Uma Arquitetura Baseada em *Sidecar* para Integração de Criptografia Pós-Quântica no Core 5G

Ricardo Alves Faval

Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2026

Ricardo Alves Faval

Uma Arquitetura Baseada em *Sidecar* para Integração de Criptografia Pós-Quântica no Core 5G

Dissertação de mestrado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Prof. Flávio de Oliveira Silva, Ph.D.

Uberlândia
2026

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

F272 Faval, Ricardo Alves, 1979-
2026 Uma Arquitetura Baseada em Sidecar para Integração de Criptografia Pós-Quântica no Core 5G [recurso eletrônico] / Ricardo Alves Faval. - 2026.

Orientador: Flávio de Oliveira Silva.
Dissertação (Mestrado) - Universidade Federal de Uberlândia, Pós-graduação em Ciência da Computação.
Modo de acesso: Internet.
DOI http://doi.org/10.14393/ufu.di.2026.244
Inclui bibliografia.
Inclui ilustrações.

1. Computação. I. Silva, Flávio de Oliveira ,1970-, (Orient.). II. Universidade Federal de Uberlândia. Pós-graduação em Ciência da Computação. III. Título.

CDU: 681.3

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:
Gizele Cristine Nunes do Couto - CRB6/2091
Nelson Marcos Ferreira - CRB6/3074

Agradecimentos

A conclusão deste trabalho não é o resultado de um esforço isolado, mas o reflexo do apoio, da paciência e da confiança depositados por pessoas fundamentais em minha trajetória.

Ao meu orientador, Dr. Flávio de Oliveira Silva, pela orientação segura e pelo apoio irrestrito desde as primeiras definições deste projeto. Sua expertise e visão foram essenciais para que eu pudesse navegar pelos desafios deste tema, e sua postura sempre solícita foi um exemplo de dedicação acadêmica.

À minha esposa, Ana Paula Gontijo Faval, por ser meu porto seguro. Pela compreensão nas ausências, pelo incentivo nos momentos de cansaço e por acreditar neste sonho tanto quanto eu. Sem o seu amor e suporte incondicionais, este caminho teria sido muito mais árduo.

Aos meus pais, Nilton Faval dos Santos (em memória) e Maria de Lourdes Alves.

Ao meu pai, por ter sido meu primeiro mestre e alicerce do que me tornei hoje. Embora seu gênio fosse, por vezes, difícil, a retidão do seu caráter e sua honradez inquestionável moldaram minha visão de mundo e me deram a disciplina e a resiliência necessárias para concluir esta etapa.

À minha mãe, Maria de Lourdes, pelo amor incondicional e pela força serena que me sustentaram nos momentos de incerteza. A vocês, dedico esta conquista, ciente de que sou o resultado lógico de seus esforços e ensinamentos.

"Sua vida pode ser dividida em dois períodos: antes de agora e a partir de agora."
(Prof. Obvious Stating)

Resumo

O rápido avanço da computação quântica representa uma ameaça significativa aos fundamentos de segurança das redes 5G, que atualmente dependem da criptografia clássica de curva elíptica (Elliptic-curve cryptography (ECC)) para a proteção de assinaturas e certificados. Esta dissertação investiga a viabilidade da integração de criptografia pós-quântica (Post-Quantum Cryptography (PQC)) no núcleo 5th Generation Mobile Network (5G) para estabelecer comunicações resistentes a ataques quânticos por meio da Service-Based Interface (SBI) e da interface N2. Utilizando o framework `free5gc-compose`, uma arquitetura de proxy *sidecar* não intrusiva foi desenvolvida e orquestrada via Docker, permitindo a atualização tecnológica PQC padronizada ("PQC-retrofitting") das Network Functions (NFs) sem a necessidade de alterar o código-fonte subjacente.

A configuração experimental avalia três implementações arquiteturais distintas: uma linha de base nativa (arquitetura original com Hypertext Transfer Protocol Security (HTTPS)/Transport Layer Security (TLS) habilitado), uma arquitetura Sidecar Proxy TLS utilizando certificados tradicionais via `OpenSSL` e uma arquitetura Sidecar Proxy PQC que utiliza algoritmos padronizados pelo National Institute of Standards and Technology (NIST) via `liboqs` por meio de microsserviços especializados (`wrapper-kem` e `wrapper-sign`).

A latência foi medida no limite da requisição-resposta Hypertext Transfer Protocol (HTTP)/2 na SBI, com métricas coletadas em trinta repetições independentes para garantir a significância estatística. Resultados preliminares estabelecem uma latência mediana de referência de 15,07 ms para a arquitetura Nativa, fornecendo um parâmetro para quantificar o *overhead* de desempenho introduzido pelos maiores tamanhos das chaves PQC. Este trabalho demonstra que uma abordagem baseada em *sidecar* oferece um caminho flexível e modular para a segurança da infraestrutura 5G contra futuros adversários quânticos, mantendo a integridade operacional da rede central.

Palavras-chave: Criptografia Pós-Quântica, Núcleo de Rede 5G, Proxy Sidecar, Inter-

face Baseada em Serviços (SBI), Virtualização de Funções de Rede (NFV), Padronização NIST PQC.

A Sidecar-Based Architecture for Integrating Post-Quantum Cryptography into the 5G Core

Ricardo Alves Faval

Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2026

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Coordenação do Programa de Pós-Graduação em Computação
Av. João Naves de Ávila, 2121, Bloco 1A, Sala 243 - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
Telefone: (34) 3239-4470 - www.ppgco.facom.ufu.br - cpgfacom@ufu.br

ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Ciência da Computação				
Defesa de:	Dissertação, 08/2026, PPGCO				
Data:	25 de Fevereiro de 2026	Hora de início:	09:00	Hora de encerramento:	12:34
Matrícula do Discente:	12312CCP030				
Nome do Discente:	Ricardo Alves Faval				
Título do Trabalho:	Uma Arquitetura Baseada em Sidecar para Integração de Criptografia Pós-Quântica no Core 5G				
Área de concentração:	Ciência da Computação				
Linha de pesquisa:	Sistemas de Computação				
Projeto de Pesquisa de vinculação:	----------				

Reuniu-se por videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Ciência da Computação, assim composta: Professores Doutores: Silvio Ereno Quincozes - Unipampa, Paulo Manuel Martins Carvalho - Universidade do Minho e Flávio de Oliveira Silva - Universidade do Minho, orientador do(a) candidato(a).

Os examinadores participaram desde as seguintes localidades: Flávio de Oliveira Silva - Braga/Portugal, Paulo Manuel Martins Carvalho - Braga/Portugal e Silvio Ereno Quincozes - Alegrete/RS . O aluno(a) participou da cidade de Uberlândia.

Iniciando os trabalhos o presidente da mesa, Prof. Dr. Flávio de Oliveira Silva, apresentou a Comissão Examinadora e o(á) candidato(a), agradeceu a presença do público, e concedeu ao(á) Discente a palavra para a exposição do seu trabalho.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir o(á) candidato(a). Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato(a):

Aprovado

Esta defesa faz parte dos requisitos necessários à obtenção do título de Mestre.

Ressalta-se que o examinador Paulo Manuel Martins Carvalho, por ser estrangeiro, residente em outro país e não possuir CPF registrado no Brasil não assinará a ata de defesa.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.

Documento assinado eletronicamente por **Flávio de Oliveira Silva**, **Usuário Externo**, em 03/07/2026, às 13:51, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Silvio Ereno Quincozes**, **Usuário Externo**, em 03/07/2026, às 14:53, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **7018105** e o código CRC **FC3F9279**.

Referência: Processo nº 23117.004498/2026-15 SEI nº 7018105

Abstract

The rapid advancement of quantum computing poses a significant threat to the security foundations of 5G networks, which currently rely on classical ECC for signaling protection. This dissertation investigates the viability of integrating PQC into the 5G core to establish quantum-resistant communication, potentially via the SBI. Utilizing the `free5gc-compose` framework, a non-intrusive sidecar proxy architecture was developed and orchestrated via Docker, enabling the "PQC-retrofitting" of standard NFs without altering their underlying source code. The experimental setup evaluates three distinct architectural deployments: a Native (HTTPS/TLS) baseline, a TLS Proxy using traditional certificates, and a PQC Proxy utilizing NIST-standardized algorithms through specialized microservices (`wrapper-kem` and `wrapper-sign`).

Latency was measured at the HTTP/2 request-response boundary on the SBI, with metrics captured across thirty independent repetitions to ensure statistical significance. The analysis incorporates both end-to-end SBI communication latency and proxy-side **Container Metrics** captured via an observability stack comprising *Prometheus* and *cAdvisor*.

Preliminary results establish a baseline median latency of 15.07 ms for the Native architecture, providing a benchmark to quantify the performance overhead introduced by larger PQC key sizes and computational requirements. This work demonstrates that a sidecar approach provides a flexible, modular pathway for securing 5G infrastructure against future quantum adversaries while maintaining the operational integrity of the core network.

Keywords: Post-Quantum Cryptography, 5G Core Network, Sidecar Proxy, Service-Based Interface (SBI), Network Function Virtualization (NFV), NIST PQC Standardization.

List of Figures

List of Tables

Acronyms list

3GPP	3rd Generation Partnership Project
5G	5th Generation Mobile Network
5GC	5G Core
6G	6th Generation Mobile Network
AMF	Access and Mobility Management Function
API	Application Programming Interface
AUSF	Authentication Server Function
AES	Advanced Encryption Standard
CA	Certificate Authority
CI	Confidence Interval
CPU	Central Processing Unit
CRQC	Cryptographically Relevant Quantum Computer
DSA	Digital Signature Algorithm
ECDH	Elliptic-Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
ECC	Elliptic-curve cryptography
E2E	End-To-End
eMBB	Enhanced Mobile Broadband
FIPS	Federal Information Processing Standards
Fuzz	(Fuzzing or Fuzz testing) - Automated Software Testing Technique
gNB	gNodeB
HNDL	Harvest Now, Decrypt Later

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Security

IoT Internet of Things

IQR Interquartile Range

IPsec Internet Protocol Security

IP Internet Protocol

I/O Input/Output

JSON JavaScript Object Notation

KEM Key Encapsulation Mechanism

LWE Learning with Errors

LTS Long Term Support

MITM Man-in-the-Middle

mTLS Mutual Transport Layer Security

mMTC Massive Machine Type Communications

NAS Non-Access Stratum

NSSF Network Slice Selection Function

NIST National Institute of Standards and Technology

NF Network Function

NFV Network Function Virtualization

NRF Network Repository Function

NVP Nearest Vector Problem

OAuth Open Authorization

OQS Open Quantum Safe

PCF Policy Control Function

PKI Public-Key Infrastructure

PQC Post-Quantum Cryptography

QoS Quality of Service

RAM Random Access Memory

RESTful REpresentational State Transfer Application Programming Interface

RAM Random Access Memory

RTT Round-Trip Time

RAN Radio Access Network

RSA	Rivest–Shamir–Adleman
RQ	Research Questions
SBA	Service-Based Architecture
SBI	Service-Based Interface
SCP	Service Communication Proxy
SCTP	Stream Control Transmission Protocol
SDN	Software-Defined Networking
SVP	Shortest Vector Problem
SMF	Session Management Function
socat	SOcket CAT
SUCI	Subscription Concealed Identifier
TCP	Transmission Control Protocol
TEE	Trusted Execution Environment
TLS	Transport Layer Security
UE	User Equipment
UDM	Unified Data Management
UPF	User Plane Function
UDR	Unified Data Repository
UDP	User Datagram Protocol
URLLC	Ultra-Reliable and Low Latency Communications
VM	Virtual Machine

Contents

I hereby certify that I have obtained all legal permissions from the owner(s) of each third-party copyrighted material included in my thesis, and that their permissions permit availability, such as deposit in public digital libraries.

student name and signature

Chapter 1

Introduction

The global transition toward 5G marks a definitive departure from the hardware-centric, monolithic architectures of the past. Unlike its predecessors, 5G is built upon a cloud-native, microservices-oriented framework known as the Service-Based Architecture (SBA). In this model, traditional point-to-point interfaces are replaced by a common bus on which NFs communicate via HTTP/2 over TLS 1.3. While this modularity facilitates unprecedented scalability and supports a heterogeneous ecosystem of billions of devices, it also introduces a centralized reliance on Public-Key Infrastructure (PKI) for the core network's internal security. As the telecommunications industry begins to architect the foundations of 6th Generation Mobile Network (6G), the cryptographic primitives that currently safeguard these infrastructures face an unprecedented existential threat: the rapid maturation of quantum computing. The security of the 5G Core (5GC) relies fundamentally on the computational hardness of mathematical problems such as integer factorization and discrete logarithms, which underpin algorithms like Rivest–Shamir–Adleman (RSA) and ECC. However, the advent of Cryptographically Relevant Quantum Computers (CRQCs) renders these classical defenses obsolete by enabling Shor's algorithm to solve these problems in polynomial time.

1.1 Motivation

The motivation for this research is the immediate and long-term security implications of the quantum threat for critical infrastructure. While large-scale, fault-tolerant quantum computers may still be years away, the vulnerability of 5G networks is a present-day concern due to the Harvest Now, Decrypt Later (HNDL) strategy. In this scenario, adversaries intercept and store encrypted 5G signaling traffic—containing Subscription Concealed Identifier (SUCI), authentication vectors, and session keys—with the intent to decrypt it once quantum hardware becomes available (SANON; SCHOTTEN, 2025). Given that telecommunications infrastructure often has a lifespan of more than a decade, security measures implemented today must remain resilient against threats in the 2030s

and beyond.

Furthermore, the centralized nature of the SBA means that a compromise of the UDM or the AUSF could result in the total collapse of subscriber privacy and network trust (SCALISE et al., 2024). Although the NIST finalized the first PQC standards in 2024, specifically Key Encapsulation Mechanism (KEM) and Digital Signature Algorithm (DSA) —their practical integration into the 5G Core remains largely unexplored.

Current literature highlights a significant performance gap: PQC algorithms typically have larger public keys, longer ciphertexts, and higher computational overhead than their classical counterparts (OLUSHOLA; MEENAKSHI, 2026). In a high-throughput, low-latency environment like the 5GC, these overheads could lead to signaling bottlenecks or increased handover latency. This research is motivated by the urgent need to empirically evaluate the feasibility of PQC in a live cloud-native 5G environment. By proposing a sidecar proxy architecture, this work seeks to provide a migration path that ensures quantum resilience for legacy 5GC functions without necessitating a disruptive overhaul of the existing software stacks.

The integration of PQC into the microservices-based telecommunications scenario introduces specific computational challenges that challenge established latency and resource-consumption parameters, as well as the overall architectural feasibility. To address this complex interaction, the present study employs a rigorous empirical methodology to systematically deconstruct the architectural and operational implications of transitioning to quantum-computing-resistant security protocols in the 5th Generation Mobile Network (5G) network core.

1.2 Research Questions (RQ)

This section presents the fundamental research questions that guide this work. They emerge from the critical need to address the potential quantum threat to telecommunications infrastructure while maintaining the current operational requirements of modern mobile networks.

- ❏ RQ-1: Is it technically feasible to integrate NIST-standardized post-quantum cryptographic algorithms into a 5G core via a sidecar proxy architecture, as measured by successful protocol completion (registration, discovery, and handshake transactions) and bounded, quantifiable overhead in latency, CPU, and memory relative to native and classical-TLS baselines?

- ❏ RQ-2: What is the quantitative impact on latency and throughput in intra-network function communications when comparing native baseline configurations, sidecar

proxies with classic TLS cryptography, and sidecar proxies with post-quantum cryptography in a 5G core environment?

- ❑ RQ-3: Is the sidecar proxy-based approach for PQC integration into the 5G core a viable, applicable solution for immediate deployment, considering performance trade-offs, architectural complexity, and transition requirements in contemporary telecommunications infrastructure?

1.3 Objectives

The overarching objective of this research is to design, implement, and empirically evaluate a sidecar proxy-based architecture for integrating NIST-standardized Post-Quantum Cryptography (PQC) algorithms into the 5GC, using the free5gc(free5GC Project, 2025a)-compose framework as a controlled experimental platform.

This work demonstrates the technical viability, performance characteristics, and operational feasibility of such an integration. It bridges the gap between theoretical cryptographic advances and practical deployment scenarios in 5G networks, ultimately assessing whether the sidecar proxy pattern is a viable pathway for immediate PQC adoption in production 5GC environments.

To achieve this overarching goal, the research is organized into four interconnected lines of work, each comprising specific objectives that collectively encompass the research questions.

- ❑ **Fundamental research and algorithm selection**: a systematic review of NIST-standardized post-quantum algorithms, focusing on implementation maturity, stability, and deployment readiness for high-availability telecommunications systems. This includes assessing compatibility with existing 5G core security protocols, namely TLS 1.3 and the 3rd Generation Partnership Project (3GPP)-defined service-based interfaces, establishing the basis for subsequent architectural decisions.

- ❑ **Architectural design and integration methodology**: a detailed specification for integrating post-quantum cryptography into the 5G core via sidecar proxy containers, intercepting inter-NF traffic transparently without modifying the NFs themselves. This covers the sidecar deployment pattern in free5gc-compose (free5GC Project, 2025b), secure proxy-to-NF channels, and compliance with 3GPP security requirements, accommodating both pure PQC and hybrid classical-quantum configurations.

- ❑ **Practical implementation and experimental deployment**: a fully functional prototype within free5gc-compose(free5GC Project, 2025b), incorporating NIST-standardized algorithms in operational sidecar proxies. The prototype supports

three comparative configurations—native, classical-TLS sidecar, and PQC sidecar—with instrumentation for metric collection and reproducible experimental conditions.

- **Empirical evaluation and comparative analysis**: systematic performance experiments measuring end-to-end latency, resource utilization, and stability under sustained load across all three configurations. The analysis determines whether the added complexity of PQC sidecar integration compromises minimum availability requirements, synthesizing these findings into an assessment of the approach's readiness for production deployment.

Through the systematic pursuit of these objectives, this research aims to provide empirical evidence on the applicability, performance implications, and operational feasibility of integrating sidecar-based PQC into 5G Core networks, directly addressing each of the established research questions and contributing reproducible experimental methodologies and validated architectural patterns to the scientific community.

1.4 Contributions

This dissertation offers significant scientific and technological contributions that collectively advance the field of quantum-computing-resistant telecommunications security. The scientific contributions of this work advance knowledge on the practical integration of quantum-resistant cryptographic mechanisms in fifth-generation (5G) core networks. Prior literature extensively addresses the theoretical foundations of Post-Quantum Cryptography (PQC) and the NIST standardization process, but a central gap remains: the near-total absence of empirical evidence on *retrofitting* PQC into already-deployed, unmodified 5G core network functions. This dissertation addresses that gap by demonstrating, through the sidecar proxy pattern, that PQC integration is architecturally testable and observable — its feasibility can be measured directly, through latency, resource consumption, and functional correctness, rather than argued theoretically.

The research contributes new empirical evidence on the performance implications of the NIST-standardized algorithms, specifically ML-KEM, ML-DSA, and Falcon, in the context of 5G signaling protocols. It further contributes to the scientific understanding of architectural patterns for cryptographic migration by evaluating the sidecar proxy pattern as a mechanism to retrofit cryptography onto telecommunications networks without deep changes to existing network function implementations.

From a technological perspective, this work contributes to the design and empirical validation of a PQC-enabled 5G Core prototype. It distinguishes itself by repurposing the sidecar proxy model, a staple of modern microservice management, into a robust mechanism for transparent cryptographic upgrading. This transition from traditional traffic

management to post-quantum security constitutes a significant architectural innovation within the 5G Service-Based Architecture (SBA).

Another significant technological contribution of this work involves the innovative use of the `socat` tool to solve the technical challenge of intercepting and encrypting Stream Control Transmission Protocol (SCTP)-based traffic between network functions in the `free5gc` (free5GC Project, 2025a) environment. SCTP presents unique challenges for proxy-based traffic interception due to its multi-stream capabilities and association-based connection model. The solution developed in this research uses `socat` as an intermediary process that transparently relays SCTP connections, enabling the injection of cryptographic processing via sidecar proxies. This approach provides a minimally invasive way to introduce quantum-resistant cryptography into existing NF implementations without requiring modifications to the underlying application code.

As an architectural contribution, this research empirically demonstrates that the sidecar proxy pattern effectively decouples cryptographic logic from functional network implementations, thereby enabling cryptographic agility. This decoupling allows network operators to upgrade to quantum-resistant algorithms without disrupting the operational continuity of existing services, representing a pragmatic migration path to protect telecommunications infrastructure against future quantum threats.

1.5 Thesis Organization

This dissertation is structured into five chapters. This first chapter introduces the work, establishes the research motivation, defines the specific Research Questions (RQ), and outlines the study's primary objectives and contributions. The other chapters are organized as follows:

- ❏ Chapter 2: Provides the theoretical foundation, covering the 5G Core, the Sidecar Pattern, and Post-Quantum Cryptography (PQC), while positioning the research within the current state of the art.
- ❏ Chapter 3: Empowering the Core 5G with PQC – Details the architectural heart of the thesis, explaining the sidecar approach, cryptographic processing layers, and the specific implementation within the free5GC framework.
- ❏ Chapter 4: Experimental Evaluation and Results – Describes the testbed configuration and experimental design, followed by a comprehensive analysis of latency and PQC performance metrics.
- ❏ Chapter 5: Summarizes the findings, lists the resulting publications, and proposes directions for future research.

