future internet

MDPI

Article

Towards Quantum-Safe O-RAN: Experimental Evaluation of ML-KEM-Based IPsec on the E2 Interface

Mario Perera [1], Michael Mackay [1], Max Hashem Eiza [1,*], Alessandro Raschella [1], Nathan Shone [1] and Mukesh Kumar Maheshwari [2]

[1] School of Computer Science and Mathematics, Liverpool John Moores University, Liverpool L3 3AF, UK; b.m.perera@2024.ljmu.ac.uk (M.P.); m.i.mackay@ljmu.ac.uk (M.M.); a.raschella@ljmu.ac.uk (A.R.); n.shone@ljmu.ac.uk (N.S.)

[2] Department of Electrical Engineering, Bahria University, Karachi Campus, Karachi 75260, Pakistan; mukeshkumar.bukc@bahria.edu.pk

* Correspondence: m.hashemeiza@ljmu.ac.uk

Abstract

As Open Radio Access Network (O-RAN) deployments expand and adversaries adopt "store-now, decrypt-later" strategies, operators need empirical data on the cost of migrating critical control interfaces to post-quantum cryptography (PQC). This paper experimentally evaluates the impact of integrating a NIST-aligned Module-Lattice Key-Encapsulation Mechanism (ML-KEM) into IKEv2/IPsec, protecting the E2 interface between the 5G Node B (gNB) and the Near-Real-Time RAN Intelligent Controller (Near-RT RIC). Using an open-source testbed built from srsRAN, Open5GS, FlexRIC and strongSwan (with liboqs), we compare three configurations: no IPsec, classical Elliptic Curve Diffie–Hellman (ECDH)-based IPsec, and ML-KEM-based IPsec. This study focuses on IPsec tunnel-setup latency and the runtime behaviour of Near-RT RIC xApps under realistic signalling workloads. Results from repeated, automated runs show that ML-KEM integration adds a small overhead to tunnel establishment, which is approximately 2.7~4.7 ms in comparison to classical IPsec, while xApp operation and RIC control loops remain stable in our experiments. These findings, produced from an open, reproducible testbed, indicate that ML-KEM-based IPsec on the E2 interface is practically feasible and inform quantum-safe migration strategies for O-RAN deployments.

Keywords: 5G network; post-quantum cryptography; PQC; OpenRAN; O-RAN; IPSec; ML-KEM

Academic Editors: Mario Di Mauro and Francesco Pascale

Received: 11 February 2026
Revised: 20 March 2026
Accepted: 23 March 2026
Published: 1 April 2026

1. Introduction

Secure communication protocols are fundamental to trustworthy mobile networks in increasingly complex fifth-generation (5G) deployments. Open Radio Access Networks (O-RANs), aka Open-RANs, have emerged as a key architectural paradigm for 5G and beyond, enabling service heterogeneity, multi-vendor interoperability, virtualisation, and AI-driven intelligence. Specifically, O-RAN disaggregates traditional base station functions into separate units which are interconnected through open interfaces and managed and optimised by centralised control entities, namely the RAN Intelligence Controller (RIC) [1]. This disaggregation, however, also significantly enlarges the attack surface, making interface protection a central security concern [2], and the O-RAN Alliance Working Group (WG11) specifies mandatory controls such as TLS and IPsec for inter-node communication [3]. Among these, the E2 interface linking the 5G Node B (gNB) with the RIC is

particularly critical, as it carries near-real-time control and telemetry data and is subject to stringent timing constraints. WG11 mandates IPsec on E2 to ensure confidentiality and integrity, but stronger cryptographic mechanisms may impact latency and resource usage in ways that are not yet fully explored [3,4]. In parallel, advances in quantum computing threaten widely deployed public-key primitives and, thus, the long-term security of IPsec and TLS. This has motivated intensive work on post-quantum cryptography (PQC) and the standardisation of quantum-resistant schemes. The National Institute of Standards and Technology (NIST) has recently finalised FIPS 203, specifying the Module-Lattice-based Key-Encapsulation Mechanism (ML-KEM, formerly CRYSTALS-Kyber) as a general-purpose KEM for quantum-safe key establishment, with parameter sets ML-KEM-512/768/1024 that trade performance against security [5]. Integrating such primitives into O-RAN interface security is essential to mitigate "store-now, decrypt-later" attacks over the lifetime of 5G and future sixth-generation (6G) systems.

Existing work has begun to address both sides of this problem. On the O-RAN side, several studies analyse architectural security risks, interface threats, and mitigation strategies and provide comprehensive reviews of O-RAN security challenges and opportunities (e.g., [6]). Other works quantify the impact of classical encryption on O-RAN interfaces, including E2 and Open Fronthaul, using testbeds and network emulators (e.g., [7]). On the PQC side, there are empirical evaluations of PQC KEMs in 5G core networks (e.g., PQC-enabled TLS in free5GC) and in other wireless contexts, focusing mainly on handshake latency and message sizes rather than RAN control interfaces [8]. Despite this progress, a critical gap remains in which there is limited empirical evidence on how ML-KEM-based key encapsulation, when integrated into IPsec in line with O-RAN WG11 guidance, affects the performance of the E2 interface in realistic O-RAN deployments. Particularly, network operators lack open, reproducible testbeds and quantitative measurements that characterise the trade-off between quantum-resistant security and near-real-time responsiveness for RIC-driven control.

Against this background, this paper investigates whether ML-KEM-based post-quantum key encapsulation imposes significant performance issues when used to protect the E2 interface of 5G O-RAN via IPsec. To achieve this aim, this paper pursues three objectives. Firstly, we review 5G O-RAN architectures, security requirements of O-RAN interfaces, and PQC algorithms and focus on ML-KEM as a suitable candidate for securing E2 in accordance with emerging standards. ML-KEM was selected, as it is currently the most mature PQC KEM supported by prototype implementations of the IKEv2/IPsec stack. Thus, it represents a practical near-term candidate for post-quantum deployment in network security protocols. Secondly, we design and deploy an open-source, reproducible experimental platform emulating a 5G O-RAN with Near-Real Time RIC (Near-RT RIC), using components srsRAN [9], Open5GS [10], FlexRIC [11], and strongSwan with ML-KEM support [12] to realise IPsec-based E2 protection. Finally, we quantify the impact of ML-KEM-based IKEv2 on the E2 interface, comparing (1) no IPsec, (2) classical ECDH-based IPsec, and (3) ML-KEM-based IPsec in terms of tunnel setup latency and the runtime behaviour of Near-RT RIC xApps.

Hence, the novel contributions of this paper are threefold: (1) it presents, to the best of our knowledge, the first empirical evaluation of ML-KEM-based IPsec on the O-RAN E2 interface using an open, standards-aligned 5G O-RAN testbed; (2) it provides a documented, reproducible framework for studying PQC-enhanced interface security in O-RAN; (3) it delivers quantitative evidence that ML-KEM integration primarily increases IPsec tunnel setup latency by a few milliseconds while introducing no observable disruption to Near-RT RIC or xApp operation, thereby informing quantum-safe migration strategies for O-RAN deployments.

https://doi.org/10.3390/fi18040188

The rest of this paper is organised as follows. Section 2 presents a literature review covering the O-RAN architecture and interface security, prior work on PQC integration in network protocols, and recent industry and standards activity relevant to quantum-safe RANs. Section 3 describes the testbed design and measurement methodology, including component selection, topology, IPsec/IKEv2 profiles, and the signalling workloads used to exercise the E2 interface. Section 4 details the implementation of the testbed and its configuration. Section 5 reports the evaluation results and analysis of tunnel-setup latency, xApp runtime behaviour under the three security modes, and a discussion of operational implications. Finally, Section 6 concludes and summarises the main findings and introduces ideas for future work.

2. Related Works

This section introduces the O-RAN architecture, including its interfaces and security. After that, it discusses the latest works on PQC evaluation in network protocols (IPsec/TLS) and recent industry and standards activity towards quantum-safe telecoms.

