Privacy-Preserving and Post-Quantum Counter Denial of Service Framework for Wireless Networks

Saleh Darzi
University of South Florida, Tampa, USA
salehdarzi@usf.edu

Attila Altay Yavuz
University of South Florida, Tampa, USA
attilaayavuz@usf.edu

MILCOM 2024 - 2024 IEEE Military Communications Conference (MILCOM) | 979-8-3503-7423-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/MILCOM61039.2024.10773665

***Abstract*—As network services progress and mobile and IoT environments expand, numerous security concerns have surfaced for spectrum access systems (SASs). The omnipresent risk of Denial-of-Service (DoS) attacks and raising concerns about user privacy (e.g., location privacy, anonymity) are among such cyber threats. These security and privacy risks increase due to the threat of quantum computers that can compromise long-term security by circumventing conventional cryptosystems and increasing the cost of countermeasures. While some defense mechanisms exist against these threats in isolation, there is a significant gap in the state of the art on a holistic solution against DoS attacks with privacy and anonymity for spectrum management systems, especially when post-quantum (PQ) security is in mind. In this paper, we propose a new cybersecurity framework `PACDoSQ`, which is the first to offer location privacy and anonymity for spectrum management with counter DoS and PQ security simultaneously. Our solution introduces the private spectrum bastion concept to exploit existing architectural features of SASs and then synergizes them with multi-server private information retrieval and PQ-secure Tor to guarantee a location-private and anonymous acquisition of spectrum information together with hash-based client-server puzzles for counter DoS. We prove that `PACDoSQ` achieves its security objectives, and show its feasibility via a comprehensive performance evaluation.**

***Index Terms*—Spectrum Management, Post-Quantum Security, Counter DoS, Privacy and Anonymity**

I. Introduction

With the progression of wireless network services such as 5G/6G, coupled with the rapid expansion of mobile and IoT applications, the significance and frequency of threats to such services, specifically spectrum access systems (SAS) are escalating [1]. Among these threats, the omnipresence of Denial of Service (DoS) attacks is becoming increasingly sophisticated and executable, due to the availability of open-source software, enhanced processing capabilities, and the proliferation of inexpensive devices. DoS attacks are particularly applicable to emerging wireless networked systems because of their inherent broadcast nature, spectrum access, and geolocation database requisites [2]. Wireless spectrum access, despite its merits, also brings profound privacy concerns for its users. More specifically, the continuous reporting of spectrum and location data to geo-location database servers raises numerous privacy concerns [3]. Finally, the emergence of quantum computers poses a significant risk to the long-term security and privacy preservation of these next-generation networks, challenging existing classical security countermeasures [4]. Efforts are underway to address security issues including counter-DoS, privacy, and PQ threats in SAS. However, existing solutions work in isolation, and do not comprehensively and effectively tackle these issues simultaneously. We outline some of the most relevant efforts to our work below.

A. Related Work

Counter-DoS and Spectrum Management for NextG Networks: The widespread growth of mobile and IoT devices has led to a shortage of spectrum resources. Cognitive Radio Networks (CRN) offer secondary users (SUs) the ability to opportunistically access unoccupied licensed channels, presenting a prospective solution for spectrum management. While spectrum management serves as a critical wireless resource allocation tool, it faces several security threats, including DoS attacks, due to their broadcast nature, database-driven architecture, and the potential malicious behavior of SUs [5], [6]. Adversaries may target system availability through DoS aiming to exhaust server resources that handle servicing requests. Numerous research endeavors exist to mitigate DoS attacks through intrusion detection systems (IDSs) and mechanisms encompassing network-based solutions, cryptographic techniques, and game theory-based approaches.

Recent progress in machine learning has propelled AI-based IDSs into the spotlight, demonstrating the ability to accurately identify abnormal behavior, with success rates surpassing 95% in some cases [2]. However, despite their merits, these methods may require knowledge and access to broad (some cases private) network topologies, user-sensitive network traffic, and continuous training on large-scale data [7], [8]. Moreover, they may be vulnerable to some AI-based loopholes exploited by attackers with substantial costs to the underlying system [9], [10]. Therefore, it is ideal that they are complemented with counter-DoS techniques that do not rely on such features and can offer additional provable security guarantees.

Client Puzzle Protocols (CPP) permit a client to access server resources only upon presenting a valid token generated by solving a puzzle like Proof of Work (PoW) [11]. CPPs significantly increase the cost of the DoS attack (e.g., computational, memory) depending on the type of puzzles (e.g., timing, AI-based), thereby substantially mitigating their impact. CPPs can offer an ideal complement to AI-based counter-DoS, but they must achieve various properties such as cost asymmetry, efficiency, statelessness, memorylessness, unforgeability, and non-parallelization [12]. Given their requisite features and the need for scalability for IoT networks, alleviating the burden of puzzle management from the users and servers is crucial. One such effort is outsourcing puzzle generation and distribution to a trusted entity called "Bastions" [13]. However, these approaches often presume the existence of Bastions in applications, presenting just abstract concepts without clarity on which entity realistically assumes the Bastion role, thus missing proper architectural incentives. To benefit from outsourced CCPs, bastions' trust level and architectural duty must be well-justified and integrated into the target application.

