Pri-Share: Enabling Inter-SAS Privacy Protection via Secure Multi-Party Spectrum Allocation

Hexuan Yu, Shanghao Shi, Yi Shi, Eric Burger, Y. Thomas Hou, and Wenjing Lou

Virginia Polytechnic Institute and State University, VA, USA

***Abstract*—Dynamic spectrum sharing has emerged as a promising solution to address the spectrum scarcity challenge. Currently, the FCC has designated several Spectrum Access Systems (SAS) administrators to deploy their SAS that coordinates the usage of the certificated shared band(s) such as the 3.55-3.7 GHz CBRS band. The SAS ensures that the incumbent's access to the shared band is guaranteed while also granting commercial users access rights when the incumbents are not present. However, explicitly sharing the spectrum band(s) information among participants raises privacy concerns. Certain participants, such as curious SAS administrators, have the ability to deduce the confidential operational patterns of the incumbents through the Environmental Sensing Capability (ESC) or Incumbent Informing Capability (IIC) notifications. Additionally, a curious SAS administrator may obtain the client's operational information of other SAS administrators throughout the process of inter-SAS coordination.**

We propose Pri-Share, a novel privacy-preserving spectrum sharing paradigm that tailors the threshold-based private set union (PSU) and homomorphic encryption (HE) techniques to address the aforementioned privacy problems. Specifically, it enables all parties to jointly compute a unified spectrum allocation plan to resolve the potential conflicts between different parties while safeguarding the confidentiality of each stakeholder's spectrum requirements and usage. Pri-Share also ensures that while a curious participant might ascertain the usage of a particular spectrum band, they are unable to deduce the precise identity of the party utilizing it. Besides, Pri-Share adheres to the key spectrum allocation regulations outlined by FCC (part 96), such as assurance of access rights for various priority levels.

Our implementation result shows that Pri-Share can be achieved with notable computational and communication efficiency, indicating the practicality and feasibility of our proposed design.

***Index Terms*—Spectrum Sharing, Privacy, Private Set Union, and Homomorphic Encryption.**

I. Introduction

The radio spectrum is a crucial resource for the wireless industry. In the US, the governance of spectrum band operations and the authorization of various wireless services, such as 5G communication, WiFi, remote sensing, radar, and satellite communications, fall under the jurisdiction of the Federal Communication Commission (FCC) and the National Telecommunications and Information Administration (NTIA). However, the recent surge in wireless communications has led to an escalating demand for spectrum resources, resulting in a spectrum scarcity issue. Consequently, regulators have started opening up bands that were previously reserved for federal usage, in an effort to enhance spectrum access opportunities. The 3.55 GHz-3.7 GHz citizens broadband radio service (CBRS) band has been authorized by the FCC and adopted by the wireless industry for commercial deployments since 2020. Furthermore, the FCC is also examining the potential for shared use of other bands, including the 3.1 GHz-3.55 GHz mid-band [1], the 12.2 GHz-12.7 GHz satellite-terrestrial band [2], and the 42 GHz-42.5 GHz millimeter wave band [3]. Currently, these spectrum bands are primarily owned by federal users (i.e. incumbents) such as Navy and Army radars, and they are either already being shared or are set to be shared with commercial entities, such as AT&T and T-Mobile, for deploying their commercial services. However, a stringent regulation is in place: commercial users must not interfere with the operations of the incumbents. Moreover, they are obligated to vacate the spectrum within a strict time frame whenever the incumbents are active and require access to these bands.

A. Spectrum Access System

The FCC has designated several SAS administrators to govern the operation of the CBRS band. These administrators are in charge of deploying their SAS server to coordinate the spectrum usage of different tiers of users following the FCC's rules. The SAS adopts a three-tiered access model for the CBRS band, including the incumbent tier, the priority access license (PAL) tier, and the general authorized access (GAA) tier. Among them, the incumbents are considered as *primary users*, who mostly are federal users; both the PAL tier and GAA tier users are commercial users (or *secondary users*). The priority level is arranged as follows: incumbents $>$ PAL $>$ GAA. The SAS shall ensure that higher-priority users have higher access rights and their operations shall not be affected by lower-tier users [4].

A simplified architecture of the current spectrum sharing system (SAS) is depicted in Figure 1. The SAS comprises of three major components: the SAS servers, the ESC/IIC, and the spectrum users. SAS servers are deployed by the SAS administrators and are responsible for fulfilling critical spectrum sharing functions including incumbent protection, spectrum assignment, database management, etc. Each SAS server manages its own spectrum users and performs local spectrum assignments for them. Spectrum users submit spectrum access requests to the SAS server, which include information such as (*id, priority, band(s), location*). Upon receiving transmission grants, spectrum users deploy Citizen Broadband Radio Service Devices (CBSDs) to operate within the allocated frequency band for their service. Notably, as various SAS administrators may overlap in *the same* service area, they must conduct an

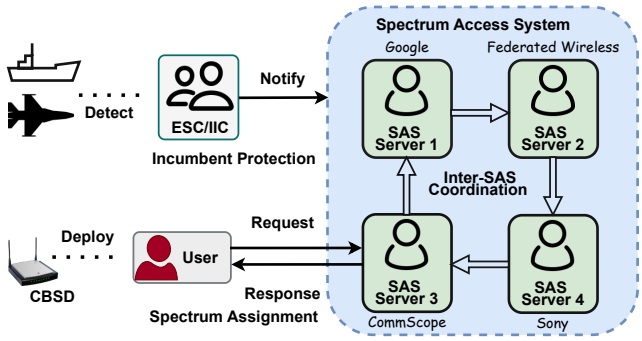


Fig. 1: SAS Architecture

inter-SAS coordination process at least once a day to keep their state synchronized and prevent possible conflicts in spectrum bands. Besides, the ESC or IIC is in charge of detecting the presence of the incumbents and notifying SAS servers about their presence within a timeframe of 300 seconds (FCC Part 96.15 (a)(4)).

B. Privacy concerns

The *ESC/IIC notification* and *inter-SAS coordination* procedures might reveal detailed spectrum usage of a particular party. However, such disclosed information can enable a curious party to infer confidential information pertaining to the operations of incumbents or commercial users under other SAS administrators. Incumbent users, particularly military users, are averse to any potential inference of their whereabouts and activities. Commercial spectrum users also wish to safeguard their operational information from access by other SAS administrators, in order to protect their privacy and commercial secrets.

A practical approach to protecting the privacy of each participant during coordination and allocation is to encrypt and hide the detailed spectrum usage information in ciphertext.

Nevertheless, implementing all functions and enforcing the complex spectrum sharing policies becomes a formidable task when all involved parties only disclose the encrypted information about their spectrum usage.