2.1. O-RAN Architecture

The O-RAN Alliance rearchitects the 5G RAN by extending the 3GPP Next Generation (NG-RAN) "split" gNB into a fully disaggregated, cloud-native, and programmable platform. Building on the logical separation of the gNB into a Central Unit (CU), Distributed Unit (DU), and Radio Unit (RU) in 3GPP Release 15, O-RAN renames these elements O-CU, O-DU, and O-RU, respectively, and standardises open, interoperable interfaces between them. This approach inherits ideas from Cloud (C-RAN) and Virtualised RAN (V-RAN) and goes further by mandating the openness of interfaces and embedding intelligence as a first-class design goal [13]. It is now the reference architecture for most academic and industrial work on Open RAN [14]. The O-RAN architecture proposed by the O-RAN Alliance is depicted in Figure 1.

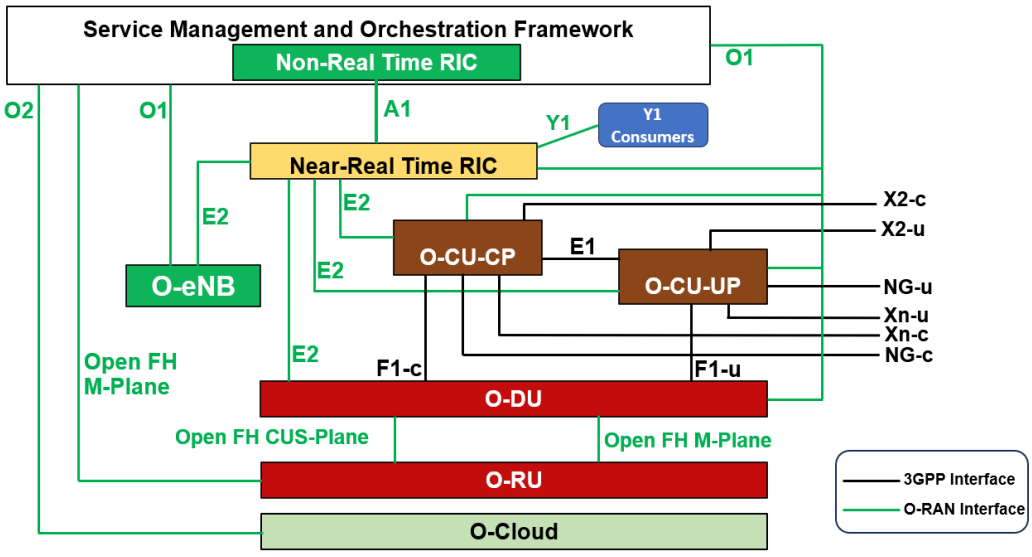

Figure 1. Logical architecture of O-RAN [15].

Within the RAN domain, the O-CU serves as the intelligent control hub that acts as the bridge between the RAN and the core network. It handles higher-layer protocol processing up to the Packet Data Convergence Protocol (PDCP), terminates the NG interface towards the 5G core, and coordinates with the O-DU over the 3GPP F1-C/F1-U interfaces. The O-DU performs mid-layer RAN functions such as Radio Link Control (RLC), MAC, and upper PHY, and it aggregates multiple O-RUs via the Open Fronthaul interface. The O-RU

is responsible for Radio Frequency (RF) processing and lower PHY, positioned close to the antenna to meet tight timing constraints while leveraging the O-DU for more compute-intensive baseband tasks. These virtualised network functions (O-CU, O-DU, and, in many deployments, the Near-RT RIC) execute on an O-Cloud infrastructure, which provides the underlying compute, storage, and networking platform for O-RAN Virtual Network Functions (VNFs) and Cloud Native Functions (CNFs).

Within the Service Management and Orchestration (SMO) domain, the SMO framework acts as the central management plane for the RAN. It consolidates traditional Operations Support System (OSS) and Network Management System (NMS) functions with cloud-native lifecycle management and coordinates RAN optimisation via the RIC. The RIC is split into a Non-Real-Time RIC (Non-RT RIC) operating at time scales above 1 s and hosted within the SMO and a Near-Real-Time RIC (Near-RT RIC) targeting 10 ms–1 s control loop for Radio Resource Management (RRM) and policy enforcement. Non-RT RIC rApps perform long-term optimisation, analytics, and model training, while Near-RT RIC xApps execute fine-grained control actions such as load balancing, interference management, and radio-bearer optimisation in close interaction with the O-CU/O-DU. This decoupled, application-centric control framework is one of the key architectural differentiators of O-RAN.

These components are interconnected through a set of standardised open interfaces that form the backbone of O-RAN's interoperability. The A1 interface links the Non-RT RIC (in the SMO) to the Near-RT RIC, carrying high-level policies, machine learning (ML) models, and aggregated key performance indicators (KPIs) for policy-driven optimisation. The E2 interface connects the Near-RT RIC to O-CU and O-DU, exposing near-real-time telemetry and control data that enables xApps to implement closed-loop optimisation while respecting RAN timing constraints. The O1 interface provides Fault, Configuration, Accounting, Performance, and Security (FCAPS) management between the SMO and O-RAN nodes (O-CU, O-DU, O-RU, and Near-RT RIC), whereas the O2 interface connects the SMO to the O-Cloud for orchestration of virtualised RAN resources. At the fronthaul, the Open Fronthaul (O-FH) interface between O-DU and O-RU is specified with distinct control, user, synchronisation, and management planes to support multi-vendor interoperability and flexible deployment options. Collectively, these components and interfaces realise a modular, cloud-native RAN that is open to multi-vendor implementations and programmable control while also introducing new interface surfaces that must be secured.

2.2. O-RAN Security and Post-Quantum Cryptography (PQC)

The openness and disaggregation of O-RAN, the virtualisation of RAN functions on O-Cloud infrastructure, the introduction of programmable RIC platforms running xApps/rApps, and the exposure of multiple interfaces all create additional points where adversaries can target control and management traffic. The increased exposure covers both passive and active threats. For instance, adversaries can perform passive eavesdropping [16], active man-in-the-middle [17], or packet-injection attacks that tamper with control messages [16], protocol-level manipulations that subvert RIC decisions [18], and resource-exhaustion attacks (e.g., Denial-of-Service (DoS) [19]). Threat modelling studies and recent security reports, such as [20], consistently highlight risks such as DoS against control-plane interfaces, zero-day exploits, supply-chain compromise of software components, and attacks on AI/ML pipelines within the RIC. In particular, the E2 interface, which carries near-real-time control and telemetry data between the Near-RT RIC and O-CU/O-DU, is recognised as highly sensitive. Successful attacks on the E2 can directly degrade RRM, QoS, and the stability of deployed xApps. Another major class of attacks involves adversarial ML, where attackers manipulate telemetry and/or training data to mislead

ML-based xApps, significantly degrading classification accuracy and network performance (e.g., exploiting traffic-steering xApps to obtain disproportionate allocations) [21].

To address these risks, O-RAN Alliance Security WG11 has defined a set of mandatory security controls per interface, specifying the use of standardised cryptographic protocols to provide confidentiality, integrity, entity authentication, and replay protection [22]. In the current framework, TLS (1.2/1.3) is mandated for many management and policy interfaces (e.g., O1, O2, and A1 in Figure 1), MACsec or IPsec is recommended for Open Fronthaul, and IPsec is mandated for E2, where tunnel-based protection aligns with the need for strong, end-to-end security between logically separate domains. From a protocol-design viewpoint, these controls are realised by combining symmetric ciphers, asymmetric key exchange and digital signatures, and cryptographic hash functions within TLS and IPsec handshakes. While TLS is widely used for secure client–server and management communications at the transport layer, IPsec operates at the network layer to protect inter-component tunnels such as E2 and is the focus of this work.