Privacy and Anonymity in SAS under DoS Attacks: The Federal Communications Commission (FCC) has instructed the utilization of centralized SAS, comprising multiple geolocation spectrum databases to foster dynamic spectrum resource access [14]. This facilitates spectrum sharing between governmental entities and commercial operators, with primary and secondary users. FCC mandates that users provide sensitive information, including precise location coordinates (longitude and latitude), desired spectrum channel, usage data, and transmission details, to access spectrum availability [3]. This not only gives rise to privacy concerns regarding users' confidential data, and identity but also facilitates the tracing and potential exposure of location privacy (e.g., revealing behavioral patterns, lifestyle choices, etc.) [15]. Moreover, the absence of authentication during private spectrum data access, coupled with the reliance on many counter-DoS solutions for authentication, underscores the critical need to prioritize user anonymity and privacy. This gap in spectrum management services calls for solutions that address anonymity, location privacy, and DoS mitigation simultaneously.

Counter-DoS and Privacy for Wireless Networks in the Post-Quantum Era: The emergence of quantum computers presents a substantial security risk to NextGen networks, potentially compromising foundational security protocols (e.g., TLS) and undermining critical aspects of SAS such as DoS protection and privacy safeguards (e.g., [1]). Furthermore, conventional cryptographic methods used in privacy-preserving techniques, anonymity networks, and counter-DoS solutions rely on cryptographic problems vulnerable to quantum computers. Hence, Post-Quantum Cryptography (PQC) becomes imperative to furnish a robust long-term security solution [4].

B. Our Contribution

We designed a novel framework that culminates various cryptographic techniques to address the complex array of privacy and security challenges stemming from SAS under DoS and quantum computer attacks. Our scheme presents a "Privacy and Anonymity preserving Counter-DoS in the post-Quantum era" (`PACDoSQ`) for spectrum management in next-generation networks. We summarize some of the desirable properties of `PACDoSQ` as follows:

- *Enabling Outsourced Counter-DoS Services with SAS Architecture Compliance:* We devised innovative counter-DoS services formed on CPP architecture featuring hash-based puzzles, where puzzle generation and distribution are delegated to database-driven entities, termed "Private Spectrum Bastions" (PSBs) in our work. Integrating Bastion services within SAS geo-location databases offers several advantages, as PSBs can supply quantum-safe puzzles alongside spectrum availability, maintaining architectural feasibility, and enhanced efficiency. Our PSB approach also paves the way for tackling the location privacy problems of spectrum management and outsourced puzzle services with enhanced robustness as described below.
- *Fault-Tolerant Location Privacy and Anonymity:* The database-driven SAS architecture, by FFC requirements, brings various privacy issues as discussed in Section I-A [1]. Therefore, despite our opportunistic integration of outsourced CCPs with existing SAS architectures, which mitigates DoS attacks, it still requires clients to obtain puzzles and spectrum data from PSBs. We address the privacy concerns as follows: *(i)* We harness distributed Private Information Retrieval (PIR) protocols [16] that synergize with multi-server PSB architecture [1]. The clients fetch spectrum information (by adhering to FFC regulations) and CCPs privately. Moreover, our choice of PIR protocol permits resiliency against network failures and some subsets of non-responding PSB servers. *(ii)* We ensure clients connect to PSBs and perform private retrieval operations through a post-quantum secure version of the Tor network [17], thereby offering anonymous access.
- *Post-Quantum Security:* `PACDoSQ` offers all the above desirable security and privacy features with a post-quantum guarantee thanks to the reliance on NIST-PQC standards in Tor and information-theoretically secure PIR operations.

II. Preliminaries and Building Blocks

In this section, we outline the notations, cryptographic primitives, and tools employed in our proposed framework.

Notations: $|x|$ and $\{0,1\}^k$ signify the bit length of a variable and k-bit binary value, respectively. $\mathbb{F}$, $GF(2)$, and $\mathbb{Z}$, denote a finite field, Galois Field with modulo 2, and a set of integers, respectively. $\{x_i\}_{i=1}^{\ell}$ and $\xleftarrow{\$} \mathcal{S}$ denote $(x_1, x_2, ..., x_\ell)$ and random selection from the set $\mathcal{S}$, respectively. The function $h(.)$ denotes a cryptographically secure hash function. sk and pk are secret and public keys, respectively.

Private Information Retrieval: The PIR construction enables a client to retrieve a block of information from a database without revealing the privacy of the retrieved item to the database server(s). We will focus on multi-server PIR since our system model includes multiple spectrum databases. We opt for the fault-tolerant IT-PIR [16] that offers ν-byzantine robustness, ensuring the reconstruction of the target block even if ν servers provide incorrect responses.

PQ-Secure Primitives: We use NIST PQC standardized lattice-based schemes for KEM and signature, *Kyber* [18] and *Dilithium* [19], respectively. The *Kyber* KEM is formed on the Module-LWE problem and is comprised of three algorithms ($\texttt{Kyber.KeyGen}, \texttt{Kyber.Encap}, \texttt{Kyber.Decap}$). The *Dilithium* signature is also formed on Module-LWE and is comprised of three algorithms: $(sk, pk) \leftarrow \texttt{Dilith.KeyGen}(1^\lambda)$; $\sigma \leftarrow \texttt{Dilith.Sign}(sk, m)$; and $\{0,1\} \leftarrow \texttt{Dilith.Verify}(pk, m, \sigma)$.

