PQClass: Classification of Post-Quantum Encryption Applications in Internet Traffic

Angelos K. Marnerides*, Chen Hajaj†, Revital Marbel‡, Ran Dubin§, Amit Dvir§
*Dept. of Electrical & Computer Engineering & KIOS CoE, University of Cyprus, Nicosia, Cyprus
marnerides.angelos@ucy.ac.cy
† Dept. of Industrial Engineering and Management & Data Science and AI Research Center, Ariel University, Israel
chenha@ariel.ac.il
‡ School of Computer Science, Holon Institute of Technology, Israel
revi85@gmail.com
§ Dept. of Computer & Software Engineering, Ariel Cyber Innovation Centre, Ariel University, Israel
[rand,amitdv]@ariel.ac.il

***Abstract*—Post-quantum cryptography (PQC) is expected to revolutionize secure communications in next-generation digital ecosystems. Previous and ongoing activities demonstrate that different PQC algorithms significantly impact traffic latency, but they do not yet provide a scheme to assess the existence of the PQC algorithm or its identification when encrypted traffic is analyzed for traffic engineering purposes. Hence, this work is the first to propose a novel PQClass pipeline for classifying encrypted Internet traffic of recently NIST-approved PQC algorithms. Hence, it establishes solid grounds for enabling engineers to optimize their networks and, in parallel, for cybersecurity practitioners to familiarise themselves with PQC algorithmic properties for enhancing or devising security architectures in diverse setups. Our pipeline demonstrates impressive performance on real-world data, achieving 86% accuracy in detecting the presence of a PQC algorithm and 91% and 98% accuracy in identifying the browser and OS, respectively, based on PQC-based traffic.**

***Index Terms*—Post-Quantum Cryptography, Classification, Encrypted Traffic**

I. Introduction

The field of cryptography stands at a critical point as the advent of quantum computing threatens to undermine the security of current cryptographic systems. This impending challenge has given rise to the rapidly evolving field of Post-Quantum Cryptography (PQC), which aims to develop cryptographic systems resistant to attacks from quantum computers. At the forefront of this endeavor is the National Institute of Standards and Technology (NIST), a U.S. federal agency renowned for developing standards, conducting research, and collaborating with industry to advance and apply new technologies [1]. Recently, NIST made a landmark announcement, selecting a set of PQC algorithms for digital signature and key establishment protocols: Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM), which is based on CRYSTALS-Kyber [2]; Module-Lattice-Based Digital Signature Algorithm (ML-DSA) which is based on CRYSTALS-Dilithium [3]; Stateless Hash-Based Digital Signature Algorithm (SLH-DSA) which is based on SPHINCS+ [4]; FFT (fast-Fourier transform) over NTRU-Lattice-Based Digital Signature Algorithm (FN-DSA) which is based om FALCON (Fast-Fourier Lattice-based Compact Signatures over NTRU) [5] and for now it is a draft.

The field of encrypted traffic classification grapples with the challenge of understanding and categorizing network traffic increasingly protected by sophisticated encryption methods. This task has grown in complexity over recent years, prompting researchers to explore innovative approaches. Specifically, integrating deep learning and artificial intelligence models has introduced additional and improved research perspectives to these approaches [6], enabling more nuanced and accurate classification of encrypted traffic.

While current PQC research has primarily focused on characterizing the performance overhead of quantum-resistant algorithms [7]–[13], the classification of metadata within PQC traffic represents a critical gap in the research, highlighting the lack of studies in this area. Addressing this gap is essential, especially given the upcoming changes to network protocols. For instance, the list of PQC algorithms a client uses, currently visible in the ”supported groups” and ”signature algorithms” fields of TLS Client Hello packets, will soon be encrypted using the Encrypted Client Hello (ECH) TLS extension. This change necessitates the development of new methods for inferring information about the cryptographic capabilities and choices of network participants [13].

In this paper, we explore if and how a set of classifiers based on packet size, direction, and timing could enrich our understanding of the properties of PQC-enabled applications and PQC flows. Moreover, how they can provide information about web browsers and applications that use PQC algorithms. The key question from the above discussion is whether we can classify the overhead introduced by new PQC algorithms from encrypted traffic. What role do Operating Systems (OS), web browsers, and applications play in contributing to PQC protocol overhead? For instance, can we classify two web browsers using the PQC algorithms? The following research questions highlight the areas of interest:

- Can we classify between encrypted traffic connections that use PQC algorithms and connections that do not use PQC algorithms?
- Can we classify the operating system, which may support different PQC algorithms?
- Can we classify the web browser, which may implement different PQC algorithms?