Furthermore, the O-RAN Next-Generation Research Group (nGRG) stresses that protecting O-RAN deployments requires moving beyond interface encryption to platform-level assurances [20]. It advocates trusted-computing techniques backed by hardware roots of trust (e.g., Trusted Platform Module (TPM), CPU enclaves) and layered remote attestation (e.g., IETF Remote ATtestation procedureS (RATS) and Trusted Computing Group Device Identifier Composition Engine (DICE)) to establish and maintain trust in O-RAN components, O-Cloud hosts, and management systems. In this view, a trusted O-RAN is one where O-RU/O-DU/O-CU, RIC platforms, and cloud infrastructure can cryptographically prove identity and integrity at boot and runtime, enabling zero-trust enforcement decisions across the architecture [23]. This platform-security perspective complements the protocol-centric view adopted by WG11 and reinforces the need to ensure that PQC-enabled IPsec and TLS can run on these trustworthy platforms.

However, the long-term security of these TLS and IPsec deployments is threatened by advances in quantum computing. Both protocols typically rely on public-key primitives such as RSA, Diffie–Hellman, and ECC for key establishment and authentication. These primitives can be solved efficiently by Shor's algorithm, while Grover's algorithm weakens the effective security of symmetric ciphers. As noted by the O-RAN nGRG report [24], although current quantum hardware cannot yet break deployed schemes, the "harvest-now, decrypt-later" threat is already relevant because mobile network traffic often has long confidentiality lifetimes and infrastructure generations last 15–20 years or more. This motivates an early transition to quantum-safe mechanisms that can be integrated into existing protocol frameworks without requiring quantum hardware.

PQC provides such mechanisms that can resist known quantum attacks while remaining implementable on classical hardware. NIST's multi-year standardisation process has now culminated in the publication of FIPS 203, 204, and 205, which specify ML-KEM and related signature schemes as the first generation of PQC standards [25]. ML-KEM offers three parameter sets (ML-KEM-512/768/1024) that trade off security strength and performance, and it is designed specifically for use as a drop-in replacement for classical KEMs in protocols such as TLS and IKEv2. Surveys and position papers (e.g., [26]) on PQC for networks emphasise that, given the long lead times historically observed when transitioning from one public-key paradigm to another (e.g., from RSA to ECC), experimentation and staged deployment must begin well before quantum adversaries become practical.

Recent research has begun to explore how PQC can be integrated into the security protocols relevant to 5G and beyond. Scalise et al. [8] extended the free5GC core network with PQC KEMs for TLS 1.3, showing modest increases in handshake time and packet size but no prohibitive overheads in VNF-to-VNF signalling. García et al. [27] proposed hybrid

and triple-hybrid TLS 1.3 schemes that combine PQC, classical cryptography, and Quantum Key Distribution (QKD) to achieve quantum-resistant authenticated key exchange while maintaining practical performance. Rathi et al. [28] presented Q-RAN, a full-stack PQC framework for O-RAN. It integrates ML-KEM for encryption and ML-DSA for signatures and quantum entropy into all control-plane protocols. Specifically, PQ-IPsec, PQ-DTLS, and PQ-mTLS are deployed on every O-RAN interface, anchored by a centralised post-quantum certificate authority. They showed that it is feasible to retrofit O-RAN with hybrid PQC handshakes (e.g., TLS/IPsec) using ML-KEM and ML-DSA while retaining compatibility. Other studies provide generic performance evaluation frameworks for PQC-enabled TLS, enabling systematic comparison of different NIST-selected KEMs and signature schemes [29].

For IPsec and lower-layer protocols, Bae et al. [30] evaluated PQC-integrated IKEv2 in IPsec, quantifying the overheads of different KEMs in tunnel establishment and showing that, although PQC increases computational load and handshake time, the impact can be acceptable in many deployment scenarios. Most recently, a study on introducing PQC algorithms in O-RAN interfaces demonstrated how PQC KEMs can be integrated into IPsec to protect Open Fronthaul links, using strongSwan and liboqs and reporting throughput, delay, jitter, and resource usage for multiple KEM choices [31]. That work confirms both the feasibility of PQC-enabled IPsec in the O-RAN context and the importance of carefully characterising performance trade-offs. The O-RAN nGRG complements these efforts by mapping PQC use cases across 3GPP and O-RAN interfaces, including E2, and outlining migration strategies such as crypto-agile designs and staged deployment of PQC and QKD [24].

2.3. PQC for O-RAN—Latest Industrial and Standardisation Efforts

A recent collaboration between PQC vendor Patero and O-RAN vendor Eridan demonstrated Patero's CryptoQoR PQC suite running on Eridan's 5G radios [32]. They set up a private 5G Open RAN network in which Patero's post-quantum encryption protected the data path. The demo showed end-to-end quantum-resistant 5G communication using cryptographic algorithms aligned with the new NIST PQC standards. This proof of concept highlights that commercial O-RAN hardware can support PQC without changing the radio stack, providing a quantum-safe solution for critical infrastructure.

In 2024, Nokia and Turkcell successfully trialled a quantum-safe mobile transport link [33]. In their demo, Nokia's IPsec Security Gateway in Turkcell's network was configured with post-quantum algorithms (NIST ML-KEM, ML-DSA, etc.) to secure subscriber data. The partners reported a "world-first" implementation of quantum-safe IPsec in a live 5G mobile network, effectively protecting current traffic against future quantum attacks. While this test was on the transport network rather than a disaggregated O-RAN interface, it demonstrates the viability of PQC-IPsec in telecom networks and sets a precedent for extending such solutions into O-RAN deployments.

In the 3GPP ecosystem, studies have begun migrating cellular cryptography to PQC [34]. 3GPP TR 33.938 provides a cryptography inventory for 5G/6G and outlines how NIST PQC algorithms might replace existing keys. Future 3GPP work (SA3 security specifications) will evaluate hybrid and pure-PQC handshakes for procedures like Subscription Concealed Identifier (SUCI) concealment and TLS/IKE tunnels.

Despite this growing body of work, an analytic gap remains at the intersection of PQC and O-RAN interface security. While recent work has explored PQC-enabled IPsec in O-RAN, its focus has been primarily on Open Fronthaul and hybrid key exchange rather than on fully ML-KEM-based key establishment for Near-RT control. In this context, this work positions itself as a focused contribution on quantum-safe O-RAN security. Building

on the cryptographic and protocol background above and leveraging NIST-standardised ML-KEM and PQC-enabled IPsec implementation strongSwan, it provides an empirical evaluation of ML-KEM-based key encapsulation for IPsec on the 5G E2 interface. By quantifying the impact on tunnel setup latency and the behaviour of Near-RT RIC xApps, this work addresses a key open question for operators: whether quantum-safe protection of critical O-RAN control interfaces can be achieved without compromising stringent performance requirements.

3. Testbed Design

3.1. Design Rationale and Components Selection

The objective of the experimental design is a reproducible, standards-aligned 5G O-RAN testbed that can compare three security configurations for the E2 interface: (1) no IPsec (i.e., a baseline), (2) classical IPsec using ECDH key exchange, and (3) IPsec using a NIST-standardised post-quantum KEM (ML-KEM) for IKEv2. As mentioned above, ML-KEM represents a practical near-term candidate for PQC deployment in network security protocols. Note that other families of post-quantum KEMs exist, including code-based and lattice-based alternatives. These schemes differ significantly in terms of public-key size, ciphertext size, and computational cost, which can influence the behaviour of network security protocols such as IKEv2. For instance, schemes with substantially larger public keys or ciphertexts may increase the size of the initial key exchange payloads, potentially introducing additional packet fragmentation or transport overhead depending on network MTU constraints. This can introduce additional latency or retransmission overhead. In the case of ML-KEM, the key and ciphertext sizes remain within a range that can typically be accommodated within a small number of IKEv2 packets, which helps limit the impact on handshake performance. However, KEMs with larger parameter sizes could lead to increased fragmentation and, therefore, higher handshake latency.