Hash-based Puzzles: We use hash-puzzles [20] that are comprised of three functions ($\texttt{Gen}, \texttt{PoW}, \texttt{Verify}$):

- $\Pi \leftarrow \mathit{Puzzle.Gen}(1^\lambda, \kappa)$: Given the security parameter λ and the difficulty level κ, it selects a random nonce $N \leftarrow \{0,1\}^\kappa$ and produces hash-based puzzles $\Pi = (N, \kappa)$.

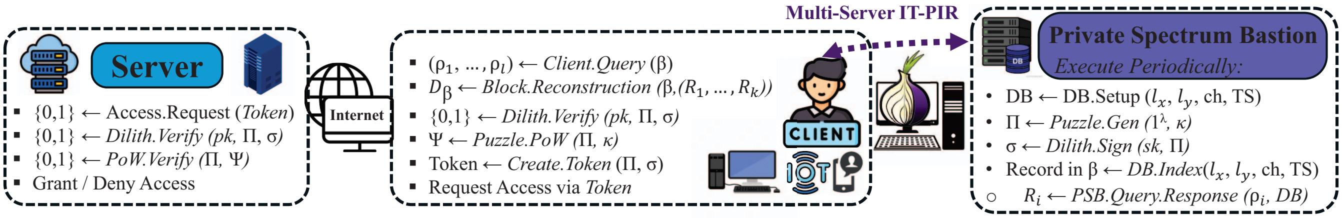
Multi-Server IT-PIR
Server
$\{0,1\} \leftarrow$ Access.Request (*Token*)
$\{0,1\} \leftarrow$ *Dilith.Verify (pk, Π, σ)*
$\{0,1\} \leftarrow$ *PoW.Verify* (Π, Ψ)
Grant / Deny Access
Internet
$(\rho_1, ..., \rho_l) \leftarrow$ *Client.Query* (β)
$D_\beta \leftarrow$ *Block.Reconstruction* $(\beta,(R_1, ..., R_k))$
$\{0,1\} \leftarrow$ *Dilith.Verify (pk, Π, σ)*
$\Psi \leftarrow$ *Puzzle.PoW* (Π, κ)
Token ← *Create.Token* (Π, σ)
Request Access via *Token*
CLIENT
IoT
Private Spectrum Bastion
Execute Periodically:
DB ← DB.Setup $(l_x, l_y$, ch, TS)
Π ← *Puzzle.Gen* $(1^\lambda, \kappa)$
σ ← *Dilith.Sign (sk, Π)*
Record in β ← *DB.Index*$(l_x, l_y$, ch, TS)
$R_i \leftarrow$ *PSB.Query.Response* (ρ_i, DB)
DB

Fig. 1: A high-level representation of the proposed architecture and workflow.

Algorithm 1 PACDoSQ Scheme

```
    Private Spectrum Bastions:
 1: $DB \leftarrow \texttt{DB.Setup}(l_x, l_y, ch, TS)$
 2: for $(l_x, l_y) \in grid$ do
 3:     for $TS \in Timeframe$ do
 4:         for $\theta \in \{1, 2, ..., \max\}$ do
 5:             $\Pi_\theta \leftarrow \texttt{Puzzle.Gen}(1^\lambda, \kappa)$
 6:             $\sigma_\Pi \leftarrow \texttt{Dilith.Sign}(sk, \Pi_\theta)$
 7:             Given $\beta \leftarrow$ DB-Index$(l_x, l_y, ch, TS)$.
 8:             Record $\Pi_\theta$ and $\sigma_{\Pi_\theta}$ in DB within index $\beta$.
    $\underline{R_i \leftarrow \texttt{PSB.Query.Response}(\rho_i, DB)}$:
 9: Upon receiving request $\rho_i$, $\texttt{PSB}_i$ computes: $R_i \leftarrow \rho_i \cdot DB$
10: return $R_i$, and respond to the client via PQ-Tor

    Clients:
    $\underline{(\rho_1, \rho_2, ..., \rho_\ell) \leftarrow \texttt{Client.Query}(\beta)}$:
 1: Obtain index $\beta \leftarrow$ DB-Index$(l_x, l_y, ch, TS)$
 2: Set $e_\beta \leftarrow \overrightarrow{1_{\beta_r}} \in Z_r$
 3: Choose $\ell$ distinct $\alpha_1, \alpha_2, ..., \alpha_\ell \in \mathbb{F}^*$
 4: Choose $r$ random degree-$t$ polynomials $f_1, f_2, ..., f_r \overset{\$}{\leftarrow} \mathbb{F}[x]$ s.t.
    $f_j(0) = e_\beta[j]$, for all $j \in [1, 2, ..., \ell]$
 5: $\rho_i \leftarrow \langle f_1(\alpha_j), f_2(\alpha_j), ..., f_r(\alpha_j)\rangle$ for all $j \in \{1, 2, ..., \ell\}$
 6: Query $\text{PSB}_i$ via transmitting $\rho_i$ over PQ-Tor for $i = 1, 2, ..., \ell$
    $\underline{D_{\beta c} \leftarrow \texttt{Block.Reconstruction}(\beta, (R_1, R_2, ..., R_k))}$:
 7: Receiving PIR responses $R_i$ from PSBs for $i = 1, 2, ..., k$
 8: if $k > t$ then
 9: for $c$ from 1 to $s$ do
10:     $R_{ic} \leftarrow R_i[c]$ for all $i \in [1, 2, ..., k]$
11:     $S_c \leftarrow \langle R_{1c}, R_{2c}, ..., R_{kc}\rangle$
12:     $D_{\beta c} \leftarrow$ EASYRECOVER$(t, \omega, [\alpha_1, \alpha_2, ..., \alpha_k], S_c)$
13:     if Recovery fails and $\nu < k - \lfloor\sqrt{kt}\rfloor$ then
14:         $S_c \leftarrow \langle R_{1c}, R_{2c}, ..., R_{kc}\rangle$
15:         return $D_{\beta c} \leftarrow$ HARDRECOVER$(t, \omega, [\alpha_1, \alpha_2, ..., \alpha_k], S_c)$
    $\underline{Token \leftarrow \texttt{Create.Token}(\Pi, \sigma_\Pi)}$: Given $\Pi$ in $D_\beta$, do
16: $\{0,1\} \leftarrow \texttt{Dilith.Verify}(pk, \Pi, \sigma_\Pi)$
17: $\Psi \leftarrow \texttt{Puzzle.PoW}(\Pi, \kappa)$
18: return Token $\leftarrow (\Pi, \sigma_\Pi, \Psi)$

    Servers: For requests, server does:
 1: $\{0,1\} \leftarrow \texttt{Dilith.Verify}(pk, \sigma_\Pi)$
 2: $\{0,1\} \leftarrow \texttt{PoW.Verify}(\Pi, \Psi)$
 3: if above holds, return 1, grant access, and record the Token
 4: otherwise, return 0 and deny access.
```