C. Our Solution

We propose Pri-Share, a privacy-preserving multi-party computation (MPC) framework built for complex spectrum-sharing systems. The core objective is to facilitate collaborative computation among various SAS participants for a unified spectrum allocation plan, while simultaneously managing and resolving any potential conflicts. Importantly, Pri-Share is designed to obscure the detailed spectrum requests and usage data of each participant. This ensures that a participant can ascertain the usage of a spectrum band without identifying the specific party occupying it.

In essence, Pri-Share is a decentralized protocol that utilizes homomorphic encryption to conceal the band request, and leverages the Private Set Union (PSU) technique to compute a union set over the encrypted requests and identify the *conflicted* and *prohibited* bands. We also incorporate a commitment scheme throughout the protocol, to prevent misbehavior SAS servers from sending dishonest queries and ensure fairness of the spectrum sharing process.

In a classic PSU protocol, the duplicated encrypted elements are identified and eliminated to ensure the singularity of each item in the union set by allowing each unique element to only appears once. In order to meet the requirements for spectrum sharing and allocation, our method modifies the conventional PSU strategy. Instead of removing duplicated elements, our focus is on identifying those duplicated elements that lead to overlapping band requests. Subsequently, we guide the allocation of these conflicted bands.

Additionally, while conventional MPC-based applications often face computational constraints due to large input sets or the number of participants involved, our MPC-based approach presents a viable solution to privacy challenges in spectrum-sharing systems for several reasons. Firstly, each participant deals with a limited size of input; secondly, the use of Pri-Share involves only a few participants, thus ensuring manageable communication overhead; and thirdly, SAS servers typically possess substantial computational resources. The results of our proof-of-concept implementation further demonstrate that this approach can be executed with significant computational and communication efficiency, underlining its practicality and feasibility.

To summarize, our contributions can be listed as follows:

1) We propose a privacy-preserving protocol that utilizes threshold-based PSU and homomorphic encryption techniques to prevent operational information leakage and inference in the spectrum sharing and allocation process.
2) Our protocol complies with standard band allocation rules and supports misbehavior detection while preserving privacy by incorporating a commitment scheme.
3) Our evaluation demonstrates this privacy-preserving spectrum sharing framework is practical in terms of computation and communication efficiency.

II. Related Work

Several existing studies also focus on safeguarding the privacy of SAS participants, although they vary in terms of threat models and protection goals. Bahrak et al. propose an obfuscation technique to protect the primary users' information from being inferred from secondary users' query results [5]. Clark et al. introduce an analytical framework to quantify the privacy of incumbents based on varying degrees of adversary capabilities and provide obfuscating strategies to protect incumbent privacy [6]. In parallel, Dou et al. [7] and Li et al. [8] utilize homomorphic encryption techniques to protect the privacy of incumbents from untrusted SAS servers. While [9], [10] focus on more unified solutions to protect the privacy of both the incumbent and the secondary users against untrusted SAS servers.

However, all the aforementioned works are designed based on a single SAS server model. Unfortunately, this goes against the present SAS service model, as FCC currently appoints *multiple* SAS administrators, who normally coexist in *the*

same geo-location. The aforementioned privacy protection mechanisms fall short of addressing the challenges created by this distributed model, such as how to preserve spectrum users' privacy without affecting inter-SAS coordination and how to resolve conflicts in accordance with the FCC's regulation rules. Several blockchain-based SAS designs [11], [12] have adopted the decentralized model that considers multiple SAS servers. However, [12] focuses on SAS fault-tolerance and auditability without taking privacy into consideration; [11] focuses on protecting the privacy of each data record retrieved by secondary users from SAS databases and maintaining the trustworthiness of these databases.

III. System Model

In this section, we will cover system entities and threat models, followed by a high-level workflow.

A. Threat Model

Prior to Pri-Share execution, our system requires that **FCC** act as the *trusted* party to generate and distribute a secret key sk among all SAS servers within the same region. However, it should be noted that the key generation and distribution is a *one-time* event. During the Pri-Share execution, encompassing conflict resolution and band allocation, all computations are executed in a *decentralized manner* and without the involvement of FCC.

The participants in Pri-Share are the **SAS Servers** and **ESC/IIC**, which may share spectrum bands in an overlapping manner within the same region.

Pri-Share regards **SAS Servers** as *honest-but-curious* parties with the intent of exploring client and operational information of other SAS servers in order to gain any advantage for band allocation; nevertheless, they will adhere to the prescribed protocols, such as encryption, decryption, and commitments.

The confidentiality and integrity of communication contents between any parties are protected by the Transport Layer Security (TLS) protocol by leveraging the existing CBRS Public Key Infrastructure (PKI) [13]. Therefore, the description pertaining to the protection against **outsiders** throughout the Pri-Share is omitted.

B. Allocation Rules and Workflow Overview

1) FCC Part 96 Rules: Generally, licensed spectrum clients have priority access in cases of spectrum scarcity. The FCC has defined comprehensive regulatory rules to govern the operations of the participants of the CBRS ecosystem. Pri-Share design considers the following major band allocation rules:

- **Priority rule**: Higher priority users are protected from lower priority users. — *§ 96.1-(b): Priority Access Licensees and General Authorized Access Users must not cause harmful interference to Incumbent Users and must accept interference from Incumbent Users. General Authorized Access Users must not cause harmful interference to Priority Access Licensees and must accept interference from Priority Access Licensees.*

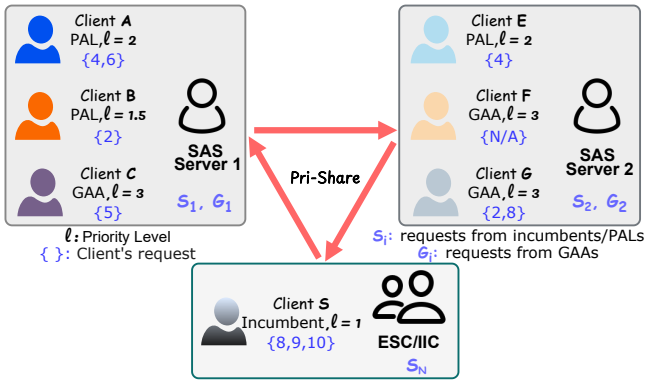


Fig. 2: An example of system entities and their inputs in Pri-Share

- **PAL channel assignment rule**: SAS shall assign continuous channels (10 MHz blocks) to the same PAL user. — *§ 96.25-(b)(2)(i): An SAS must assign multiple channels held by the same Priority Access Licensee to contiguous channels in the same License Area, to the extent feasible.*
- **PAL number limitations**: PAL users have maximum spectrum usage number limitations. — *§ 96.31-(a): Priority Access Licensees may aggregate up to four PAL channels in any License Area at any given time. § 96.13-(a)(1): No more than seven PALs shall be assigned in any given License Area at any given time.*

2) Workflow Overview: We show a simple example in Fig. 2 to demonstrate the system entities and a high-level workflow of Pri-Share. In a designated region, there is usually more than one SAS administrator (e.g., Google, Federated Wireless), and at least one ESC or IIC system. Each SAS administrator (or SAS server) manages a group of commercial clients and routinely gathers all spectrum band requests from them. In Pri-Share, it should combine these requests into a single private set.