The testbed should align with O-RAN alliance WG11 security specifications for interface protection and be built on widely available commercial-off-the-shelf (COTS) hardware and open-source components to enable reproducibility. Moreover, it must implement realistic RAN control and user-plane behaviour, including Near-RT RIC xApp signalling, and allow instrumentation of tunnel setup latency and steady-state resource metrics. Finally, it should allow straightforward substitution of KEM implementations (e.g., liboqs/strongSwan). These constraints will inform component selection for our designed testbed. To balance realism and reproducibility, our design will utilise an open-source, containerisable software stack including srsRAN for RAN and UE emulation [10], Open5GS as the 5G Core [35], FlexRIC as the Near-RT RIC platform running representative xApps [11], and strongSwan (patched with liboqs) for IPsec/IKEv2 with both classical and PQC KEMs [12]. The choice of srsRAN/Open5GS/FlexRIC provides a standards-conformant E2 emulation while remaining deployable on commodity server hardware. Alternative options such as the Colosseum testbed [36], Free5GC [37], O-RAN Software Community (OSC) RIC [38], and the Liverpool 5G simulator [39] were explored but not selected based on pragmatic considerations, ensuring feasibility, reliability, and reproducibility within the scope of this work.

It is worth noting that this work focuses on a standalone ML-KEM-based integration to isolate and evaluate the impact of post-quantum key exchange on IKEv2/IPsec performance within an O-RAN testbed. In practical deployments, PQC mechanisms are expected to be introduced using hybrid approaches, where classical and post-quantum algorithms are combined rather than used in isolation. For instance, within the IKEv2 framework, hybrid key exchange can be realised by combining a classical Diffie–Hellman exchange (e.g., Curve 25519) with a PQC KEM such as ML-KEM. In such a configuration, both key exchanges

contribute to the derivation of the shared secret used to generate SKEYSEED, typically by concatenation or a key derivation function that incorporates both inputs. As a result, the established IKE SA remains secure as long as at least one of the underlying key exchange mechanisms remains unbroken. Evaluating such hybrid modes is outside the scope of this work. However, the proposed testbed is extensible to hybrid configurations.

3.2. Testbed Components

An outline of the software components that collectively enable the deployment and evaluation is as follows.

srsRAN—In our testbed, O-RAN with its subcomponents and internal interfaces is based on srsRAN and is a complete O-RAN solution developed by Software Radio Systems. srsRAN adheres to the 3GPP 5G system architecture by implementing functional splits between DU and CU. Furthermore, the CU is disaggregated into the control plane (CU-CP) and the user plane (CU-UP), enabling greater flexibility and scalability in the network design, making it a complete O-RAN implementation [9].

Open5GS is an open-source implementation for the 5G Core, complying with the 3GPP Release-17 specification. Documentation is available for building a 5G Core using Open5GS here [10]. Most notably, the proposed framework features a lightweight, containerised deployment of the Open5GS 5G Core, simplifying the development of the testbed and minimising the effort required for 5G Core deployment and integration.

srsUE—The srsRAN implementation does not include a UE implementation by default. Instead, the prototype 5G UE (srsUE) of srsRAN 4G can be used for the end-to-end test network [40]. srsUE is suitable for testing srsRAN in preliminary testing and proof-of-concept scenarios without using expensive hardware. In our testbed, neither UE devices nor hardware radio units are used. Instead of transmitting/receiving real RF signals using hardware modules, srsRAN implements an abstraction to emulate the radio channel between the UE and the gNB using ZeroMQ sockets.

ZeroMQ is a messaging library suitable for the high-performance requirements of distributed or concurrent applications [41]. ZeroMQ's support for a range of common messaging patterns across multiple transport mechanisms enables it to be effectively used in radio channel emulation. This implementation makes it perfect for lab testing and development.

FlexRIC is an open-source near-RT RIC and an E2 node emulator agent. It has implementations of E2SM-KPM v2.01/v2.03/v3.00 and E2SM-RC v1.03 service models and includes several xApps compliant with O-RAN Alliance specifications. FlexRIC is developed with the flexibility of development and deployment at its core [42]. It enables building specialised service-oriented controllers while running its components and xApps in a set of orchestrated containers in isolation.

All these open-source components are widely used in the research community for evaluating the deployment and performance of O-RAN networks and, therefore, provide a reliable basis for our experiments.

3.3. Testbed Topology and IPSec Placement

The topology for our testbed had evolved through three design stages:

(1) Minimal Functional Topology (Proof of Concept): In this stage, a single host runs srsRAN and Open5GS with a collocated RIC to validate basic E2 signalling and IKEv2/IPsec mechanics. This stage verified functional compatibility between RAN, RIC, and IPsec stacks and established measurement baselines.
(2) Distributed Software Topology (Isolation and Instrumentation): Functions were separated into distinct virtual machines/containers, including (a) O-CU/O-DU (srsRAN),

(b) 5G Core (Open5GS), (c) Near-RT RIC (FlexRIC + xApps), and (d) security gateway endpoints (strongSwan). Separating these roles enabled realistic network hops, latency characterisation, and per-node resource monitoring. It also allowed the IPsec tunnel endpoints to be placed in realistic locations. For instance, between the O-CU host and the Near-RT RIC host to emulate cross-domain E2 protection.

(3) The final reproducible testbed is shown in Figure 2 below. The final design consolidates lessons from earlier stages into a modular topology suitable for repeatable experiments.

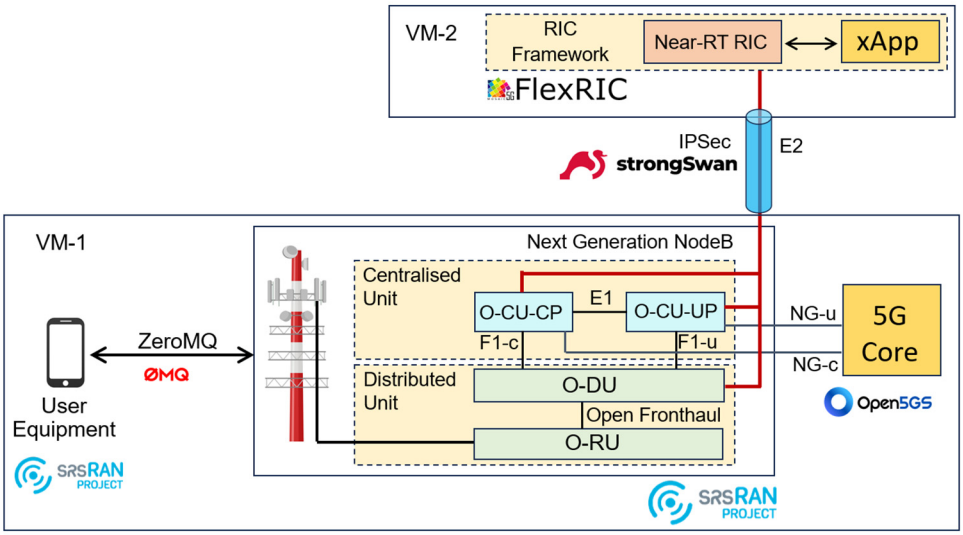

Figure 2. Final design of the experimental system.

The key elements are as follows:

- O-RU/O-DU/O-CU (srsRAN Split): srsRAN implements the gNB split (DU + CU functions). In the testbed, the logical gNB endpoint of E2 is hosted on the O-CU node.
- Near-RT RIC (FlexRIC): This is hosted on a separate node reachable over the emulated network. It runs representative xApps to generate realistic E2 control traffic and closed-loop signalling.
- IPsec Endpoints (strongSwan): Configured to protect the E2 interface between the O-CU and Near-RT RIC. The IKEv2 configuration can be toggled between classical (ECDH) and ML-KEM key exchanges. ML-KEM is provided by liboqs/strongSwan integration and uses NIST-recommended parameter sets for experimentation.
- Core Network (Open5GS) and UE Traffic: deployed to produce normal session activity and user plane load where required by specific experiments, but the E2 path is the primary measurement focus.
- Measurement and Orchestration Hosts: Dedicated hosts and scripts for triggering tunnel setups, capturing packet traces, and measuring tunnel setup latency (from IKE_SA_INIT to IKE_AUTH completion and IPsec SA/ESP readiness). All nodes are time-synchronised to ensure accurate latency measurement.

