenarios.

The maximum quantization levels X_N, Y_N, F_N, and H_N are selected based on specific requirements of performance in different scenarios, e.g., the balance between the accuracy of spectrum allocation and the computational complexity. For instance, the quantization tuple (50, 50, 20, and 20) may provide a very high level of accuracy at the expense of high computational complexity (refer to Table 4). Thus, that tuple can be selected as the maximum boundary of the optimization variables for scenarios that can tolerate a higher error ratio. For the above integer programming, we apply the branch and bound method [25] to solve it. Since this optimization procedure is completed offline by the SAS administrator, it does not increase the online overhead of PSEO during the all later spectrum requests.

Note that each grid generated from equation (31) involves original attenuation values, referring to the attenuation values calculated by the SAS administrator using the terrain data. These data are sent to all users in the system before the stage of spectrum requesting period. We provide three different methods to estimate the attenuation value for each grid, which are Maximum value, Minimum value, and Average value estimation methods.

The Maximum value estimation method can be expressed as

$$A = \max(A^1, A^2, ...A^L), \tag{32}$$

where $A^1, A^2, ...A^L$ are original attenuation values in one grid area. L means the total number of original attenuation values in the grid. Obviously, this estimation method selects the maximum value from L values as the value of the grid.

For the Minimum value estimation method, the attenuation value for each grid is the minimum value in that gird, which is

$$A = \min(A^1, A^2, ...A^L). \tag{33}$$

In terms of the Average value estimation method, the attenuation value of each grid is the average of all original values in that grid, which is

$$A = \frac{1}{L}\sum_{l=1}^{L} A^l. \tag{34}$$

These three estimation methods would result in different impacts on actual attenuation values. For example, "Maximum value" estimation method would cause over-estimation of attenuation values, and "Minimum value" would lead to under-estimation of actual attenuation values. Different estimation methods can be selected according to various specific requirements in different scenarios.

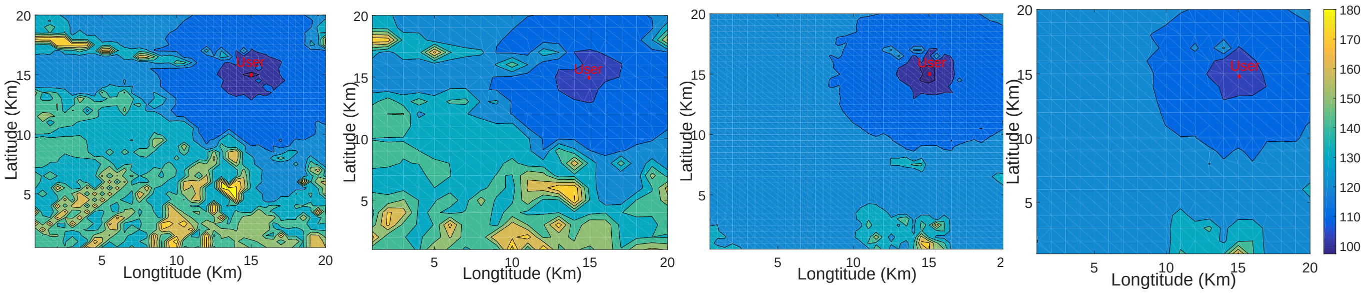
5.2 Effect of Quantization Selection on Attenuation Map

To illustrate the effect of the quantization selection on the attenuation map under different scenarios (e.g., different antenna heights), we provide Fig. 5. Different colours in each subfigure indicate the various attenuation values from the user to the corresponding locations, which vary in a range of $100 - 180$dB, as shown in the color bar. The subfigures in Fig. 5 show the attenuation map of the same user with different quantization tuples and antenna heights. Specifically, the user's antenna heights in both Fig. 5(a) and Fig. 5(b) are 6 m, but quantization tuples are $(40, 40, 10, 10)$ and $(20, 20, 6, 6)$, respectively. For Fig. 5(c) and Fig. 5(d), the antenna heights are 60 m, while the quantization tuples are $(40, 40, 10, 10)$ and $(20, 20, 6, 6)$, respectively.

It is clear that the effect of quantization on the attenuation maps varies with different antenna heights. As can be seen in Fig. 5(a) and Fig. 5(b), when the antenna height is 6m, decreasing the number of quantization grids (e.g., increasing the grid size) would lose a lot of details in the attenuation map. In this case, there will be a big error between the actual attenuation value and the estimated one, which degrades the accuracy of the proposed schemes. However, when the antenna height is 60m, as shown in Fig. 5(c) and Fig. 5(d), less details in the attenuation map are lost when decreasing the number of quantization grids, resulting in less degradation of the performance of proposed schemes. Therefore, in different scenarios, the effect of the quantization method on the attenuation maps varies, leading to different effects on the performance of the proposed schemes.

5.3 Effect of c_3 and c_4 on Spectrum Utilization

As discussed in Section 3.3, the obfuscating factors, i.e., c_3 and c_4, are used to obfuscate users' actual private information, which can protect the operational privacy. Notably, unlike the conventional obfuscation methods (e.g., [10]), in our method, using obfuscation factors does not degrade the spectrum utilization.



(a) (b) (c) (d)

Fig. 5. The effect of quantization selection (or grid size) on the attenuation map varies with different antenna heights. (a) The user's antenna height is 6m, quantization tuple (40, 40, 10, 10); (b) The user's antenna height is 6m with quantization tuple (20, 20, 6, 6); (c) The user's antenna height is 60m with quantization tuple (40, 40, 10, 10); (d) The user's antenna height is 60m with quantization tuple (20, 20, 6, 6).

For the conventional obfuscation methods (e.g., [10]), the obfuscating parameters are added to the user's actual data to preserve the user's actual operational information. The spectrum decisions are hence made based on the obfuscated values, leading to degradation of the spectrum utilization.

By contrast, for our proposed schemes, the spectrum decision 'YES/NO' (as described above) is made by counting the number of positive and negative terms, instead of the specific value of each term. Moreover, the obfuscating parameters c_3 and c_4 that are positive integers, used to multiply the numbers, do not change the positiveness/negativeness property of numbers. Therefore, the obfuscation does not degrade the spectrum utilization in our proposed schemes.