ESC/IIC systems are responsible for detecting the presence of federal incumbent signals (e.g., military radar) and triggering the interference protection mechanism. In practice, upon a successful detection of incumbent activity, SAS servers shall be notified and the underlying clients will receive a termination grant and cease operations in the conflicted frequency range. In Pri-Share, ESC/IIC also forms a private input set containing the bands that are currently utilized by incumbents. Pri-Share is executed among SAS Servers and ESC/IIC, without the involvement of the spectrum clients. The private input sets from SAS servers and ESC/IIC are inputted into Pri-Share, enabling the collaborative computation of a universal allocation plan.

We denote the 15×10 MHz available CBRS bands as an ordered list of integers:$\{1, 2, 3, \ldots, 15\}$. In the current round, for example, *ESC/IIC* detected that client S (incumbent) is using bands $\{8, 9, 10\}$, therefore, *ESC/IIC* shall input a private set S to the Pri-Share scheme to convey to other parties the prohibition needs due to incumbent activities. Both SAS Server 1 and SAS Server 2 have three clients, in every round,

each SAS server collects all PAL client requests as a set S and another set G for all GAA users. All the requests made to Pri-Share shall be accompanied by a commitment of the designated priority level l.

We let $l = 1$ denote the incumbent user's priority, $l = 2$ denote PALs' priority level, and $l = 3$ denote GAAs's priority level, i.e., a smaller number indicates a higher priority for a particular request/prohibition. Besides, if a band requested by a PAL client is adjacent to any bands currently in use by the same PAL client, its SAS server is responsible for validating and assigning the priority level for this requested band as $l = 1.5$. For example, if the client *B* is requesting the band $\{2\}$ that is adjacent to a band currently used by itself, then this request is given a priority level $l = 1.5$ by the *SAS Server 1*.

Pri-Share then utilizes an over-threshold PSU technique to resolve spectrum-sharing requirements. This method allows the participants (SAS Servers and ESC/IIC) to input their private set and locate any requested band(s) that appear more than once in the combined private input set union. When this process is initiated, the actual elements present in each private input set and the committed priority level are not disclosed. Only in the event of a conflict in a particular requested band, the concerned competitors are required to reveal their corresponding committed priority level for fair competition, the request with the highest priority level will then be granted access to the conflicted band. In an ideal situation, where no conflicts arise among requests, the allocation of frequency bands and the priority levels associated with each request remain completely concealed from all the participants.

IV. Cryptographic Building Blocks

In this section, we introduce the homomorphic encryption scheme and the basic privacy computing operations that are utilized in Pri-Share.

A. Paillier Cryptosystem

The Paillier encryption scheme is considered to be semantically secure. It is based on Carmichael function $\lambda(n)$, or reduced Euler's totient function, over $\mathbb{Z}^*_{n^2}$. One notable characteristic of the Paillier cryptosystem is its additive homomorphic features, rendering it well-suited for applications involving computations over encrypted messages, such as multi-party computation (MPC) applications. Our approach utilizes a distributed(threshold) variant of the Paillier cryptosystem, enabling joint decryption among multiple parties.

A plain version of the Paillier cryptosystem executes as follows:

Key Generation.
Choose an RSA modulus $n = pq$, where $gcd(n, \phi(n)) = 1$. The **public key** pk is (n, g) where $g = n + 1$ (we set p and q with the same length for efficiency purposes). The **secret key** $sk = \lambda(n)$, where $\lambda(n) = lcm(\phi(p), \phi(q)) = (p-1)(q-1)$ when p, q have the same length.

Encryption.
To encrypt a plaintext $M \in \mathbb{Z}_n$, select a random number $x \in \mathbb{Z}^*_{n^2}$, the ciphertext is $C = g^M x^n \bmod n^2$.

Decryption.
With knowledge of the $sk = \lambda(n) = (p-1)(q-1)$, the plaintext M can be recovered by computing:

$$M = \frac{L\left(C^{\lambda(n)} \bmod n^2\right)}{L\left(g^{\lambda(n)} \bmod n^2\right)} \bmod n \tag{1}$$

where the L-function $L(u) = \frac{u-1}{n}$. The correctness can be validated according to the properties that: $C^{\lambda(n)} = 1 \bmod n$ and $C^{n\lambda(n)} = 1 \bmod n^2$ for any $C \in \mathbb{Z}^*_{n^2}$.

B. A Distributed Version

1) Key Generation and Distribution: During system setup, our scheme requires a *one-time* key generation and distribution process, with the FCC serving as the trusted dealer.

Key Generation.

- FCC generates an RSA modulus $n = pq$, where $gcd(n, \phi(n)) = 1$. p and q are strong prime integers, i.e., $p = 2p' + 1$, $q = 2q' + 1$, where p' and q' are also prime.
- FCC randomly generates an element $\beta \in \mathbb{Z}^*_{n^2}$, and set $m = p'q'$. For all $x \in \mathbb{Z}_{n^2}$, $x^{2nm} \equiv 1 \bmod n^2$, since $\lambda(n^2) = lcm(\phi(p^2), \phi(q^2)) = 2nm$.
- The **public key** pk consists of (n, g, θ), where $g = n + 1$, and $\theta = \beta m \bmod n$.

Key Distribution.
The **secret key** $sk = \beta m$ is shared among all participants through the *Shamir Secret Sharing* mechanism:

- To distribute the secret key among all the participants, FCC generates $a_0 = \beta m$, and randomly picks a_i from $\{0, \ldots, nm - 1\}$. A polynomial $f(x) = \sum_{i=0}^{t} a_i X^i = \beta m + a_1 X + a_2 X^2 + \cdots + a_t X^t \bmod nm$ will be sent to each participant P_i $(1 \le i \le N)$ through a secure channel. For each participant P_i, the share s_i is $f(i) \bmod nm$.
- FCC also generates and publishes the **verification keys** vk and vk_i (for participant P_i), which are necessary later time for each participant to prove that their decryption is performed correctly. FCC randomly generates $r \in \mathbb{Z}^*_{n^2}$, and calculate $vk = v = r^2 \bmod n^2$, $vk_i = v^{\Delta f(i)} \bmod n^2$. Note that $\Delta = N!$, where N is the total number of participants within one region.

Encryption.
The encryption of a plaintext M can be computed by any party who has access to the public key (n, g, θ) as $C = g^M x^n \bmod n^2$, where x is a random number in $\mathbb{Z}^*_{n^2}$.

2) Distributed Paillier Decryption: The Paillier encryption and decryption procedure takes place during each spectrum-sharing round while running our Pri-Share protocol. It is threshold-bounded and can tolerate up to t corrupted (e.g., coalition) servers among all the N servers.