In Figure 2, IPsec is realised using strongSwan with IKEv2. For classical experiments, baselinecurve 25519 key exchange is used. For PQC experiments, the IKEv2 exchange is altered to use ML-KEM (CRYSTALS-Kyber/NIST ML-KEM) either as a pure KEM or in hybrid mode (classical + PQC). The implementation leverages liboqs support compiled into strongSwan's IKE daemon, allowing selection among ML-KEM parameter sets (512/768/1024) to study performance trade-offs. The design keeps ESP protection (symmetric encryption) unchanged across configurations so that the measured differences focus on the key-establishment phase. The IPsec endpoint placement emulates a realistic cross-

domain tunnel between the O-CU host and the Near-RT RIC host, consistent with WG11 guidance. Finally, note that the ML-KEM-512 parameter set was not included in our evaluation, as it provides a lower security level and is generally not considered suitable for long-term security in critical infrastructure scenarios such as O-RAN.

3.4. Workloads and Metrics

Workloads are designed to stress both the control plane and the cryptographic handshake. First, the E2 signalling workload involves xApps on the Near-RT RIC that generate periodic control messages and subscriptions to exercise E2 procedures at realistic rates to observe if tunnel establishment disrupts control loops. Secondly, UE/session traffic and uplink/downlink data flows are generated to validate that IPsec payload processing does not create steady-state overheads under PQC. In terms of measured metrics, we have identified the following. Tunnel setup latency measures the time between initiating IKEv2 and the completion of IPsec Security Association (SA) establishment, which is measured from packet captures and IKE logs. xApp stability/control-loop impact is used to monitor message processing times and any error/recovery events at the Near-RT RIC.

Note that estimating CPU utilisation during the IPsec handshake using standard Linux monitoring tools available within the virtualised testbed was not feasible. The obtained measurements proved difficult to interpret reliably because the experiments were conducted on VMs sharing host resources, where hypervisor scheduling and background processes introduce variability that obscures the incremental cost of individual cryptographic operations. As a result, the collected CPU utilisation data did not provide sufficiently stable or reproducible results to support a rigorous quantitative comparison. Despite this limitation, the observed tunnel establishment and message latency results provide an indirect indication of computational overhead. In future work, we plan to extend this evaluation to hardware-based testbeds and edge computing platforms representative of operational O-RAN deployments, where more precise measurements of CPU utilisation and energy consumption can be obtained. Finally, the following instruments are used: tcpdump to capture packets at E2 endpoints, IKEv2 debug logs, system resource monitors, and automated scripts to repeat experiments across multiple runs for statistical confidence.

4. Testbed Implementation

The implementation creates a multi-host software testbed using open-source components: srsRAN (gNB/UE), Open5GS (5G Core), FlexRIC (Near-RT RIC and xApps), and strongSwan (IPsec/IKEv2). The strongSwan instance is compiled with liboqs to support the NIST-standardised ML-KEM (formerly CRYSTALS-Kyber) parameter sets. The testbed was deployed on a high-performance server with dual Intel Xeon Gold 6248 CPUs at 2.50 GHz, 256 GB of DDR4 ECC RAM, and VirtualBox v7.1.12 to support the creation of VMs. The following VMs were configured as shown in Figure 3 below. VM1 has two vCPUs and 16GB of memory, runs Ubuntu 22.04 LTS, and includes two network interfaces: (1) enp0s8, internal network only for interconnection; and (2) enp0s9, internal network only for management. VM2 has a similar configuration to VM1 except for 8GB of memory. Finally, a third virtual machine, which is shown as “server” in Figure 3, with the same configurations as that of VM2 was configured to act as the Certificate Authority (CA) for IPSec authentication.

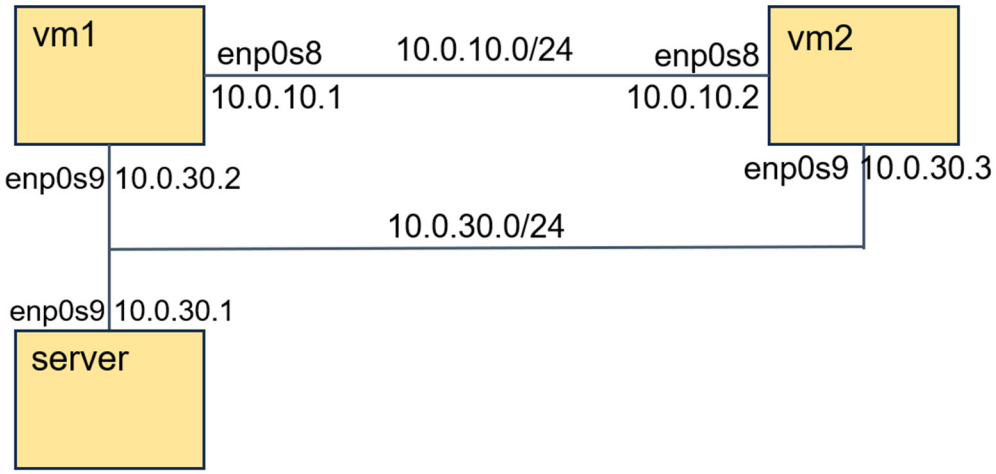

Figure 3. VM configurations & connectivity.

4.1. Deployment of 5G Network Components

4.1.1. srsUE

srsUE is part of srsRAN 4G, but the included experimental 5G UE can be used for our purposes. The installation can be done following the steps shown below, assuming that dependencies are already installed.

`# git clone` https://github.com/zeromq/czmq.git (accessed on 12 August 2025)
`# make`
`# git clone` https://github.com/srsRAN/srsRAN_4G.git (accessed on 12 August 2025)
When compiling srsRAN 4G, ZMQ should be enabled.

```
/srsRAN_4G/build# cmake ../-DENABLE_EXPORT=ON -DENABLE_ZEROMQ=ON
/srsRAN_4G/build# make
/srsRAN_4G/build# make test
```

The last command verifies the installation by running the built-in test. Upon successful installation, the executable file srsue should be available in `srsRAN_4G/build/srsue/src/`.

4.1.2. srsRAN

After installing the build tools and mandatory requirements, we clone the git repository.

`# git clone` https://github.com/srsRAN/srsRAN_Project.git (accessed on 12 August 2025)
ZeroMQ is disabled by default, so it should also be enabled here when building as follows:

```
# cmake ../-DENABLE_EXPORT=ON -DENABLE_ZEROMQ=ON
# make -j $(nproc)
# make install
```

4.1.3. Open5GS

The srsRAN project provides a containerised deployment of the Open5GS core network, a Docker container, significantly simplifying the process of setting up a functional 5G core. Assuming that Docker Compose is installed, the 5G core included in the srsRAN project can be built as follows.

```
/srsRAN_Project/docker# docker compose --build 5gc
/srsRAN_Project/docker# docker compose start 5gc
[+] Running 1/1
Container open5gs_5gc started
```

4.1.4. FlexRIC

This stage involves the installation and configuration of FlexRIC, followed by verification of xApp integration, to ensure correct operation and interaction with the deployed 5G network components. These steps establish the foundation for subsequent performance evaluations involving secure E2 interface communication. After installing the necessary dependencies, we install FlexRIC as follows.

```
# git clone https://gitlab.eurecom.fr/mosaic5g/flexric.git (accessed on 12 August 2025)
~/flexric# git checkout br-flexric
~/flexric/build# cmake -DKPM_VERSION=KPM_V3_00 -DXAPP_DB=NONE_XAPP ../
~/flexric/build# make
~/flexric/build# make install
~/flexric/build# ctest
```

The last command will run the built-in test to verify the installation. Upon successful implementation, the executable file, nearRT-RIC, should be available in `flexric/build/examples/ric`. The final setup is shown in Figure 4 below, with established routes between gNodeB and RIC.