Chapter 2

Background and Related Work

2.1 Background

2.1.1 5G CORE

The network communication 5GC, as defined by 3rd Generation Partnership Project (3GPP) in the TS 23.501 technical specification, represents a significant transformation in mobile network architecture, introducing a cloud-native and software-driven framework capable of supporting diverse service requirements from a unified infrastructure (BROWN et al., 2017) (LEI et al., 2021). This architectural evolution enables the simultaneous delivery of Enhanced Mobile Broadband (eMBB), Ultra-Reliable and Low Latency Communications (URLLC), and Massive Machine Type Communications (mMTC) through a shared physical infrastructure.

The 5GC is characterized by the adoption of the SBA paradigm, which moves away from the point-to-point interface model of previous network generations towards a distributed and loosely coupled architecture, in which network functions expose their capabilities as services through standardized Application Programming Interfaces (APIs) (BROWN et al., 2017). This approach employs HTTP/2 and REpresentational State Transfer Application Programming Interface (RESTful) principles, enabling the independent scaling, evolution, and deployment of network functions across a distributed cloud infrastructure. The NRF provides dynamic service discovery, enabling network functions to locate and communicate with their peers without static configuration (LEI et al., 2021)(TAID, 2021).

Network slicing (FOUKAS et al., 2017) is a transformative capability that creates multiple logical networks from shared physical infrastructure, each optimized for specific use cases or vertical sectors. This isolation ensures differentiated services while maximizing resource utilization across the physical infrastructure.

The technological foundations of Network Function Virtualization (NFV), Software-Defined Networking (SDN), and cloud-native principles provide the implementation basis

for 5GC, enabling the transition from hardware-dependent networks to software-driven systems that can dynamically adapt (ABDELWAHAB et al., 2016). As illustrated in Figure 1, SBA describes the interconnection of essential network functions through service-based interfaces, demonstrating the communication model between control plane and user plane components. In this way, the core architecture of 5G establishes a comprehensive framework that meets diverse service requirements, offers flexibility for continuous evolution through service-based design and network functions, and supports enabling technologies.

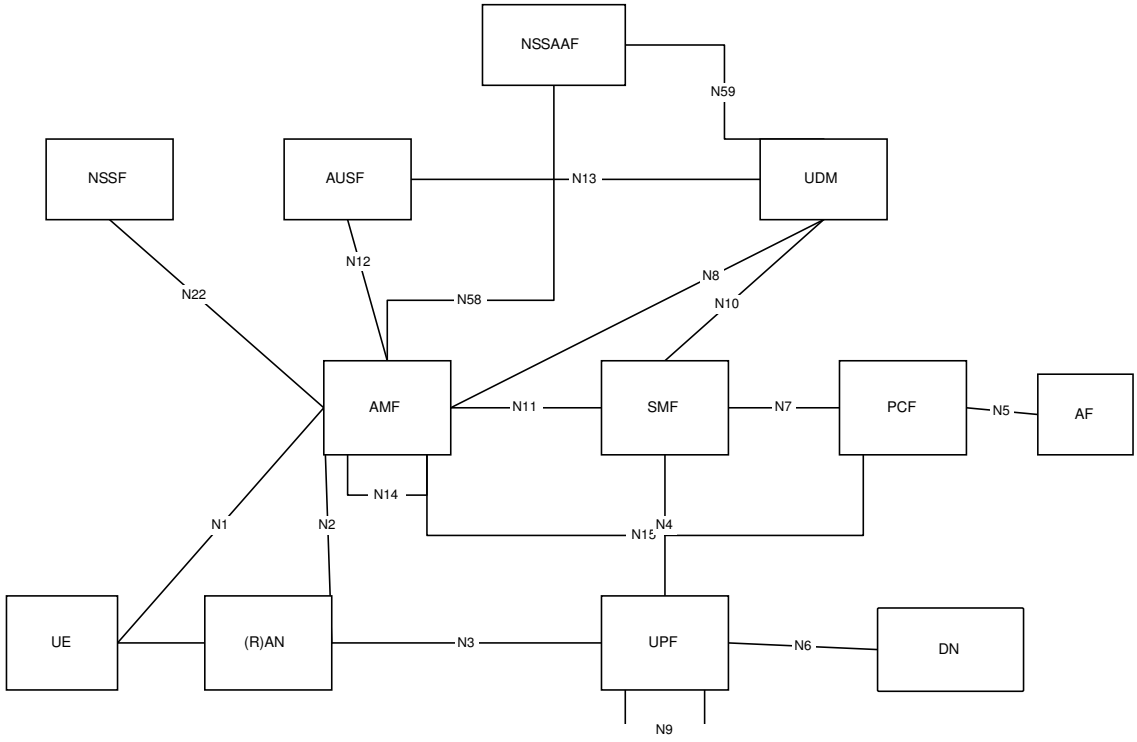

Figure 1 – 5G Core architecture adapted from Figure 4.2.3-2 of the 3GPP TS 23.501 (3rd Generation Partnership Project (3GPP), 2023)

The relevance of the 5GC architecture to this thesis lies in its SBA foundation: because NFs communicate exclusively through standardized HTTP/2-based SBIs, cryptographic protection can, in principle, be applied uniformly at the transport boundary rather than inside each NF. This uniformity is precisely what makes the sidecar-based PQC retrofitting strategy proposed in this dissertation viable, as it relies on a well-defined, protocol-consistent interception point between network functions.

2.1.2 Sidecar Pattern

The Sidecar pattern derives its name from the analogy to a motorcycle sidecar, in which a secondary accessory is attached to a main vehicle to extend its functionality without

modifying the main entity itself (BURNS; OPPENHEIMER, 2016)(BURNS; OPPENHEIMER, 2016). In the context of microservices architecture, the sidecar container operates as an independent process that coexists with the main application container, sharing the same host environment while maintaining functional independence in implementation and update cycles (SALCEDO-NAVARRO; GARCIA-PINEDA; GUTIÉRREZ-AGUADO, 2025).

The pattern was formally codified in the seminal work of Burns and Oppenheimer at Google, who identified it as one of the three main multi-container patterns observed in container-based distributed systems (BURNS; OPPENHEIMER, 2016). It is based on the principle of orthogonal separation of concerns, in which multi-functional infrastructure capabilities (such as logging, monitoring, networking, and security) are abstracted from the application's business logic and delegated to a dedicated sidecar component. This approach allows application developers to focus exclusively on core business logic while maintaining consistent infrastructure management across services and programming languages (BURNS; OPPENHEIMER, 2016).

The Sidecar architecture is characterized by a multi-container deployment model on a single host, in which the main application container and one or more sidecar containers run on the same host and share critical system resources. In container orchestration platforms like Kubernetes, this structural arrangement is observed through the Pod abstraction, which represents the smallest "reservable" unit and encapsulates one or more containers that share a network namespace, storage volumes, and execution context (BURNS; OPPENHEIMER, 2016)(SALCEDO-NAVARRO; GARCIA-PINEDA; GUTIÉRREZ-AGUADO, 2025). As illustrated in Figure 2, the sidecar container is deployed alongside the application container within the same Pod, enabling seamless interprocess communication via localhost loopback interfaces or shared volume mounts (BURNS; OPPENHEIMER, 2016). The shared lifecycle is the defining characteristic of the Sidecar pattern: the sidecar container and the main application container are instantiated, scaled, and terminated as a single atomic unit. This tight coupling of lifecycles ensures that infrastructure functions remain consistently available throughout the application's execution, enabling independent development and deployment of each component (BURNS; OPPENHEIMER, 2016). The container orchestration platform manages the scheduling of these co-located containers as a single unit, abstracting the complexity of managing multiple application developer processes.

Communication between the application container and the sidecar is implemented via local network interfaces, where both containers communicate through the localhost loopback address or via shared file system volumes (BURNS; OPPENHEIMER, 2016)(SAHU et al., 2025). This communication pattern allows the sidecar to intercept, modify, or augment requests directed to the application without requiring explicit modifications at the application level. For example, a service-mesh sidecar proxy can transparently

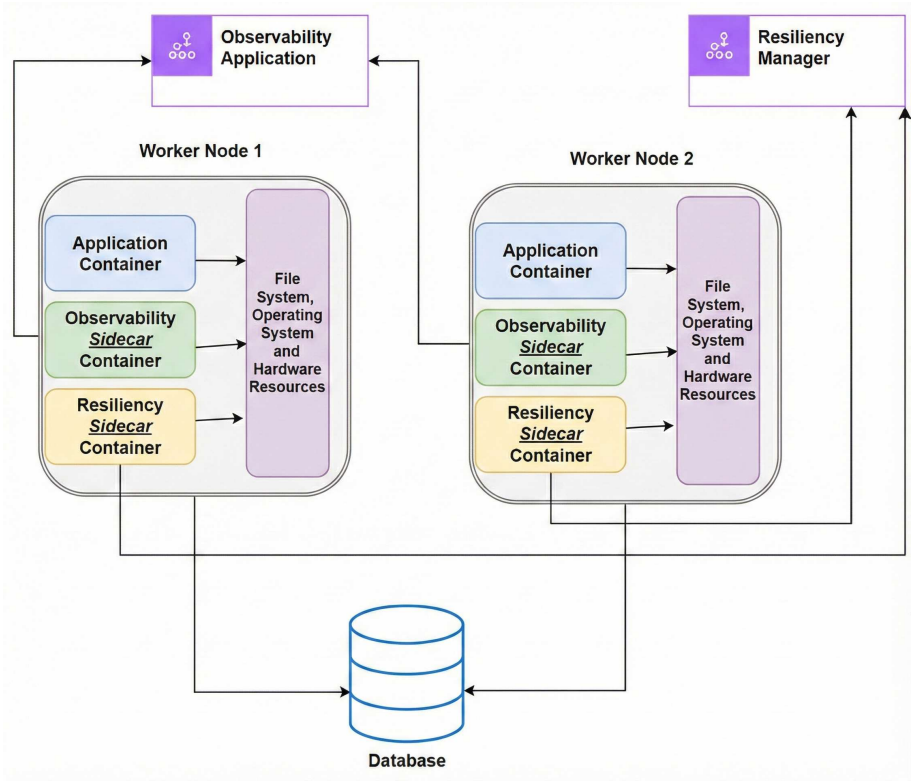

Figure 2 – Sidecar pattern Figure Sidecar-systems-architecture (PUNUGOTI RAJASRIKAR E MANAM, 2025)

handle Mutual Transport Layer Security (mTLS) termination, traffic routing, and load balancing, abstracting network complexity from the application container. (SAHU et al., 2025)(BURNS; OPPENHEIMER, 2016).

The main functional categories addressed by sidecar implementations encompass observability, traffic management, and security (BURNS; OPPENHEIMER, 2016). Observability sidecars implement logging, metrics collection, and distributed tracing agents that collect operational data without polluting the application code with infrastructure-specific implementations. Security-oriented sidecars handle cryptographic operations, identity management, and policy enforcement at the infrastructure layer. These sidecars can implement mutual termination of TLS (mTLS), certificate rotation, and network policy enforcement, thus centralizing security concerns while maintaining application-agnostic operation (BURNS; OPPENHEIMER, 2016). Architectural separation enables security policies to be updated and applied independently of application code, thereby significantly reducing the attack surface and simplifying compliance requirements.

The multi-language support enabled by the Sidecar pattern represents a particularly significant benefit for organizations that employ heterogeneous technology stacks in their microservices ecosystem. Since the sidecar is implemented as a standalone container, a single sidecar implementation can be reused across services written in different programming languages, without requiring language-specific libraries or modifications to the

application code (BURNS; OPPENHEIMER, 2016). This feature allows organizations to adopt microservices architectures that leverage the most appropriate technology for each service, while maintaining consistent infrastructure management across the system.

Despite its advantages, the Sidecar pattern has disadvantages that should be considered in architectural design decisions. Key concerns include resource overhead, increased startup latency, and configuration complexity (SALCEDO-NAVARRO; GARCIA-PINEDA; GUTIÉRREZ-AGUADO, 2025)(SAHU et al., 2025), as each sidecar container consumes extra CPU, memory, and storage resources, which can become significant in large-scale deployments with multiple microservices. Additionally, the shared lifecycle model implies that the total Pod startup time is determined by the slowest container to initialize, which can increase application deployment and scaling times.

Configuration complexity represents another significant consideration, particularly when multiple sidecars are deployed in a single Pod. Managing interactions among sidecar components, ensuring proper startup order, and debugging issues in multi-container environments require specialized operational knowledge.

The relevance of the sidecar pattern to this thesis is direct: it is the architectural mechanism that allows PQC to be retrofitted onto unmodified 5GC network functions, which is precisely the capability that RQ-1 and RQ-3 interrogate. The overhead and complexity trade-offs described above are also revisited empirically in Chapter 4, where they are measured rather than assumed.

2.1.3 Post-Quantum Cryptography (PQC)

The emergence of quantum computing represents an existential threat to contemporary cryptographic systems that underpin the global communications infrastructure. Classical cryptographic algorithms, such as RSA and ECC, derive their security from mathematical problems involving integer factorization and discrete logarithms, which remain computationally intractable for classical computers, but can be efficiently solved by quantum machines using Shor's algorithm (XIAO et al., 2022). Furthermore, Grover's algorithm provides a quadratic speedup for unstructured search problems, thereby halving the security level of symmetric-key algorithms such as Advanced Encryption Standard (AES). These vulnerabilities led the cryptographic community to initiate research and development of PQC, also called quantum-resistant or quantum-attack-secure cryptography, which refers to cryptographic algorithms that can be executed on classical computers while maintaining security against both classical and quantum adversaries (ALAGIC et al., 2025) (SABANI; SAVVAS; GARANI, 2024).

The origin of PQC dates back to the early 2000s, when researchers began systematically exploring mathematical structures they believed were resistant to quantum attacks. Unlike quantum key distribution, which requires specialized quantum hardware for communication, PQC represents a software-based solution that can be implemented

on existing classical computing infrastructure (XIAO et al., 2022). The urgency for standardization intensified following advances in quantum computing hardware, culminating in the NIST Post-Quantum Cryptography (PQC) Standardization Process, initiated in 2016 and finalized in August 2024.

Researchers and industry experts recognized the urgent need to develop cryptographic systems that could withstand classical and quantum computational attacks, which led to extensive research on mathematical problems considered resistant to quantum algorithms, particularly those based on lattice structures, error-correcting codes, and hash functions (XIAO et al., 2022).

The fundamental principle of perceptual quantum cryptography PQC involves replacing the number-theoretic problems used in classical public-key cryptosystems with mathematical constructs that are resistant to quantum attacks. Lattice-based cryptography has emerged as the predominant paradigm in PQC, deriving its security from the computational complexity of problems related to structured grids in high-dimensional spaces (SABANI; SAVVAS; GARANI, 2024) (SABANI; SAVVAS; GARANI, 2024) (WANG; XU; YU, 2023). The Learning with Errors (LWE) problem and its ring-based variant (Ring-LWE) constitute the fundamental mathematical problems underpinning many leading PQC candidates, including CRYSTALS-Kyber and CRYSTALS-Dilithium (SABANI; SAVVAS; GARANI, 2024)(SABANI; SAVVAS; GARANI, 2024).

Lattice-based cryptographic schemes exploit the complexity of problems such as the Shortest Vector Problem (SVP) and the Nearest Vector Problem (NVP), which remain computationally challenging even for quantum computers (SABANI; SAVVAS; GARANI, 2024). The addition of controlled errors (referred to as "learning from errors") to linear equations creates a problem structure that obscures the underlying mathematical relationships, thus providing security guarantees under reasonable assumptions (SABANI; SAVVAS; GARANI, 2024). This approach offers several advantages, including relatively compact key sizes, efficient computational performance, and versatility across multiple cryptographic primitives, such as key encapsulation and digital signatures (SABANI; SAVVAS; GARANI, 2024) (XIAO et al., 2022).

Additionally, the PQC portfolio encompasses cryptographic families beyond code-based cryptography (e.g., the McEliece cryptosystem), hash-based signatures (ZONG, 2025), and multivariate polynomial systems (XIAO et al., 2022). Each family offers distinct trade-offs among security assumptions, performance characteristics, and implementation complexity, enabling the construction of heterogeneous cryptographic ecosystems tailored to diverse application requirements.

The National Institute of Standards and Technology (NIST) initiated its Post-Quantum Cryptography (PQC) Process in 2016, marking the most comprehensive international effort to develop and standardize quantum-resistant cryptographic algorithms (SABANI; SAVVAS; GARANI, 2024). This process consisted of multiple stages and attracted sub-

missions from cryptography researchers worldwide, with initial candidates undergoing rigorous security analysis, performance evaluation, and implementation assessment (XIAO et al., 2022) (SABANI; SAVVAS; GARANI, 2024).

The evaluation criteria across the stages of this process emphasized resistance to classical and quantum attacks, as well as practical considerations such as key size, computational efficiency, and suitability for deployment in diverse computing environments (XIAO et al., 2022). The distinction between Key KEM for asymmetric cryptography and digital signature algorithms constituted a fundamental organizational principle throughout the standardization process (SABANI; SAVVAS; GARANI, 2024).

Following extensive cryptanalysis and community feedback, NIST announced its selection of finalist algorithms in 2022 and subsequently published standardized specifications in 2024 (ALAGIC et al., 2025)(SABANI; SAVVAS; GARANI, 2024). The first three finalized standards, designated Federal Information Processing Standards (FIPS) 203 (ML-KEM), FIPS 204 (ML-DSA), and FIPS 205 (SLHDSA), were published in August 2024, establishing the basis for the commercial deployment of quantum-resistant cryptography (ALAGIC et al., 2025)(SABANI; SAVVAS; GARANI, 2024).

The NIST standardization process resulted in the selection of algorithms from the CRYSTALS (Cryptographic Suite for Algebraic Lattices) family, alongside SPHINCS+ and FALCON, each serving distinct cryptographic purposes and each with a specific standardized identifier (ALAGIC et al., 2025)(SABANI; SAVVAS; GARANI, 2024).

For general encryption and key encapsulation, CRYSTALS-Kyber was standardized as the ML-KEM (Module-Lattice-Based Key-Encapsulation Mechanism), serving as the primary algorithm for protecting sensitive data transmissions against quantum attacks (ALAGIC et al., 2025)(SABANI; SAVVAS; GARANI, 2024). The "Module-Lattice" designation reflects the mathematical structure underlying the algorithm, wherein lattice problems are instantiated using module lattices to optimize efficiency while maintaining security.

For digital signatures, CRYSTALS-Dilithium was standardized as MLDSA (Module-Lattice-Based Digital Signature Algorithm), offering a balance of security, performance, and signature size suitable for most applications (ALAGIC et al., 2025) (SABANI; SAVVAS; GARANI, 2024). FALCON was standardized as FN DSA (Fast Fourier-NBDSA), providing a more compact signature alternative for bandwidth-constrained environments, though it requires a more complex implementation (SABANI; SAVVAS; GARANI, 2024). SPHINCS+ was standardized as SLH DSA (Stateless Hash-Based Digital Signature Algorithm), offering a conservative security foundation based solely on hash function assumptions, thereby providing a fallback option with minimal cryptographic assumptions (SABANI; SAVVAS; GARANI, 2024).

The relevance of PQC to this thesis follows directly from this standardization landscape: with ML-KEM and ML-DSA now finalized as FIPS standards, the open question

addressed by this dissertation is no longer which algorithms to adopt, but whether they can be practically retrofitted into an operational 5G core without disrupting existing network function implementations. The algorithm choices evaluated experimentally in Chapter 4 (ML-KEM-768, ML-DSA variants, and Falcon) are drawn directly from this standardized portfolio.

2.2 Related Work

Mahyoub et al. (MAHYOUB et al., 2024) present a security analysis of critical interfaces in the Fifth Generation architecture, identifying interfaces that exchange sensitive data or are externally exposed, and mapping threats and protections across Service-Based Architecture and non-Service-Based Architecture connectivity. It discusses defenses such as Transport Layer Security, Internet Protocol Security, and Open Authorization for Network Function interactions, and highlights post-quantum considerations for key exchange and tunnel establishment.

Mehic et al. (MEHIC et al., 2024) provide a comprehensive overview of quantum cryptography for Fifth Generation networks, covering Quantum Key Distribution deployment options, integration challenges, and standardization aspects in cellular architectures. It contrasts Quantum Key Distribution with PQC and discusses how quantum-resilient key establishment can complement mechanisms such as TLS and Internet Protocol Security (IPsec) in Fifth Generation systems.

Lawo et al. (LAWO et al., 2024) design and implement a Post Quantum Cryptography network stack on a Data Processing Unit platform, combining Falcon with Kyber and Dilithium with Kyber for authentication and key establishment. It benchmarks performance on the NVIDIA Data Processing Unit platform, reporting latency-throughput trade-offs for PQC-protected communication.

Scalise et al. (SCALISE et al., 2024) offer a systematic survey of security considerations in Fifth- and Sixth-Generation systems, organizing challenges and trends across confidentiality, integrity, and availability objectives, and introducing a Zero Trust Architecture perspective. It highlights emerging directions, including Post-Quantum Cryptography for key exchange, as well as the security implications of Software-Defined Networking, Network Function Virtualization, and Multi-access Edge Computing deployments.