Group Decryption.
In a threshold version of the Paillier cryptosystem. To decrypt a ciphertext $C = g^M x^n \bmod n^2$ and recover the plaintext M, the participants are required to execute the following steps collaboratively:

1) Each participant P_i calculates and publishes a partial decryption share $C_i = C^{2\Delta f(i)} \bmod n^2$ by using its secret share $s_i = f(i)$. Besides, P_i shall prove the validity of this partial decryption through a zero-knowledge proof π_i of equality relationship: $f(i) = \log_{v^\Delta} vk_i = \log_{C^{4\Delta}} {C_i}^2$. (Using $f(i) = \log_{C^{4\Delta}} {C_i}^2$ instead of $f(i) = \log_{C^{2\Delta}} C_i$ can make sure that the ZKP process is working in Q_{N^2}, which is a subgroup of $\mathbb{Z}^*_{n^2}$, thus to ensure soundness as in [14]).
2) The algorithm will abort if there are less than t valid partial decryption shares (i.e., does not pass the ZKP process). Otherwise, the plaintext M can be obtained by combining $t+1$ valid shares:

$$M = L\left(\prod_{i\in S} C_i^{2\mu_i} \bmod n^2\right) \times \frac{1}{4\Delta^2\theta} \bmod n \quad (2)$$

where $\mu_i = \Delta \times \lambda^S_{0,i} \in \mathbb{Z}$, $\lambda^S_{x,i} = \prod_{i'\in S\setminus\{i\}} \frac{x-i'}{i-i'}$, and $L(u) = \frac{u-1}{n}$. The proof of correctness can be found in [15].

3) Additively Homomorphic: Pri-Share requires extensive operations over encrypted values, which are made possible due to the homomorphism of the Paillier cryptosystem. Let $E(\cdot)$ denotes an Paillier encryption function using public key pk, and $D(\cdot)$ denotes the decryption function using secret key sk, Paillier cryptosystem exhibits the following homomorphisms:

- $D(E(M_1, x_1) \times E(M_2, x_2) \bmod n^2) = (M_1 + M_2) \bmod n$: Given ciphertexts over the plaintexts M_1 and M_2, we can obtain the encryption over $M_1 + M_2$ efficiently, i.e., the product of two ciphertexts will decrypt to the sum of their corresponding plaintexts.
- $D(E(M_1, x_1) \cdot g^{M_2} \bmod n^2) = (M_1 + M_2) \bmod n$: The product of a ciphertext with a plaintext M_2 raising by a basis g will decrypt to the sum of the corresponding plaintexts.
- $D(E(M)^c \bmod n^2) = c \times M \bmod n$: Given an encryption over a message M raised by the power of a constant value c , we can decrypt and obtain the product $c \times M$.

C. Private Set Union

PSU is applied when mutually distrustful parties wish to find the union of their respective private sets [16]. We utilize a special over-threshold version of the classic PSU protocol and adapt its function to fit into the spectrum sharing scenarios.

1) Polynomial Representation of Sets: In Pri-Share, each participants generates a set S representing the bands $\{S^1, S^2, \ldots, S^k\}$, with $|S| = k$ (we set $k = 15$ as the number of available bands). Note that each SAS server identifies its requested bands, whereas ESC/IIC encodes its prohibited incumbent bands. We denote the jth element in the set S_i as S_i^k. A polynomial representation with roots S^k encodes a server's requested bands and is calculated through interpolation:

$$f(x) = (x - S^1)(x - S^2)\ldots(x - S^k) = \sum_{j=0}^{k} a_j x^j \quad (3)$$

The key pillar of our Pri-Share scheme is performing multi-party computations over encrypted polynomials among multiple servers. A homomorphic encryption over a polynomial f_i is denoted by an ordered list of ciphertexts over its coefficients a_i for $i = 0, \ldots, k$, and each coefficient is encrypted individually, i.e.,

$$E(f) = E(\sum_{j=0}^{k} a_j x^j) : \{E(a_0), \ldots, E(a_k)\} \quad (4)$$

2) Basic operations over encrypted polynomials: We introduce the operations over encrypted polynomials involved in Pri-Share, by using the homomorphic properties of the Paillier cryptosystem.

1) **Summation of encrypted polynomials.** Given two polynomials $f_1 = \sum_{j=0}^{k} a_j x^j$ and $f_2 = \sum_{j=0}^{k} b_j x^j$, the encryption of $f_1 + f_2$ can be computed by multiplying their ith coefficients respectively, i.e.,

$$E(f_1 + f_2) : \{E(a_0)E(b_0), \ldots, E(a_k)E(b_k)\} \quad (5)$$

2) **Product of two polynomials, one in ciphertext and one in plaintext.** Given an encrypted polynomial $E(f_1)$ and an unencrypted polynomial f_2, we can efficiently obtain an encryption of their product $f_1 \times f_2$ with an degree of $2k$:

$$\begin{aligned} E(f_1 \times f_2) :\{&E(a_0)^{b_0}, E(a_0)^{b_1} + E(a_1)^{b_0}, \ldots, \\ &E(a_0)^{b_k} + E(a_1)^{b_k - 1} + \ldots + E(a_k)^{b_0}, \\ &\ldots, E(a_k)^{b_k}\} \end{aligned} \quad (6)$$

3) **Derivative of an encrypted polynomial.** Given an encrypted polynomial $E(f)$, the derivative polynomial $\frac{d}{dx}f$ with an degree of $k-1$ can be efficiently computed as:

$$E(f') : \{E(a_1)^1, E(a_2)^2 \ldots, E(a_k)^k\} \quad (7)$$

The above operations are the key techniques utilized throughout Pri-Share to perform secure and private computations.

D. Pedersen Commitment Scheme

To prevent unscrupulous participants from altering their private input throughout the protocol, our system incorporates a traditional cryptographic commitment scheme, known as Pedersen commitment [17]. Pedersen's commitment scheme is based on the hardness of the Discrete Logarithm (DL) problem and is computationally binding and unconditionally hiding.

Let $\mathbb{G}$ denote a group of prime order p. g and h are two generators of the group $\mathbb{G}$. The committer randomly generates a blinding factor $r \in \mathbb{Z}_p$, and computes $Com(M) = g^M h^r$, where $M \in \mathbb{Z}_p$.

The commitment $Com(M)$ allows a participant to bind to a message M without revealing it to other parties.

V. Pri-Share

Pri-Share is a decentralized privacy-preserving scheme without relying on a trusted third party, SAS servers and ESC/IIC are the participants in Pri-Share. During each

SAS synchronization cycle, ESC/IIC reports the activity of incumbents, and each SAS server gathers band requests from all of its clients.