II. Contributions

We provide a novel method, termed hereafter PQClass, for classifying encrypted network traffic deployed using PQC algorithms[1], addressing a critical gap in our understanding of next-generation secure communications. In order to effectively assess PQClass, we create a comprehensive dataset of PQC- and Non-PQC-based traffic patterns (various OSs and Web browsers), which the research community may reuse to study PQC traffic classification patterns. By virtue of the fact the PQClass correlates the behavior of network traffic with system and application-level characteristics, we enable the identification of PQC/Non-PQC sessions and user environments from encrypted data streams, providing unprecedented insights into the performance and security implications of quantum-resistant cryptography in real-world scenarios. Finally, PQClass is delivered as an open-source analysis toolkit [14], paving a tangible path towards standardizing PQC traffic performance practices. Thus fostering collaborative advancements in cybersecurity and laying the critical groundwork for more resilient and efficient network infrastructures in the post-quantum era.

The main contributions of this work are summarized as follows:

- **Dataset Composition**: We introduce a unique dataset that facilitates the reproduction of our results and enables the establishment of benchmarks for PQC application traffic classification.
- **Encrypted Traffic Classification**: We accurately classify encrypted traffic, distinguishing between PQC and non-PQC traffic while also identifying application-level (e.g., browsers) and system-level properties (e.g., OS).
- **Open-Source Tool**: We provide the first open-source tool for traffic classification under the PQC paradigm, [14], which can serve as a foundation for further research and development.

The remainder of this paper is structured as follows: Section III reviews the background and existing work on PQC and traffic classification, whereas Section IV introduces the pipeline and composition of the PQClass mechanism. Section V is dedicated to discussing our results. Finally, Section VI summarises and concludes this work.

III. Background and Related Work

A. PQC algorithms overview

First, we briefly describe the NIST PQC algorithms for digital signature, encryption, and key establishment in the following section, followed by describing the works in this field. **CRYSTALS-Kyber**, the security of Kyber is based on the hardness of solving the learning-with-errors problem in module lattices. **CRYSTALS-Dilithium** algorithm is a lattice-based cryptographic algorithm designed to create secure digital signatures against both classical and quantum computing attacks. **FALCON** algorithm is a post-quantum digital signature scheme engineered to provide robust security against quantum attacks while maintaining computational and storage efficiency. It is grounded in the NTRU lattice problem, which relies on the difficulty of finding the shortest vector in a lattice generated by a polynomial ring and is considered difficult to solve by both classical and quantum computers. **SPHINCS+** algorithm is an advanced version of the original SPHINCS (Stateless Practical Hash-based Incredibly Nice and Compact Signatures) scheme, recognized for its robust security.

B. PQC Characterisation & Network Traffic

Recently, scientific works focused on the intersection of PQC with core internet networking technologies, where researchers have examined the performance and feasibility of PQC ciphers with key enablers of secure internet networking, including Public-Key Infrastructure (PKI) for trust and protocols like TLS 1.3 and QUIC for secure session establishment. Such works are crucial to understanding how PQC algorithms integrate and impact the existing network infrastructure [7]–[13].

Sikeridis et al. [7] conducted a comprehensive performance study on the impact of post-quantum signature algorithm candidates on TLS 1.3 under realistic network conditions, providing valuable data on how these new algorithms might affect real-world network performance. Building on this, Raavi et al. [8], [9] presented an in-depth analysis of post-quantum

[1] From now on we refer to traffic deploying PQC algorithms as PQC encrypted network traffic

digital signature performance overheads in terms of computational and memory requirements. The authors' work also included the development of a security comparison model for understanding and comparing the security of algorithms with different hardness problems, offering a pipeline for evaluating the trade-offs between security and performance in PQC algorithms. A study by Henrich et al. [10] delved into the specifics of the TLS handshake and Key Encapsulation Mechanisms (KEMs) under varying network characteristics, aiming to identify suitable quantum-safe algorithms within TLS 1.3 and considering various PQC KEMs and KEM parameters. Given the prevailing network quality, the authors provided valuable recommendations regarding the use of various algorithms and configurations. Notably, their work demonstrated how unwanted scheduling of operating system processes may affect performance and how TCP control mechanisms heavily influence handshake performance.