- $\Psi \leftarrow \texttt{Puzzle.PoW}(\Pi, \kappa)$: Given a puzzle Π, it brute forces a nonce $\Psi = N_x$ to obtain a hash value with κ-bit leading zeros, $0_1 0_2 0_3 ... 0_\kappa Y \leftarrow h(\Pi, N_x)$, where $Y \in \{0,1\}^{|h|-\kappa}$.
- $\{0,1\} \leftarrow \texttt{PoW.Verify}(\Pi, \Psi)$: The verifier checks if the first κ bits of the hash value of $h(\Pi, N_x)$ are zero.

III. The Proposed System & Framework: PACDoSQ

This section delineates our system setup and framework.

A. PACDoSQ Architecture and Initial Setup

Our system model has three main entities: 1) *Private Spectrum Bastions (PSBs):* consist of multiple geo-location spectrum databases [1], [15] that provide spectrum availability information. They maintain synchronicity and consistency under FCC guidelines. 2) *Clients:* are secondary users equipped with mobile devices (e.g., laptops). They connect to the servers for network services by obtaining spectrum availability from PSBs. 3) *Servers:* are various network servicing platforms (e.g., web/cloud servers), to which clients seek to connect.

We outline the initial setup of PSB and *PQ-Tor* below.

Database (DB) Structure & Setup: The PSBs synchronize their DB by incorporating various parameters like location coordinates (l_x, l_y), frequency channel number (ch), and spectrum data. DB is conceptualized (and simplified) as a matrix with dimensions $r \times s$, where each row represents one data block comprising b bits. Each block consists of s words, each with a size of w bits formatted as $GF(2^w)$ (as in [2], [5]). PSBs maintain other relevant information stipulated by the FCC (as in [15]) like row index of coordinates with proper subroutines, for brevity herein referred as *DB-Index(.)*.

***PQ-Tor* Configuration:** Our *PQ-Tor* variant has the following alterations over conventional Tor: (i) *RSA* signature is replaced with *Dilithium* signing in the consensus part. (ii) *RSA* KEM is substituted with *Kyber* KEM in circuit creation. (iii) *AES-128* is replaced with *AES-256* to double symmetric key size against Grover's algorithm [21].

B. PACDoSQ Framework

We illustrate the flow of PACDoSQ framework in Fig. 1, provide its algorithmic description in Algorithm 1, and further elaborate on its steps as follows:

*1) **PSBs - Puzzle Management and Private Spectrum Service:*** *(i)* PSBs setup DB by generating spectrum management context (e.g., coordinates, channels), puzzles, and their PQ signatures as in Step 1-8. Within defined segments of the grid marked by specific coordinates for multiple time frames, they generate hash-based puzzles and sign them according to predetermined indices derived from $((l_x, l_y), ch, TS)$. The puzzles and their *Dilithium* signatures are updated periodically according to the puzzle difficulty/validity interval (e.g., every hour). The quantity of puzzles generated depends on factors such as the number of servers and their maximum capacity (max). *(ii)* PSBs handle the spectrum query first via the fault-tolerant multi-server PIR [16] that permits an information-theoretically private retrieval of coordinate availability, puzzle, and their signatures (Step 9). The PIR response is sent to the client via *PQ-Tor* to ensure anonymity (step 10).