Each **SAS server** P_i $(1 \leq i \leq N-1)$ preserves two private input sets: **(i)** S_i, encodes all the requested bands collected from its *PAL* clients, and **(ii)** G_i, encodes all the requested bands collected from its *GAA* clients; **ESC/IIC** P_N constitutes one private input set S_N encompassing the bands with immediate *incumbent* activity, namely, the bands that are currently prohibited. Note that, **ESC/IIC** is unnecessarily to be the Nth party, it can be any P_i for $i = 1 \leq i \leq N$ we denote it in this way for illustration convenience.

To put it briefly, Pri-Share aims to identify the elements that appear more than once in the joint multisets $S_1 \cup \ldots \cup S_N$ and $G_1 \cup \ldots \cup G_{N-1}$ under the collaboration among N participants, and to automatically producing a universal bands allocation plan for all participants simultaneously without showing each private input set explicitly. These operations are all executed over polynomial representations discussed previously, and the duplication element detection is conducted via evaluating the polynomial derivatives according to *Gauss–Lucas theorem*: if a polynomial $f(x)$ has a root a of multiplicity greater than one, then its derivative $f'(a)$ also has this root, i.e., $f'(a) = 0$. Additionally, to prevent private input information from being leaked, all these operations shall be performed in an encrypted manner among all participants.

A. Set Constraints of the Private Input Sets

For each SAS server, it preserves two sets, S_i and G_i, while ESC/IIC preserves one set S_N. $|S_i| = |G_i| = k$ for $1 \leq i \leq N$. In the current setting, we set $k = 15$, i.e., the total number of available CBRS bands.

We denote the 15 available CBRS bands (10MHz for each) as an ordered list of integers in $\mathcal{A} : \{1, 2, 3, \ldots, 15\}$, such that $S_i, G_i \subseteq \mathcal{A}$. If there is a request for band j from PALs or GAAs, a SAS server will set the jth element in their sets to $Sj = j$ or $Gj = j$, respectively. ESC/IIC sets $S^j = j$ if there is any incumbent activity detected on the jth band. The rest of the elements are randomly picked from a redundant set $\mathcal{R} : \{16, 17, \ldots, 30\}$. This is to ensure set alignment and prevent any party from deducing that a specific participant possesses an empty set.

For example, in Fig. 2, participant 1 (a SAS server) has a PAL client *A* requesting the bands 4 and 6, and a PAL client *B* requesting band 2, then the 2nd element S_1^2, 4th element S_1^4 and 6th element S_1^6 shall be 2, 4 and 6, respectively, whereas the remaining 12 elements of the set S_1 shall be randomly selected from R. Similarly, participant 3 (ESC/IIC) detected the presence of an incumbent user over bands 8, 9, and 10 within the designated region, it shall constitute a set $S_3 : \{..., 8, 9, 10, ...\}$.

B. Allocation Regulations and Enforcement

To meet the regulatory requirements mentioned in Section III-B1, we define *priority level* l and introduce a commitment vector $Com(l) : \{Com(l^1), \ldots, Com(l^k))\}$. Each participant shall commit to the priority level l^j from a set $\mathcal{L} : \{1, 1.5, 2, 3\}$ to its jth request or prohibition, such that $l^j \in \mathcal{L}$; for any element picked from the redundant set $\mathcal{R}$, each participant commits it to a random value such that $l^j \notin \mathcal{L}$. Due to the cryptographic properties of the Pedersen commitment scheme, two commitments $Com(l)$, $Com(l')$ will appear uniformly distributed even if the commitments are computed over the same priority level, i.e., $l = l'$. The commitment vector $\{Com(l^1), \ldots, Com(l^k)\}$ has dual purposes:

- **Tier control**: Once there is a conflict or prohibition over a band j, all the participants who had a collision over band j shall reveal the priority level of the underlying clients for competition.
- **Misbehavior protection and Auditability**: It is not permissible for any participant to alter the priority level for unfair competition after they have formed their own private input set and entered the band allocation process. The priority l stated in the commitment and the priority of the client who ultimately acquires the band must always match; an authorized authority, such as the FCC, can conduct future audits based on earlier commitments and actual spectrum operating information.

C. Pri-Share

1) Workflow: The comprehensive workflow of Pri-Share is illustrated in Fig. 3. At a high level, it can be summarized as follows:

In **Step 1**, each server maps its private input sets to the corresponding private polynomial representations, and generates the commitment vectors. At the **Step 2** and **3**, each server encrypts its private polynomial representations (i.e., a list of coefficients), and transmits them to the next server. The ultimate encrypted polynomial representations p_f, p_g is broadcasted to each party at **Step 4**; p_f, p_g can be seen as the encryption of the product of all the private polynomials f, g, respectively. They are primarily calculated using the method introduced in equation 6. Each server computes the derivatives of the polynomials p_f, p_g through the equation 7 method at **Step 5**. All participants then use these derivatives in **Step 6** to conduct over-threshold PSU and to detect duplicated elements in the final union set.

2) Outputs: The joint set V_f in step 6(c)i contains all the bands that have been requested by PALs or occupied by incumbents, including bands that could have potentially been requested by more than one PAL. Any element (i.e., band) that appears in $V_{f'}$ at step 6(c)ii indicates that it is subject to a conflict or prohibition that necessitates an additional resolution. Conversely, an element that is absent from set $V_{f'}$ indicates that it was only requested by one party and can be allocated to the requested client without issues, the associated party can then occupy that band without disclosing any information to others.

3) Allocation Plan after identifying bands conflicts and prohibition: When it comes to conflict resolution, the *Universal Allocation Plan* at step 6d executes according to the following principles:

Pri-Share Workflow

System Setup: All SAS servers and ESC/IIC within one region share the secret key sk (i.e., Distributed Paillier), to which pk is the corresponding public key.

Input: There are $N \geq 3$ honest-but-curious parties, each SAS server i has two private input sets S_i, G_i, ESC/IIC has one private input set S_N. $|S_i| = |G_i| = k$ for $1 \leq i \leq N$.