Mangla et al. (MANGLA et al., 2023) review Fifth Generation security vulnerabilities in a future quantum computing environment and survey quantum-based solutions intended to mitigate these challenges, while motivating the transition toward Sixth Generation. It discusses Quantum Key Distribution and related quantum security techniques and argues that classical public-key cryptography must be complemented by quantum-resistant alternatives for next-generation industrial deployments.

Dolente et al. (DOLENTE; GARROPPO; PAGANO, 2024) conduct an experimental

vulnerability assessment of open-source Fifth Generation Core Network Function implementations, focusing on externally exposed Network Functions such as the Access and Mobility Management Function, the Network Repository Function, and the Network Exposure Function. It applies Application Programming Interface injection and fuzzing-style tests to Open5GS and OpenAirInterface, and reports robustness issues that can lead to denial-of-service attacks and other security risks.

In Table 1, 5G / SBA indicates an explicit focus on the 5G and SBI and TLS, IPsec, and Open Authorization (OAuth) indicate whether TLS, IPsec, and OAuth are discussed or used; "Exp." indicates studies that implemented practical trials; 5G Stack indicates use of an open-source Fifth Generation implementation or testbed; API/(Fuzzing or Fuzz testing) - Automated Software Testing Technique (Fuzz) indicates API injection or fuzzing-style testing; and PQC indicates explicit PQC discussion or implementation.

Table 1 – Related Work.

Paper	5GC /SBA	TLS	IPsec	OAuth	Exp.	5G Stack	API /Fuzz	PQC
Mahyoub et al. (MAHYOUB et al., 2024)	●	●	●	●	○	○	○	●
Mehic et al. (MEHIC et al., 2024)	●	●	●	○	○	○	○	●
Lawo et al. (LAWO et al., 2024)	○	○	○	○	○	○	○	●
Scalise et al. (SCALISE et al., 2024)	●	●	●	●	○	○	○	●
Mangla et al. (MANGLA et al., 2023)	○	●	●	○	○	○	○	●
Dolente et al. (DOLENTE; GARROPPO; PAGANO, 2024)	●	●	●	●	○	●	●	○

The data presented in Table 1 illustrate the current landscape of research on 5G security and PQC, highlighting several critical research gaps that this work seeks to address.

While most of the cited works explore the integration of PQC into traditional security protocols, such as TLS and IPsec, or into the SBA of the 5G core, there is a notable absence of empirical experimentation in these studies. On the other hand, the few articles that provide experimental results and in-depth analyses of the 5G Core or the use of APIs, such as (DOLENTE; GARROPPO; PAGANO, 2024) and (PELL et al., 2023), do not yet incorporate PQC solutions. These "implementation gaps," where the theoretical

benefits of post-quantum resilience have not been rigorously validated through practical experimentation in a full 5G environment, constitute the scope that this research's experimental methodology aims to fill.

Specifically, no work reviewed in Table 1 combines all four of the following elements: (i) an open-source, 3GPP-compliant 5G core as the experimental substrate; (ii) the sidecar proxy pattern as the integration mechanism; (iii) actual PQC retrofitting applied to unmodified NFs, rather than a theoretical or protocol-level discussion; and (iv) systematic, reproducible instrumentation of the resulting latency and resource overhead. This four-way combination is what distinguishes the present work from the related literature and defines the specific scope of this dissertation.

Chapter 3

Empowering the Core 5G with PQC

The integration of PQC into existing telecommunications infrastructure represents one of the most significant challenges in contemporary network security research. The 5G core network, with its service-based architecture and distributed network-function topology, presents unique challenges for PQC integration that differ substantially from those in traditional cryptographic migration scenarios. This session provides a comprehensive conceptual explanation of how the sidecar proxy pattern was applied to enable PQC protection in the free5gc-compose implementation of the 5G Core (5GC) network, without requiring modifications to the existing network function implementations.

3.1 The Sidecar Approach

The sidecar proxy pattern (MAIA; CORREIA, 2022), originally developed within the microservices architecture paradigm, provides an elegant solution for extending functionality across distributed systems without invasive code changes. According to research on service mesh architectures, the sidecar pattern was formalized as one of six core patterns identified in contemporary service mesh implementations, serving as the foundation for data plane communication in distributed systems (CALCOTE; JACKSON, 2023). This deployment model ensures consistent network behavior across all services without requiring modifications to individual service implementations. Research comparing service mesh implementations has identified the sidecar pattern as a commonly used architecture model that enables sophisticated traffic management and security capabilities (SAOKAR et al., 2023).

From a functional perspective, when a sidecar proxy receives a request originating from a microservice, it can perform a variety of operations, including service discovery, security enforcement, load balancing, and metrics collection (ASHOK; GODFREY; MITTAL, 2021). In the context of 5GC networks, this standard allows for the transparent insertion of cryptographic processing between network functions, creating a security layer that operates independently of the central network function logic, thus maintaining back-

ward compatibility and the integrity of 3GPP-compatible standards while offering robust protection against quantum attacks for communications between network functions.

The application of the sidecar proxy pattern in the free5gc-compose architecture, with support for post-quantum cryptography, provides a practical, functional approach to securing 5GC network communications. Rather than modifying the source code of the network functions themselves to incorporate PQC processing, the architecture deploys proxy containers alongside each network function that intercept all inter-network-function traffic. These proxies perform the cryptographic operations required for quantum-resistant protection while maintaining complete transparency to the protected network functions.

3.1.1 Service-Based Architecture Fundamentals

The conceptual architecture follows the same principles as service mesh implementations (CHANDRAMOULI; BUTCHER; CALLAGHAN, 2024), in which sidecar proxies form a distributed data plane that handles all network communications on behalf of the services they accompany. Each NF in the 5GC is associated with a pair of proxies (see Figure 3): a server-side proxy that receives requests destined for the network function, and client-side proxies that intercept requests originating from it. This bidirectional proxy deployment ensures comprehensive coverage of all SBI communications.

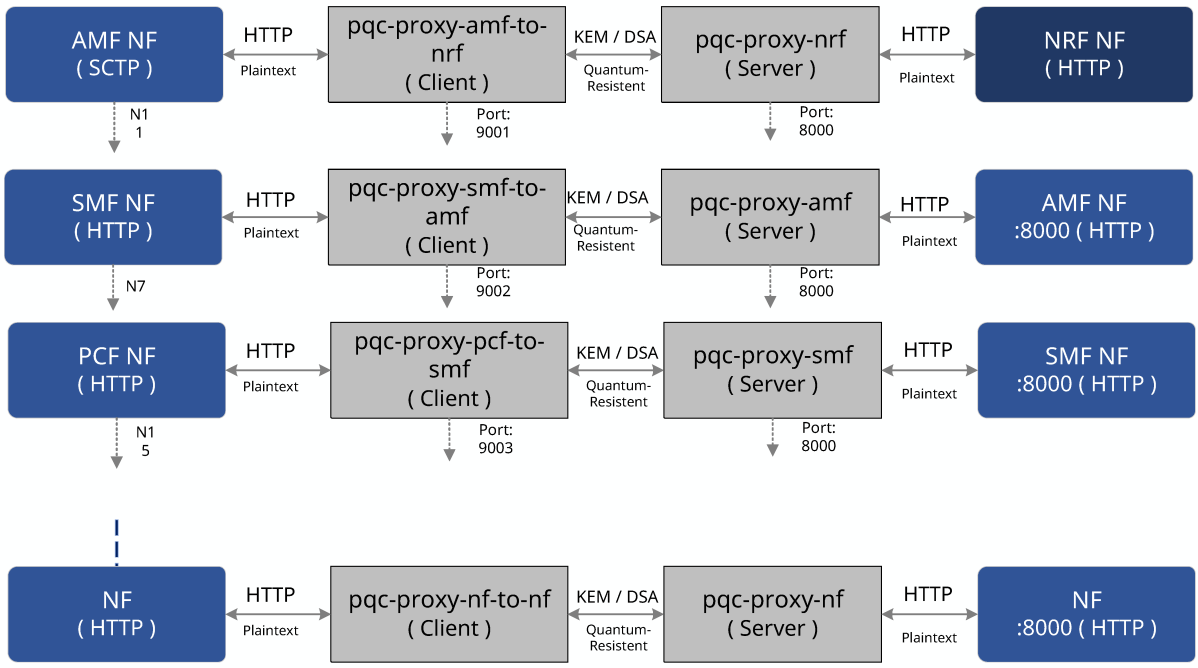

Figure 3 – PQC Secure 5G Architecture (partial view)

In this research, the `free5gc` project (free5GC Project, 2025a) was chosen, as it provides an open-source implementation of the 3GPP 5G core specifications, offering a platform for research and development that closely follows the standards established

by the 3GPP (3rd Generation Partnership Project (3GPP), 2023). The platform implements all major NFs specified for the 5G core, including the AMF, Session Management Function (SMF), NRF, UDM, AUSF, Policy Control Function (PCF), and others. Each of these functions communicates with its peers through SBIs that utilize HTTP/2 for request-response interactions. Alternative open-source cores, notably Open5GS (Open5GS Project, 2025) and OpenAirInterface (OpenAirInterface Software Alliance, 2025), were considered but not selected as the primary experimental platform: free5gc-compose (free5GC Project, 2025b) provides a ready-made, Docker Compose-based deployment of the full NF set used in this study, which reduced integration effort for the sidecar proxy layer, and it is the platform with which the authors had prior operational experience, reducing the risk of confounding integration defects with the PQC measurements themselves.

Comparative analyses of open-source 5G core platforms have demonstrated that deploying free5gc (free5GC Project, 2025a) and similar implementations entails complex setup procedures that require extensive expertise in network architecture, virtualization, and 3GPP-compliant configurations (3rd Generation Partnership Project (3GPP), 2023). The platform has been used extensively in academic research and industry evaluations, providing a reliable foundation for investigating novel security mechanisms without the complexity of commercial equipment deployments.

3.1.2 Inter-NF Communication Patterns

Communication between NFs in the 5G SBA follows a client-server model, in which each NF can operate as both a service consumer and a service provider. The NRF plays a central role in this architecture, maintaining a registry of available NFs and their capabilities, which enables dynamic service discovery. When an NF requires services from another function, it queries the NRF to discover available providers and then establishes direct communication with the selected instance.

3GPP defines two communication models for the SBA: a *direct* model, in which each NF queries the NRF and communicates with its peers directly, and an *indirect* model, in which a Service Communication Proxy (SCP) mediates discovery and routing on behalf of the NFs, which need not be aware of peer instance addresses. This work adopts the direct communication model, consistent with the `free5gc-compose` deployment used as the experimental platform; the sidecar proxy pair intercepts NF-to-NF traffic in the same way regardless of which model is used, so extending the architecture to an SCP-mediated indirect deployment is expected to require only reconfiguration of proxy target addresses, not a change to the cryptographic processing layer itself.

The Service-Based Interface utilizes HTTP/2 as its transport protocol, carrying JavaScript Object Notation (JSON)-encoded message bodies that conform to 3GPP-defined schemas. This approach provides several benefits, including widespread tool support, human-readable

message formats, and efficient multiplexing of multiple streams over single connections. However, it also means that all inter-NF communications are potentially vulnerable to the same cryptographic attacks that threaten traditional web communications.

The application of the sidecar proxy pattern in the free5gc-compose (free5GC Project, 2025b) implementation with PQC support represents a novel approach to securing 5G core network communications. Rather than modifying the source code of the NFs themselves to incorporate PQC processing, the architecture deploys proxy containers alongside each NF that intercept all inter-NF traffic. These proxies perform the cryptographic operations required for quantum-resistant protection while maintaining complete transparency to the protected Network Function (NF).

The proxy operates at the application layer, processing HTTP/2 requests and responses as they flow between network functions. When a request passes through a proxy, the proxy performs the necessary PQC operations before forwarding it to its destination. Similarly, responses returning through the proxy undergo cryptographic verification and processing. This approach enables end-to-end quantum-resistant protection for all inter-network-function communications without requiring any changes to the network function implementations.

3.2 Cryptographic Processing Layer

The cryptographic processing within each proxy is performed by integrating with a dedicated cryptographic services layer that provides post-quantum operations. This layer utilizes the liboqs (Open Quantum Safe Project, 2025) library, version 0.15.0, from the Open Quantum Safe (OQS) project, which provides reference implementations of NIST-standardized post-quantum algorithms. The architecture employs CRYSTALS-Kyber (specifically ML-KEM-768) for key encapsulation and CRYSTALS-Dilithium (specifically ML-DSA and Fancon) for digital signatures, providing quantum-resistant security levels equivalent to AES-192 and AES-256, respectively.

Separating cryptographic processing into dedicated wrapper services provides several architectural benefits. First, it isolates the complexity of PQC integration into specialized components that can be independently developed, tested, and updated. Second, it enables centralized key management, as all cryptographic operations access keys through standardized interfaces rather than maintaining separate key stores within each proxy. Third, it provides a clear abstraction boundary that facilitates algorithm substitution as PQC standards continue to evolve.

The key generation process produces cryptographic key pairs, which are distributed to all proxies that require them for communication protection. These keys are stored in shared volumes accessible to both proxies and cryptographic wrapper services, enabling consistent key access across the distributed deployment. The architecture supports key

rotation procedures that enable periodic key updates without service interruption, an operational requirement for long-running telecommunications infrastructure.

3.2.1 Traffic Interception and Processing Flow

The traffic interception mechanism operates through explicit configuration of proxy listen addresses and target addresses within the `docker-compose` deployment. Each proxy is configured to listen on a specific network address and port, while the target address specifies the ultimate destination for forwarded requests. NFs are configured to communicate with proxy addresses rather than directly with peer NFs, ensuring that all traffic passes through the cryptographic processing layer. This includes discovery traffic directed at the NRF: as shown in Table 2, each consumer NF (e.g., the AMF) is configured with a dedicated client-side proxy toward the NRF (`pqc-proxy-nrf`), so discovery queries are protected by the same mechanism as any other SBI exchange; no pre-established, unprotected discovery phase is assumed.

When an NF initiates communication with a peer function, the request is directed to the appropriate client-side proxy. The proxy receives the request, processes it through the PQC cryptographic layer (which includes key encapsulation and digital signature operations), and forwards the protected request to the corresponding server-side proxy. The server-side proxy receives the protected request, performs the complementary cryptographic operations (decapsulation and signature verification), and forwards the original request to the destination NF.

This bidirectional processing flow ensures that both the request and response paths are protected by quantum-resistant cryptography. The processing is entirely transparent to the NFs, which receives and sends unmodified HTTP/2 messages as they would in a conventional deployment. The only observable difference is the additional latency introduced by the cryptographic processing, which can be measured and characterized to assess the practical impact of PQC integration.

Explanation of the sequence diagram (part 1)
The sequence diagram presented in Figure 4 illustrates the communication flow between NF 1 (NF1) acting as a client and the Network Repository Function (NRF) within a 5G Core architecture, mediated by PQC-enabled sidecar proxies. This diagram demonstrates how post-quantum cryptographic protection is transparently applied to inter-network function communications without requiring modifications to the core network functions themselves.

The process begins when NF1, acting as a client, issues an HTTP/2 request to the NRF. Rather than communicating directly, the traffic is intercepted by the Proxy Client sidecar (NF1 $\rightarrow$ NRF), which serves as the cryptographic intermediary for outbound communications from NF1. In Step 1, the HTTP/2 Request is forwarded from the Proxy

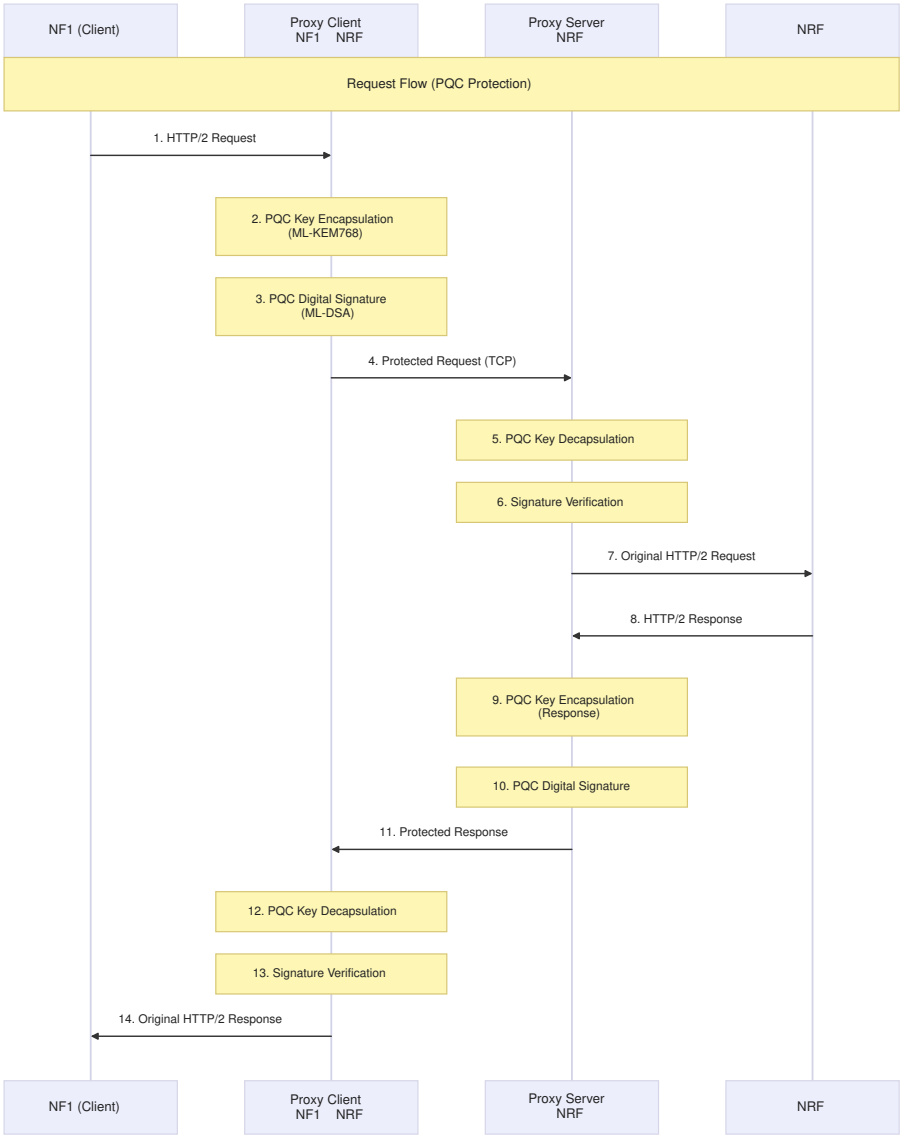

Figure 4 – PQC Secure 5G Architecture Sequence Diagram 1

Client to the Proxy Server sidecar (NRF), establishing the initial communication channel between the two proxy instances.

Following the initial request, Step 2 initiates the post-quantum cryptographic key establishment through PQC Key Encapsulation using ML-KEM-768. This step involves executing the key encapsulation mechanism, in which the Proxy Client generates ephemeral key material to establish a shared secret with the Proxy Server. The D-768 algorithm, standardized by NIST, provides quantum-resistant key establishment through lattice-based cryptography, replacing or supplementing classical Diffie-Hellman key exchange.

In Step 3, the Proxy Client transmits a PQC Digital Signature using ML-DSA to authenticate its identity to the Proxy Server. This digital signature, generated using the ML-DSA, ensures the authenticity and non-repudiation of the communicating party,

providing quantum-resistant authentication that withstands attacks from both classical and quantum adversaries. The signature covers the previously exchanged key material and relevant context to prevent replay attacks and ensure session integrity.

Once cryptographic authentication and key establishment are complete, Step 4 transmits the Protected Request over Transmission Control Protocol (TCP). At this stage, the original HTTP/2 request from NF1 has been encapsulated within the secure channel established through the preceding PQC handshake. The request travels through the protected tunnel between the Proxy Client and the Proxy Server, ensuring confidentiality and integrity via the shared secret derived from the ML-KEM-768 encapsulation.

Upon receiving the protected request, the Proxy Server initiates Step 5, PQC Key Decapsulation, to recover the shared secret from the encapsulated key material received in Step 2. This decapsulation process uses the Proxy Server's private key to decrypt the encapsulated secret, completing the key agreement protocol and enabling the server-side proxy to derive identical session keys for cryptographic operations.

Step 6 involves Signature Verification, where the Proxy Server validates the ML-DSA signature received in Step 3 against the Proxy Client's public key. This verification confirms the requesting party's authenticity and ensures that the key establishment process has not been tampered with by malicious intermediaries. Successful verification establishes trust in the client identity and the integrity of the negotiated session parameters.

With the secure channel fully established and authenticated, Step 7 involves transmitting the Original HTTP/2 Request from the Proxy Server to the NRF. At this point, the protective encapsulation is removed, and the original request is presented to the destination network function in its native format, preserving complete transparency from the NRF's perspective.

The NRF processes the request and generates an HTTP/2 Response, which is returned to the Proxy Server in Step 8. This response travels in cleartext between the NRF and its local sidecar proxy, as the protection is applied at the inter-network function boundary rather than within the network function itself.

Step 9 initiates response protection using PQC- based Key Encapsulation for the return path. The Proxy Server generates new encapsulated key material to establish a fresh shared secret for the response communication, ensuring that both directions of communication benefit from independent keying material and forward secrecy.

In Step 10, the Proxy Server transmits a PQC Digital Signature to authenticate the response's origin to the Proxy Client. This signature ensures that the client-side proxy can verify the authenticity of the responding NRF-side infrastructure, maintaining mutual authentication throughout the bidirectional communication flow.