The application of PQC to specific domains has also been researched. Sajimon et al. [11] analyzed cryptography algorithms and protocols in the context of the Internet of Things (IoT), evaluating the performance of PQC schemes for IoT and suggesting suitable approaches. This work is particularly relevant given the growing importance of the IoT in various sectors and the need to secure these often resource-constrained devices against future quantum threats.

Finally, in the mobile domain, Mankowski et al. [12] analyzed TLS usage by the highest-ranked apps from the Google Play Store. This work assessed the potential overhead from adopting post-quantum algorithms in mobile applications, providing insights into the practical implications of transitioning to PQC in the mobile ecosystem.

IV. PQClass

In this section, we introduce the core elements of our PQC classification (PQClass) pipeline. Our approach combines sophisticated data collection methodologies with advanced feature extraction techniques to develop a robust classification system to identify subtle patterns in encrypted traffic. We begin by detailing the dataset collection process, which provides the empirical foundation for our research. Following this, we explain the feature extraction process—a crucial step that distills traffic representations into fundamental characteristics such as packet size, timing, and directional flow. These features form the basis for training a comprehensive suite of classification algorithms. Each algorithm is trained on various classification tasks, ranging from the binary PQC/Non-PQC distinction to more complex multi-class tasks, including OS and browser classification.

A. Dataset creation

The data collection process for our research was designed in several well-defined phases to ensure a comprehensive and accurate representation of PQC-based network traffic across various environments.

Environment Setup: In this phase, we configure the diverse operating systems (e.g., Windows, macOS, Linux) and web browsers (e.g., Chrome, Firefox), with the PQC flag at the web browsers true in the case of PQC traffic or false in the case of non-PQC traffic. This setup ensures we capture various environments to simulate real-world conditions accurately. **Traffic Capture:** During this phase, we capture encrypted network traffic generated by the simulated interactions. The traffic includes the combinations of operating systems and web browsers. **Packet Analysis:** Once the traffic is captured, we extract and analyze it. The analysis is divided into two distinct stages: the handshake phase and the data transfer phase. This separation allows us to observe the specific impact of PQC algorithms. This dataset forms the foundation for our subsequent analysis and classification efforts, offering a valuable resource for understanding the impact of PQC algorithms on network traffic patterns across various protocols and software environments.

Feature Extraction Our approach relies on three fundamental features: Packet timestamp, Packet direction (client-to-server or server-to-client) and Packet size. The rationale behind choosing these features is twofold: (1) traditional features such as TCP window size and Server Name Indication (SNI) will soon be inaccessible due to the increasing adoption of QUIC, which uses UDP, and focusing on distinguishing between; and (2) adding authenticity and encryption introduces overhead [8], [9], which affects packet transmission times.

B. Learning Models

We evaluated several machine learning classifiers on the vector representation of our data, employing a diverse range of modeling techniques. Specifically, we tested tree-based models, ensemble methods, distance-based classifiers, discriminative models, and neural networks to provide a comprehensive analysis of classification performance. The classifiers included Decision Tree, Random Forest, XGBoost, Logistic Regression, K-Nearest Neighbors (KNN), Gaussian Naive Bayes, Multi-Layer Perceptron, AdaBoost, and Gradient Boosting.

Tree-based models such as Random Forest and Decision Tree leverage decision nodes to partition the feature space, with Random Forest employing an ensemble of trees grown on randomly selected subsets of features to improve generalizability. Ensemble methods, including XGBoost, AdaBoost, and Gradient Boosting, iteratively build multiple models, with each successive model focusing on correcting the errors of the previous iterations. XGBoost, in particular, is notable for its computational efficiency and effectiveness in handling large datasets. On the other hand, distance-based models, such as KNN, classify data points based on the proximity to their nearest neighbors in the feature space, offering an intuitive approach for classification tasks. Logistic Regression, a discriminative model, estimates the posterior probability of class membership by modeling the decision boundary directly. We also employed the MLP classifier, a type of neural network that captures complex, non-linear relationships by propagating information through multiple hidden layers. This diverse selection of classifiers allows for a broad exploration of the underlying data structure, enabling us to compare model performance across different methodologies and identify the most suitable approach for our specific problem domain.

V. Evaluation

A. Dataset

Using the PQClassify pipeline, we generated a dataset to study communication across different operating systems and browsers with PQC encryption. The dataset includes 100 samples for each combination of the following operating systems—Windows, Linux, macOS, and iOS—and browsers—Firefox and Chrome. For each combination, where technically feasible, we simulated transactions with and without PQC, allowing for classification across various environments. Table I presents the different classes included in the dataset and the number of samples collected for each. In the spirit of open science, this dataset and our data collection and feature extraction methodology are publicly available on our pipeline' GitHub repo [14].