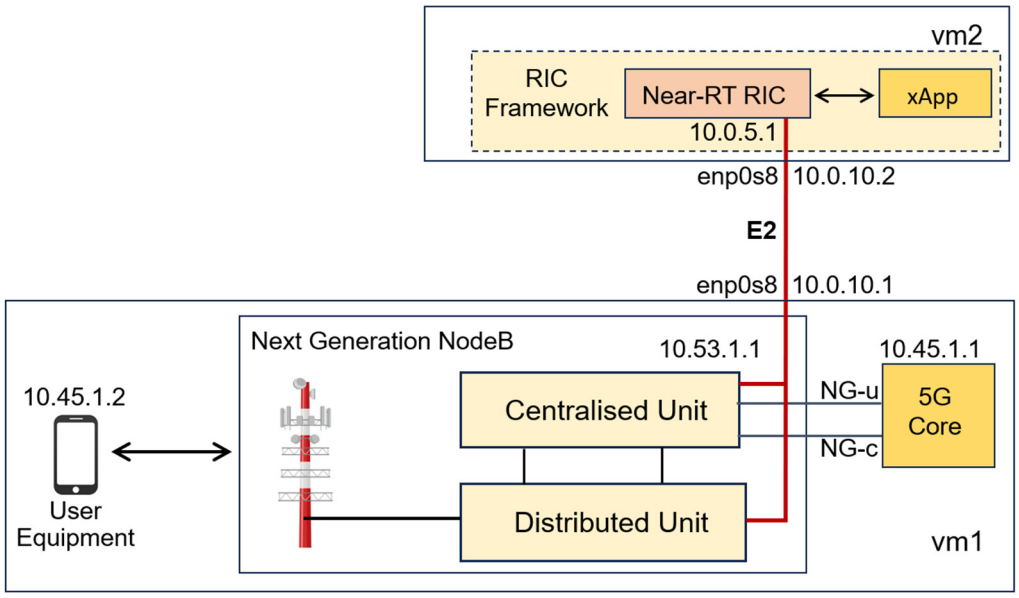

Figure 4. The final setup of the experimental system.

4.2. Implementation of Interface Security

In this section, we explain the integration of PQC into the O-RAN E2 interface using strongSwan with ML-KEM support. Note that support for PQC algorithms in strongSwan is available only from version 6.0.0 onwards. After installing dependencies, the latest source code of strongSwan 6.0.1 is installed via its website: https://download.strongswan.org/strongswan-6.0.1.tar.bz2 (accessed on 12 August 2025). While preparing for building, it is important to ensure that --enable-ml is used as an option to prepare the installation with ML-KEM cryptographic primitives. After compiling and installing strongSwan, the installation and ML-KEM algorithm support can be verified, as shown in Figure 5.

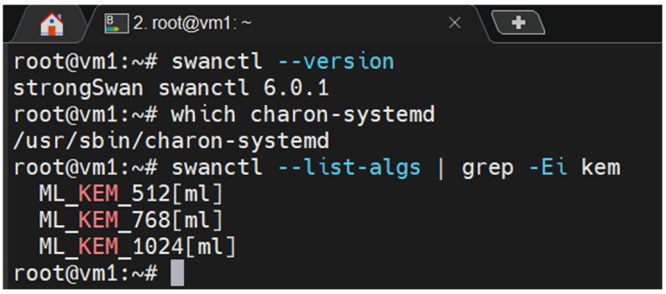

Figure 5. strongSwan version with ML-KEM support.

Key Confirmation and Failure Handling in IKEv2 with ML-KEM

In IKEv2, the initial message (IKE_SA_INIT) negotiates the IKE SA algorithms, exchanges nonces, and transports key-exchange material in the KE payload. In our PQC-enabled system, the KE payloads are used to carry the ML-KEM key-exchange material as implemented by the strongSwan ML-KEM branch (i.e., ML-KEM material is exchanged during IKE_SA_INIT and used to derive the shared secret). This shared secret is then fed into the standard IKEv2 key schedule to compute SKEYSEED and derive the per-direction encryption and integrity keys (SK_e/SK_a) used to protect subsequent IKEv2 messages [RFC7296]. IKEv2 provides implicit key confirmation at the start of the IKE_AUTH exchange. In this way, the IKE_AUTH messages are encrypted, and integrity is protected using keys derived from SKEYSEED, which was established in IKE_SA_INIT. They also authenticate the previous exchange via the AUTH payload. As a result, the successful decryption and integrity verification of IKE_AUTH and the validation of AUTH demonstrate that both sides derived the same key material from the negotiated key exchange. In a case where the secrets derived differ, the receiver will fail integrity/authentication checks, and IKE SA will not be established.

Since ML-KEM includes an explicit decapsulation step, our prototype treats decapsulation failure as a hard failure of IKE_SA establishment. Concretely, if the initiator cannot decapsulate the responder's ciphertext or obtains a shared secret that is not validated in subsequent checks, it aborts the handshake locally and clears the partial state. Consequently, the responder times out and deletes the half-open IKE_SA state. If an inconsistency occurs later (e.g., keys derived on each side do not match), the first protected IKE_AUTH message cannot be successfully verified, and the IKE SA creation fails in accordance with IKEv2 authentication/error-handling behaviour (i.e., authentication failure caused by an invalid shared secret).

Based on the setup of the two VMs, the interconnecting link IP address plans, and the subnets used in O-RAN and RIC can be used for designing IKE/IPSec policies for SAs. Configuration parameters for IPSec were as follows. ESP is used as the method to encrypt payloads for data secrecy, and the mode is set to tunnel mode. Authentication is performed using PKI, where the VM "server", as shown in Figure 3, is used as the CA. Encryption uses AES-256-GCM, KEM is the baseline Curve 25519, while PQC profile uses ML-KEM, and integrity is verified using prfsha256. Once all certificates are generated and signed, the configuration file/etc/swanctl/swanctl.conf is updated as per the designed IPSec policy. Figure 6 shows an example of the IPSec configuration file of VM1. Note that we had to change the IP address of the Near-RT RIC to run on 172.16.2.34 because, by default, both gNB and Near-RT RIC are configured to run on a single VM (i.e., the RIC runs on 127.0.0.1 by default).

```
connections {
   e2 {
      proposals = aes256gcm16-prfsha256-curve25519
      remote_addrs = 10.0.10.2
      local {
       auth = pubkey
       certs = vm1Cert.pem
     }
     remote {
       auth = pubkey
       id = "C=CH, O=strongswan,
CN=vm2.strongswan.org"
     }
     children {
       app1 {
         local_ts = 10.0.10.1/24, 172.16.1.0/27
         remote_ts = 10.0.10.2/24, 172.16.2.0/27
         esp_proposals = aes256gcm16-prfsha256-
curve25519
         start_action = trap
       }
     }
   }
}
```

Figure 6. IPSec configuration file of VM1.

4.3. Execution and Validation of the Testbed

The network components should be started in the following sequence to ensure proper operation:

(1) 5G Core: Initialise 5G core using Docker: `docker compose up 5gc`.

(2) Near-RT RIC: Run the FlexRIC executable: `~/flexric/build/examples/ric#./nearRT-RIC`

(3) gNB: Run the gnb executable, with the configuration file specific for E2 settings:`../srsRAN_config# gnb -c gnb_zmq_e2.yaml`. Upon starting the gNB, the following can be observed, as shown in Figure 7:

- The "Connection to AMF on 10.53.1.2:38412" message indicates that the gNB initiated a connection to the 5G core.
- The "Connection to NearRT-RIC on 172.16.2.34:36421" message indicates that the gNB initiated a connection to the NearRT-RIC successfully.

Figure 7. Successful initiation of the gNB.

In the 5G core terminal, the gNB attachment is observed, as shown in Figure 8 below.

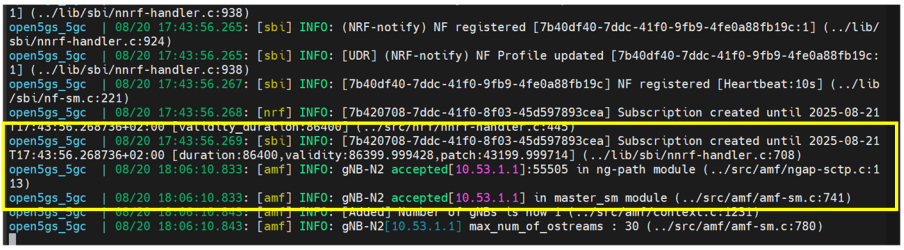

Figure 8. Successful attachment of the gNB to 5G core.