Step 11 transmits the Protected Response from the Proxy Server back to the Proxy Client, encapsulating the HTTP/2 response within the quantum-resistant protective channel. This protected transmission completes the secure transaction cycle, allowing NF1 to

receive the data as if it had been sent through a standard, albeit non-quantum-resistant, channel.

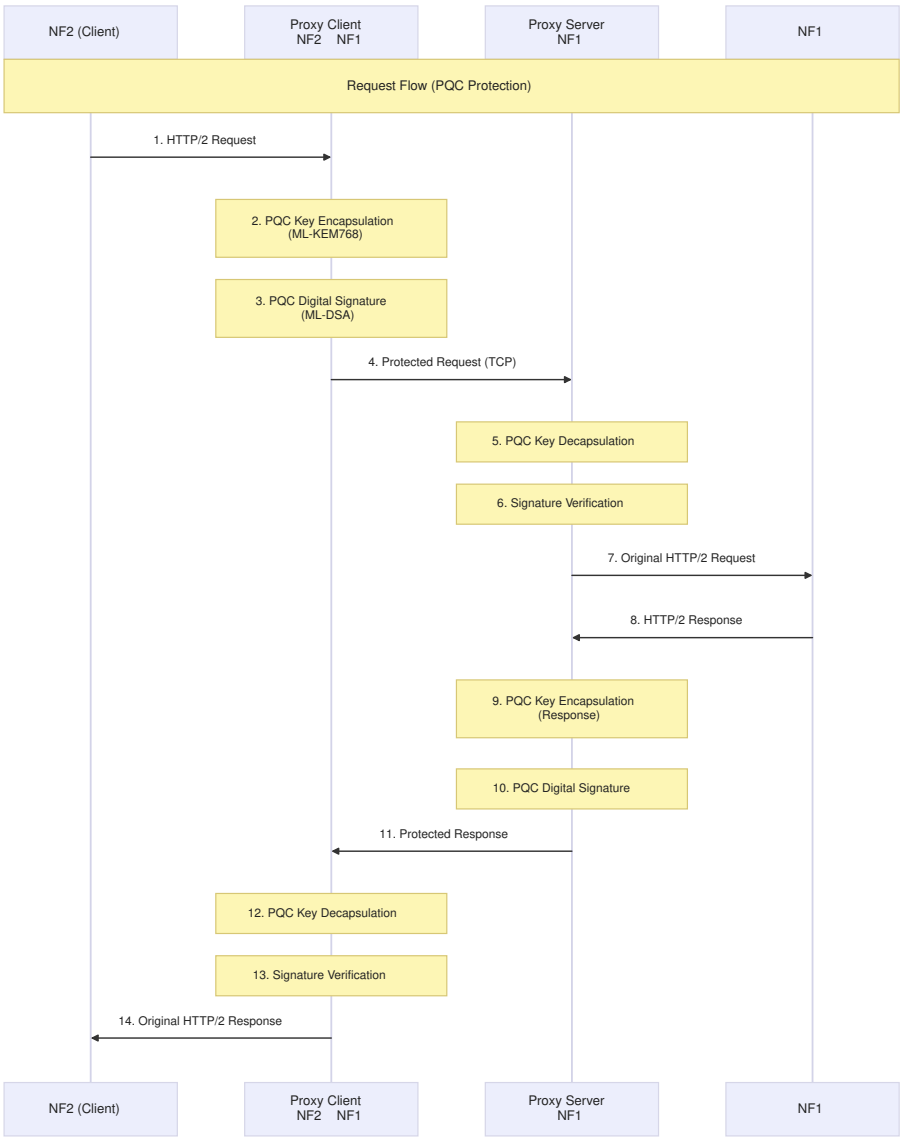

Figure 5 – PQC Secure 5G Architecture Sequence Diagram 2

Explanation of the sequence diagram (part 2)

The sequence diagram presented in Figure 5 illustrates the communication flow between NF 1 (NF1) acting as a client and the NRF within a 5G Core architecture, mediated by PQC-enabled sidecar proxies. This diagram demonstrates how PQC protection is transparently applied to inter-NF communications without requiring modifications to the core NFs themselves.

The process begins when NF1, acting as a client, issues an HTTP/2 request to the NRF. Rather than communicating directly, the traffic is intercepted by the Proxy Client sidecar (NF1 $\rightarrow$ NRF), which serves as the cryptographic intermediary for outbound

communications from NF1. In Step 1, the HTTP/2 Request is forwarded from the Proxy Client to the Proxy Server sidecar (NRF), establishing the initial communication channel between the two proxy instances.

Following the initial request, Step 2 initiates the PQC key establishment through PQC Key Encapsulation using ML-KEM-768. This step represents the execution of the key encapsulation mechanism, in which the Proxy Client generates ephemeral key material to establish a shared secret with the Proxy Server. The ML-KEM-768 algorithm, standardized by NIST in August 2024, provides quantum-resistant key establishment through lattice-based cryptography, replacing or supplementing classical Diffie-Hellman key exchange.

In Step 3, the Proxy Client transmits a PQC Digital Signature using ML-DSA to authenticate its identity to the Proxy Server. This digital signature ensures the authenticity and non-repudiation of the communicating party, providing quantum-resistant authentication that withstands attacks from both classical and quantum adversaries. The signature covers the previously exchanged key material and relevant context to prevent replay attacks and ensure session integrity.

Once cryptographic authentication and key establishment are complete, Step 4 transmits the Protected Request over TCP. At this stage, the original HTTP/2 request from NF1 has been encapsulated within the secure channel established through the preceding PQC handshake. The request travels through the protected tunnel between the Proxy Client and the Proxy Server, ensuring confidentiality and integrity via the shared secret derived from the ML-KEM-768 encapsulation.

Upon receiving the protected request, the Proxy Server initiates Step 5, PQC Key Decapsulation, to recover the shared secret from the encapsulated key material received in Step 2. This decapsulation process uses the Proxy Server's private key to decrypt the encapsulated secret, completing the key agreement protocol and enabling the server-side proxy to derive identical session keys for cryptographic operations.

Step 6 involves Signature Verification, where the Proxy Server validates the ML-DSA signature received in Step 3 against the Proxy Client's public key. This verification confirms the requesting party's authenticity and ensures that the key establishment process has not been tampered with by malicious intermediaries. Successful verification establishes trust in the client identity and the integrity of the negotiated session parameters.

With the secure channel fully established and authenticated, Step 7 involves transmitting the Original HTTP/2 Request from the Proxy Server to the NRF. At this point, the protective encapsulation is removed, and the original request is presented to the destination NF in its native format, preserving complete transparency from the NRF's perspective.

The NRF processes the request and generates an HTTP/2 Response, which is transmitted back to the Proxy Server in Step 8. This response travels in cleartext between the

NRF and its local sidecar proxy, as the protection is applied at the inter-NF boundary rather than within the NF itself.

Step 9 initiates response protection using PQC-based Key Encapsulation for the return path. The Proxy Server generates new encapsulated key material to establish a fresh shared secret for the response communication, ensuring that both directions of communication benefit from independent keying material and forward secrecy.

In Step 10, the Proxy Server transmits a PQC Digital Signature to authenticate the response's origin to the Proxy Client. This signature ensures that the client-side proxy can verify the authenticity of the responding NRF-side infrastructure, maintaining mutual authentication throughout the bidirectional communication flow.

Step 11 transmits the Protected Response from the Proxy Server back to the Proxy Client, encapsulating the HTTP/2 response within the quantum-resistant protective channel. This protected transmission ensures that sensitive information contained in the NRF response remains confidential against eavesdropping by quantum-capable adversaries.

Upon receiving the protected response, the Proxy Client performs the necessary decryption and verification operations to recover the original response content. Finally, Step 12 represents the delivery of the Original Response to NF1, completing the full request-response cycle. A NF original recebe a resposta da NRF em seu formato nativo, totalmente alheia às transformações criptográficas realizadas pelos proxies sidecar intervenientes.

3.3 The N2 Interface Challenge

The 5G architecture employs distinct protocol stacks for different interface categories, creating inherent complexity when applying uniform cryptographic protection via a single proxy architecture. The Service-Based Interface (SBI), which handles all inter-network-function communications within the 5G core, utilizes HTTP/2 over TCP as its transport protocol. This design choice enables the RESTful communication patterns that characterize the SBA, facilitating interoperability, monitoring, and troubleshooting through standard web technologies. All NFs within the 5G core communicate with their peers via these SBI interfaces, using JSON-encoded message bodies over HTTP/2 connections.

In contrast, the N2 interface, which handles control plane signaling between the gNodeB (gNB) in the radio access network and the AMF in the core network, operates on fundamentally different protocol foundations. The N2 interface uses SCTP as its transport-layer protocol, a choice motivated by SCTP's inherent reliability and support for multi-streaming, which align well with the multiplexing requirements of radio control signaling. SCTP provides ordered delivery of messages within streams while maintaining independence between streams, characteristics that are particularly valuable for managing the multiple concurrent signaling procedures that occur during typical mobile network

operations.

When the sidecar proxy architecture was deployed to protect SBI communications, a significant problem emerged: the proxy components were architecturally designed to process TCP-based HTTP/2 traffic rather than SCTP datagrams. The proxies implemented within the architecture are expected to handle request-response patterns characteristic of HTTP/2, including connection pooling, stream multiplexing, and message framing specific to TCP-based communications.

TCP and SCTP differ in two respects that shape their behavior under loss and failure. TCP delivers data as a single ordered byte stream, so a lost segment blocks all subsequent segments until it is retransmitted (Fig. 6A). SCTP instead partitions data into independent streams within a single association: a loss in one stream stalls only that stream, while the others continue delivering data unaffected (Fig. 6B).

The two protocols also differ in path resilience. A TCP connection is bound to a single Internet Protocol (IP) address pair, so a failure anywhere along that path terminates the connection (Fig. 7A). SCTP endpoints may instead bind multiple IP addresses and negotiate a primary and backup path; if the primary path fails, traffic is automatically redirected to the backup path and the association survives (Fig. 7B).

3.3.1 Conceptual Analysis of the Protocol Mismatch

The SCTP differs from TCP in several characteristics that affected the proxy implementation approach. While both are transport-layer protocols that provide reliable data delivery, SCTP introduces the concepts of streams and multi-homing, which are not present in traditional TCP connections. SCTP associations can involve multiple IP addresses on each endpoint, providing inherent redundancy and failover capabilities that are valuable in telecommunications deployments where network reliability is paramount. The message-oriented nature of SCTP, where data is organized into discrete messages rather than a continuous byte stream, also differs from TCP's stream-oriented semantics. Notably, SCTP's ordered-delivery guarantee applies *per stream*, not across the association as a whole, which is what allows independent signaling procedures to progress without head-of-line blocking one another.

The SCTP differs from TCP in several characteristics that affected the proxy implementation approach. While both are transport-layer protocols that provide reliable data delivery, SCTP introduces the concepts of streams and multi-homing, which are not present in traditional TCP connections. SCTP associations can involve multiple IP addresses on each endpoint, providing inherent redundancy and failover capabilities that are valuable in telecommunications deployments where network reliability is paramount. The message-oriented nature of SCTP, where data is organized into discrete messages rather than a continuous byte stream, also differs from TCP's stream-oriented semantics. Notably, SCTP's ordered-delivery guarantee applies *per stream*, not across the association

Figure 6 – SCTP vs TCP - Streams head of line blocking

as a whole, which is what allows independent signaling procedures to progress without head-of-line blocking one another.

The proxy implementation within the PQC-protected architecture expects to interact with connections at the TCP level, managing socket states, handling TCP-specific flow control mechanisms, and processing HTTP/2 frame structures that are built atop TCP's byte-stream abstraction. SCTP's stream identifiers and per-stream ordered delivery guarantees do not have direct equivalents in the proxy's processing logic, resulting in a fundamental incompatibility between the proxy's designed behavior and the N2 interface requirements. Attempts to directly apply the same proxy architecture used for SBI communications to N2 traffic would result in protocol parsing errors, connection failures, and a complete breakdown of communication between the radio access network and the core NFs.

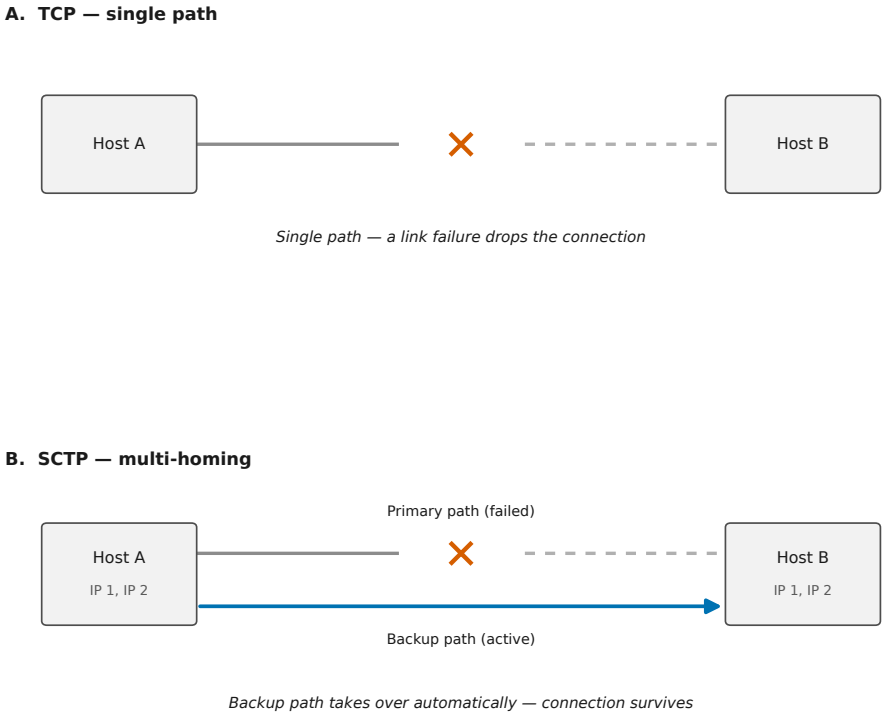

Figure 7 – SCTP vs TCP - Multihoming path redundancy

Furthermore, the `Docker` container networking environment presented additional constraints. `Docker`'s native networking support is optimized for TCP and User Datagram Protocol (UDP) traffic; SCTP is not one of the protocols its bridge networking natively forwards, and enabling it requires explicit kernel module support that may not be consistently available across all deployment environments. This is a protocol- and kernel-support limitation rather than a routing problem: the constraint is which transport protocols the container networking stack understands, not how packets are directed once accepted, and it directly motivated the protocol-translation approach adopted below.

3.3.2 Protocol Translation with socat

The architectural solution implemented to address the N2 interface challenge employs a protocol translation bridge using the `socat` utility (RIEGER, 2025), which functions as a bidirectional data relay between two network endpoints. The bridge architecture creates a mediation layer that converts the gNB's SCTP traffic into TCP traffic that can be processed by the PQC-protected proxy layer and, conversely, converts the proxy layer's TCP traffic back to SCTP for delivery to the AMF.

Using this approach, the gNB (either a physical appliance or the `UERANSIM` simulator for testing purposes) establishes an SCTP association with a designated port on the host system. The `socat` bridge component listens on this SCTP port and receives SCTP datagrams. Upon receiving them, the bridge translates the SCTP traffic into a single TCP

byte stream and forwards the converted data to the appropriate PQC proxy component. This translation is a functional simplification rather than a full multi-stream-preserving TCP implementation: the bridge does not maintain separate TCP streams per SCTP stream identifier, so SCTP's per-stream ordering and message-boundary framing are not preserved across the translation, and only a single logical channel between the gNB and the AMF is bridged at a time. This is an accepted functional limitation of the current prototype, discussed further in Chapter 4.

The proxy processes TCP traffic through the cryptographic layer, applying protection against quantum attacks, and forwards the protected data to the NF destination. For return traffic, the process is reversed: the AMF sends responses through its client-side proxy, which are forwarded via TCP to the bridge, which then translates the TCP stream back into SCTP format for delivery to the gNB.

This bridge architecture introduces additional latency and processing overhead, as each SCTP-to-TCP conversion requires buffering, protocol parsing, and message reconstruction in the alternate format. However, the impact on overall system performance remains manageable, as the N2 interface handles only control-plane signaling, not high-volume user data traffic. This work does not experimentally separate the `socat` translation cost from the PQC cryptographic cost within the N2 path; this limitation is discussed in Section 4.2.1 and identified as future work.

3.3.3 Flow Intra-NF communication via PQC Sidecar

The following figures illustrate the communication flows between the 5GC and the NF, mediated by PQC sidecar proxies. Figure 4 describes the interaction between an NF client and the NRF, representing a fundamental service discovery scenario for 5G SBA, while Figure 5 shows the communication flow between two NF peers, demonstrating direct service communication between NF.

The resulting N2-domain deployment, including the `socat` bridges and the PQC proxies flanking them, is illustrated in Figure 8.

This bridge architecture introduces additional latency and processing overhead, as each SCTP-to-TCP conversion requires buffering, protocol parsing, and message reconstruction in the ac TCP format. However, the impact on overall system performance remains manageable because the N2 interface handles only control plane signatures, not high-volume user data traffic. With this approach, the PQC proxy components designated for N2 communication (pqc-proxy-gnb and pqc-proxy-amf-n2) are configured to operate over the translated TCP protocol, preserving cryptographic protection and accommodating the protocol translation performed by the socat (RIEGER, 2025) bridges.

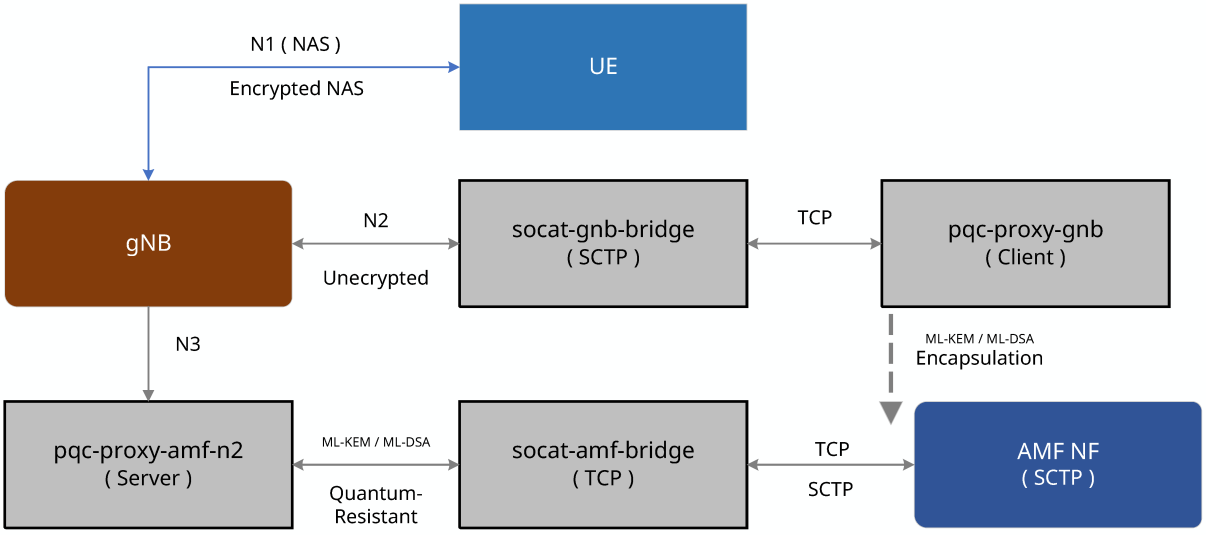

Figure 8 – PQC RAN domain (partial view).

3.4 Threat Model and Security Goals

The threat model underlying this research addresses the evolving security landscape in which 5G core networks must operate, specifically considering the emergence of quantum computing capabilities that fundamentally challenge the cryptographic protections currently deployed in telecommunications infrastructure. This section delineates the adversary capabilities, attack vectors, and system components considered in the design and evaluation of the proposed PQC-enabled architecture.

The primary threat factor in this model is the potential development of cryptographically relevant quantum computers capable of executing Shor's algorithm to solve the integer factorization and discrete logarithm problems that underlie RSA cryptography and the elliptic curve problem, respectively. Although such computers do not yet exist commercially, the concept of "collect now, decrypt later" attacks recognizes that attackers with robust capabilities and resources may already be collecting and storing encrypted communications, expecting them to become decryptable as quantum computing capabilities advance. This temporal dimension is particularly relevant for 5G networks, which are designed to support mission-critical applications with operational lifecycles spanning decades. Data transmitted today on 5G networks may remain sensitive for thirty years or more, creating a substantial window of vulnerability should quantum-capable adversaries emerge during the network's lifespan.

The threat model considers a passive adversary capable of recording encrypted traffic for later decryption in a HNDL scenario, as well as an active adversary performing Man-in-the-Middle (MITM) attacks to intercept, replay, or modify signaling data. We assume that the NFs and their host environments are trusted, while the underlying transport

network is inherently insecure. This is a deliberate scoping decision: the architecture protects the confidentiality and integrity of data in transit, but it does not defend against a compromised network function or host, for instance a supply-chain attack on a container image or a container-escape exploit. In such cases, an attacker with code execution inside the trust boundary could access key material or plaintext regardless of the cryptographic protocol used on the wire. This limits the practical security value of the proposed architecture in operational deployments where endpoint integrity cannot be independently guaranteed, and motivates future integration with attestation or confidential-computing mechanisms (e.g., Trusted Execution Environments (TEEs)) as a complementary control rather than a substitute for the sidecar-based protection studied here.