B. Metrics

The performance of each model is assessed based on common classification evaluation metrics: accuracy, precision, recall, and F1-score. These metrics are crucial for determining the most effective and practical models for real-world encrypted internet traffic classification. Accuracy is calculated using the following formula: $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$. Precision measures the proportion of correctly predicted positive instances out of all instances predicted as positive. The Precision formula is as follows: $Precision = \frac{TP}{TP+FP}$. Recall measures the proportion of correctly predicted positive instances out of all actual positive instances. The Recall formula is as follows: $Recall = \frac{TP}{TP+FN}$. F1-score is a weighted average of precision and recall that considers both metrics to provide an overall measure of the model's performance. The F1-score formula is: $F1-score = 2 \times \frac{Precision \times Recall}{Precision+Recall}$. Finally, AUC (Area Under the Curve) measures the overall ability of the model to distinguish between positive and negative classes across various thresholds. It is calculated as the area under the Receiver Operating Characteristic (ROC) curve, which plots the True Positive Rate (Recall) against the False Positive Rate. Higher AUC values reflect a better model performance, showing a greater capacity to accurately identify positive and negative instances at multiple threshold settings. Including AUC in our evaluation offers a broader perspective on the model's performance, as it encapsulates the trade-offs between Recall and False Positive Rate across thresholds.

C. Results

To better understand the viability of PQC classification across multiple platforms, we conducted four experiments, for each experiment we used 10-fold stratified cross-validation to verify the robustness of our results: **PQC binary classification**, where the ability to classify if the encrypted traffic is using PQC algorithm is tested. **Browser classification**, where different browsers that use the PQC algorithm are classified. **Operation system classification**, where the same experiment is performed on different operation systems. **Tuple Classification (Browser, OS, and PQC)** where each specific case (e.g., specific OS with a specific browser and using PQC or not using PQC) is considered a class for classification. **PQC binary classification.** In the first experiment, we aim to classify encrypted traffic to determine if a user employs a PQC algorithm. Consequently, this experiment involves a binary classification task, with results summarized in Table II. Our evaluation indicates that a simple decision tree can successfully identify the presence of a PQC algorithm with an accuracy of 86%. It demonstrates a high recall, accurately identifying 85% of the samples that utilize the PQC algorithm.

TABLE I: Dataset

OS	**Browser**	**PQC**	**Samples**
Windows	Firefox	X	100
	Firefox	✓	100
	Chrome	X	100
	Chrome	✓	100
Linux	Firefox	X	100
	Firefox	✓	100
	Chrome	X	100
	Chrome	✓	100
MacOS	Firefox	X	100
	Firefox	✓	100
	Chrome	X	100
	Chrome	✓	100

We further analyzed the subtle distinctions between three operating systems—Windows, Linux, and macOS—and two

TABLE II: PQC, Non-PQC classification Results

Model	Acc.	Prec.	Rec.	F1	AUC
AdaBoost	0.82	0.82	0.82	0.81	0.86
Decision Tree	0.86	0.85	0.85	0.85	0.87
Gradient Boosting	0.84	0.84	0.84	0.84	0.9
KNN	0.76	0.76	0.76	0.76	0.81
Logistic Regression	0.75	0.76	0.75	0.74	0.79
MLP	0.76	0.75	0.74	0.74	0.78
Naive Bayes	0.55	0.57	0.55	0.48	0.76
Random Forest	0.83	0.82	0.82	0.83	0.89
XGBoost	0.85	0.85	0.85	0.85	0.92

TABLE III: Given PQC, Browsers Classification Results

Model	Acc.	Prec.	Rec.	F1	AUC
AdaBoost	0.89	0.91	0.89	0.88	0.94
Decision Tree	0.91	0.93	0.91	0.91	0.92
Gradient Boosting	0.91	0.93	0.92	0.9	0.95
KNN	0.82	0.82	0.82	0.81	0.87
Logistic Regression	0.73	0.74	0.73	0.73	0.79
MLP	0.8	0.81	0.81	0.82	0.83
Naive Bayes	0.76	0.77	0.76	0.75	0.79
Random Forest	0.89	0.9	0.89	0.88	0.92
XGBoost	0.91	0.92	0.91	0.9	0.93