6 Performance Analysis of PSEO and PF-PSEO

6.1 Privacy Analysis of PSEO and PF-PSEO

In this subsection, we analyze the privacy performance of proposed PSEO and PF-PSEO. Notably, PSEO and PF-PSEO achieve the same performance according to privacy-preserving. That is because these two schemes leverage the same BICS with similar obfuscation methods and additive homomorphic properties of the Paillier cryptosystem to preserve users' privacy. Therefore, we only analyze the privacy performance of PSEO.

Theorem 1. *PESO is capable of protecting IUs' and SUs' operational privacy from the first three privacy threats described in Section 2.2.*

Proof: We take the *Privacy threat 3)* in Section 2.2 as a study case to prove this theorem. For other two privacy threats, similar conclusions can be easily obtained. If an adversary is able to eavesdrop the ciphertext of IU or SU, it will attempt to decrypt the ciphertext to access user's actual operational information. According to the semantically secure of Paillier cryptosystem, it is impossible to decrypt the ciphertext correctly unless with the corresponding secret key [20]. However, as the secret keys are only kept by their owners (i.e., IUs or SUs), the adversary is not able to access neither the IU's nor the SU's secret key. The probability of generating the correct secret key is $1/2^K$, where K is the bit length of the secret key which is usually a large number. For instance, in this paper $K = 2048$, then the probability for the SAS administrator to get the correct key is only $1/2^{2048}$. As a result, the adversary cannot infer any actual operational information from the user's ciphertext. Similarly, even if the SAS administrator obtains the ciphertexts of IU or SU, it cannot infer any relevant information about the IU or SU. ■

As mentioned in the fourth type of attack, rogue SUs and a compromised SAS administrator may attempt to collude to reveal/obtain IUs' operational information. We have the following theorem.

Theorem 2. *PSEO is able to protect IU's operational information under a collusion attack.*

Proof: Throughout all the steps in PSEO above, if there is a collusion between the SU and the SAS administrator, the SAS administrator or the SU may obtain $\mathbf{R}_g$ by decrypting $E_{K_{SP}}[\mathbf{R}_g]$ (refer to equation (28)). Note that $\mathbf{R}_g$ is the obfuscated value of the interference budget $\mathbf{R}'_g$ that is calculated with IU's and the SU's operational information. Thus, the IU's precise operational information can be revealed as long as $\mathbf{R}'_g$ is known. However, neither the SAS administrator nor the SU is able to access the actual value of $\mathbf{R}'_g$, because the obfuscating factor c_4 is only accessible to the IU. As a result, the IU's precise operational privacy is preserved even if the SAS administrator colludes with the SU. Similarly, if an adversary successfully eavesdrops information from the SAS administrator and the SU, it is unable to infer any precise operational information about the IU without the obfuscating factor c_4 (known only by the IU).

Additionally, neither the SAS administrator nor SU g can infer IU i's operational information (e.g., location) based on the positiveness/negativeness of numbers/grids in $\mathbf{R}_g$, even in an extreme case with only one negative grid. In other words, if the value of a grid is negative in $\mathbf{R}_g$, it does not necessarily imply that the IU is located in this grid or nearby grids. This is because in addition to the distance, various complex parameters (e.g., terrain, antenna height) are taken into account when calculating the radio attenuation map. Consequently, the interference map $\mathbf{G}_g$ and the interference threshold map $\mathbf{X}_i$ are not monotonically increasing with distance/location from the SU g and IU i, respectively (refer to Fig. 5). We can completely remove the correlation between the positiveness/negativeness of numbers/grids in $\mathbf{R}_g$ and the IU's location is by randomly rearranging all the elements in $E_{K_{SP}}[\mathbf{R}_g]$ (refer to the 6th step in PSEO) (other users' precise operating information is protected via the obfuscation procedure). ■

Notably, the scheme proposed in [12] is vulnerable to the collusion attack. When the SAS administrator and the key distributor (e.g., introduced in [12]) collude, the IUs' information can be decrypted. Even if the two entities do not collude, attackers can attempt each of the two separately then obtain the necessary information that allows them to interpret the operational data of IUs.

THEOREM 3. *Under PSEO, SUs' operating information is protected from the IU (as long as the IU does not collude with the SAS administrator).*

Proof: Throughout all PSEO's steps, the step 4 is the only potential one during which the IU can infer the SU's operational information. According to equation (19), IU i can infer SU g's actual information $\mathbf{B}_g$ if it gets the access to $\mathbf{G}_g$. To deal with it, a positive number c_3 (in equation (20) and (21)), which is randomly generated by the SAS administrator, is used to obfuscate the actual information of $\mathbf{B}_g$. Since there is no collusion between the SAS administrator and IU i, IU i is not able to get c_3. As a result, after the decryption process, IU i would only obtain the obfuscated data $\mathbf{B}'_g$ instead of $\mathbf{B}_g$. Note that, similar to the **Theorem** 2, IU i cannot infer SU g's operational information (e.g., location) using the positiveness/negativeness property of numbers in $\mathbf{B}'_g$. Another easy option to protect the SU's location is that the SAS administrator randomly rearranges all the elements in $E_{K_{IP}}[\mathbf{C}^{'}]$ and $E_{K_{IP}}[\mathbf{B}^{'}_g]$ (following the same arrangement rule) before sending $E_{K_{IP}}[\mathbf{B}^{'}_g]$ and $E_{K_{IP}}[\mathbf{C}^{'}]$ to IU (refer to the 4th step in PSEO). ■

From the above, we note that if the IU and the SAS administrator collude, the SU's operational privacy would be compromised (e.g., refer to the 4th step of PSEO). That is the weakness of PSEO, which is also our future work. However, in practice, IUs are often the agencies of defense (e.g., military radars) that are highly-trusted and most-protected components. The chance for IUs to be compromised or collude with the SAS server is negligibly small.

6.2 Complexity Analysis of PSEO and PF-PSEO

In this subsection, we analyze the complexity of proposed schemes that involves the online overhead and the offline complexity. As aforementioned, the online overhead refers to the computational complexity during the spectrum requesting stage. The offline complexity refers to the computational complexity before the spectrum requesting stage.