1) For server in $i = 1, \ldots, N$
   a) Each **SAS server** calculates two polynomials: $f_i = (x - S_i^1) \ldots (x - S_i^k)$ and $g_i = (x - G_i^1) \ldots (x - G_i^k)$; **ESC/IIC** calculates one polynomial: $f_N = (x - S_N^1) \ldots (x - S_N^k)$
   b) Commits to the priority level l^j behind each of the input in S_i as $\{Com_f(l_i^1), \ldots, Com_f(l_i^k)\}$
2) Server 1 encrypts its two polynomials and sends $E(f_1), E(g_1)$ to server 2
3) For server $i = 2, \ldots, N-1$
   a) receives the encryption of the polynomial $E(f_{i-1}), E(g_{i-1})$ from server $i-1$
   b) encrypt the products of the polynomials as $\lambda_i = E(f_{i-1} \times f_i)$ and $\omega_i = E(g_{i-1} \times g_i)$
   c) sends the encryption of the polynomials λ_i and ω_i to server $i + 1 \bmod N$
4) server N (i.e., **ESC/IIC**)
   a) receives $p_g = \omega_{N-1} = E(\prod_{i=1}^{N-1} g_i)$ from server $N-1$
   b) computes the encryption of the multiplied polynomials as $p_f = \lambda_N = E(\prod_{i=1}^{N} f_i)$
   c) distributes p_f, p_g to the remaining parties $2, \ldots, N-1$
5) For server $i = 1, ..., t+1$ $(t \leq N-2)$
   a) calculates the derivatives p'_f, p'_g
   b) chooses two random polynomials (blinding factors) r_i^0, r_i^1 with degree of Nk, and one random polynomial v_i with degree of $(N-1)k$
   c) calculates $p_f \times r_i^0$, $p'_f \times F \times r_i^1$ and $p'_g \times v_i$ and sends them to all other servers
6) Each server $i = 1, \ldots, N$
   a) computes $\Lambda = p_f \sum_{i=1}^{t+1} r_i^0$, $\Theta = p'_f \sum_{i=1}^{t+1} r_i^1$ and $\Omega = p'_g \sum_{i=1}^{t+1} v_i$
   b) performs an $(N, N)-$Paillier group decryption to obtain the polynomials $Q_f = D(\Lambda)$, $Q_{f'} = D(\Theta)$ and $Q_g = D(\Omega)$
   c) evaluates $Q_f(j)$, $Q_{f'}(j)$ and $Q_g(j)$ for each $j = 1, \ldots, k$
      i) j is an element appears in the union set $S_1 \cup \ldots \cup S_N$ if and only if $Q_f(j) = 0$, record all such elements in a joint set V_f.
      ii) j is a *duplicated* element in the union set $S_1 \cup \ldots \cup S_N$ if and only if $Q_{f'}(j) = 0$, record all such elements in another joint set $V_{f'}$. $V_{f'}$ contains the **conflicted** bands (between **PALs**) and **prohibited** incumbent band(s).
      iii) j is a *duplicated* element in the union set $G_1 \cup \ldots \cup G_{N_1}$ if and only if $Q_g(j) = 0$, record all such elements in another joint set V_g. V_g contains the **conflicted** bands (between **GAAs**).
   d) reveals the corresponding commitment(s) against any input element(s) that appear in $V_{f'}$, and obtains a *universal allocation plan* according to the principles defined in section V-C3.

Fig. 3: Pri-Share – A Privacy-Preserving Spectrum Sharing Scheme

1) Incumbents always retain the greatest priority, i.e., any incumbents that appear in $V_{f'}$ will win the competition according to our commitment comparison method.
2) After resolving conflicts between PALs, we allocate the remaining available bands to GAAs based on their requests.

Solving band conflicts. While resolving a certain conflict element (band) appears in $V_{f'}$ at step 6, the SAS server or ESC/IIC associated with an element j that is present in V'_f must disclose the jth commitment $Com_f(l^j)$ in their commitment vector. They then conduct a priority comparison to aid in further decision-making: access to this specific band is always granted to the party with the smallest priority number l; when all the conflicting requests hold the same priority level, a random draw is conducted by using a random oracle to generate the winner. For example, $H(\cdot)$ is a deterministic hash function, let c = the total number of conflicting parties associated with the band j, each relevant party can compute $y = H(c + j) \bmod c$, then the yth party among c parties wins the band j.

Meeting PALs' requirement prior to GAAs' assignment. According to the regulations discussed in Section III-B1, PALs have priority over GAAs of being assigned to the first 10 continuous bands, we denote such bands as a set $\mathcal{P} : \{1, \ldots, 10\}$. In case a PAL fails to obtain its requested band due to incumbents' occupation during step 6(c)ii, it shall be granted with a band from $\mathcal{P}$. We introduce two relative

complement sets: $\mathcal{P} \setminus V_f$ to denote the currently available band(s) that PALs have higher priority than GAAs, and the failed PAL shall have a chance to be reassigned a band appears in $\mathcal{P} \setminus V_f$, we denote the reassigned bands as $\mathcal{R}$ while $\mathcal{R} \subseteq \mathcal{P} \setminus V_f$, besides, the elements of $\mathcal{R}$ need to be explicitly revealed to the resting parties for preventing conflicts among GAAs in the next round.

Band allocation to GAAs. The relative complement set $\mathcal{A} \setminus (V_f \cup \mathcal{R})$ denotes the remaining bands available to GAAs after the allocation among PALs is finished. $V_f \cup \mathcal{R}$ is known to all SAS servers, any GAA requests that collide with $V_f \cup \mathcal{R}$ are automatically rejected. The rest requested bands that do not appear in the conflict set V_g can be directly occupied by the corresponding GAAs without revealing any information to other participants. The conflicted bands in V_g that have not collided with $V_f \cup \mathcal{R}$ shall also be allocated according to the commitment comparisons of the committed priority level.

Constraints on the number of bands used by a PAL user. In practice, the FCC limits the number of licenses sold in a county to no more than 7 and permits each licensee to aggregate no more than 4 spectrum bands concurrently. A primary challenge arises from the possibility that a PAL user might simultaneously be a client of multiple SAS administrators within the same region. As a decentralized conflict resolution framework, Pri-Share does not limit PALs' request number directly. Instead, a post-hoc auditing method with the help of trusted authorities is necessary to enforce such rules.

Considering the transparency of actual spectrum operating records to the trusted authority, such as the FCC, it is possible for these authorities to audit potential violations. This can be achieved through the verification of PAL licenses, which are presented by PAL users during the course of spectrum operations. More specifically, violations of number restrictions are detectable and will consequently discourage any PAL from exceeding the number limitations. In the event that a PAL has managed to secure more than four bands by the end of Pri-Share, it shall relinquish the surplus bands.

D. Security Analysis

1) Privacy and Auditability:

Theorem 1: (SECURITY AND PRIVACY) In the Pri-Share protocol of Fig. 3, each honest-but-curious participant learns information no more than the elements appear in the union set of all participants' private inputs, $S_1 \cup \ldots \cup S_N$, and the duplicated elements of all SAS server's private inputs $G_1 \cup \ldots \cup G_{N_1}$ as well as the priority levels of conflicted requests, with overwhelming probability.

Pri-Share is secure against any PPT (Probabilistic Polynomial Time) adversaries $\mathcal{A}$. The Theorem 1 is based on the assumption that the additive homomorphism of the threshold Paillier cryptosystem is semantically secure [15], [14], and the Pedersen commitment scheme is computationally binding [17], with overwhelming probability.

Theorem 2: (AUDITABILITY AND UNDENIABILITY) In Pri-Share protocol, once a commitment $Com(l)$ over a priority level l is constructed, the committer cannot deny the l they committed to.