*2) **Client's Private Availability Information and Quantum-Safe Puzzle Retrieval**:* To comply with FCC regulations and participate in the counter-DoS mechanism for accessing a networking services server, the clients retrieve puzzles and spectrum information from the PSBs. Clients use their coordinates, frequency channel, and timestamp to determine the target index β within the PSB's DB (Step 1). Subsequently, the client constructs a PIR request by selecting a basis vector $\overrightarrow{1_{\beta_r}}$, where all elements are zero except for index β, which is set to one (Step 2). Furthermore, considering ℓ PSBs and utilizing Shamir's secret sharing technique, the client selects ℓ random elements from $\mathbb{F}^*$ (Step 3), generates r random polynomials with a degree of t satisfying $f_j(0) = e_\beta[j]$ (Step 4), and creates ℓ PIR requests ρ (Step 5). Finally, the client dispatches the PIR requests to each PSB's DB_i via *PQ-Tor* (Step 6).

Steps 7-15 involve the client's query recovery phase. Assuming that k out of ℓ PSB servers respond to the client, the client can reconstruct the block using the *EASYRECOVER* subroutine as described in [16], which relies on the Lagrange interpolation technique. If a sync/transmission error occurs or an incorrect block is returned by $\nu < k$ servers (e.g., Byzantine (compromised) server), the client can use *HARDRECOVER* algorithm [16] based on error-correction codes to handle the error. By reconstructing the block item with one of these recovery algorithms, the client retrieves the puzzle, whose validity can be confirmed by verifying PSB's signature.

*3) **PoW & Token Creation**:* The online phase of the framework begins at this stage, where the client performs the PoW and generates the *Token*. Given the hash-based puzzle (Π) and the target network service ID (ID_S), the client must conduct a brute-force search through a nonce (N_C) to discover a hash value $h(ID_S, TS, N_B, N_C)$ with κ-bit leading zeroes (Step 17-18). Then, upon identifying a solution, it generates the *Token*, which comprises the PSBs' and client's nonces along with the TS and ID_S, and transmits it to the server.

*4) **Access Requests**:* The client submits a request to the server with *Token* for a given time interval. The server first verifies the puzzle's validity by checking the PSB's signature (Step 1), followed by efficiently verifying the *Token* using a hash operation (Step 2). Only if the puzzle solution is valid and authentic, the access is granted.

IV. Security Analysis

Threat Model and Security Objectives: Our threat model captures a vast range of attacks at the intersection of counter-DoS, privacy, anonymity, and basic security services, all under quantum computing threat: *(i)* Clients may launch DoS attacks on the servers. To mitigate such attacks, we consider the counter DoS threat model in outsourced puzzle settings, wherein PSBs carry over the Bastion role for puzzle management. *(ii)* Client's location privacy and identity information are under threat due to the FCC's requirement of sharing coordinate and device specs with spectrum management databases. In our model, PSBs carry out this duty along with puzzle management. Hence, we consider that PSBs are curious about the location and identity information of the clients. *(iii)* Some (but small set of) PSBs might be compromised and therefore may act as Byzantine servers (do not respond or provide incorrect input). *(iv)* The attacker is quantum computing capable and can use it to launch attacks considered in *(v)* as well as to threaten basic security services such as confidentiality, authentication, and integrity (which are usually achieved through essential services like TLS). Given the threat model, PACDoSQ aims to achieve the following security objectives:

- *Client Privacy and Anonymity:* Clients' location privacy (i.e., coordinates), device specs, and identity remain confidential and anonymous during spectrum availability and puzzle retrieval from the PSBs and external attackers.
- *Resilience to Partial Failure and Byzantine Behavior:* The client can retrieve and reconstruct the intended block item (including spectrum and puzzle data) even if some subset of the PSBs act non-responsive or malicious.
- *DoS Mitigation:* A measurable and provable counter DoS measures are employed.
- *PQ-security:* All the above objectives are achieved in the presence of quantum computing capable adversaries.

Security Analysis: We give a series of security proofs capturing the threat model as follows:

Lemma 1. *PACDoSQ ensures t-private k-out-of-ℓ information-theoretically secure location privacy and computationally secure anonymity via onion routing.*

Proof. By utilizing (ℓ, t)-Shamir secret sharing, with the assumption of k honest responses from ℓ PSBs where $k > t$, the target index β, along with the client's private information, including location and transmission details, remains confidential during the block retrieval process, even in the event of collusion among t PSBs with $0 \leq t \leq \ell - 1$. The deployment of onion routing with a minimum of three intermediate nodes, each possessing knowledge solely of its predecessor and successor, alongside communication through a circuit with layers of symmetric encryption using *AES-256* keys derived via a *Module-LWE*-based KEM scheme, ensures the anonymity and untraceability of the client's identity and activities against both PSBs and eavesdropping adversaries. □

Corollary 1. *PACDoSQ attains ν-Byzantine-Robustness with $\nu < k - \lfloor\sqrt{kt}\rfloor$.*

Proof. PACDoSQ offers block reconstruction from client-received query responses (e.g., communication failures, malicious drop) by employing Guruswami-Sudan list decoding algorithm capable of correcting $\nu < k - \lfloor\sqrt{kt}\rfloor$ errors and (ℓ, t)-Shamir secret sharing with k responding PSBs ($k > t$). □