(4) User Equipment: First, we need to create a namespace for the UE as follows: `~/project-lab# ip netns add ue1`. Then, we initiate the UE by running the executable, with the configuration file modified with configuration settings for the ZMQ-based RF driver and E2:
`#./project-lab/srsRAN_4G/build/srsue/srcsrsue/root/project-lab/srsRAN_config/ue_zmq_e2.conf`
Once the srsUE is successfully attached, a message should appear on the 5G Core console. Finally, after configuring routing among these components, traffic between the UE and 5G Core is generated using ipref3 to validate the experimental setup. Note that we ran the iperf3 server on the 5G core as a Docker container, while the iperf3 client is running in the UE network namespace.

(5) xApp: To verify the proper working of the experimental system, we ran an xApp available with the FlexRIC installation. The xApp `xapp_oran_moni` connects to the Near RT-RIC and uses E2SM_KPM service modules to subscribe to measurement data. The metric names to be passed to the xApp are in the srsRAN configuration file, which was updated with the Near RT-RIC IP address, as shown in Figure 9 below.

```
root@vm2:~/project-lab/flexric/build/examples/xApp/c/monitor# cat xapp_mon_e2sm_kpm.conf
SM_DIR = "/usr/local/lib/flexric/"

# supported name = xApp
Name = "xApp"
NearRT_RIC_IP = "172.16.2.34"
E42_Port = 36422
```

Figure 9. xApp config file modified with near RT-RIC.

5. Evaluation and Result Analysis

In this section, we present a measurement-based evaluation of the performance impact of integrating PQC KEM into the IPsec protecting the O-RAN E2 interface.

5.1. Performance Metrics

Before we dive into the experiments' configurations, we provide details of the three primary metrics we chose to focus on in this evaluation. For each metric, we develop a test case and configuration to measure it effectively:

Latency Introduced by IPsec: This provides a direct measure of the overhead imposed on traffic between two endpoints due to securing an O-RAN interface. It is particularly relevant for latency-sensitive applications using the E2 interface.
IPsec Security Association (SA) Setup Time: This measures the impact of using PQC KEM algorithms in comparison to classical KEM, as an additional overhead. It reflects how PQC adoption may impact session initiation delays in practice.
Latency with IPSec SA setup for xApp. This measures the latency on xApp traffic during the IPSec SA setup. It reflects how PQC adoption may impact xApp's operation during the shorter period of session initiation.

To conduct a controlled experiment and isolate the performance impact of different cryptographic primitives, the testbed was systematically configured for the following scenarios: (1) baseline operation without IPSec, (2) IPsec with classical KEM, and (3) IPsec with PQC KEM. For each scenario, we performed 100 iterations/runs.

5.2. Experimental Scenario Setup

5.2.1. Latency Introduced by IPSec

The test setup consists of two VMs running Ubuntu (version 24.04.2 LTS), as shown below in Figure 10. sockperf is used as the latency testing tool. sockperf [43] is a utility designed for network benchmarking, including latency and throughput.

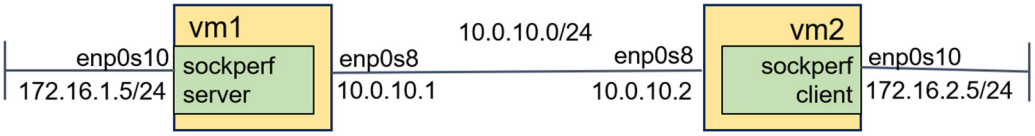

Figure 10. VM connectivity for network latency observation.

As shown in Figure 10, the sockperf server is running on VM1, while the client is running on VM2. Here, we run 100 iterations of the following command with and without IPSec: `sockperf ul --ip ‘‘$SERVER_IP’’ --client_ip ‘‘$CLIENT_IP’’ > ‘‘$OUTPUT_DIR/iteration-$i.txt’’`.

5.2.2. IPSec SA Setup Time

In this test case, the time taken to set up an IPsec tunnel established between two VMs is measured to evaluate whether the use of different KEM leads to variations in tunnel setup time. Note that IPSec setup time includes certificate validation that is generated through the pki command-line utility, which is part of strongSwan's integrated tools. We used the same setup as shown in Figure 10 above. However, we utilise tcpdump to analyse the timestamps of IKEv2 messages exchanged during the initiation process. For this analysis, the following packets were marked for the respective IPsec setup phases:

- IKE Initiation: {`start`} *parent_sa ikev2_init[I]* and {`end`} *parent_sa ikev2_init[R].*
- IKE Authorisation: {`start`} *child_sa ikev2_auth[I]* and {`end`} *child _sa ikev2_auth[R].*
- Child SA Setup: {`start`} *child_sa child_sa[I]* and {`end`} *child _sa child_sa[R].*

After this step, we run the following test cases: (1) IPSec setup with classical KEM (i.e., ECDH is used in the IKE SA proposal, namely, *aes256gcm16-prfsha256-curve25519*); (2) IPSec setup latency with ML-KEM-768 (ML-KEM is used with *aes256gcm16-prfsha256-mlkem768*); and (3) IPSec setup latency with ML-KEM-1024 (ML-KEM is used with *aes256gcm16-prfsha256-mlkem1024*).

5.2.3. Latency with IPSec SA Setup for xApp

In this test case, the time difference between the time of initiation of xApp and the time of the first control packet generated by xApp entering the encrypted E2 interface is measured. This measurement is taken as an estimate of the latency overhead introduced by the IPsec setup. Any delay in the packet stream initiated by xApp is reflected in the first packet. This is done to evaluate whether the use of ML-KEM leads to a delay in the start of control traffic flow. For comparison, two test cases were performed for the IPsec on E2: (1) with ECDH as the KEM and (2) with ML-KEM1024 as the KEM. Using the same setup in Figure 10 above, we use tcpdump to analyse the timestamps of xApp control messages, which are transported over the E2 interface using the Stream Control Transmission Protocol (SCTP). On VM2, we start packet capturing as follows: tcpdump -i enp0s8 -c 5. Then, on VM1, the xApp was initiated as follows:

`/root/project-lab/flexric/build/examples/xApp/c/monitor/xapp_oran_moni -cxapp_mon_e2sm_kpm.conf.`

Two test cases were evaluated here. First, IPSec was configured between VM1 and VM2, with ECDH as the KEM. For the second case, IPSec was configured with ML-KEM1024.

5.3. Results and Analysis

After running the test cases illustrated in the previous sub-section, we collected performance data from the results files to analyse them in this paper. We first analyse the latency introduced by IPSec. Figure 11 illustrates the latency distribution observed with and without IPsec protection using AES-256. The results show a consistent shift in the latency distribution when IPsec is enabled, indicating a measurable but bounded overhead introduced by encryption and encapsulation. This is an expected result, as encryption of data involves additional processing for transforming plaintext into ciphertext. More importantly, the overall shape of the distribution remains similar, suggesting that IPsec does not introduce significant variability or instability in packet delivery. This indicates that while IPsec increases baseline latency, it does not adversely affect the predictability of communication, which is critical for near-real-time O-RAN control loops.