TABLE IV: Given PQC, Operation Systems Classification Results

Model	Acc.	Prec.	Rec.	F1	AUC
AdaBoost	0.95	0.96	0.95	0.95	0.98
Decision Tree	0.96	0.96	0.96	0.95	0.97
Gradient Boosting	0.98	0.98	0.98	0.98	1
KNN	0.9	0.91	0.9	0.9	0.96
Logistic Regression	0.89	0.89	0.89	0.88	0.93
MLP	0.87	0.88	0.89	0.88	0.94
Naive Bayes	0.9	0.9	0.9	0.89	0.95
Random Forest	0.98	0.99	0.98	0.97	1
XGBoost	0.98	0.98	0.98	0.98	0.99

TABLE V: Classify tuple: Browser, OS and Algorithm (PQC, nonPQC)

Model	Acc.	Prec.	Rec.	F1	AUC
AdaBoost	0.83	0.83	0.84	0.83	0.92
Decision Tree	0.81	0.82	0.82	0.8	0.9
Gradient Boosting	0.84	0.84	0.84	0.83	0.98
KNN	0.63	0.64	0.63	0.62	0.91
Logistic Regression	0.62	0.62	0.62	0.61	0.91
MLP	0.67	0.65	0.65	0.64	0.91
Naive Bayes	0.46	0.42	0.46	0.39	0.91
Random Forest	0.82	0.82	0.82	0.81	0.98
XGBoost	0.84	0.84	0.84	0.83	0.98

web browsers, Firefox and Chrome, while using PQC encryption. The objective was to determine whether traffic could be accurately classified, assuming that the encrypted traffic employed PQC. For these experiments, we utilized a subset of the dataset consisting solely of samples with PQC. **Browser classification.** Table III presents the results for classifying different web browsers. According to the table, the Gradient Boosting model outperforms other classifiers, achieving the highest accuracy and AUC. **Operation system classification** The classification results for different operating systems are shown in Table IV. The Random Forest model achieves the highest accuracy and the best AUC. **Tuple Classification (Browser, OS, and PQC).** In the fourth experiment, we addressed the most complex task: classifying encrypted traffic samples based on whether they use PQC and determining the user's parameters, such as operating system and browser. This experiment aimed to identify behavioral differences in real-life scenarios. As shown in Table V, XGBoost and Gradient Boosting demonstrate superior performance in both accuracy and AUC, confirming their robustness in handling multi-dimensional classification tasks.

Overall, these results demonstrate that traffic encrypted with PQC can be accurately classified under various scenarios. The XGBoost classifier consistently delivers strong performance, particularly when tackling complex classification tasks that involve multiple parameters. A summary of our accuracy results for the different classification tasks is provided in Fig. 1. WLOG, and given that the dataset is balanced, our primary focus is on the accuracy metric; nevertheless, all other relevant metrics are reported in Tables II–V. Next, we examine how the number of packets within a flow influences model accuracy for classifying the tuple (browser, OS, PQC). As illustrated in Fig. 2, the top-performing models already yield strong results after the first ten packets—a pattern closely linked to the handshake. To summarize, we introduce several key innovations in classifying whether the user uses PQC and user environments. Our research utilizes actual encrypted traffic implementing PQC for classification, moving away from the common practice of simulating network behaviors. This approach provides more accurate and realistic insights into the performance of PQC protocols in real-world scenarios, thereby enhancing the reliability and applicability of our classification models. Second, we have developed a model that distinguishes between PQC and non-PQC algorithms, adding a critical layer of specificity to the classification process. Given the assistance of a PQC algorithm, we show that both the browser and the

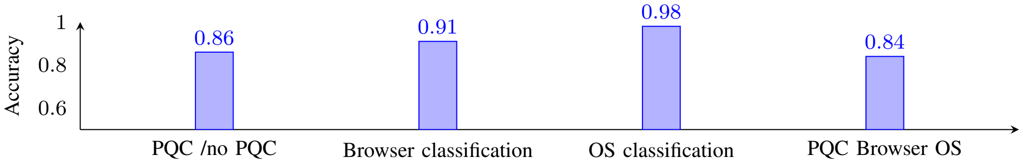

Fig. 1: Summary of Accuracy Results for All Experiments

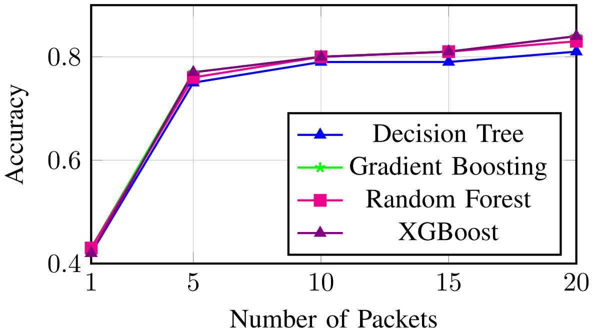

Fig. 2: The influence of the number of packets on the model accuracy

OS used by the user can be accurately detected.