Corollary 2. *PACDoSQ offers enhanced counter-DoS for the servers via client-server puzzles.*

Proof. The server only accepts puzzle solutions with PSB's signature, eliminating the possibility of puzzle forgery. Since PoW requires $O(2^n)$ trial (for classical settings), the adversary needs an average of $O(2^\kappa)$ hash operations to acquire a valid *token* for server, where a puzzle is only valid for a designed amount of time depending on the difficult level κ. □

Lemma 2. *PACDoSQ achieves the objectives in Lemma 1 and Corollary 1-2 with PQ-security.*

Proof. (i) The location privacy guarantees and robustness features in Lemma 1 and Corollary 1, respectively, are information-theoretically secure, and therefore remain unaffected by the adversary's computational power, including quantum computers [16]. *(ii)* The onion routing anonymity in Lemma 1 relies on 128-bit PQ security of the *AES-256* [22] given Grover's algorithm and the hardness of the *Module-LWE* problems, which can be closely reduced from the worst-case *Module-SIVP* problem in the random oracle model [18]. *(iii)* The end-to-end security and PQ-TLS security of PQ-Tor and authentication of puzzles also achieve the same level of PQ-security via NIST PQC framework [4]. *(iv)* The hash-based puzzle in Corollary 2 offers $O(2^{\kappa/2})$ level of PQ security due to Grover's probabilistic algorithm, and by adjusting the time validity of the PoW accordingly, the hash-based puzzles offer robust PQ counter-DoS mitigation for `PACDoSQ`. □

V. Performance Evaluation

We present our evaluation metrics and experimental results.

A. Metrics, Selection Rationale, and Configurations

Evaluation Metrics and Rationale: We consider the computational, communication, and storage overhead of `PACDoSQ` for multi-server PIR, puzzle generation, PoW, token verification, and overhead of PQC, including *PQ-Tor* components. We also investigate scalability aspects such as end-to-end delay perceived by the client for an increased number of users, networking conditions, and PSB configurations. The configuration of PSB servers specifies the privacy levels achieved during block retrieval; for instance, $(3, 2)$ indicates that privacy is maintained if any 2 out of 3 PSBs collude. To the best of our knowledge, `PACDoSQ` is the first to offer location privacy, anonymity, and resiliency for puzzle-based counter DoS with PQ-security. Therefore, a vis-a-vis performance comparison with counterparts is not feasible. Instead, we focus on providing a detailed performance evaluation for given metrics to assess the potential feasibility of our framework. We detail our performance evaluation as follows.

Hardware, Software Libraries, and Parameters: We used a desktop equipped with an 11^{th} Gen Intel Core *i9-11900K*@3.50 GHz, 64.0 GiB RAM, a $1TB$ SSD, and Ubuntu 22.04.4 LTS. We employed varying number of Virtual Machines (VMs) with Ubuntu to simulate multiple PSB/PIR/PQ-ToR interactions. We used *percy++* library[1] for the multi-server PIR, the *Open Quantum-Safe* library[2] for PQC primitives, and the *OpenSSL*[3] for hash. The PSB used *SQLite*[4] and *Python3*. We used *AES-256*, *Kyber* for the KEM part, and *Dilithium* for the signature part of *PQ-Tor*. The hash-based puzzles are formed on *SHA-256*. We rely on NIST-PQC level I security for *Kyber* [18] and *Dilithium* [19].

Data and Format Selection: The database structure is a matrix with varying row sizes (e.g., 2^{10}, 2^{12}, 2^{14}, 2^{17}), where each row represents a single block of data. Utilizing publicly available raw data from the FCC[5], we estimated that each block in the database would contain approximately 560 bytes of information, excluding puzzles and signatures. Within a designated grid segment defined by coordinates l_x and l_y, we populated databases with synthetic data representing spectrum information and signed hash-based puzzles stored in PSBs, synchronized as mandated by the FCC [1].

B. Experimental Results

Computational Costs: We outline our analysis in Table I and elaborate it as follows: *(i)* The *Dilithium* signature with the puzzle entails key generation, signing, and verification of $29\mu s$, $84\mu s$, and $30\mu s$, respectively. Puzzle generation and verification each require approximately one hash, while solving (PoW) demands brute force corresponding to the difficulty levels denoted by κ. The difficulty level is $\kappa/2$ for quantum attacks with Grover's algorithm. *(ii)* With $t_\oplus$ representing the time for one *XOR*, analytical costs are $(n/w) \cdot t_\oplus$ for PIR computations on the client side and $\ell \cdot (\ell - 1) \cdot r \cdot t_\oplus + 3\ell \cdot (\ell + 1) \cdot t_\oplus$ on the PSB side. The empirical costs, as detailed in Table I, show that the expenses for PIR increase linearly with the size of the database. *(iii) PQ-Tor*'s costs are circuit build and applying encryption layers dominated by the three *Kyber* and *AES* operations. Notably, *Kyber* key generation, encapsulation, and decapsulation each take $10\mu s$, $13.4\mu s$, and $9\mu s$, respectively, while *AES-256* only costs 7 μs for key generation and 8 μs for encryption.