We emphasize that under SAS and all existing work (e.g.,[12]), the interference map has to be computed for *every* SU's request (even when the SU does not move or the topology does not change). This is because the interference map is part of the interference budgeting at each IU, and this budget may change (e.g., a request from other SUs is granted) even if the SU or the topology does not change (related to the interference map). This is a shortcoming of [12] as well as all other SAS-based frameworks. We observe this problem and propose our approach in which we separate the interference map calculation (above referred to as the offline overhead/complexity) from the interference budget update/calculation (the online overhead) so that the interference map does not have to be updated as frequently as the IU interference budgeting. Note that the offline complexity under PSEO and PF-SEO is only required once for very single SU and the map will be reused for all the future spectrum requests from that SU (i.e., well amortized by all future spectrum requests). Thus, the total complexity under our model is significantly reduced.

TABLE 1
Total complexity per SU's request

Operation		PSEO	PF-PSEO	Scheme[12]
Offline	Subtractions	N/N_F	N/N_F	None
Online	Encryptions	$(4N+1)$	$(4N+2)$	$(8N+1)$
	Decryptions	$(3N+1)$	$(3N+2)$	$(2N+1)$
	Hom Add	N	N	$4N$
	Hom Sub	N	N	$4N$
	Hom S-Mul	$2N$	$2N$	$4N$

Table 1 shows the total complexity for each SU's request. In this table, N is the total number of grids. "Hom Add", "Hom Sub" and "Hom S-Mul" denote the homomorphic additions, homomorphic subtractions and homomorphic scalar multiplications, respectively. Regarding the offline overhead, since only N subtraction operations (one for each grid) in the plaintext domain are conducted (refer to the equation (15)), as shown in Table 1, for proposed PSEO and PF-PSEO, the average number of subtractions (i.e., the offline overhead) per SU's request is N/N_F (N_F is the number of all future requests of this SU). That is negligible (especially when N_F is a very large number).

For the online overhead, since the homomorphic operations (such as encryption, decryption, homomorphic additions, homomorphic subtractions, and homomorphic scalar multiplications) are very much computationally demanding, we use the number of these operations as metrics. Unlike PSEO and PF-PSEO, [12] conducts all interference calculations in the ciphertext domain during the spectrum requesting stage that more than doubles the online overhead, as seen in the Table 1. For the case of multiple IUs that use different public keys, the requesting SU encrypts its interference map using all IUs' public keys (so that any IU can decrypt it). However, in practice, multiple or even all IUs (e.g., military radars from DoD) may share the same public key, e.g., [12]. Thus, the complexity scaling with the number of IUs in this case is practically marginal. We take PSEO as an example to demonstrate how these homomorphic operations above are counted:

- At the 1st and the 2nd steps in PSEO, the SU and the IU encrypt their own operational information, requiring $2N$ times of encryption.
- At the 3rd step, the SAS administrator calculates the interference budget in the ciphertext domain, incurring N times of encryption, addition and subtraction.
- At the 4th step, the SAS administrator obfuscates the interference budget, resulting in $2N$ times of scalar multiplication.
- At the 5th step, the IU decrypts the received information with $2N$ times of decryption.
- At the 6th step, the IU encrypts the obfuscated calculation with N times of encryption.
- At the 9 step, the SAS administrator encrypts the request decision, requiring 1 time of encryption.

- At the 10th step, the SU decrypts the received information, requiring $(N+1)$ times of decryption.

Therefore, the numbers of encryption, decryption, addition, subtraction, and scalar multiplication of PSEO for processing each SU's request are $(4N+1)$, $(3N+1)$, N, N, and $(2N)$, respectively.

It is similarly straightforward to compute the online overhead of PF-PSEO or [12]. Notably, for PF-PSEO, in addition to the operations above, its online overhead also includes one encryption and decryption operations, which occur at the "Authentication of SU". It is clear that the number of homomorphic operations of PSEO or PF-PSEO is much smaller than those in [12], making the computation time of PSEO or PF-PSEO more than 50% faster than that in [12]. For instance, if $N = 40\times40\times12\times8$, the total numbers of encryptions, online homomorphic additions, homomorphic subtractions, and homomorphic scalar multiplications for [12] are 1.2288×10^6, 6.144×10^5, 6.144×10^5, and 6.144×10^5, respectively. By contrast, those numbers for our proposed schemes are only about 6.144×10^5, 1.536×10^5, 1.536×10^5 and 3.072×10^5, respectively. Note that, IUs and SUs under SAS are usually military radar systems and base stations, respectively [3], thus they have relatively strong computing capability and power to effectively facilitate those encryption and decryption operations. The experiment results are shown in Section 7.3.

6.3 PSEO and PF-PSEO with Coexistence of Multiple SASs

In practice, there may be multiple SASs which work in parallel to provide nationwide spectrum sharing service. For the case that SU a in SAS A wants to request the spectrum band of IU b in the SAS B, the proposed PSEO and PF-PSEO are also capable of protecting users' operational privacy. Specifically, for the systems with multiple SASs, the operating frequency range of each SAS is broadcast, i.e., public knowledge, available for all SAS administrators and users [26]. Thus, the SU a knows that its requesting frequency falls in the frequency range of SAS B, but it does not know which IU owns the spectrum nor IU's bandwidth. As such, SU a encrypts its interference map using all IUs' public keys in SAS B before sending the encrypted data (together with the unencrypted information guiding SAS A to forward the request to SAS B) to the SAS administrator A. Then the SAS administrator A forwards the encrypted data to the SAS administrator B for processing SU a's request. Note that SU a would not be able to send the request directly to SAS B as SUs can only talk to its administering SAS [26]. After that, the SAS administrator B will process SU a's encrypted request, which are the same as $2-7$ steps of our proposed PSEO. After the decision of SU a's request is made, the SAS administrator B sends it to SU a via the SAS administrator A. Then SU a verifies the request decision and sends a confirmation to the SAS administrator B via the SAS administrator A.

It is worth noting that for multiple SAS coexistence, SAS administrators do not need to exchange interference thresholds of IUs with each other. This is because each IU is administered by only one SAS who solely manages the IU's interference budget/threshold. Each SAS administrator updates the interference thresholds of IUs under its management based on the decision for each SU's request, and keeps the threshold by itself. All SUs' requests to this IU is processed solely by the IU's SAS, not any other one.