The attack surface within the 5G core architecture encompasses multiple interfaces and communication paths that require protection. The SBI communications between NFs represent a primary attack vector, as these HTTP/2-based communications carry sensitive control-plane signaling, including authentication credentials, subscription data, session state information, and network topology details. The N2 interface connecting the radio access network to the core presents another significant attack surface, as it carries mobility management messages and radio resource control signaling that could enable tracking of user equipment or disruption of connectivity. The N1 interface, which handles Non-Access Stratum (NAS) signaling between the User Equipment (UE) and the AMF, similarly requires protection against interception and manipulation.

The threat model specifically considers the characteristics of cryptographic algorithms currently protecting 5G communications. TLS 1.3, which provides transport layer security for SBI communications, relies on classical Elliptic-Curve Diffie-Hellman (ECDH) key exchange, which is vulnerable to quantum attacks; static RSA key transport, permitted in earlier TLS versions, was removed from TLS 1.3, which mandates ephemeral (elliptic-curve) Diffie-Hellman key exchange. Digital signatures used for authentication, including Elliptic Curve Digital Signature Algorithm (ECDSA) and RSA-PSS, similarly lack quantum resistance. The compromise of these cryptographic primitives would enable adversaries to decrypt historical communications, impersonate network functions, and potentially establish persistent access to network signaling infrastructure.

The security goals for the PQC-enabled 5G core architecture address the fundamental objectives of confidentiality, integrity, and authentication, and extend them to incorporate quantum resistance as a core requirement. These goals guide the design decisions throughout the architecture and define the success criteria for the implemented solution.

- ❏ The primary security goal is confidentiality protection for all inter-network-function communications within the 5G core. This goal requires that sensitive information exchanged between network functions, including user identity data, authentication credentials, session keys, and network topology information, remain protected from unauthorized disclosure even in the presence of quantum-capable adversaries.

The implementation achieves this goal by applying post-quantum key encapsulation mechanisms (specifically ML-KEM768) to establish session keys that encrypt communication between NFs. The `sidecar proxy architecture` ensures that all SBI traffic is encrypted using quantum-resistant algorithms without requiring modifications to the network function implementations themselves.

- Integrity protection constitutes the second major security goal, ensuring that communications between network functions cannot be modified by adversaries without detection. This goal addresses scenarios where attackers may attempt to alter signaling messages to manipulate network behavior, trigger unintended state changes, or disrupt network operations. The architecture implements integrity protection by applying post-quantum digital signatures (specifically ML-DSA and Falcon) to authenticate message origin and detect unauthorized modifications. Each message traversing the proxy layer receives cryptographic protection, enabling the recipient to verify both the source and the integrity of the received data.

- Authentication represents the third core security objective, ensuring that network functions can verify the identity of their communication partners before accepting and processing requests. The quantum-resistant authentication mechanism relies on digital signature verification to confirm that messages purportedly originating from a specific network function were indeed generated by that function using its private key. This prevents impersonation attacks in which adversaries attempt to impersonate legitimate network functions to inject malicious signaling messages or extract sensitive information.

- The fourth security objective addresses non-repudiation, ensuring that network functions cannot deny having sent messages that they actually transmitted. The cryptographic signature mechanisms implemented in the architecture provide mathematical evidence of message origin that can be verified by any party possessing the corresponding public key. This property is essential for accountability in telecommunications networks, where regulatory requirements may mandate the ability to demonstrate that specific network actions were authorized.

Beyond these classical security objectives, the architecture incorporates cryptographic agility as a forward-looking security goal. The separation of cryptographic processing into dedicated wrapper services, combined with the abstraction provided by the sidecar proxy pattern, enables the replacement or upgrading of cryptographic algorithms without affecting network function implementations. This agility is critical given the evolving nature of post-quantum cryptography standards and the likelihood that algorithm selections will need to be adjusted as cryptanalysis advances and standardization processes mature.

The threat model and security goals operate within defined boundaries that establish the scope of protection provided by the architecture. The primary scope encompasses

inter-network-function communications within the 5G core, specifically targeting the SBI interfaces that connect network functions such as AMF, SMF, NRF, UDM, AUSF, PCF, Unified Data Repository (UDR), and Network Slice Selection Function (NSSF). The N2 interface connecting the radio access network to the AMF is included in scope, while the N3 interface carrying user-plane data between the RAN and User Plane Function (UPF) is explicitly excluded from the current implementation.

The architecture assumes that the underlying network functions (the free5gc implementations) operate correctly and do not contain malicious code. The sidecar proxy pattern provides protection against network-level attacks but cannot compensate for compromised or malicious implementations of network functions. Additionally, the model assumes that the cryptographic keys used for PQC operations are generated using appropriate random number generation and are stored securely throughout their lifecycle.

Physical security of the underlying infrastructure, including protection of data centers, network equipment, and the hosts running containerized network functions, falls outside the scope of this architecture. Similarly, the threat model does not address denial-of-service attacks that aim to disrupt network availability through resource exhaustion or traffic flooding, instead focusing on the confidentiality and integrity of communications that reach their intended destinations.

The architectural approach demonstrated in this study with `free5gc-compose` (free5GC Project, 2025b) can be extended to other open-source 5G core implementations, including `Open5GS` (Open5GS Project, 2025) and `OpenAirInterface` (OpenAirInterface Software Alliance, 2025), with minor configuration adjustments to accommodate the specific NF addressing and SBI characteristics of each platform.

The fundamental sidecar proxy pattern remains applicable regardless of the underlying 5G core implementation, as the proxy architecture operates independently of the specific NF code, focusing instead on the standardized HTTP/2 communications that characterize the SBI defined by 3GPP specifications (3rd Generation Partnership Project (3GPP), 2023).

3.5 The PQC Enabled Free5GC

To experiment with our approach, we selected the free5GC platform, an open-source implementation of 3GPP 5G Core specifications designed for cloud-native deployments (3rd Generation Partnership Project (3GPP), 2023). Utilizing the free5gc-compose project (free5GC Project, 2025b), we implemented a modular post-quantum infrastructure using a sidecar proxy pattern that strictly separates cryptographic operations from the NF business logic. As illustrated in the architecture (see Figure 3), the deployment relies on a specialized cryptolib container that integrates liboqs (Open Quantum Safe Project, 2025) v0.15.0 and the OpenSSL OQS-provider, providing the foundation for quantum-

resistant primitives. To abstract these low-level interactions, we developed wrapper-kem and wrapper-sign services that expose RESTful APIs for ML-KEM and ML-DSA operations. This structure (see Figure 9), orchestrated via Docker Compose, ensures that cryptographic lifecycles, including the generation of entity-specific keys and certificates by a dedicated PQC-key-generator, are managed independently of the NF.

The core of the PQC integration lies in the pqc-proxy instances deployed in both server and client modes. NFs communicate via plain-text HTTP/2 over TCP with local proxies within isolated Docker bridge networks (172.17.0.0/16), while inter-proxy traffic across network edges is encapsulated in ML-KEM-768 tunnels and authenticated via ML-DSA-65. A critical technical challenge addressed was the protection of the N2 interface; since NGAP signaling typically relies on SCTP, we deployed SOcket CAT (socat) (RIEGER, 2025) bridges to perform SCTP-to-TCP conversion, enabling PQC encapsulation between the pqc-proxy-gNB and pqc-proxy-amf-n2 (see Figure 8). To ensure traffic redirection, the free5GC configuration files (e.g., amfcfg.yaml, nrfcfg.yaml) were modified to route SBI requests through the local proxy's listener port.

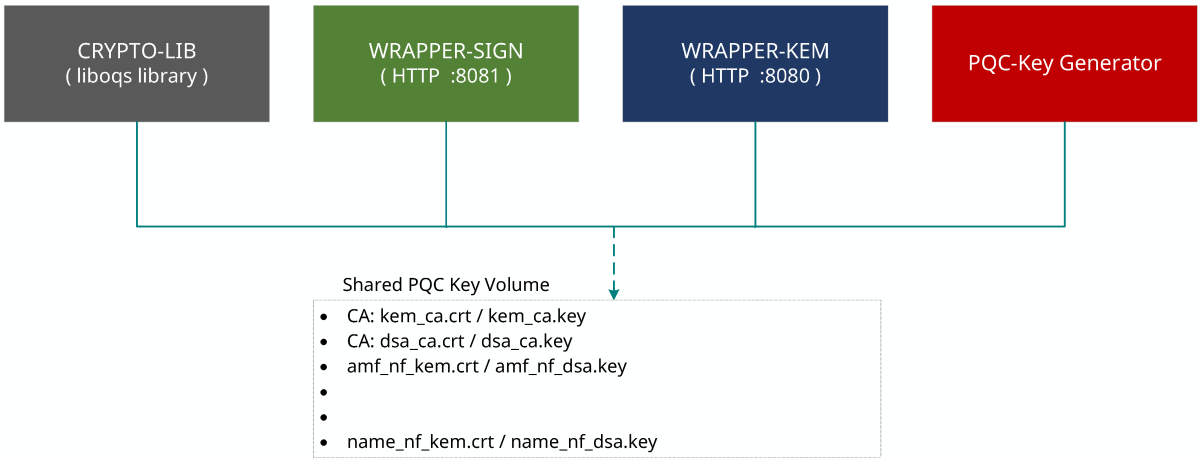

Figure 9 – Cryptographic infrastructure (partial view).

To enhance the existing infrastructure without altering the free5GC NFs 's original source code, this work implements a sidecar proxy approach. This method involves deploying a suite of auxiliary service containers that intercept SBI traffic to provide a PQC-encrypted layer.

As defined in the customized `docker-compose-pqc.yaml` orchestration, the architecture introduces several essential components:

– **PQC-Proxy Suite:** Specialized containers acting as SBIs clients and servers to perform transparent PQC encryption/decryption of HTTP/2 traffic.

– **Crypto-Services Infrastructure:** A dedicated `crypto-lib` base, supported by `wrapper-kem` and `wrapper-sign` microservices, provides the necessary cryptographic primitives to the proxies (see Figure 9).

– **Protocol Bridging:** For the N2 interface, which utilizes the SCTP, **SCTP-TCP bridges** (implemented via `socat` (RIEGER, 2025)) are utilized to route traffic through the TCP-based proxy layer, preserving message delivery and content while collapsing SCTP's multi-stream framing into a single TCP channel, as discussed in Section 3.3.

– **Observability Stack:** The architecture is integrated with **Prometheus**, **cAdvisor**, and **Grafana** to capture granular metrics, including CPU utilization and end-to-end transaction latency.

Table 2 – Proxy Configuration and Port Mapping

Network Function	Proxy Role	Port	Target Port	Identity
NRF	Server (SBI)	8000	nrf:8000	nrf
AUSF	Server (SBI)	9012	ausf:8000	ausf
UDM	Server (SBI)	9013	udm:8000	udm
SMF	Server (SBI)	9014	smf:8000	smf
PCF	Server (SBI)	9015	pcf:8000	pcf
UDR	Server (SBI)	9016	udr:8000	udr
NSSF	Server (SBI)	9017	nssf:8000	nssf
AMF → NRF	Client (SBI)	9001	pqc-proxy-nrf:8000	nrf
AMF → AUSF	Client (SBI)	9002	pqc-proxy-ausf:9012	ausf
AMF → UDM	Client (SBI)	9003	pqc-proxy-udm:9013	udm
AMF → SMF	Client (SBI)	9004	pqc-proxy-smf:9014	smf
AMF → PCF	Client (SBI)	9005	pqc-proxy-pcf:9015	pcf
AMF → UDR	Client (SBI)	9006	pqc-proxy-udr:9016	udr
AMF → NSSF	Client (SBI)	9007	pqc-proxy-nssf:9017	nssf
gNB → AMF	Client (SCTP)	9020	pqc-proxy-amf-n2:9020	amf

Table 2 details the network architecture designed to intercept and protect communications in a 5G Core environment. The first section of the table defines the SBI roles for essential NFs, including NRF, AUSF, UDM, and SMF. Each function receives a unique external port (e.g., 8000 for NRF and a range of 901x for the others) that is mapped to an internal destination port, usually port 8000. This is an example of a configuration applied to a sidecar proxy deployment, where the proxy acts as an entry point to the SBI of

each NF, allowing centralized application of security protocols (PQC) without requiring changes to the underlying NF source code. In the last part of this table, the client-side configurations are shown, illustrating the specific communication flows between the AMF and various peer NFs, as well as the gNB 's connection to the AMF. The AMF traffic is routed through dedicated client proxies on ports 9001 to 9007, which in turn direct the traffic to the corresponding PQC-enabled server proxies. Observing the inclusion of the gNB $\rightarrow$ AMF flow, which uses the SCTP protocol on port 9020, it becomes clear that the architectural approach goes beyond HTTP-based service interfaces, including the N2 interface, ensuring that both SBA and traditional 5G stack signaling are covered by the proxy-based security layer. The listed ports are internal addresses on the isolated Docker bridge network rather than public-facing endpoints, so they follow the sequential internal numbering scheme of the `free5gc-compose` deployment rather than the conventional port 443 used for public-facing HTTPS services.

Chapter 4

Experimental Evaluation and Results

4.1 Experimental Setup

This section presents the experimental methodology adopted to assess the performance implications of integrating PQC into the free5GC using the proposed sidecar architecture. The evaluation is designed to ensure reproducibility, isolate cryptographic overhead from wrapper-induced processing costs, and enable a controlled comparison between Native, classical TLS, and PQC-enabled deployments. The following subsections describe the workload definition, testbed configuration, experimental design, and statistical reporting framework.

4.1.1 Evaluation Scope and Workload

To quantify the performance impact of the sidecar architecture and PQC, the evaluation focuses on control-plane SBI interactions between NFs and the NRF, including registration, discovery, and heartbeat procedures. By isolating the control plane from UE emulation overhead, the setup enables direct comparison between the Native architecture, a classical TLS sidecar proxy, and a PQC sidecar proxy integrated via liboqs.

The workload consists of sequential HTTP/2 request–response exchanges per NF pair. Each experimental repetition executes 100 transactions without artificial concurrency to isolate cryptographic overhead from scheduling or load-induced effects. A transaction is considered successful when the exchange completes without timeouts or retransmissions.

Because transactions are executed sequentially, the reported latencies characterize per-transaction cryptographic and proxy overhead in isolation, but do not support conclusions about behavior under concurrent load. A minimal experimental design capable of addressing concurrency would require issuing overlapping registration and discovery bursts from multiple simulated UEs (e.g., 10–50 concurrent UE sessions against a shared AMF/NRF) and measuring queuing delay and tail latency ($P95/P99$) in addition to the median reported here.

4.1.2 Testbed Configuration

All architectures were deployed in standardized Virtual Machines (VMs) running Ubuntu 22.04 Long Term Support (LTS), each allocated 4 CPU cores and 8 GB of Random Access Memory (RAM). The 5G Core was orchestrated using free5GC-compose v4.2.0. The PQC implementation relied on OpenSSL v3.4.0 integrated with liboqs v0.15.0, enabling ML-KEM-768 and ML-DSA through sidecar proxies. All experiments were conducted without hardware acceleration to isolate software-side overheads.

4.1.3 Experimental Design

To evaluate the performance impact of integrating PQC into a 5G mobile core network using the proxy/wrapper approach, a comparative analysis was conducted encompassing three distinct architectural configurations (Native, TLS Proxy, and PQC Proxy), evaluated through 17 experimental sets (EP01 to EP17).

To achieve statistical stability and capture SBI behavior, each experiment followed a two-phase protocol, consisting of 30 repetitions (independent deployments) and 100 iterations (service requests) per deployment. The PQC configuration was evaluated on 15 parameter sets combining CA (ML-DSA-44, ML-DSA-65, and ML-DSA-87), KEM (ML-KEM-768) for encapsulation, and ML-DSA-44, ML-DSA-65, ML-DSA-87, `Falcon512`, and `Falcon1024` (see Table 3).

This approach was essential to distinguish functional initialization from steady-state signaling performance, ensuring that the latency overhead of the sidecar proxy architecture and the `liboqs` (Open Quantum Safe Project, 2025) version 0.15.0 implementation was accurately captured.

KEM was held fixed at ML-KEM-768 across all 15 PQC configurations as an experimental control: varying the signature and CA algorithms while holding key encapsulation constant isolates the performance contribution of signature choice from that of key exchange. ML-KEM-768 was selected as the fixed parameter because it corresponds to NIST Security Level 3, the mid-point security category and the one generally recommended by NIST for general-purpose deployment, whereas the signature algorithms were varied because they represent the primary axis of trust-model flexibility (CA versus leaf certificate) explored in this study.

- **Phase 1:** Occurs immediately after executing the `docker-compose up` command, which runs the orchestration file for the containers of each architecture. This stage is characterized by automated system triggers initiated by the native *free5gc* source code. Within the 5G architecture, specifically in the `github.com/free5gc/openapi` library, the `callAPI` function in `client.go` acts as the main dispatcher for all SBI communications.

Table 3 – Settings applied in the experiments

Experiments	Approach	KEM	Signature	CA	Technology
EP01	PQC + Proxy	KEM-768	DSA44	DSA44	PQC
EP02	PQC + Proxy	KEM-768	DSA65	DSA44	PQC
EP03	PQC + Proxy	KEM-768	DSA87	DSA44	PQC
EP04	PQC + Proxy	KEM-768	Falcon512	DSA44	PQC
EP05	PQC + Proxy	KEM-768	Falcon1024	DSA44	PQC
EP06	PQC + Proxy	KEM-768	DSA44	DSA65	PQC
EP07	PQC + Proxy	KEM-768	DSA65	DSA65	PQC
EP08	PQC + Proxy	KEM-768	DSA87	DSA65	PQC
EP09	PQC + Proxy	KEM-768	Falcon512	DSA65	PQC
EP10	PQC + Proxy	KEM-768	Falcon1024	DSA65	PQC
EP11	PQC + Proxy	KEM-768	DSA44	DSA87	PQC
EP12	PQC + Proxy	KEM-768	DSA65	DSA87	PQC
EP13	PQC + Proxy	KEM-768	DSA87	DSA87	PQC
EP14	PQC + Proxy	KEM-768	Falcon512	DSA87	PQC
EP15	PQC + Proxy	KEM-768	Falcon1024	DSA87	PQC
EP16	OpenSSL + Proxy	NA	mTLS OAuth	mTLS OAuth	Classic
EP17	Free5GC reference	NA	mTLS OAuth	mTLS OAuth	Classic

As each NF is initialized, internal Go routines trigger registration requests to the NRF to announce its availability. For example, when the AMF reaches an operational state, it internally invokes the `RegisterNFInstance` function, resulting in the following automated request:

```
Method:  PUT
Path:  "/nnrf-nfm/v1/nf-instances/{amfInstanceId}"
```

The goal here is to validate the cryptographic stack's functional integrity. A successful registration confirms that the mTLS handshake (whether classic or PQC-based) was successfully negotiated and that the proxy sidecar correctly handled the internal `net/http` dispatch of the native Go binary.

- **Phase 2**, defined in this project as "Controlled Iterative Sampling," is used to evaluate the performance of the architectures and begins once the main network reaches a stable state. Unlike Phase 1, which relies on automatic system initializations, Phase 2 uses manual triggers to execute 100 iterations per deployment. This enables intensive data collection at a large scale (3,000 samples per experimental set) without the overhead and time penalty of a complete system restart for each sample. For

example, to simulate service traffic, a series of `Nnrf_NFDiscovery` requests are sent via automated scripts using `curl`, simulating an AMF or SMF querying the NRF to obtain the location of a specific service provider:

```
Command: curl -X GET "http://localhost:8000/nnrf-disc/v1/nf-
instances?target-nf-type=AUSF"
```

By maintaining the system in a "steady state," Phase 2 isolates the computational cost of processing the cryptographic payload (e.g., signing and PQC encryption) from the initial connection establishment costs. This provides a clear distinction between the overhead introduced by the sidecar application-layer security logic (TLS Proxy and PQC Proxy) and that of the Native architecture.

4.1.4 Metrics and Statistical Reporting

Metrics were collected using Prometheus, cAdvisor, and Node Exporter. For statistical reporting, each repetition is treated as an independent sample. Results are expressed using median and Interquartile Range (IQR) across repetitions, and 95% confidence intervals were computed over per-run median latency values using the t-distribution. Reporting the median and IQR, rather than the mean and standard deviation, was a deliberate choice to reduce sensitivity to the occasional outlier samples expected in containerized, virtualized testbeds. The monitored performance and security indicators are summarized in Table 4.

All latency values reported in this chapter, including L_{SBI}, are round-trip measurements (Round-Trip Time (RTT)): the client-side timestamp is captured immediately before request transmission, and immediately after the corresponding response is received, so the reported value is not a one-way delay. L_{Srv} and L_{Cli} denote, respectively, the processing time added by the server-side and client-side sidecar proxy for a given transaction, computed as $\Delta t = t_{\text{egress}} - t_{\text{ingress}}$ at each proxy. These two quantities are components *within* the end-to-end L_{SBI} measurement, not additional latency incurred on top of it: L_{SBI} already includes the time spent inside both proxies, since the proxies sit inline on the request-response path. The Memory (MB) metric reported throughout this chapter corresponds to resident memory actually allocated to the container process, as reported by cAdvisor, not free or available memory.