Entity	Operations	Parameter			
PSB	*Puzzle Generation& Sign*	**|DB|**			
		$\mathbf{2^{10}}$	$\mathbf{2^{12}}$	$\mathbf{2^{14}}$	$\mathbf{2^{17}}$
		31 ms	310 ms	3.1 s	31 s
	Query Response	2.3 ms	5.4 ms	17.3 ms	109.9 ms
Client	*Query & Reconstruction*	0.9 ms	2.1 ms	5.7 ms	12.5 ms
	Puzzle Signature Verify	30 μs			
	PQ-Tor Computations	255.6 μs			
	Proof of Wok	$\boldsymbol{\kappa}$: 14	$\boldsymbol{\kappa}$: 18	$\boldsymbol{\kappa}$: 20	$\boldsymbol{\kappa}$: 23
		5.73 ms	91.7 ms	367 ms	2.93 s
Server	*Puzzle Signature Verify*	30 μs			
	Token Verification	0.35 μs			

One client, one PSB, and a server in a (3,2) configuration setting, fixed block size of 2.93 KB, and varying database entries ($|DB|$).

TABLE I: Computational Costs of `PACDoSQ`

Communication and Storage Overhead: We summarized our findings in Table II and explain them as follows: *(i)* Multi-server PIR is the predominant cost due to its communication overhead. The communication cost of retrieving $\sqrt{nw}$ bits from ℓ PSBs is approximately $\ell \times \sqrt{nw}$. The transmitted data volume increases linearly with the number of database entries. *(ii)* The storage overhead at the client side is minimal, but that of PSBs increases linearly with the number of puzzles and signatures. The hash-based puzzle $\Pi = (\kappa, N_B)$ features a difficulty level κ of 4 bytes and a nonce N_B of κ bits, with a *Dilithium* signature size of 2.363 KB. Given these specifications, each block has a fixed size of 2.93 KB, resulting in database sizes of 4.1 MB, 16.8 MB, 67.3 MB, and 538.2 MB respectively for a grid segment. *(iii)* The communication aspect

[1] https://percy.sourceforge.net/

[2] https://openquantumsafe.org/

[3] https://www.openssl.org/

[4] https://www.sqlite.org/

[5] https://enterpriseefiling.fcc.gov/dataentry/public/tv/lmsDatabase.html

Operation	|DB|			
	2^{10}	2^{12}	2^{14}	2^{17}
Total Communication	12.98 KB	25.99 KB	77.69 KB	605.92 KB
Client's Storage	4.19 KB	17.2 KB	68.9 KB	597.13 KB
PSB's Storage	4.1 MB	16.8 MB	67.3 MB	538.2 MB
Communication Delay	≈ 145 ms	≈ 175 ms	≈ 275 ms	≈ 650 ms
Circuit RTT Latency	≈ 250 ms			

TABLE II: Communication/Storage Overhead of PACDoSQ

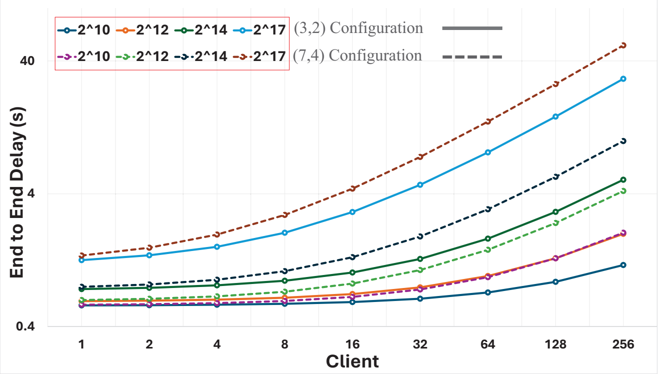

Fig. 2: End-to-End delay of PACDoSQ for increasing clients.

of *PQ-Tor* closely mirrors conventional Tor, with negligible differences (e.g., *Kyber* ops). Thus, we utilized conventional Tor network metrics for communication delay estimation [23]. Despite *Kyber* being faster than conventional *RSA* used in Tor, employing *Kyber* necessitates two packet transmissions due to Tor's default packet size of 512 bytes, resulting in an average bound of 300 ms for circuit build time. The communication delay entails the average timing of sending PIR requests and receiving PIR responses via PQ-Tor within a built circuit.

Scalability Assessment: We assess the performance of PACDoSQ for a growing number of clients and different PSB configurations, offering different privacy-speed trade-offs. Our evaluation combines computational and communication overhead to analyze the perceived end-to-end delay for the clients and PSB servers. The client can retrieve its puzzle along with spectrum availability information *offline*. The process involves fetching the signed puzzle from PSBs using multi-server PIR over PQ-Tor. The end-to-end delay encompasses the PIR computation on both the client and PSB sides plus the communication delays due to PQ-Tor when fetching a block of data from multiple PSB databases. The experimental analysis of PACDoSQ for numerous clients with various privacy configurations is depicted in Fig. 2. Upon successfully retrieving the puzzle, the client can connect with the server efficiently by solving PoW and sending its solution with *Token*. This (online) phase is swift, and mirrors standard client-server puzzle settings, with the key difference being that the request is transmitted through a PQ-secure TLS channel.