TABLE 2
Simulation Parameters

Number of IUs	20
Number of SUs	1500
Value of X	20 km
Value of Y	20 km
Value of H	60 m
Bandwidth of each SU	10MHz
Frequency band F	3.55 $\sim$ 3.65 GHz
Transmitting power of each SU	$-40\sim$ 20dBm
Sensitive interference threshold	-80 $\sim$ -120 dBm

7 Simulation Results

We deploy a protection zone of SAS in a 20 km by 20 km rectangular area. We experiment the computation task using parallelization implementation on four distributed desktops with Intel $XeonE5-2690$ and 64GB RAM. The Paillier cryptosystem is constructed based on the GMP library [27]. The radio propagation model in this work is the Longley-Rice (L-R) model [28] that takes multiple parameters into account when calculating the attenuation value, such as locations, operating frequencies, polarization, terrain data, etc, and we generate it using $SPLAT!$ [29]. The attenuation value is calculated using the real terrain data in Washington D.C. from $SRTM3$ [30]. The simulation results are averaged over 2000 independent runs. In each run, the user's operational information (e.g., users' locations, antenna heights, transmitting powers, etc) is uniformly distributed within the corresponding ranges. Take the user's antenna height as an instance, each user's antenna height is uniformly distributed between $1\sim60\ m$. The relevant height of the user's antenna to the average terrain, e.g., the height above average terrain (HAAT), is also different, depending on the user's specific locations. Other important simulation parameters are in the Table. 2. Besides, we select the number of grids by applying the equation (31). Specifically, we use three different types of metrics to measure the objective function, $Er(.)$: PDE, NDE and TDE, for achieving optimal grids. Moreover, we apply three estimation methods to obtain the estimated values of girds (refer to equations (32)-(34)). Consequently, we can obtain 9 groups of optimal grids considering the different combinations of $Er(.)$ and estimation methods. We select three groups of grids in this section to evaluate the performance of proposed schemes.

To evaluate the level of greediness of each greedy SU, we introduce a parameter α, which is

$$\alpha = \frac{G_r}{G_a}, \tag{35}$$

where G_r is the reported interference in the SU's request, G_a is actual interference caused by that SU. $0<\alpha\leq1$. If $\alpha=1$, the SU's behavior conforms to its report, i.e., the SU is honest, otherwise, the SU is dishonest.

We evaluate the performance of proposed schemes and other work using three metrics: privacy performance, spectrum utilization and online overhead.

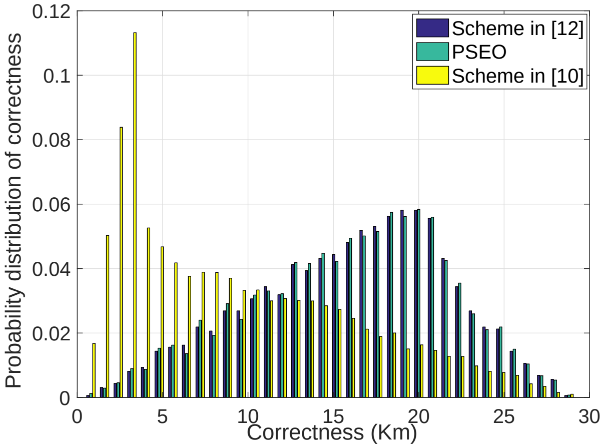


Fig. 6. Privacy performance of IU for three methods

7.1 Privacy Performance of PSEO and PF-PSEO

To evaluate how PSEO/PF-PSEO protect IUs'/SUs' information, we use the location privacy as the representative. The same conclusions for other operating information (e.g., operating frequency bands) then follow. We adopt the correctness [31], which is the distance between the estimated location and the actual location, as the metric to evaluate the privacy preservation capability. The larger correctness, the better/higher privacy performance. Due to the space limitation, we only present the performance of PSEO while PF-PSEO achieves similar performance regarding IUs'/SUs' privacy protection.

Fig. 6 and Fig. 7 compare the privacy performance of PSEO with two latest methods (e.g., the Perturbation with additive noise method in [10] and the method in [12]). The sum of all the probabilities for each method is 1. X_q, Y_q, H_q, and F_q in these two figures are 40, 40, 12, and 10 grids, respectively. The estimated attenuation value for the quantized gird is calculated using the "average value estimation method" (refer to equation (34)).

Fig. 6 compares the probability distribution of correctness of the three methods (PSEO and those in [10][12]). Since the secret keys are only held by users themselves, adversaries can only generate the secret keys via random guessing. Thus the probabilities of correctness under PSEO and [12] are distributed as the random guessing in Fig. 6, indicating that adversaries cannot infer additional information about the IU, without the correct secret keys. However, for [10], adversaries can roughly infer the user's location information, i.e., extracting user information from the data preserved by perturbation (e.g., with the additive noise method) [10]. For example, in Fig.6, under the scheme in [10], when the correctness is less than or equal to 5km, the corresponding probability is around 0.364. In contrast, the probabilities for PSEO and [12] are both lower than 0.042. In other words, for [10], there is still a high probability of revealing IU's rough location. By contrast, the corresponding probabilities for the scheme in [12] and our proposed PSEO are much lower.

For both IU and SU location protection (in Fig.6 and Fig.7, respectively), PSEO and the method in [12] achieve nearly the same correctness. That is because both leverage the properties of homomorphic cryptosystems to protect users' privacy. However, as mentioned before, the scheme in [12] is vulnerable to the collusion attack. By contrast, our proposed PSEO is able to protect IUs operational information under that attack. Moreover, PSEO requires much less online overhead (thanks to the proposed BICS) than [12], which will be shown in Section 7.3.

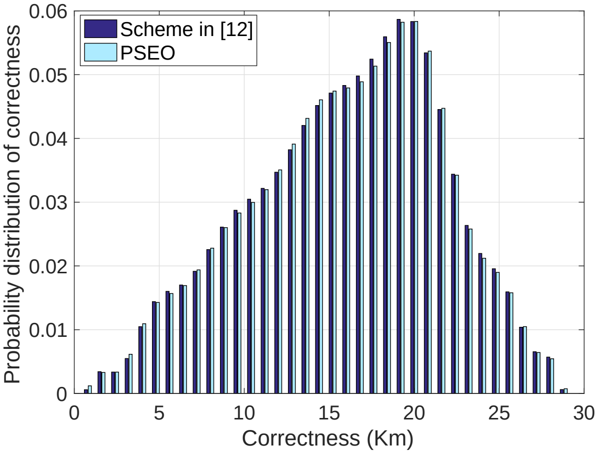


Fig. 7. Privacy performance of SU for two methods

7.2 Spectrum Utilization for PSEO and PF-PSEO

The accuracy of accepting or rejecting a spectrum request is the decisive factor to the proper operation of SAS. As discussed above, we first use the positive decision error ratio (PDER), the negative decision error ratio (NDER), and the total decision error ratio (TDER) to evaluate the spectrum utilization of the proposed schemes.