To evaluate the resulting cryptographic overhead and system stability, a robust monitoring framework was integrated directly into the architecture. This stack comprises Prometheus for metrics collection, Grafana for real-time visualization, Node Exporter for hardware telemetry, and cAdvisor for container-level resource tracking. This integrated observability framework enables precise, high-fidelity analysis of CPU and memory consumption, as well as latency variations across the PQC mechanisms deployed within the

Table 4 – Selected metrics for evaluation.

Visual Analysis	Selected Metrics	Evaluation Goal
Latency Distribution	L_{SBI}	Assess delay scaling and jitter across architectures.
Resource Efficiency	CPU% Mem_{MB}	Validate hardware sustainability for PQC workloads.
Latency Breakdown	L_{Srv} L_{Cli}	Isolate sidecar overhead from cryptographic processing.
SBI Matrix	L_{SBI} per NF pair	Identify bottlenecks in 5G signaling (e.g., AMF, NRF).
PQC Operations	T_{enc} T_{sig} T_{ver}	Correlate KEM/DSA complexity with E2E performance.

5G Core, providing the necessary data to validate the performance trade-offs of the proposed sidecar approach (see Figure 10).

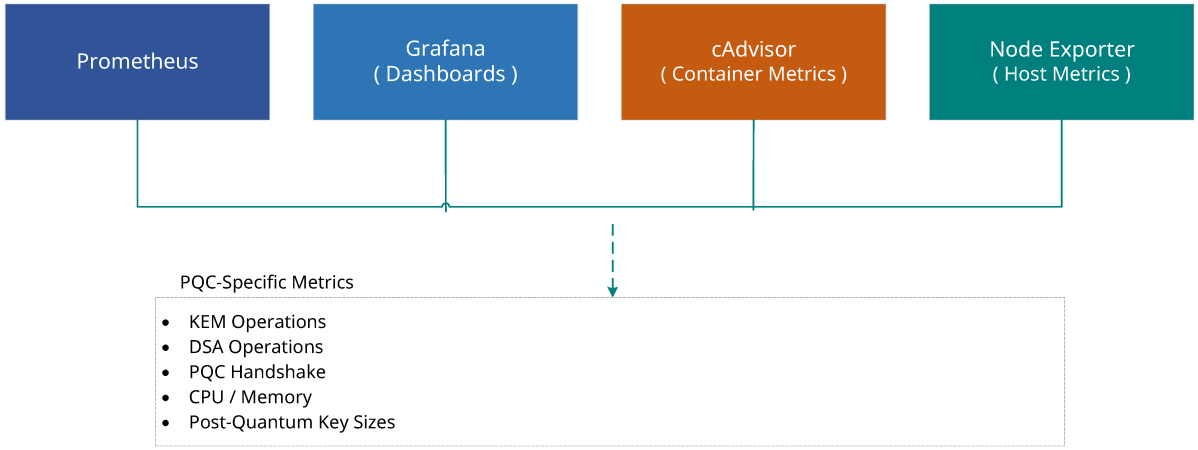

Figure 10 – Monitoring infrastructure layer.

4.2 Results and Analysis

4.2.1 Latency Analysis

This latency analysis compares the three architectures to quantify the performance implications of integrating Post-Quantum Cryptography. We evaluated three distinct free5GC deployments: (i) *Native* setup using standard HTTPS/TLS, (ii) *TLS Proxy* configuration utilizing a sidecar model with classic OpenSSL certificates, and (iii) *PQC Proxy* deployment that enables PQC through the proposed sidecar approach.

Latency is measured at the HTTP/2 request–response boundary on the SBI: timestamps are captured immediately before request transmission and after complete response reception at the application layer. The SBI Communication metric, therefore, captures end-to-end control-plane transaction latency for one NF-to-NF interaction. Container Metrics capture proxy-side processing behavior, while Overall Statistics correspond to the per-run aggregate of all collected SBI request–response latencies across monitored NF pairs (Figure 11). Per-NF-pair heatmaps report mean latencies for individual paths (e.g., 5–6 ms in the Native case), whereas Figure 11 summarizes distributional medians aggregated across interactions within each run.

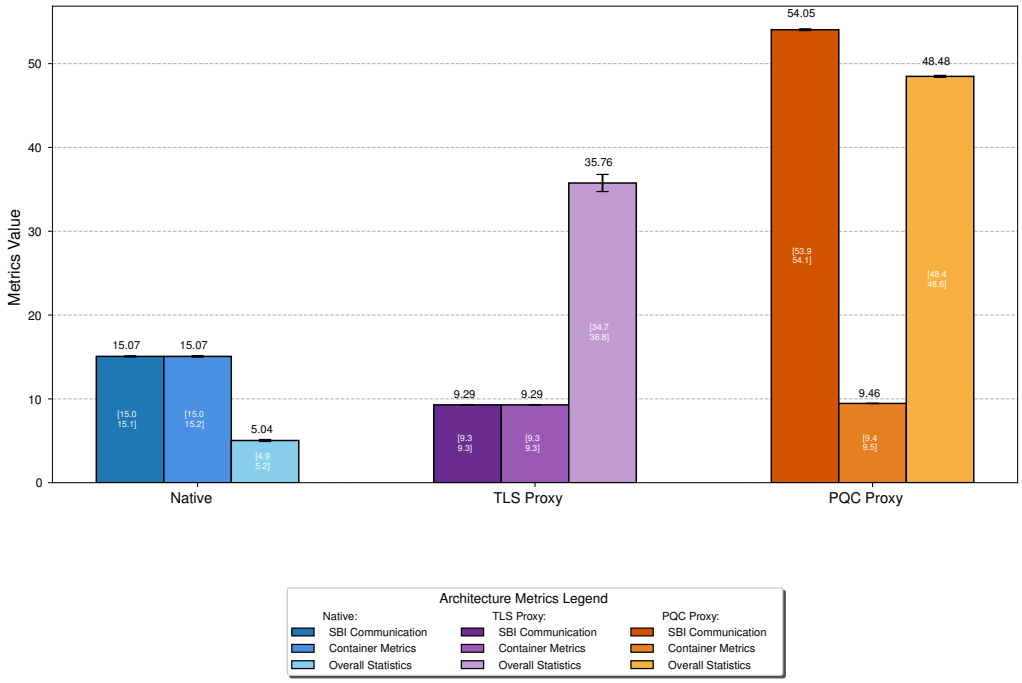

Figure 11 – Metrics comparison across architectures Core 5G.

For statistical reporting, 95% confidence intervals were computed over per-run median latency values using the t-distribution across 30 independent repetitions, while error bars represent the IQR ($Q1$–$Q3$) around the median. The native approach establishes the baseline, with SBI Communication and Container Metrics medians of 15.07 (Confidence Interval (CI): $[15.0, 15.1]$ and $[15.0, 15.2]$) and minimal quartile deviation (± 0.05–0.1). Overall Statistics records 5.04 (CI: $[4.9, 5.2]$), indicating stable conventional TLS behavior.

The TLS Proxy converges SBI Communication and Container Metrics at 10.20 ms (CI: $[10.2, 10.2]$ for both), corresponding to a 32.3% reduction relative to Native and near-zero IQR. This reduction does not imply lower cryptographic complexity; it is consistent with the sidecar centralizing connection management and reusing pooled HTTP/2/TLS sessions across repeated SBI exchanges, thereby amortizing handshake and transport setup costs. This is an optimistic scenario for the TLS Proxy: it reflects a steady-state workload

with sustained connection reuse (Phase 2, Section 4.1), and the relative advantage would be expected to shrink under workloads with frequent connection churn or short-lived sessions. To ensure comparability, latency was measured at the same request–response boundary in all architectures, using identical keep-alive and connection-reuse settings across runs.

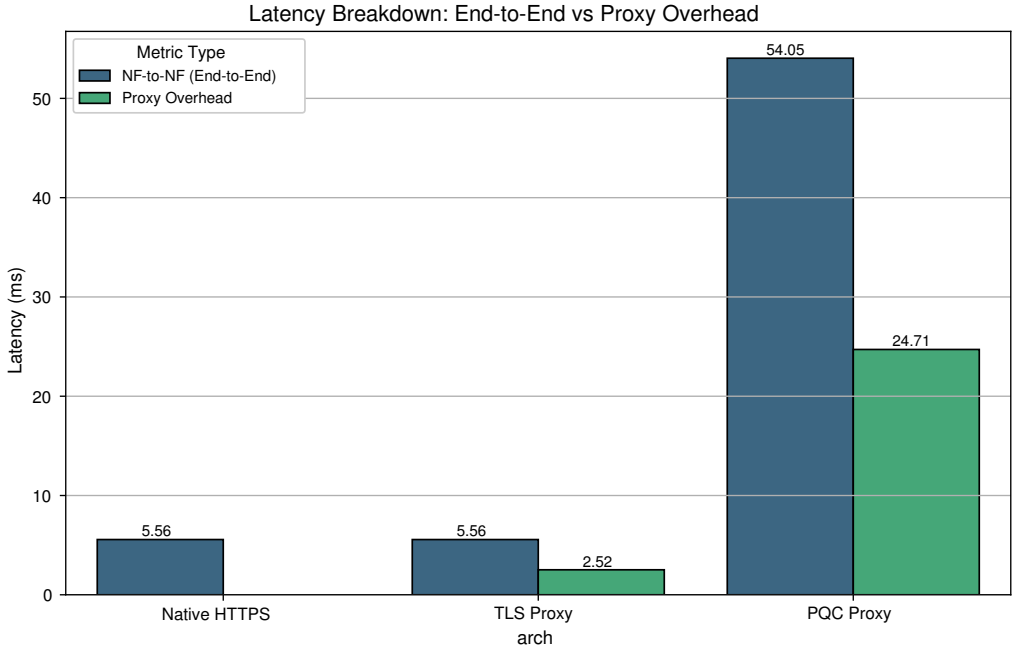

Figure 12 – Processing latency End-to-End.

In contrast, the PQC Proxy increases SBI Communication to 54.05 ms (CI: $[53.9, 54.1]$) while preserving tight quartile bounds (± 0.1), quantifying a stable overhead of approximately 48–49 ms relative to classical TLS. Container Metrics remain stable at 9.46 ms (CI: $[9.4, 9.5]$), suggesting bounded resource behavior even under ML-KEM and ML-DSA processing (Figure 12). In this and subsequent breakdown figures, the proxy-processing bars are not additive on top of the SBI Communication bar: they decompose it, showing how much of the total round-trip time is attributable to sidecar processing versus the remaining network and NF application time.

From a standards perspective, the measured ~ 54 ms end-to-end SBI latency remains well within 5G control-plane timing tolerances, which are governed by NAS procedure timers and are typically on the order of seconds. Therefore, the additional $\sim$ 48–49 ms introduced by the PQC sidecar is operationally compatible with registration and authentication signaling, while still motivating latency-sensitive optimizations.

Quartile analysis highlights distinct distributional behavior across architectures. Native exhibits low-variance distributions, while TLS Proxy shows compressed quartiles for SBI and container-level metrics but expanded quartiles for Overall Statistics, indicating variability in aggregation within the proxy pipeline. By contrast, PQC Proxy

maintains narrow quartile ranges across all metrics ($IQR \leq 0.2$), demonstrating higher absolute latency with bounded variance—a key property for predictable Quality of Service (QoS). The separation is clear: PQC SBI quartiles $[53.9, 54.1]$ do not overlap with Native $[15.0, 15.1]$ or TLS Proxy $[10.2, 10.2]$, and the observed $CV < 0.4\%$ confirms stable behavior.

Heatmaps further corroborate these trends. Native maintains uniformly low SBI latencies (5.5–6.0 ms) across NF pairs (Figure 13), while TLS Proxy remains within the same order of magnitude (5.0–7.7 ms) and exhibits stable behavior consistent with optimized connection reuse (Figure 14). PQC Proxy exhibits a nearly uniform increase to 53.0–54.4 ms across all NF pairs (Figure 15), indicating deterministic overhead largely independent of the communication pattern and dominated by ML-KEM-768 encapsulation/decapsulation and ML-DSA processing.

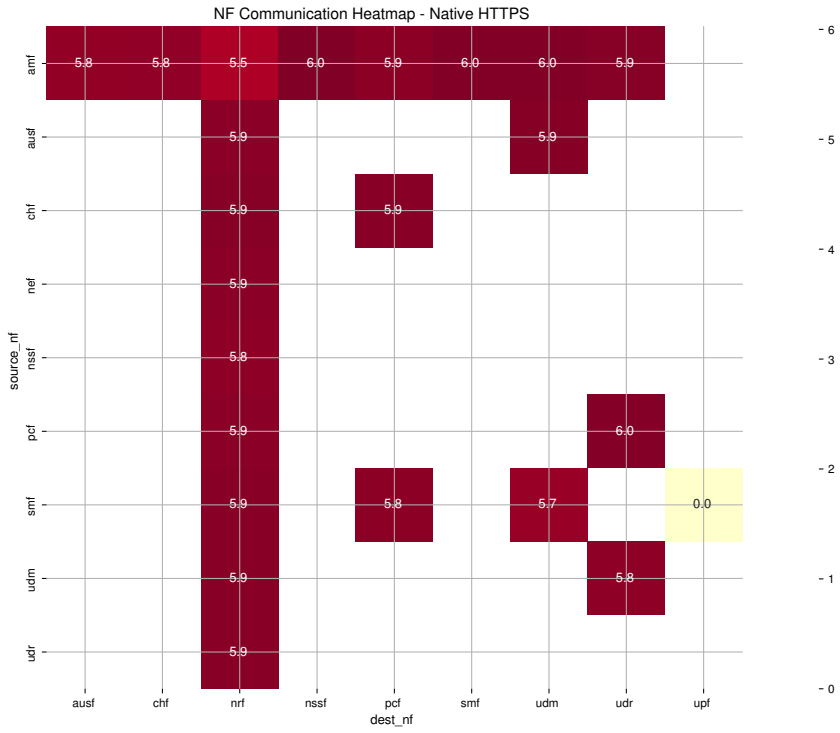

Figure 13 – Mean latency in the HTTPS baseline.

Detailed latency breakdown for representative NRF-centric exchanges (Figures 16–18) shows the same pattern: Native remains near 5–6 ms, TLS Proxy remains near 5 ms with a small proxy processing component (2.43 ms), and PQC Proxy increases to 53–54 ms. The proxy processing estimate is stable (24.71 ms) across the evaluated pairs, indicating deterministic overhead consistent with encapsulation/decapsulation dominance. The proxy processing estimate is computed inside the sidecar as $\Delta t = t_{\text{egress}} - t_{\text{ingress}}$, capturing cryptographic encapsulation/decapsulation and certificate validation time within the proxy/wrappers, independent of NF application processing.

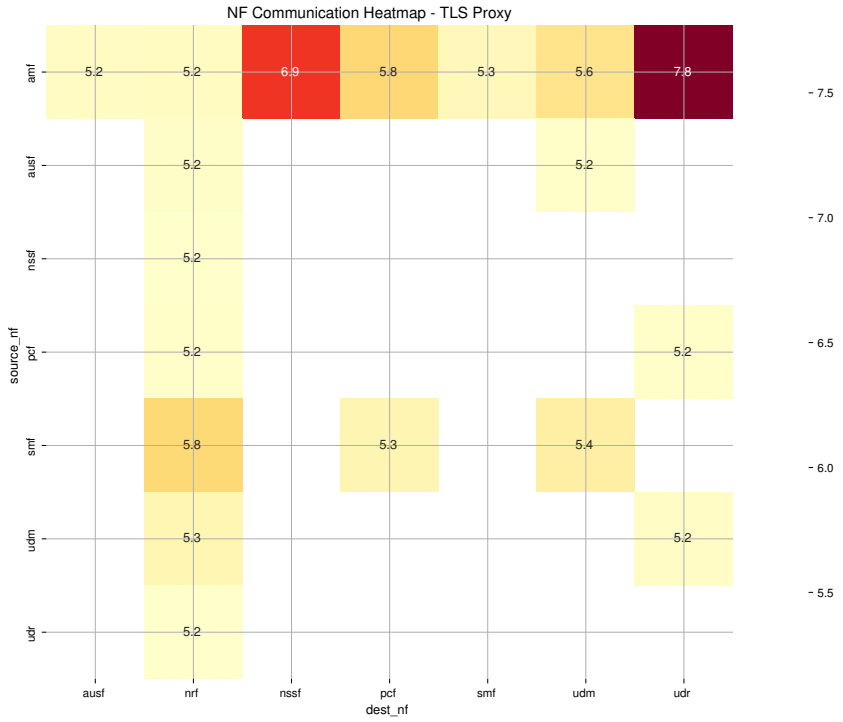

Figure 14 – Mean latency in the TLS Proxy.

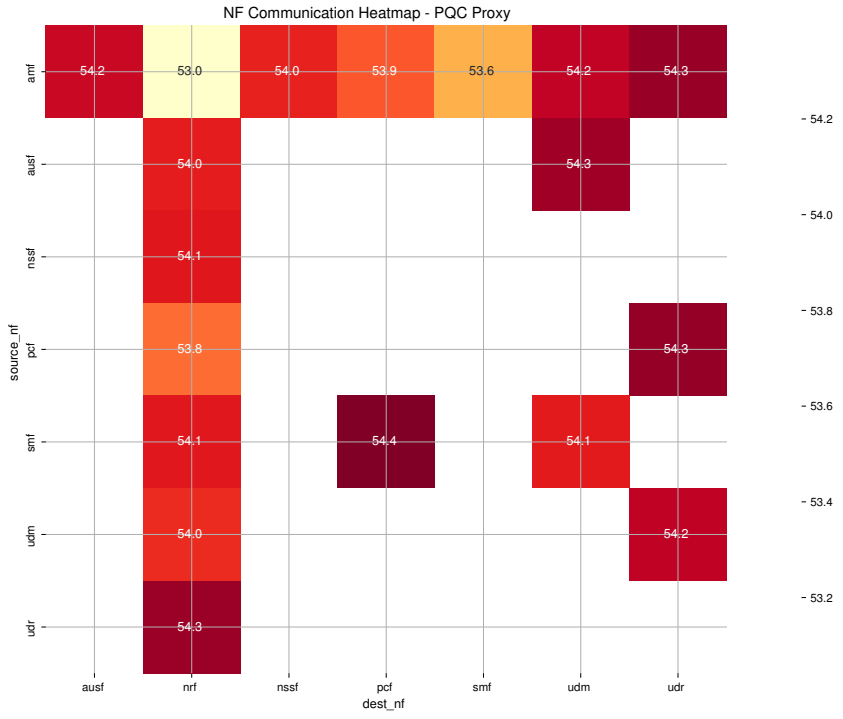

Figure 15 – Mean latency in the PQC Proxy

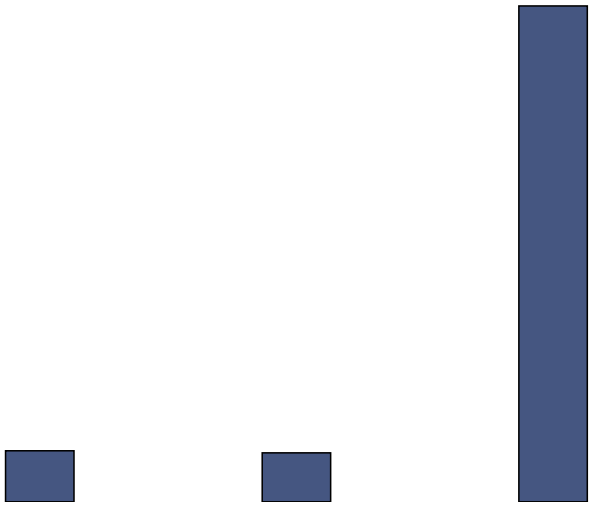

Figure 16 – Processing latency - AMF to NRF

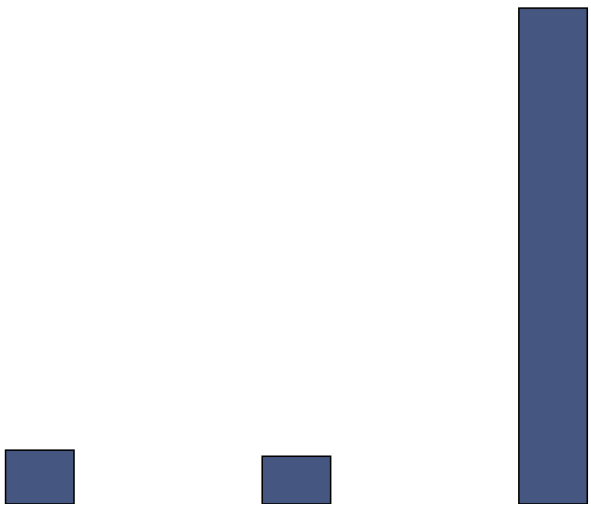

Figure 17 – Processing latency - AUSF to NRF

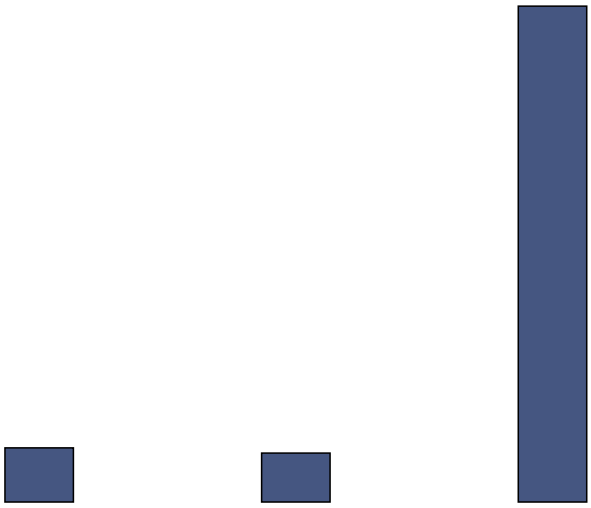

Figure 18 – Processing latency - UDM to NRF

Table 5 – Impact of CA security level on server-side validation latency (PQC Proxy).