VI. Conclusion

We present PACDoSQ, a novel cybersecurity framework designed to address the multifaceted challenges of security, privacy, and DoS attacks in SAS amidst the expanding mobile and IoT landscape and the looming threat of quantum computing. By integrating PSBs with multi-server PIR, PQ-secure Tor, and hash-based client-server puzzles, PACDoSQoffers a comprehensive solution that ensures location privacy, anonymity, and resilience against DoS attacks in the PQ era. Formal security proofs validate the security of PACDoSQ, while comprehensive performance evaluations underscore its feasibility and efficiency. As network services continue to evolve, PACDoSQ stands as an important step towards establishing a holistic cybersecurity framework safeguarding spectrum management systems from a myriad of cyber threats with reasonable overhead.

VII. Acknowledgment

This research is supported by Cisco Research Award (220159) and partially supported by the NSF CNS-2350213.

References

[1] M. Grissa, A. A. Yavuz, B. Hamdaoui, and C. Tirupathi, "Anonymous dynamic spectrum access and sharing mechanisms for the cbrs band," *IEEE Access*, vol. 9, pp. 33860–33879, 2021.

[2] T. Chakraborty, S. Mitra, and S. Mittal, "Capow: Context-aware ai-assisted proof of work based ddos defense," *arXiv preprint*, 2023.

[3] D. K. Jasim and S. B. Sadkhan, "Cognitive radio network: Security and reliability trade-off-status, challenges, and future trend," in *2021 1st BICITS*, pp. 149–153, IEEE, 2021.

[4] S. Darzi, K. Ahmadi, and S. Aghapour, "Envisioning the future of cyber security in post-quantum era: A survey on pq standardization, applications, challenges and opportunities," *arXiv:2310.12037*, 2023.

[5] T. Chakraborty, A. Islam, and V. King, "Bankrupting dos attackers despite uncertainty," *arXiv preprint arXiv:2205.08287*, 2022.

[6] M. Grissa, A. Yavuz, and B. Hamdaoui, "Lpos: Location privacy for optimal sensing in cognitive radio networks," in *2015 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2015.

[7] S. Darzi and A. A. Yavuz, "Counter denial of service for next-generation networks within the artificial intelligence and post-quantum era," *arXiv preprint arXiv:2408.04725*, 2024.

[8] P. H. Masur, J. H. Reed, and N. K. Tripathi, "Artificial intelligence in open-radio access network," *IEEE Aerospace and Electronic Systems Magazine*, vol. 37, no. 9, pp. 6–15, 2022.

[9] R. Doriguzzi-Corin and D. Siracusa, "Flad: adaptive federated learning for ddos attack detection," *Computers & Security*, vol. 137, 2024.

[10] M. Mittal, K. Kumar, and S. Behal, "Deep learning approaches for detecting ddos attacks: A systematic review," *Soft computing*, vol. 27, no. 18, pp. 13039–13075, 2023.

[11] V. Bostanov, "Client puzzle protocols as countermeasure against automated threats to web applications," *IEEE Access*, vol. 9, 2021.

[12] I. M. Ali, M. Caprolu, and R. D. Pietro, "Foundations, properties, and security applications of puzzles: A survey," *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1–38, 2020.

[13] F. N. Kiruthika, "A new approach to defend against ddos.," *Computer Science & Telecommunications*, vol. 31, no. 2, 2011.

[14] M. Grissa, A. Yavuz, and B. Hamdaoui, "An efficient technique for protecting location privacy of cooperative spectrum sensing users," in *2016 IEEE conf. on comp. comm. workshops*, IEEE, 2016.

[15] P. Agarwal, M. Manekiya, T. Ahmad, A. Yadav, A. Kumar, M. Donelli, and S. T. Mishra, "A survey on citizens broadband radio service (cbrs)," *Electronics*, vol. 11, no. 23, p. 3985, 2022.

[16] I. Goldberg, "Improving the robustness of private information retrieval," in *2007 IEEE Symposium on Security and Privacy (SP'07)*, IEEE, 2007.

[17] Z. Tujner, "Quantum-safe tor, post-quantum cryptography," Master's thesis, University of Twente, 2019.

[18] J. Bos, L. Ducas, E. Kiltz, and Lepoint, "Crystals-kyber: a cca-secure module-lattice-based kem," in *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 353–367, IEEE, 2018.

[19] L. Ducas, E. Kiltz, and T. Lepoint, "Crystals-dilithium: A lattice-based digital signature scheme," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 238–268, 2018.

[20] T. Aura and P. Nikander, "Dos-resistant authentication with client puzzles," in *Intl. workshop on sec. prtcl.*, Springer, 2000.

[21] B. Glas and J. Guajardo, "Signal-based automotive communication security and its interplay with safety requirements," in *Proceedings of Embedded Security in Cars Conference*, Citeseer, 2012.

[22] X. Bonnetain and Naya-Plasencia, "Quantum security analysis of aes," *IACR Tran. on Sym. Cryp.*, vol. 2019, no. 2, 2019.

[23] "Tor metrics." https://metrics.torproject.org/torperf.html, 2024. Accessed: May, 2024.