Table. 3 illustrates TDER, PDER and NDER. The grid tuple of (X_q, Y_q, H_q, F_q) is $(40, 40, 12, 11)$, i.e., X_q, Y_q, H_q, F_q in the equation (31) are 40, 40, 12, and 11 grids, respectively. We use three estimation methods, e.g., equation (32), (33) and (34), to calculate the estimated attenuation value for each gird. From this table, it is clear that PSEO is able to guarantee a very small error rate, which is similar to that of [12], but with much lower complexity. Besides, the error rates of PSEO and [12] are much smaller than those of [10]. This is because the scheme in [10] preserves users' private data by adding noise signals into users' operational information, causing extra errors to users' data. By contrast, thanks to the properties of the cryptosystem, our proposed PSEO enables secure interference budgeting/calculation without inducing extra errors to user's operational information. This table also shows the decision error ratios for different estimation methods (e.g., equation(32), (33) and (34)) used in the attenuation map quantization.

TABLE 3
Decision Error Ratios

Result Methods	TDER	PDER	NDER
PSEO equ (32)	0.031	0.000	0.031
PSEO equ (33)	0.029	0.029	0.000
PSEO equ (34)	0.035	0.019	0.016
Perturbation in [10]	0.061	0.037	0.024
Scheme in [12]	0.027	0.027	0.000

To further evaluate the spectrum utilization of different schemes, we provide Fig. 8 adopting the SU's averaged

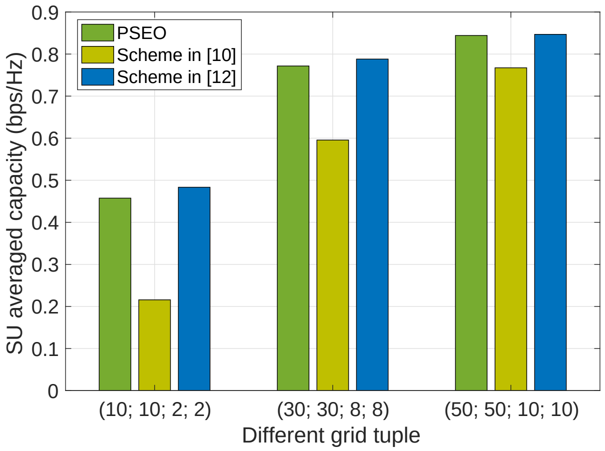


Fig. 8. SU's averaged capacity of different schemes

capacity [32] as a metric. From this figure, it is clear that the SU's averaged capacity of our proposed PSEO is more than 30% (on average) higher than that of [10]. This is because our proposed PSEO can achieve smaller decision errors than [10], thereby SU can obtain a higher probability to utilize IU's spectrum. It is noteworthy that the proposed PSEO achieves similar spectrum utilization to that of [12] but with much lower complexity. Besides, for each scheme, increasing the number of grids (e.g., the grid's size is smaller) can increase the SU's averaged capacity.

From Table. 3 and Fig. 8, either the decision accuracy and SU's averaged capacity can demonstrate the spectrum utilization of different schemes. Due to the space limitation, in the sequel, we only discuss the decision accuracy.

Fig. 9 shows TDER of different schemes considering the user mobility. In this case, the SUs move but do not report their new locations, i.e., not comply with the FCC's requirement (explained in Section 2.1) and are treated as dishonest users. In this figure, the grid tuple is (50, 50, 10, 10). The total number of SUs is 200, and the number of mobile SUs is 100. The processing time for SU's request is 16.5s (refer to Table 4). The speed of mobile SU (such as Vehicle-mounted base stations) is $100 - 120km/h$, so SU can move up to 600m during this period. If SU moves straight, it can move up to two grids (with the side length of 400 m) horizontally. As such, we select the number of grids moved by SU is $N_m = 0, 1, 2$. As can be seen from Fig. 9, TDER for each scheme dramatically increases with more grids moved by SU. This is because the SU's mobility results in a difference between its reported data and the actual information, thereby severely impacting the decision accuracy of SU's requests. As aforementioned, this kind of dishonest SUs is recognized and later banned under PSEO.

Table. 4 shows the tradeoff between the complexity and the accuracy performance. It is clear that the quantization tuples with the less number of grids (e.g., the grid's size is larger) require less processing time, while its performance degrades (e.g., the decision errors increase). Take the quantization tuple $(10, 10, 2, 2)$ as an example, its processing time for each SU's request is 0.108s, while its TDER is 0.4700. Moreover, the TDER for the tuple $(50, 50, 10, 10)$ is 0.0220, requiring processing time 16.50s for each SU request. Therefore, the performance of spectrum utilization can be improved at the expense of the online complexity. In the practical deployment, difference sizes of grids can be selected depending on the online complexity and the accuracy requirements.

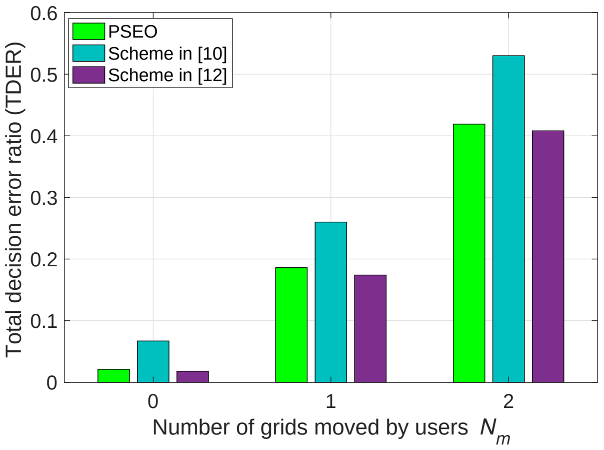


Fig. 9. Impact of SU's mobility on total decision error ratio

TABLE 4
Tradeoff between the complexity and the decision error ratio.