CA Algorithm	Server Proxy (ms)	Total SBI Median (ms)	Δ vs ML-DSA-44 (ms)
ML-DSA-44	6.48	53.72	-
ML-DSA-65	7.03	54.05	+0.33
ML-DSA-87	8.21	55.12	+1.40

The proxy processing breakdown (Figure 19) reveals asymmetric costs for TLS Proxy (server-side 3.99 ms vs. client-side 1.00 ms). Both sides perform full mTLS certificate-chain validation; the asymmetry is not due to the client skipping validation, but is consistent with server-dominated private-key operations (the server performs the private-key signing operation during the handshake, while the client's corresponding operation is comparatively cheaper verification), and is left as an open question for further profiling. For the PQC Proxy, server-side (7.03 ms) and client-side (6.56 ms) costs become nearly symmetric (13.59 ms total), indicating that encapsulation and decapsulation impose comparable computational burdens and contribute directly to end-to-end latency. Overall PQC overhead comprises (i) ML-KEM encapsulation/decapsulation, (ii) ML-DSA signature generation/verification, and (iii) proxy-side certificate-chain validation.

Detailed analysis indicates that higher security levels in the CA (e.g., ML-DSA-87) directly increase server-side validation latency. To quantify the impact of security margins on operational latency, we evaluated three CA parameter levels within the PQC Proxy configuration.

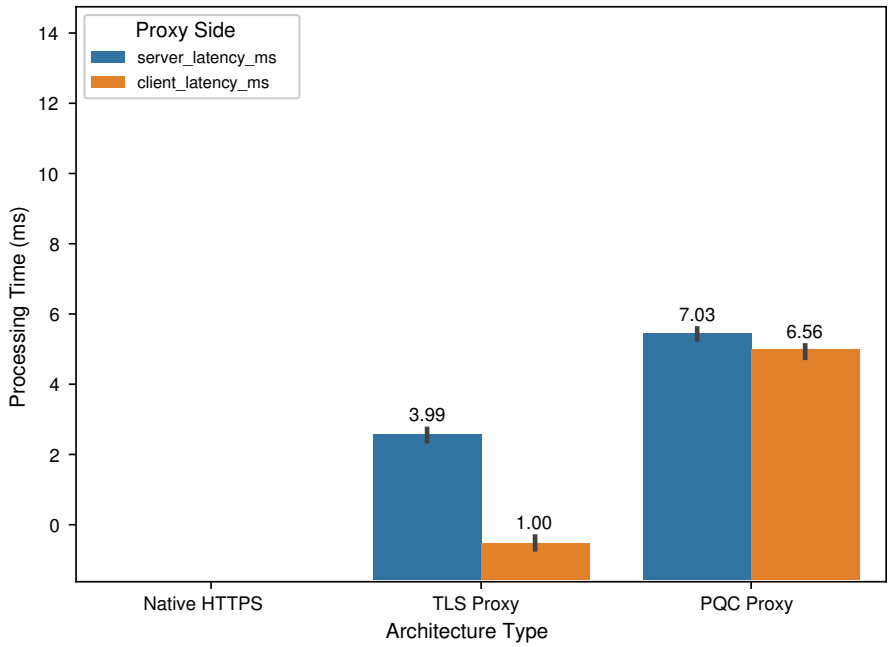

Figure 19 – PQC latency operations

As shown in Table 5, increasing the CA security level yields a monotonic rise in server-side validation latency: the transition to ML-DSA-87 adds +1.40 ms to the median total SBI latency relative to ML-DSA-44. While this variation is smaller than the dominant PQC overhead, it is repeatable and measurable, thereby confirming that certificate-chain validation is a tunable contributor to end-to-end delay. Consequently, CA parameter selection becomes a concrete deployment lever for balancing post-quantum security margin and control-plane latency constraints.

4.2.2 PQC Analysis

The experimental evaluation of PQCs integration within the free5gc-compose architecture comprises fifteen distinct experimental configurations (EP01 to EP15), each representing a unique combination of PQC algorithms. These experiments were designed to systematically evaluate the performance implications of deploying post-quantum cryptographic primitives as sidecar proxies in 5G network functions. The primary objective was to characterize the computational overhead, latency characteristics, and resource utilization patterns associated with different PQC algorithm configurations, thereby providing empirical evidence for the feasibility of PQC adoption in next-generation cellular networks.

All fifteen PQC experiments share a common KEM: ML-KEM-768, which serves as the foundational algorithm for key establishment. However, the experiments differ significantly in their signature algorithms and CA configurations, enabling a comprehensive analysis of how different post-quantum signature schemes impact overall system perfor-

mance. The signature algorithms evaluated include ML-DSA variants (mldsa44, mldsa65, mldsa87) representing different security levels, as well as Falcon signatures (falcon512, falcon1024), which offer alternative approaches to post-quantum digital signatures.

The evaluation of the PQC wrapper components reveals a distinct performance profile that characterizes the computational cost of quantum-resistant algorithms (see Figure 20). As evidenced by the performance data, both the Key Encapsulation Mechanism (`wrapper-kem`) and the Signature Mechanism (`wrapper-sign`) exhibit uniform latency, with `wrapper-kem` registering approximately 24.5 ms and `wrapper-sign` slightly higher at 24.8 ms. This symmetry in latency suggests that the computational overhead is evenly

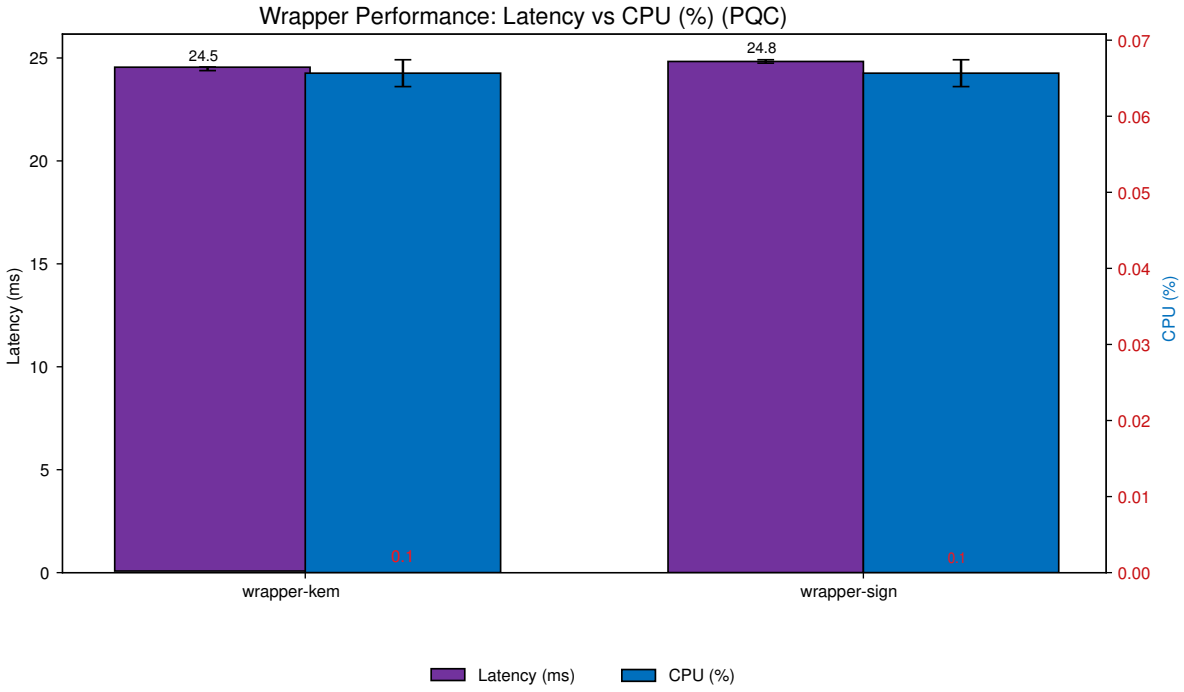

Figure 20 – Analysis of PQC Wrapper Components (CPU)

distributed between the encapsulation and signing phases of the cryptographic handshake. Despite this latency, CPU utilization for both components remains negligible at 0.1%, suggesting that latency is likely limited by the algorithmic complexity of cryptographic operations or by Input/Output (I/O) waits, rather than by thread blocking or CPU saturation.

Furthermore, the memory footprint for these operations is highly efficient, stabilizing at approximately 10.5 MB for both KEM and signature wrappers (see Figure 21). These results demonstrate that, although PQC incurs a measurable per-operation time cost (approximately 25 ms per component), it does not impose a significant memory or processor-cycle burden, making the sidecar approach viable under resource constraints.

To better understand the impact of the added security layer, Table 6 presents a break-

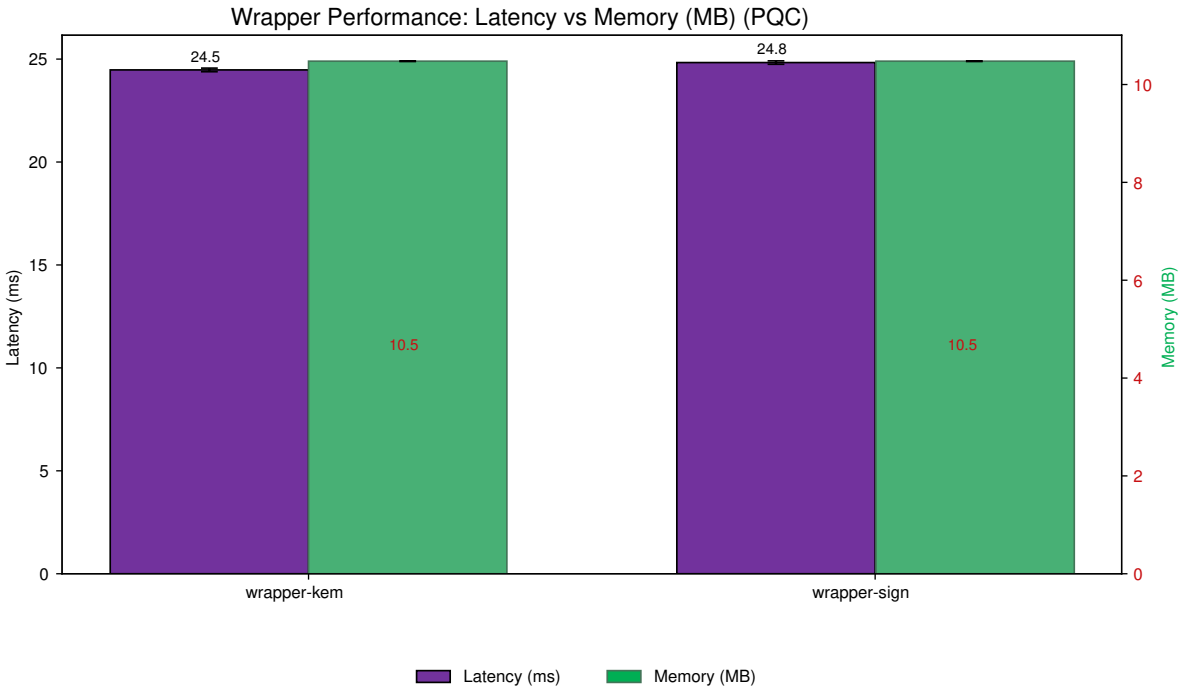

Figure 21 – Analysis of PQC Wrapper Components (Memory)

down of the performance metrics collected for the internal wrapper components. The data details the average latency, CPU usage, and memory consumption for both the Key Encapsulation Mechanism (`wrapper-kem`) and the Digital Signature (`wrapper-sign`) processes. These values serve as a baseline for estimating the computational cost of cryptographic operations, excluding network transmission overhead.

Table 6 – Performance Metrics of PQC Wrapper Components

Parameter	Label Value	Description
Wrapper Type	`wrapper-kem`, `wrapper-sign`	Identifies the specific PQC component: Key Encapsulation Mechanism or Digital Signature.
Latency (ms)	$\approx$ 24.5 ms (KEM), $\approx$ 24.8 ms (Sign)	The average time taken to execute the specific cryptographic operation.
CPU (%)	0.1%	The percentage of CPU capacity consumed during the operation.
Memory (MB)	$\approx$ 10.5 MB	The amount of RAM used by the wrapper process.

Comparing latency across functions provides a clear quantification of the overhead introduced by PQC integration. In the native reference architecture (`free5gc-compose` with HTTPS), the system demonstrates high-performance communication, with latencies between NFs, such as AMF and NRF, averaging approximately 5.8 ms, and other paths,

such as SMF to NRF, stabilizing around 5.4 ms (see Figure 22). This establishes a baseline for standard 5G core control plane traffic.

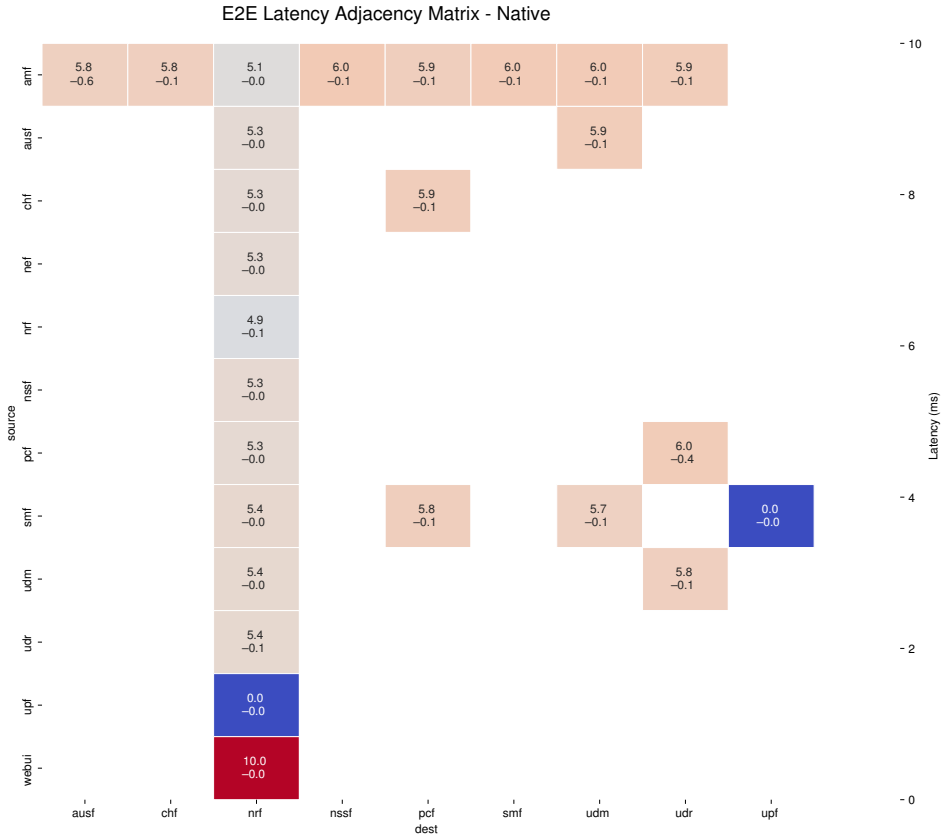

Figure 22 – E2E Network Latency in the baseline

Introducing the TLS sidecar proxy (see Figure 23) has minimal impact on latency. The data show that latencies remain comparable to native execution, with the AMF to NRF path registering 5.4 ms, and other interactions ranging from 5.3 ms to 7.5 ms. This confirms that the architectural change of adding a sidecar proxy for standard TLS termination does not inherently degrade network performance.

However, the PQC sidecar architecture introduces a substantial and consistent increase in latency (see Figure 24). The interaction times between the NFs increase significantly, with the AMF to NRF path rising to 52.5 ms, and other functions, such as AUSF and SMF, averaging around 53.8 ms. This represents an approximately tenfold increase in latency compared to the Native and TLS sidecar scenarios.

Correlating this with the previously analyzed wrapper performance data, this increase is consistent with the cumulative overhead of PQC operations (approximately 24.5 ms for KEM plus 24.8 ms for Signature, totaling $\approx$ 49.3 ms), confirming that the latency penalty is strictly cryptographic and not architectural. Despite the higher latency, the uniformity of delay across all NFs suggests a deterministic, stable performance characteristic, which is crucial for the predictability of network planning.

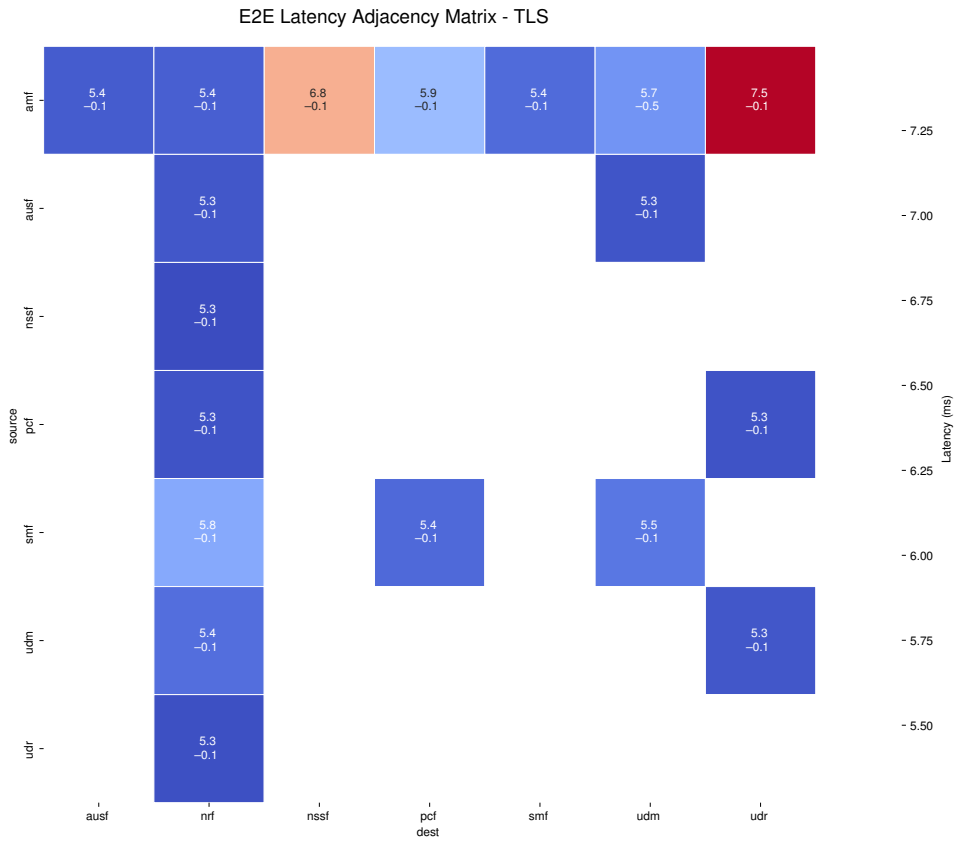

Figure 23 – E2E Network Latency in TLS Proxy

Table 7 – Comparative Latency Summary Across Architectures

Parameter	Label Value	Description
Architecture	Native, TLS, PQC	The deployment mode of the 5G Core: Standard HTTPS, Standard TLS Sidecar, or Quantum-Resistant Sidecar.
Source/Dest	`amf`, `smf`, `nrf`, `ausf`	The source and destination NFs involved in the communication path.
Native Latency	≈ 5.0 – 6.0 ms	Baseline latency for internal 5G Core communication using standard HTTPS.
TLS Latency	≈ 5.3 – 7.5 ms	Latency observed when traffic is routed through a standard TLS sidecar proxy.
PQC Latency	≈ 52.5 – 53.9 ms	Latency observed when traffic is protected by PQC via the sidecar architecture.