VI. Conclusions

This research is significant because it has the potential to enhance the security and efficiency of Internet communications in the era of quantum computing. As quantum computers advance, traditional cryptographic methods face the threat of becoming obsolete. Our research addresses this by classifying encrypted traffic that uses PQC, which is designed to be secure against quantum attacks. Accurately classifying these algorithms is crucial for identifying vulnerabilities and ensuring robust security measures. In addition, by analyzing the effects of browsers used by the user on the identity of the PQC algorithm, critical aspects of user privacy and service optimization further contribute to the security and efficiency of Internet communications. This paper's findings are based on PQC traffic recorded in a controlled laboratory setting. As PQC traffic becomes more prevalent, performing similar analyses on real-world ("in the wild") traffic from multiple locations will be crucial. Looking ahead, future research could expand this work by leveraging a broader dataset drawn from widely used applications, such as YouTube and Facebook. A key objective is to classify PQC traffic as a whole and distinguish specific NIST algorithms within these different applications. Achieving such granularity would deepen our understanding of PQC performance in diverse environments and foster the development of comprehensive security solutions tailored for the post-quantum era.

VII. Acknowledgments

The EU Horizon COCOON Project supported this work under Grant Agreement No 101120221. The authors would like to thank Noam Leshem and Eylon Yaakov Katan for their help with the data creation.

References

[1] NIST, "Nist publications." https://https://csrc.nist.gov/publications/fips, 2024. 30 Oct 2024.

[2] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, "Crystals-kyber," *NIST, Tech. Rep*, 2017.

[3] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, "Crystals-dilithium: A lattice-based digital signature scheme," *IACR TCHES*, pp. 238–268, 2018.

[4] D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, and P. Schwabe, "The sphincs+ signature framework," in *ACM SIGSAC*, pp. 2129–2146, 2019.

[5] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, *et al.*, "Falcon: Fast-fourier lattice-based compact signatures over ntru," *NIST's post-quantum cryptography standardization process*, vol. 36, no. 5, pp. 1–75, 2018.

[6] H. Zhang, L. Yu, X. Xiao, Q. Li, F. Mercaldo, X. Luo, and Q. Liu, "Tfe-gnn: A temporal fusion encoder using graph neural networks for fine-grained encrypted traffic classification," in *ACM Web Conference*, pp. 2066–2075, 2023.

[7] D. Sikeridis, P. Kampanakis, and M. Devetsikiotis, "Post-quantum authentication in tls 1.3: A performance study," *IACR Cryptol. ePrint Arch.*, vol. 2020, p. 71, 2020.

[8] M. Ravvi, *Enabling Post-Quantum Cryptography for Secure Internet Networking: Performance, Feasibility Analyses, and Solutions*. PhD thesis, University of Colorado Colorado Springs, 2023.

[9] M. Raavi, S. Wuthier, X. Zhou, and S.-Y. Chang, "Post-quantum quic protocol in cloud networking," in *European Conference on Networks and Communications and 6G Summit*, pp. 573–578, 2023.

[10] J. Henrich, A. Heinemann, A. Wiesmaier, and N. Schmitt, "Performance impact of pqc kems on tls 1.3 under varying network characteristics," in *Information Security, vol 14411*, pp. 267–287, 2023.

[11] S. P C, K. Jain, and P. Krishnan, "Analysis of post-quantum cryptography for internet of things," in *International Conference on Intelligent Computing and Control Systems*, pp. 387–394, 2022.

[12] D. Mankowski, T. Wiggers, and V. Moonsamy, "Tls $\rightarrow$ post-quantum tls: Inspecting the tls landscape for pqc adoption on android," in *European Symposium on Security and Privacy Workshops*, pp. 526–538, 2023.

[13] T. Liu, G. Ramachandran, and R. Jurdak, "Post-quantum cryptography for internet of things: a survey on performance and optimization," *arXiv:2401.17538*, 2024.

[14] ACIC-ARIEL, "Pqclass dataset and code." GitHub, 2024. https://github.com/ArielCyber/PQClass.