Result Tuple	TDER	Online processing time
$(10, 10, 2, 2)$	0.4700	0.108s
$(20, 20, 4, 4)$	0.2150	0.952s
$(30, 30, 8, 8)$	0.1060	3.81s
$(40, 40, 10, 10)$	0.0360	9.72s
$(50, 50, 10, 10)$	0.0220	16.50s

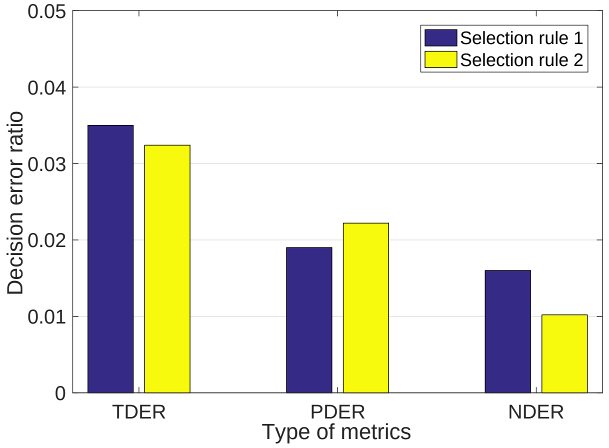


Fig. 10. Impact of selecting IU's threshold on spectrum utilization

Fig. 10 illustrates the effect of different rules in selecting IU's thresholds (refer to the 2nd step in PSEO) on decision accuracies. The selection rule 1 means selecting the smallest value in the grid as the interference threshold, which is also the method used in the 2nd step in PSEO. The selection rule 2 denotes calculating the interference threshold from IU's location to the centers of other grids. As can be seen, the NDER under the rule 1 is slightly higher than that under the rule 2. This is because although this operation can protect IU from the interference caused by SU, it minimizes the interference threshold in each grid, increasing the NDER. By contrast, the PDER under the rule 1 is slightly smaller than that under the rule 2. Therefore, the user can select different selection rules based on their requirements.

Fig. 11 shows the impact of converting a non-integer

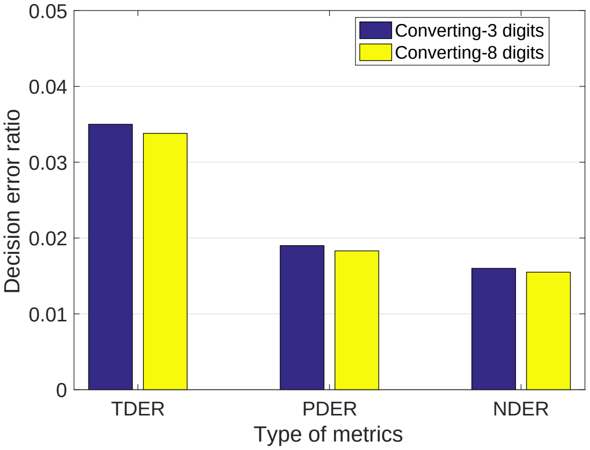


Fig. 11. Impact of conversion on spectrum utilization

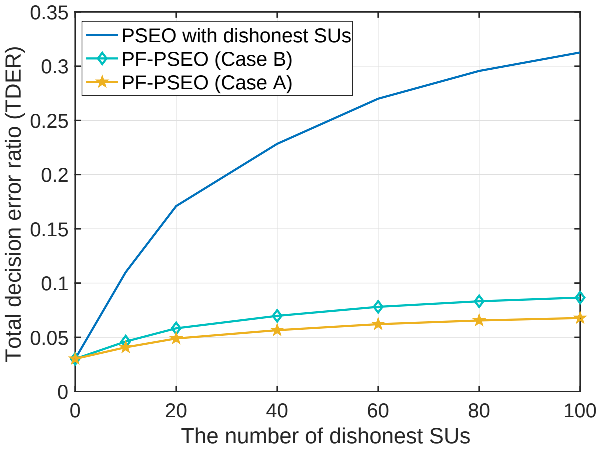


Fig. 12. TDER of PF-PSEO and PSEO with number of dishonest SUs

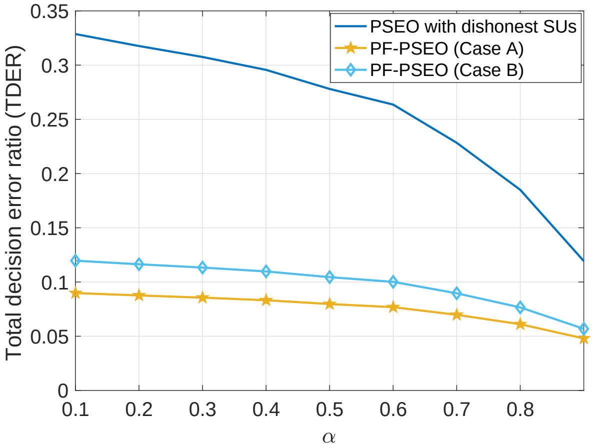


Fig. 13. TDER of two schemes with different values of α

to an integer (refer to the paragraph below the equation (11)) on the spectrum utilization. Note that this conversion process, which slightly degrades the spectrum utilization, is required because the Paillier cryptosystem is only suited for the non-negative integer numbers. From this figure, it is clear that the decision error ratio for converting a non-integer to an integer by rounding up to 8 digits is slightly smaller than by rounding up to 3 digits, with more computational complexity.

Fig. 12 and Fig. 13 show the decision accuracy of PF-PSEO with dishonest SUs. We use TDER to evaluate PF-PSEO's performance. In these two figures, all SUs can apply for the spectrum access initially, but those who are dishonest will being punished and potentially banned under PF-PSEO. The TDER is calculated by averaging all values within a period of time, depending on different cases (i.e., Case A and Case B under PF-PSEO). Take the Case A as an instance, the period is from the first dishonest user behaves dishonestly to the last dishonest user gets banned. We assume the dishonest SUs prefer to behave dishonestly when they can apply for the spectrum access. This leads to more decision errors on spectrum allocation, which can better evaluate the performance of the proposed PF-PSEO in eliminating the severe impact caused by those dishonest SUs. The quantization tuple (X_q, Y_q, H_q, F_q) is $(40, 41, 12, 8)$. We use equation (34) to achieve the attenuation value for each grid. The total number of SUs in these three figures is 200.