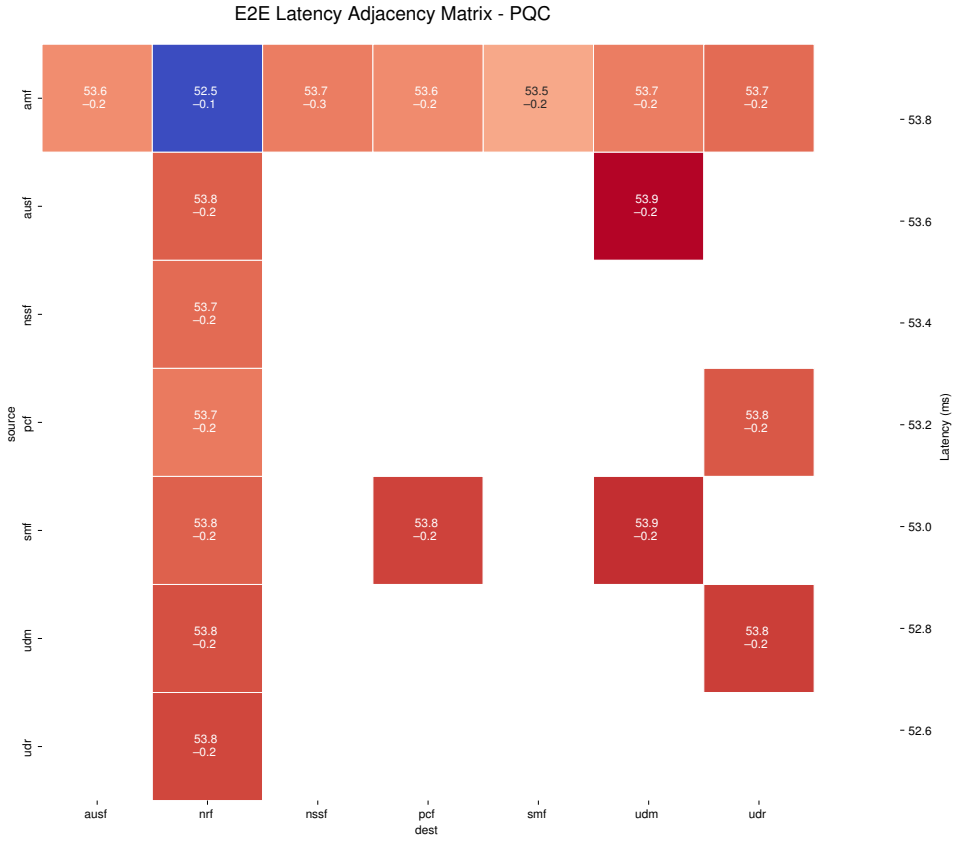

Figure 24 – E2E Network Latency in PQC Proxy

Table 7 presents a comparative summary of E2E latency across the three scenarios evaluated: Native (HTTPS enabled), Sidecar TLS, and Sidecar PQC. It highlights the delay introduced by different security implementations relative to the base latency of standard communication paths and those routed through sidecar proxies. The values represent the average time required to exchange packets between the source and destination networks in the 5G core.

Table 8 – Resource Consumption and Latency Status of NF

Parameter	Label Value	Description
Metric Type	Latency vs CPU / Memory	Correlates the communication delay with the system resources consumed by the NF.
PQC CPU Usage	0.0% – 0.1%	The CPU load on the NF when operating with the PQC sidecar proxy.
Native Memory	≈ 8 – 20 MB	Baseline memory usage of NFs in the default configuration.
PQC Memory	≈ 11 – 19 MB	Memory usage of NFs with PQC enabled, showing a comparable footprint to native.
Latency to NRF	≈ 5 ms (Native) vs ≈ 53 ms (PQC)	Specific latency metric used to correlate with resource usage, reinforcing the E2E findings.

Finally, Table 8 details the resource consumption of individual NFs functions in relation to their communication latency with the Network Repository Function (NRF) function. This table correlates CPU and memory usage for critical components, such as AMF and AUSF, with latency observed in native environments and with PQC enabled. In this way, it is possible to assess whether integrating the PQC sidecar imposes additional resource overhead on the main logic of the 5G function application.

4.2.3 Qualitative Analysis

Table 9 presents a qualitative comparison between this work and the related work presented in Section 2.2.

This table indicates an explicit focus on the 5G and SBI and TLS, IPsec, and OAuth indicate whether TLS, IPsec, and OAuth are discussed or used. 5G Stack indicates use of an open-source; Exp. indicates studies that have conducted practical experiments to test the solution.; Fifth Generation implementation or testbed; API/Fuzz indicates API injection or fuzzing-style testing; and PQC indicates explicit PQC discussion or implementation.

Table 9 – Qualitative Comparison with the Related Work.

Paper	5GC /SBA	TLS	IPsec	OAuth	Exp.	5G Stack	API /Fuzz	PQC
Mahyoub et al. (MAHYOUB et al., 2024)	●	●	●	●	○	○	○	●
Mehic et al. (MEHIC et al., 2024)	●	●	●	○	○	○	○	●
Lawo et al. (LAWO et al., 2024)	○	○	○	○	○	○	○	●
Scalise et al. (SCALISE et al., 2024)	●	●	●	●	○	○	○	●
Mangla et al. (MANGLA et al., 2023)	○	●	●	○	○	○	○	●
Dolente et al. (DOLENTE; GARROPPO; PAGANO, 2024)	●	●	●	●	○	●	●	○
This work	●	●	○	●	●	●	●	●

In this context, the present work provides a holistic approach that bridges the gap between theoretical security and practical deployment. It makes a valuable contribution

by simultaneously addressing 5GC/SBA, TLS, and OAuth while integrating a fully functional 5G stack. IPsec is outside the scope of this work, as the proposed architecture protects SBI and N2 signaling at the application layer rather than the network layer. Notably, this research advances the state of the art by proposing and executing a technical experiment that implements PQC in the 5GC. This is achieved through a novel approach based on the sidecar proxy pattern, which allows the addition of quantum-resistant algorithms in the communication plane without requiring radical modifications to the underlying network functions.

By leveraging the sidecar architecture, this work demonstrates a scalable, modular approach to protecting 5G infrastructure against emerging quantum threats, thereby offering a robust, differentiated solution compared with those identified in previous comparative studies.

4.3 Limitations

Although this research successfully demonstrates the technical feasibility and performance characteristics of integrating PQC into the 5G core via sidecar proxies, certain limitations inherent to the experimental scope and methodological approach restrict the generalization of the results. These limitations, rather than diminishing the contributions of this work, identify specific domains requiring further investigation and lay the groundwork for subsequent research.

The experimental validation in this study was conducted under controlled laboratory conditions, using the `free5gc-compose` platform for limited-scale NF deployments. This may limit understanding of how PQC-enabled sidecar proxies behave under realistic traffic-aggregation patterns characteristic of operational 5G networks.

The security assessment in this work focused on architectural correctness and protocol compliance rather than resilience against adversarial attacks. The study did not incorporate simulated attack scenarios, penetration testing, or formal security verification of the implemented cryptographic mechanisms. Although the ML-KEM-768 and ML-DSA algorithms employed are standardized by NIST and provide security against quantum adversaries, their practical implementations have not undergone an offensive security assessment. This leaves open questions about the proposed architecture's limitations regarding side-channel attacks, implementation vulnerabilities, or sophisticated threat-actor techniques.

Stability assessments were conducted under steady-state conditions, rather than dynamic reconfiguration scenarios, leaving the architecture's behavior unanalyzed during certificate rotation, algorithm migration, or sidecar instance scaling operations.

Dynamic tests simulating the registration of multiple UE devices were not conducted, potentially compromising the evaluation of the infrastructure's performance and behavior

in real-world implementation scenarios. The tests focused exclusively on adding PQC security to the internal communications of the 5G Core, thereby ensuring the stability of the intra-NF communication architecture.

Additionally, the overhead reported for the N2 interface (Sections 3.3 and 4.2.1) conflates two distinct sources: the cryptographic cost of the PQC sidecar and the buffering/parsing cost of the `socat` SCTP-to-TCP translation bridge. The current experimental design does not isolate these two components; doing so would require a dedicated control run measuring the `socat` bridge alone (protocol translation with no cryptographic proxy in the path) against the full PQC-plus-`socat` configuration. This decomposition is left for future work.

The `socat`-based bridge also does not preserve SCTP's multi-stream semantics: it collapses the association into a single TCP byte stream, so per-stream ordering and message-boundary framing are not maintained across the translation. The present evaluation only exercises a single logical N2 channel, so this simplification does not affect the reported results, but it would need to be addressed (e.g., via a multi-stream-aware bridge or one TCP connection per SCTP stream) before this component of the architecture could support production N2 signaling at scale.

These identified limitations define the boundaries of the present investigation and directly guide the recommended trajectory for future research, which should address large-scale performance validation, comprehensive safety assessment, expansion of protocol coverage, and operational resilience under dynamic conditions.

Chapter 5

Conclusion

This research systematically addressed the fundamental questions raised at the outset of this investigation, yielding valuable insights into integrating PQC into 5G Core networks via sidecar proxy architectures.

Regarding RQ-1, technical feasibility was conclusively demonstrated through the successful implementation and operation of a functional prototype in the `free5gc-compose` environment. The integration of the NIST-standardized ML-KEM-768 and ML-DSA algorithms into sidecar proxy containers enabled transparent interception and protection of communications between NFs, without requiring modifications to existing implementations of 5G Core NFs. The architectural approach proved compatible with the SBA interfaces standardized by 3GPP, validating that PQC mechanisms can be effectively deployed in containerized telecommunications infrastructure using established service-mesh standards. In quantitative terms, this feasibility is evidenced by a deterministic SBI latency increase from a 15.07 ms native baseline to 54.05 ms under PQC, a bounded, repeatable overhead rather than a source of instability, directly answering RQ-1's technical-feasibility question with measured evidence rather than an architectural claim alone.

Regarding RQ-2, quantitative empirical measurements revealed clear performance differences among the evaluated configurations. The native baseline configuration exhibited the lowest latency, serving as a benchmark for comparative analysis. The sidecar proxy with classical TLS cryptography introduced a measurable overhead attributable to additional network hops and cryptographic processing. More significantly, the sidecar proxy with PQC demonstrated additional latency increases consistent with the computational complexity of lattice-based cryptographic operations. These measurements provide the first documented benchmarks for PQC performance in 5G Core environments, enabling data-driven assessment of the relationship between performance and security in quantum-computing-resistant deployment scenarios.

Regarding RQ-3, the feasibility of immediate deployment was evaluated against multiple operational criteria. The sidecar proxy approach offers architectural advantages in deployment flexibility and cryptographic agility, supporting hybrid configurations that

enable transitional deployment strategies. However, the empirical performance overhead, resource consumption implications, and additional operational complexity introduced by sidecar mediation indicate that immediate universal deployment may be limited by specific latency and resource availability requirements in production environments. The solution is more applicable in scenarios where security requirements justify performance trade-offs or where infrastructure capacity accommodates the computational demands of post-quantum algorithms. The architectural pattern itself is validated as a viable mechanism for integrating PQC, with deployment readiness depending on specific operator requirements and transition planning.

If the architecture is feasible and the measured overhead is operationally tolerable, an open question remains: why is PQC retrofitting not yet standard practice in deployed 5G cores? The answer is not purely technical: cryptographically relevant quantum computers remain speculative on an uncertain timeline, standardization and library maturity (`liboqs`, OQS-OpenSSL) are recent, and telecommunications operators historically adopt security upgrades only after 3GPP standardization and vendor certification, rather than ahead of them. The HNDL threat model, however, means that data-confidentiality requirements, and not adversary capability today, should drive the migration timeline, since signaling captured now could remain sensitive well into the period when quantum decryption becomes feasible.

Taken together, these findings confirm that integrating PQC into 5G Core networks via sidecar proxies is technically feasible and conditionally applicable for immediate deployment, thereby fulfilling the investigative objectives of this research.

5.1 Publications

With a focus on the latency analysis presented in Section 4.2.1, the paper "Empowering Mobile Networks Security Resilience by Using Post-Quantum Cryptography" (FAVAL; MOREIRA; SILVA, 2026) was published at the *2026 European Conference on Networks and Communications*. It is currently available for reading and consultation on the IEEE Xplore platform. The results presented in Section 4.2.2 and the limitations identified in Section 4.3 will serve as input for a comprehensive study aimed at submission to a high-impact journal in the field.

5.2 Future Work

Future research should extend this work to include user-plane traffic protection, explore the integration of PQC into data-plane processing pipelines, and evaluate the feasibility of hardware acceleration for ML-KEM and ML-DSA operations. Investigations into hybrid cryptographic schemes that combine classical and post-quantum algorithms dur-

ing transition periods, as well as formal security proofs of the sidecar architecture across various adversary models, would strengthen the theoretical foundations of this practical approach.

Additionally, evaluating PQC overhead in massive Internet of Things (IoT) deployment scenarios with significantly higher device density is a critical next step toward 6G readiness.

Bibliography

3rd Generation Partnership Project (3GPP). *System Architecture for the 5G System (5GS)*. [S.l.], 2023. Release 17. Disponível em: <https://www.3gpp.org/ftp/Specs/archive/23_series/23.501/>.

ABDELWAHAB, S. et al. Network function virtualization in 5g. **IEEE Communications Magazine**, IEEE, v. 54, n. 4, p. 84–91, 2016. Disponível em: <https://doi.org/10.1109/MCOM.2016.7452271>.

ALAGIC, G. et al. **Status report on the fourth round of the nist post-quantum cryptography standardization process**. US Department of Commerce, National Institute of Standards and Technology ..., 2025. Disponível em: <https://doi.org/10.6028/NIST.IR.8545>.

ASHOK, S.; GODFREY, P.; MITTAL, R. Leveraging service meshes as a new network layer. In: **Proceedings of the 2021 Workshop on Hot Topics in Networks**. [s.n.], 2021. p. 229–236. Disponível em: <https://doi.org/10.1145/3484266.3487379>.

BROWN, G. et al. Service-based architecture for 5g core networks. **Huawei White Paper**, v. 1, 2017. Accessed: 2025-11-01. Disponível em: <https://www.3g4g.co.uk/5G/5Gtech_6004_2017_11_Service-Based-Architecture-for-5G-Core-Networks_HR_Huawei.pdf>.

BURNS, B.; OPPENHEIMER, D. Design patterns for container-based distributed systems. In: **8th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 16)**. [s.n.], 2016. Disponível em: <https://www.usenix.org/conference/hotcloud16/workshop-program/presentation/burns>.

CALCOTE, L.; JACKSON, N. **Service Mesh Patterns: Best Practices for Cloud-Native Applications**. New York, NY, USA: Association for Computing Machinery and O'Reilly Media, Inc., 2023. ISBN 9798400711111. Disponível em: <https://dl.acm.org/doi/book/10.5555/3631481>.

CHANDRAMOULI, R.; BUTCHER, Z.; CALLAGHAN, J. **Service mesh proxy models for cloud-native applications**. US Department of Commerce, National Institute of Standards and Technology, 2024. Disponível em: <https://doi.org/10.6028/NIST.SP.800-233>.

DOLENTE, F.; GARROPPO, R. G.; PAGANO, M. A vulnerability assessment of open-source implementations of fifth-generation core network functions. **Future Internet**, v. 16, n. 1, 2024. ISSN 1999-5903. Disponível em: <https://doi.org/10.3390/fi16010001>.

FAVAL, R. A.; MOREIRA, R.; SILVA, F. de O. Empowering Mobile Networks Security Resilience by using Post-Quantum Cryptography. In: **2026 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)**. [s.n.], 2026. p. 1046–1053. ISSN 2575-4912. Disponível em: <https://ieeexplore.ieee.org/document/11577513>.

FOUKAS, X. et al. Network slicing in 5g: Survey and challenges. **IEEE Communications Magazine**, v. 55, n. 5, p. 94–100, 2017. Disponível em: <https://doi.org/10.1109/MCOM.2017.1600951>.

free5GC Project. **free5GC: An open-source 5G core network**. 2025. Accessed: 2025-11-01. Disponível em: <https://www.free5gc.org/>.

_____. **free5gc-compose: Docker-compose orchestration for free5GC**. 2025. Accessed: 2025-11-01. Disponível em: <https://github.com/free5gc/free5gc-compose>.

LAWO, D. C. et al. Falcon/kyber and dilithium/kyber network stack on nvidia's data processing unit platform. **IEEE Access**, v. 12, p. 38048–38056, 2024. Disponível em: <https://doi.org/10.1109/ACCESS.2024.3374629>.

LEI, W. et al. 5g system architecture. In: **5G System Design: An End to End Perspective**. Springer, 2021. p. 297–339. Disponível em: <https://doi.org/10.1007/978-3-030-73703-0>.

MAHYOUB, M. et al. Security analysis of critical 5g interfaces. **IEEE Communications Surveys & Tutorials**, v. 26, n. 4, p. 2382–2410, 2024. Disponível em: <https://doi.org/10.1109/COMST.2024.3377161>.

MAIA, J. T. D.; CORREIA, F. F. Service mesh patterns. In: **Proceedings of the 27th European Conference on Pattern Languages of Programs**. [s.n.], 2022. p. 1–12. Disponível em: <https://doi.org/10.1145/3551902.3551962>.

MANGLA, C. et al. Mitigating 5g security challenges for next-gen industry using quantum computing. **Journal of King Saud University - Computer and Information Sciences**, v. 35, n. 6, p. 101334, 2023. ISSN 1319-1578. Disponível em: <https://doi.org/10.1016/j.jksuci.2022.07.009>.

MEHIC, M. et al. Quantum cryptography in 5g networks: A comprehensive overview. **IEEE Communications Surveys & Tutorials**, v. 26, n. 1, p. 302–346, 2024. Disponível em: <https://doi.org/10.1109/COMST.2023.3309051>.

OLUSHOLA, A.; MEENAKSHI, S. P. Design and implementation of an authenticated post-quantum session protocol using ml-kem (kyber), ml-dsa (dilithium), and aes-256-gcm. **Frontiers in Physics**, Volume 13 - 2025, 2026. ISSN 2296-424X. Disponível em: <https://doi.org/10.3389/fphy.2025.1723966>.

Open Quantum Safe Project. **liboqs: C library for quantum-resistant cryptographic algorithms, version 0.15.0**. 2025. Accessed: 2025-11-01. Disponível em: <https://github.com/open-quantum-safe/liboqs/releases/tag/0.15.0>.

Open5GS Project. **Open5GS: C-language Open Source implementation for 5G Core and EPC**. 2025. Accessed: 2025-11-01. Disponível em: <https://open5gs.org/>.

OpenAirInterface Software Alliance. **OpenAirInterface: An open-source platform for 5G and 4G wireless networks**. 2025. Accessed: 2025-11-01. Disponível em: <https://openairinterface.org/>.

PELL, R. et al. Service classification of network traffic in 5g core networks using machine learning. In: **2023 IEEE International Conference on Edge Computing and Communications (EDGE)**. [s.n.], 2023. p. 309–318. Disponível em: <https://doi.org/10.1109/EDGE60047.2023.00053>.

PUNUGOTI RAJASRIKAR E MANAM, S. Observabilidade e audição em sistemas distribuídos multi-tenant escaláveis. In: _____. IntechOpen, 2025. Disponível em: <https://doi.org/10.5772/intechopen.1011667>.

RIEGER, G. **socat: Multipurpose relay (SOcket CAT)**. 2025. Accessed: 2025-11-01. Disponível em: <http://www.dest-unreach.org/socat/>.

SABANI, M. E.; SAVVAS, I. K.; GARANI, G. Learning with errors: a lattice-based keystone of post-quantum cryptography. **Signals**, MDPI, v. 5, n. 2, p. 216–243, 2024. Disponível em: <https://doi.org/10.3390/signals5020012>.

SAHU, P. et al. Understanding sidecars in cloud orchestration. In: **Proceedings of the 3rd Workshop on Serverless Systems, Applications, and Methodologies**. [s.n.], 2025. p. 1–11. Disponível em: <https://doi.org/10.1145/3721465.3721862>.

SALCEDO-NAVARRO, A.; GARCIA-PINEDA, M.; GUTIÉRREZ-AGUADO, J. K8sidecar: A modular kubernetes chain of sidecar proxies for microservices and serverless architectures. **Software: Practice and Experience**, Wiley Online Library, v. 55, n. 8, p. 1305–1319, 2025. Disponível em: <https://doi.org/10.1002/spe.3423>.

SANON, S. P.; SCHOTTEN, H. D. Securing Mobile Networks in the Quantum Era: Imperative Role of Post-Quantum Cryptography. In: **2025 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)**. [s.n.], 2025. p. 721–726. ISSN 2575-4912. ISSN: 2575-4912. Disponível em: <https://doi.org/10.1109/EuCNC/6GSummit63408.2025.11037057>.

SAOKAR, H. et al. ServiceRouter: Hyperscale and minimal cost service mesh at meta. In: **17th USENIX Symposium on Operating Systems Design and Implementation (OSDI 23)**. Boston, MA: USENIX Association, 2023. p. 969–985. ISBN 978-1-939133-34-2. Disponível em: <https://www.usenix.org/conference/osdi23/presentation/saokar>.

SCALISE, P. et al. A systematic survey on 5g and 6g security considerations, challenges, trends, and research areas. **Future Internet**, v. 16, n. 3, 2024. ISSN 1999-5903. Disponível em: <https://www.mdpi.com/1999-5903/16/3/67>.

TAID, R. 5g core network architecture. In: _____. **Mobile Communications Systems Development: A Practical Introduction to System Understanding, Implementation and Deployment**. https://ieeexplore.ieee.org/servlet/opac?bknumber=10523201, 2021. p. 447–472. Disponível em: <https://doi.org/10.1002/9781119778714.ch20>.

WANG, X.; XU, G.; YU, Y. Lattice-based cryptography: A survey. **Chinese Annals of Mathematics, Series B**, Springer, v. 44, n. 6, p. 945–960, 2023. Disponível em: <https://doi.org/10.1007/s11401-023-0053-6>.

XIAO, L. et al. Distributed shor's algorithm. **arXiv preprint arXiv:2207.05976**, 07 2022. Disponível em: <https://doi.org/10.48550/arXiv.2207.05976>.

ZONG, C. The mathematical foundation of post-quantum cryptography. **Research**, v. 8, p. 0801, 2025. Disponível em: <https://spj.science.org/doi/abs/10.34133/research.0801>.