The Theorem 2 is ensured by the binding property of the Pedersen commitment scheme [17]. The UNDENIABILITY also ensures AUDITABILITY of Pri-Share, in such a way that any participant cannot alter their input once enter the protocol. In the later stage, the FCC, as the auditing authority, has the capability to disclose the identity of the winner over a certain frequency band j by examining the actual operational status. This information can then be compared to the previously committed priority level $Com(l_j)$ of the winner in order to determine if a participant has provided consistent information.

2) Coalition Resistance: Pri-Share is inherent $(t, N)-$secure, i.e., the derivative computation and composition round (i.e., step 5) only requires contributions of $t+1$ out of N participants, whereas the correctness of the output can be guaranteed even if there are t dishonest participants colluding. Nevertheless, given the limited number of SAS administrators currently authorized by the FCC, Pri-Share operates under the assumption of a $(N, N)-$ model. However, as the spectrum-sharing market continues to expand rapidly, we anticipate that the number of industrial enterprises participating in SAS will increase. In this scenario, the utilization of a $(t, N)-$ solution can be employed to offer both fault tolerance and efficiency.

VI. EXPERIMENT AND EVALUATION

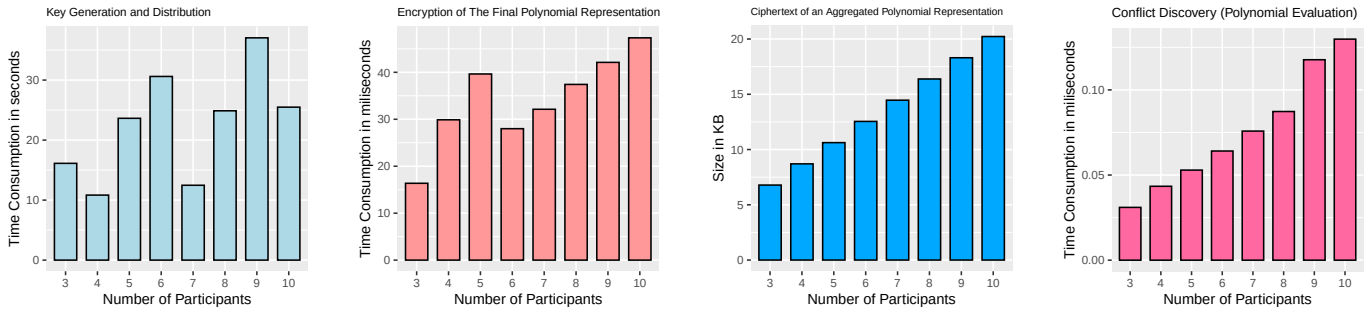
A. Experimental Environment

We implemented our proof-of-concept prototype on a standard PC (Intel Core i7-11700k, 3.6GHz, 8-core, 64-bit CPU with a Linux OS). We emulate each participant's service point as an independent Python application, while the peer-to-peer communications among participants are deployed using the TCP/IP socket.

The Paillier key generation, encryption, and decryption functions are implemented based on cryptographic library `charm` [18]; we also leverage the library `gmpy2` (C-based modules [19]) to accelerate multiple-precision arithmetic operations involved in cryptographic operations, e.g., computing exponentiation for homomorphic multiplication. The length of the security parameters (RSA modulus) p and q are both set as 1024 bits in the experiment.

B. Computation and Communication Overhead

1) Key Generation and Distribution: The *key generation and distribution* process is a one-time event that is independent of Pri-Share. During the system setup phase, a program emulates FCC will generate the pk and sk, and then the secret key shares are distributed among SAS participants. We run the key generation and distribution function 10 times based on the three-party cases ($N = 3$), and the time consumption spans from 6 to 24 s, with no apparent pattern. We also measured the computation workload when N ranges in $[3, 10]$, as shown in Fig. 4a, the time consumption of the key generation and distribution process still does not have an explicit distribution. This is mainly due to the inherent uncertainty and stochastic



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

Fig. 4: (a) Time Consumption of the *Key Generation and Distribution* process, (b) Time Consumption of final polynomial representation encryption, (c) Ciphertext size of the final polynomial representation, (d) Time Consumption of evaluating the final polynomial representation.

TABLE I: Time Consumption

Different Stages in Pri-Share	Time (ms)
Polynomials and Sets Mapping (1)	0.09
Encryption of the Polynomial Product (2)(3)(4)	10.59
Polynomial Aggregation and Group Decryption (5)(6ab)	310.25
Polynomial Evaluation (6cd)	0.02
Total	320.95

*Average of 10 runs.

TABLE II: Communication Overhead

Transmitted Messages Between Different Parties	Size (KB)
HE of the Private Input Set ($E(f_i), E(g_i)$)	2.944×2
HE of a Polynomial Product (λ_i, ω_i)	4.864×2
HE of the Final Product of All Polynomials (p_g, p_f)	$6.784 + 4.864$

nature of the prime number generation algorithm and the primality test procedure [20]. In the overall context of time consumption, the number of parties N is a negligible factor when compared to the additional time required for generating a large prime number.

2) Pri-Share: We implemented the full process of Pri-Share in three-party settings, the time consumption and communication overhead of different stages are listed in Table I and II, respectively. We also measured the computation overhead on different functions based on the different number of parties N, the results are shown in Fig. 4. It is worth noting that $N = 3$ (i.e., 2 SAS servers and 1 ESC/IIC in one region) is also consistent with the current number of active SAS vendors operated in the US, and currently, there are five SAS administrators have been designated by FCC in the US [21].

According to our measurement, the total time consumption of executing a full Pri-Share protocol is around $320.95\ ms$, which includes the communication overhead among participants through TCP/IP. Table II shows the size of the messages that are being transmitted between different participants. The size of the $E(f_1)$ and $E(g_1)$ are both $2.944\ KB$, while λ_2 and ω_2 are $4.864\ KB$. An ESC is set as the Nth participant during our implementation. It only holds a single polynomial g_3 that denotes incumbents' activity, and will calculate the λ_3 in $6.784\ KB$ and broadcast λ_3, ω_2 to others.

3) Impact of N and Optimization Methodologies: In the whole protocol, the main computation tasks of each participant include coefficients encryption and commitments generation (i.e., Step 1), as well as the multiplications between an encrypted polynomial and a polynomial in plaintext (i.e., Step 3). The Paillier group decryption (Step 6b) is the most expensive step in the protocol due to its fully distributed nature; each participant must perform a local decryption to generate a partial decryption share, and the final decryption result $Q_f, Q_{f'}$ and Q_g must be obtained by combining the remaining $N-1$ shares gathered from the remaining parties. As there is no dependency in computing $Q_f, Q_{f'}$ and Q_g at Step 6, we utilize *multi-threading* programming to parallelize the three decryption tasks. Each participant executes the partial decryption simultaneously, which is essentially conducted in parallel from a global perspective, such that overall latency does not increase significantly even with an increased number of parties N.