Fig. 12 shows the TDER of PF-PSEO and PSEO v.s. the number of dishonest SUs. β and α are 3 and 0.7, respectively. The γ's initial value is 0. We show the TDER of PF-SPEO with two cases, e.g., $CaseA$ and $CaseB$ in Section 4.1. According to this figure, PF-PSEO is able to achieve a much smaller TDER than PSEO. For example, when the dishonest SUs' number is 100, the TDER for PSEO is 0.313 that is much higher than PF-PSEO (e.g., 0.068 for $CaseA$ and 0.087 for $CaseB$). Additionally, for PF-PSEO, $CaseA$ achieves a smaller value of TDER than $CaseB$. That is because $CaseA$ is capable of rejecting regularly dishonest SUs' requests permanently drawing support from γ, which mitigates the negative impact caused by dishonest SUs.

Fig. 13 describes the effect of α on TDER of PF-PSEO and PSEO. The number of dishonest SUs is 40. The decrease of α means that the reputation of SU's honesty is becoming poor. Obviously, the TDER of PSEO increases much more dramatic than that of PF-PSEO (including $CaseA$ and $CaseB$) with the decrease of α. For example, if α decreases from 0.9 to 0.1, reflecting the decreasing dishonesty of the SU, the TDER of PSEO rises from 0.119 to 0.327. By contrast, the corresponding TDER of PF-PSEO only grows slightly, e.g., from 0.048 to 0.089 for $CaseA$ and from 0.057 to 0.119 for $CaseB$. This implies that PF-PSEO is more robust to the impact of α or dishonest SUs than PSEO.

We can conclude from Fig. 12 and Fig. 13 that the performance of PSEO dramatically suffers from dishonest SUs. By contrast, PF-PSEO is capable of achieving much better performance with the help of the punishment and forgiveness scheme.

7.3 Online Overhead of PSEO and PF-PSEO

In this subsection, we evaluate the online overhead of PSEO and PF-PSEO.

As shown in Table 1, with the same N, the numbers of homomorphic addictions and subtractions of PSEO and PF-SPEO are about one-fourth of those in [12]. Moreover, PSEO and PF-PSEO require only half the number of encryptions and scalar multiplications of the scheme in [12]. Notably, the value of N is usually large in SAS, making the complexity reduction of PSEO and PF-PSEO very much significant, compared with that of [12]. For instance, in our simulation (run on four distributed desktops with Intel XeonE5-2690 and 64GB RAM), when the total number of grids $N = 40 \times 40 \times 12 \times 8$, PSEO requires 9.61s to process a SU's request, and the total time to serve 1000 SUs is around 2.67h (the online processing time of PF-PSEO is the same as PSEO due to the very close number of operations). That can be significantly reduced with specialized hardware implementation/design in practice. For the method in [12], the online processing time for each SU's and 1000 SUs' requests are 18.94s and 5.26h, respectively (i.e., doubling the processing time of PSEO). Note that the SAS administrator is a powerful database/server [3] that can efficiently handle a great number of complex operations. The IUs are military radar systems and SUs can be base stations of cellular systems [3] that are also rich in computing resource.

8 Conclusion and Future Work

In this paper, we investigated the operational privacy issue of IUs and two types of SUs (e.g., honest SUs and dishonest SUs) in the SAS architecture. Specifically, for the case of IUs and honest SUs, we provided an operational information preserving scheme for IUs and SUs in SASs. The cores of our scheme are the blind interference calculation scheme (BICS) and a novel information exchange procedure using homomorphic encryption and obfuscation. BICS facilitates the interference budgeting without requiring IUs or SUs to reveal their operational information. That also reduces the online overhead of our scheme. For dishonest SUs, we proposed a punishment and forgiveness mechanism to encourage SUs to provide truthful information. Extensive simulation results showed that PSEO and PF-PSEO are able to preserve the operational privacy of IUs and honest/dishonest SUs with much less online overhead, compared with the state of the art schemes.

In this paper, we only investigated the privacy issue based on the condition that the SAS administrator processes SUs' requests sequentially. The scenario that the SAS administrator processes multiple SUs' requests simultaneously has not been considered and left for future work. One may also leverage the properties of blockchain [33] to protect the user's operational privacy.

References

[1] A. Osseiran, F. Boccardi, V. Braun, K. Kusume, P. Marsch, M. Maternia, O. Queseth, M. Schellmann, H. Schotten, H. Taoka *et al.*, "Scenarios for 5g mobile and wireless communications: the vision of the metis project," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 26–35, 2014.

[2] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks*, vol. 50, no. 13, pp. 2127 – 2159, 2006.

[3] FCC, "Report and order and second further notice of proposed rulemaking," *Federal Communication Commission*, April 2015.

[4] P. Zhou, W. Wei, K. Bian, D. O. Wu, Y. Hu, and Q. Wang, "Private and truthful aggregative game for large-scale spectrum sharing," *IEEE Journal on Selected Areas in Communications*, vol. PP, no. 99, pp. 1–1, 2017.

[5] M. M. Sohul, M. Yao, T. Yang, and J. H. Reed, "Spectrum access system for the citizen broadband radio service," *IEEE Communications Magazine*, vol. 53, no. 7, pp. 18–25, 2015.

[6] T. Institute, "Mobile broadband services in the 2300-2400 mhz frequency band under licensed shared access regime," *France: ETSI*, vol. V1.1.1, 2013.

[7] M. Clark and K. Psounis, "Can the privacy of primary networks in shared spectrum be protected?" in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, 2016.

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

[9] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: Architecture and algorithms," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 1–18, Jan 2008.

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

[11] J. Liu, C. Zhang, H. Ding, H. Yue, and Y. Fang, "Policy-based privacy-preserving scheme for primary users in database-driven cognitive radio networks," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–6.

[12] Y. Dou, K. Zeng, H. Li, Y. Yang, B. Gao, K. Ren, and S. Li, "p^2 -sas: Privacy-preserving centralized dynamic spectrum access system," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 1, pp. 173–187, Jan 2017.

[13] R. K. Sharma and D. B. Rawat, "Advances on security threats and countermeasures for cognitive radio networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 1023–1043, 2015.

[14] X. Zhou and H. Zheng, "Trust: A general framework for truthful double spectrum auctions," in *IEEE INFOCOM 2009*, April 2009, pp. 999–1007.

[15] F. Wu and N. Vaidya, "A strategy-proof radio spectrum auction mechanism in noncooperative wireless networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 5, pp. 885–894, May 2013.

[16] X. Feng, Y. Chen, J. Zhang, Q. Zhang, and B. Li, "Tahes: A truthful double auction mechanism for heterogeneous spectrums," *IEEE Transactions on Wireless Communications*, vol. 11, no. 11, 2012.

[17] M. R. Hassan, G. Karmakar, and J. Kamruzzaman, "Reputation and user requirement based price modeling for dynamic spectrum access," *IEEE Transactions on Mobile Computing*, vol. 13, no. 9, pp. 2128–2140, 2014.

[18] K. Chamberlin and R. Luebbers, "An evaluation of longley-rice and gtd propagation models," *IEEE Transactions on Antennas and Propagation*, vol. 30, no. 6, pp. 1093–1098, 1982.

[19] PCAST, "Report to the president realizing the full potential of government-held spectrum to spur economic growth," 2012.

[20] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 1999, pp. 223–238.

[21] M. Cotton and R. Dalke, “Spectrum occupancy measurements of the 3550 3650 mhz maritime radar band near san diego, ca,” NTIA Report TR-14-500, 2014.

[22] A. Nika, Z. Li, Y. Zhu, Y. Zhu, B. Y. Zhao, X. Zhou, and H. Zheng, “Empirical validation of commodity spectrum monitoring,” in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, ser. SenSys ’16, 2016, pp. 96–108.

[23] L. Yang, Z. Zhang, B. Y. Zhao, C. Kruegel, and H. Zheng, “Enforcing dynamic spectrum access with spectrum permits,” in *Proceedings of the Thirteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc ’12, 2012, pp. 195–204.

[24] J. Park, J. H. Reed, A. A. Beex, T. C. Clancy, V. Kumar, and B. Bahrak, “Security and enforcement in spectrum sharing,” *Proceedings of the IEEE*, vol. 102, no. 3, pp. 270–281, March 2014.

[25] E. L. Lawler and D. E. Wood, “Branch-and-bound methods: A survey,” *Operations research*, vol. 14, no. 4, pp. 699–719, 1966.

[26] FCC, “Amendment of the commission rules with regard to commercial operations in the 3550 - 3650 mhz band, federal communications commission,” *Federal Communication Commission*, September 2015.

[27] T. Granlund *et al.*, *GNU MP 6.0 Multiple Precision Arithmetic Library*. Samurai Media Limited, 2015.

[28] *Longley-Rice Methodology for Evaluating TV Coverage and Interference*, document FCC 69, Office of Eng. and Technol. (OET) Bulletin, 2004.

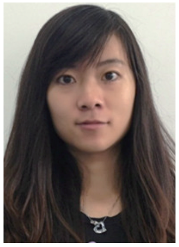
[29] *http://www.qsl.net/kd2bd/splat.html*.

[30] $http : //dds.cr.usgs.gov/srtm/version2_1/SRTM3$ $/North_America/$.

[31] R. Shokri, G. Theodorakopoulos, J. L. Boudec, and J. Hubaux, “Quantifying location privacy,” in *2011 IEEE Symposium on Security and Privacy*, 2011.

[32] J. Kim, Y. Shin, T. W. Ban, and R. Schober, “Effect of spectrum sensing reliability on the capacity of multiuser uplink cognitive radio systems,” *IEEE Transactions on Vehicular Technology*, vol. 60, no. 9, 2011.

[33] F. Tschorsch and B. Scheuermann, “Bitcoin and beyond: A technical survey on decentralized digital currencies,” *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 2084–2123, thirdquarter 2016.



Qingqing Cheng received her M.E. degree from the Harbin Institute of Technology, China in 2014 and her Master of Research (MRes) degree from the Macquarie University, Australia, in 2016. She is currently working as a PhD candidate in the School of Electrical and Data Engineering at the University of Technology Sydney, Australia. Her research interests include 5G security, privacy preservation, cognitive radio, deep learning, and massive MIMO.

Diep N. Nguyen received his M.E. and PhD degrees in electrical and computer engineering from the University of California, San Diego, CA and The University of Arizona, Tucson, AZ, USA, respectively. He was a DECRA Research Fellow with Macquarie University, a member of Technical Staff with Broadcom, San Diego, CA, USA, and ARCON Corporation, Boston, MA, USA, and a Consultant for the Federal Administration of Aviation on turning detection of UAVs and aircraft, for the US Air Force Research Laboratory on Anti-Jamming. He is currently a faculty member at the Faculty of Engineering and Information Technology, University of Technology Sydney, Australia.

Eryk Dutkiewicz received his B.E. degree in Electrical and Electronic Engineering from the University of Adelaide in 1988, his M.Sc. degree in Applied Mathematics from the University of Adelaide in 1992 and his PhD in Telecommunications from the University of Wollongong in 1996. His industry experience includes management of the Wireless Research Laboratory at Motorola in early 2000s. He is currently the Head of School of Electrical and Data Engineering at the University of Technology Sydney, Australia. He has held visiting professorial appointments at several institutions including the Chinese Academy of Sciences, Shanghai JiaoTong University and Macquarie University. His current research interests cover 5G networks and medical body area networks.

Markus Mueck received the Engineering degrees from the University of Stuttgart, Stuttgart, Germany, and the Ecole Nationale Suprieure des Tlcommunications (ENST), Paris, France, and the Doctorate degree in communications from ENST. He oversees Intels technology development, standardization, and partnerships in the field of spectrum sharing. In this capacity, he has contributed to standardization and regulatory efforts on various topics such as spectrum sharing within numerous industry standards bodies, including the European Telecommunications Standards Institute (ETSI), Third-Generation Partnership Project, the IEEE, the Wireless Innovation Forum, and the European Conference of Postal and Telecommunications Administrations (CEPT). He is an Adjunct Professor of engineering with University of Technology, Sydney, Australia, and acts as an ETSI Board Member supported by Intel and as the general Chairman of the ETSI RRS Technical Body (Software Radio and Cognitive Radio Standardization).