Except for group decryption, a larger N will also affect the final size of the polynomial product, because a multiplication between a polynomial of degree k and a polynomial of degree k' requires $(k+1)\times(k'+1)$ multiplicative operations among their corresponding coefficients, and the degree of the polynomials p_f, p_g increases linearly with N. As shown in Fig. 4c, the size of the ciphertext of the final polynomial representation is increased consistently with N. The Paillier encryption time is normally affected by the input data size, yet our experiment result (Fig. 4b) shows that it does not have an obvious trend with respect to N, this is mainly due to the randomly selected x used during each encryption operation. Given that N in the current SAS system is often in single-digit, a larger x will outweigh the influence of N.

Furthermore, we implemented the polynomial evaluation function utilizing *Horner's rule*, which can greatly accelerate the evaluation process of Step 6. Consequently, the polynomial evaluations (i.e., finding the conflicted or prohibited bands) can be performed efficiently even when dealing with a larger N according to our measurement, as shown in Fig. 4d.

VII. Discussion

A. Zone Management Systems

In this work, we focus on the privacy of the SAS as it is a well-established system working on the CBRS band. Recently, the FCC has proposed more shared bands including the 3.1 GHz-3.55 GHz, 12 GHz-12.7 GHz, and 42 GHz-42.5 GHz bands to provide more spectrum access opportunities for commercial users. To govern their operations, the zone management system (ZMS) is proposed to jointly manage all spectrum users in shared bands as an overall system, which requires more complex allocation algorithms. Considering ZMS is likely to follow a model similar to SAS, the techniques proposed in Pri-Share for SAS can be readily adjusted or applied to ZMS.

B. System Overhead

According to our evaluation, the time consumption of Pri-Share remains well below the standard ESC/IIC notification window of 300s, even when accommodating a higher number of participants, demonstrating that it is a feasible and practical solution for the current SAS systems.

Besides, some potential optimization techniques can be applied to our system. For example, previous literature [22], [23], [24] have proposed alternative optimized variants of the Paillier scheme in order to accelerate the Paillier encryption and decryption process. Such variants can be utilized to modify the distributed Paillier scheme implemented in our design, thereby further reducing the overall overhead of Pri-Share. Crucially, since the implementation of SAS platforms typically does not entail hardware constraints, the efficiency of Pri-Share can be enhanced beyond our proof-of-concept implementation. This enhancement is achievable through the use of more efficient programming languages and more powerful hardware.

VIII. Conclusions

Pri-Share is a practical privacy-preserving spectrum sharing scheme that is designed based on a decentralized SAS model and adheres to the primary allocation rules. It enables private computation for band allocation and conflict resolution among SAS servers and ESC/IIC within a single region, while keeping the activity of incumbents and private choices made by spectrum clients confidential. Besides, Pri-Share also provides misbehavior detection on private inputs, allowing for further auditing by trusted authorities.

Acknowledgment

This work was supported in part by the US National Science Foundation under grants 2331936, 2332675, 2312447, 2247560, 2154929, and 1916902, and the Virginia Commonwealth Cyber Initiative (CCI).

References

[1] NTIA, *Feasibility of Commercial Wireless Services Sharing with Federal Operations in the 3100-3550 MHz Band.*, July 2020.

[2] FCC, *FCC MOVES FORWARD ON 12 GHz PROCEEDING-New Rules Preserve Portion of the Band for Advanced Satellite Broadband While Considering How to Promote Advanced Terrestrial Broadband in the Rest of the Band*, May 2023.

[3] FCC, *FCC EXPLORES SPECTRUM SHARING APPROACHES FOR THE 42 GHZ SPECTRUM BAND - Innovations in this Spectrum Could Inform Future Sharing Approaches*, June 2023.

[4] S. Shi, Y. Xiao, W. Lou, C. Wang, X. Li, Y. T. Hou, and J. H. Reed, "Challenges and new directions in securing spectrum access systems," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6498–6518, 2021.

[5] 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)*, pp. 236–247, IEEE, 2014.

[6] 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*, pp. 1–9, IEEE, 2016.

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

[8] H. Li, Y. Dou, C. Lu, D. Zabransky, Y. Yang, and J.-M. J. Park, "Preserving the incumbent users' location privacy in the 3.5 GHz band," in *2018 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 1–10, IEEE, 2018.

[9] 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, 2016.

[10] H. Li, Y. Yang, Y. Dou, J.-M. J. Park, and K. Ren, "PeDSS: Privacy enhanced and database-driven dynamic spectrum sharing," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 1477–1485, IEEE, 2019.

[11] M. Grissa, A. A. Yavuz, and B. Hamdaoui, "TrustSAS: A trustworthy spectrum access system for the 3.5 GHz CBRS band," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 1495–1503, IEEE, 2019.

[12] Y. Xiao, S. Shi, W. Lou, C. Wang, X. Li, N. Zhang, Y. T. Hou, and J. H. Reed, "BD-SAS: Enabling Dynamic Spectrum Sharing in Low-trust Environment," *IEEE Transactions on Cognitive Communications and Networking*, 2023.

[13] WinnForum, *CBRS Communications Security Technical Specification.Document WINNF-TS-0065*, November 2020.

[14] V. Shoup, "Practical threshold signatures," in *Advances in Cryptology—EUROCRYPT 2000: International Conference on the Theory and Application of Cryptographic Techniques Bruges, Belgium, May 14–18, 2000 Proceedings 19*, pp. 207–220, Springer, 2000.

[15] P.-A. Fouque, G. Poupard, and J. Stern, "Sharing decryption in the context of voting or lotteries," in *Financial Cryptography: 4th International Conference, FC 2000 Anguilla, British West Indies, February 20–24, 2000 Proceedings 4*, pp. 90–104, Springer, 2001.

[16] L. Kissner and D. Song, "Privacy-preserving set operations," in *Annual International Cryptology Conference*, pp. 241–257, Springer, 2005.

[17] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Annual international cryptology conference*, pp. 129–140, Springer, 1991.

[18] "Charm-crypto 0.50." https://jhuisi.github.io/charm/install_source.html.

[19] "Python gmpy2." https://gmpy2.readthedocs.io/en/latest/index.html, 2023.

[20] J. M. Pollard, "Theorems on factorization and primality testing," in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 76, pp. 521–528, Cambridge University Press, 1974.

[21] S. Partners, "CBRS SAS players: Who are they and what do they do?." https://gmpy2.readthedocs.io/en/latest/index.html, 2023.

[22] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International conference on the theory and applications of cryptographic techniques*, pp. 223–238, Springer, 1999.

[23] H. Ma, S. Han, and H. Lei, "Optimized paillier's cryptosystem with fast encryption and decryption," in *Annual Computer Security Applications Conference*, pp. 106–118, 2021.

[24] M. J. Jurik, *Extensions to the paillier cryptosystem with applications to cryptological protocols*. Citeseer, 2003.