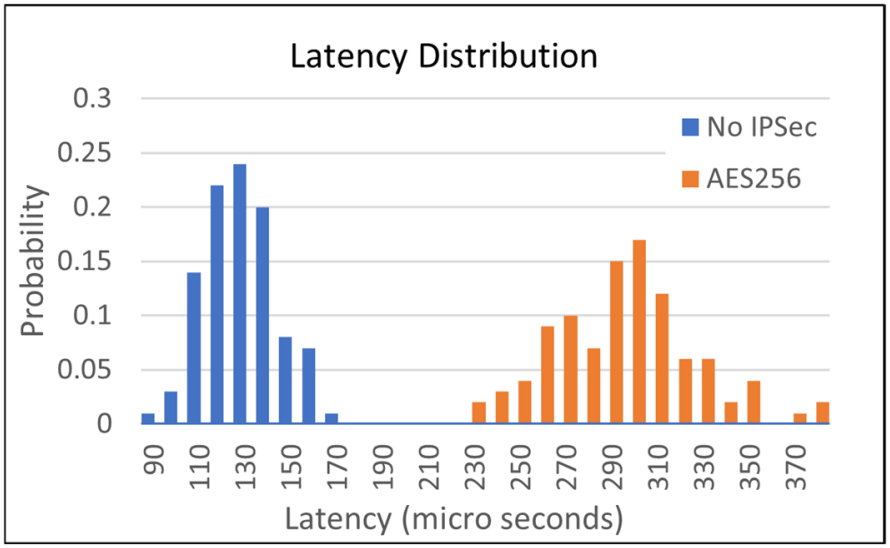

Figure 11. Latency distribution—no IPSec vs. IPSec (AES256).

In terms of IPSec SA setup time, we calculated the *ike_init*, *ike_auth*, and *child_sa* setup times extracted from packet header timestamps in each result file of the experimental runs. Figures 12–14 show IPSec initiation times with ECDH Curve 25519, ML-KEM 768, and ML-KEM 1024, respectively, while Table 1 shows the average time for the SA setup across all IKE initiation, IKE authorisation, and child SA setup stages for each scenario. Figure 12 presents the IPsec tunnel establishment times using the classical ECDH Curve 25519 key exchange. These results provide a baseline for comparison with post-quantum alternatives, showing the expected distribution of IKEv2 handshake completion times under conventional cryptographic settings. The relatively tight distribution indicates stable and repeatable performance, which serves as a reference point for evaluating the additional overhead introduced by post-quantum key exchange mechanisms.

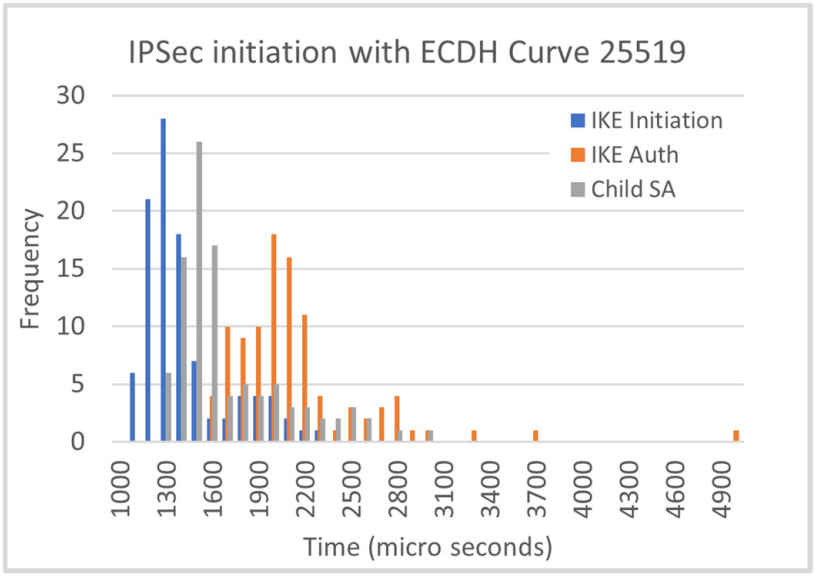

Figure 12. IPSec Initiation times with ECDH Curve 25519.

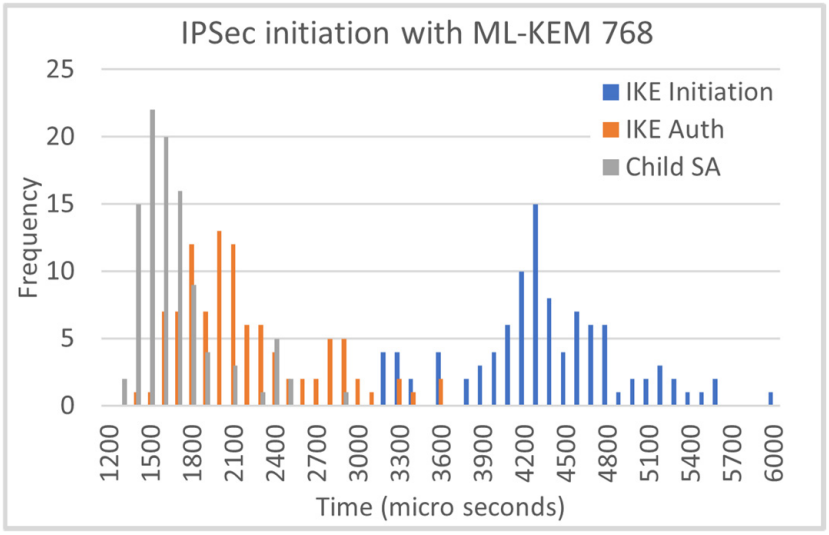

Figure 13. IPSec Initiation times with ML-KEM 768.

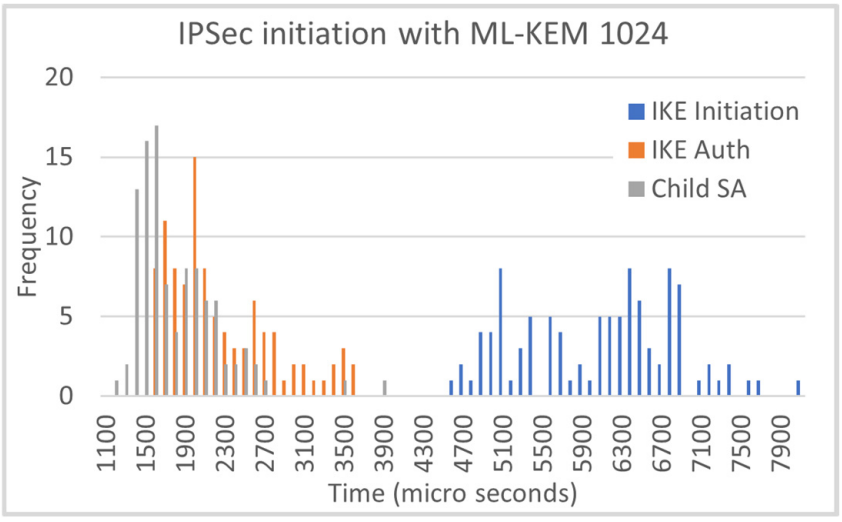

Figure 14. IPSec Initiation times with ML-KEM 1024.

Table 1. Average and standard deviation initiation times for each scenario in µs (avg, std).

Scenario	IKE Initiation	IKE Authorisation	Child SA Setup
IPSec (ECDH Curve 25519)	(1363, 276)	(2093, 466)	(1658, 371)
IPSec (ML-KEM 768)	(4306, 577)	(2141, 490)	(1633, 304)
IPSec (ML-KEM 1024)	(6026, 787)	(2198, 535)	(1768, 435)

On the other hand, Figure 13 shows the tunnel establishment times when ML-KEM-768 is used as the key exchange mechanism. Compared to the ECDH baseline, the distribution is shifted towards higher values, especially for IKE initiation, reflecting the additional computational and message-processing overhead associated with post-quantum key encapsulation. However, the distribution spread remains relatively consistent, indicating that the introduction of ML-KEM does not significantly increase variability. This suggests that the overhead is systematic and predictable, supporting the feasibility of ML-KEM-based IPsec in controlled O-RAN environments.

Finally, Figure 14 shows the tunnel establishment times for ML-KEM-1024, which provides a higher security level than ML-KEM-768. As expected, the results show a further increase in initiation time compared to both ECDH and ML-KEM-768, reflecting the larger parameter sizes and increased computational cost. Despite this increase, the distribution remains well-bounded, with no evidence of extreme outliers or instability. This demonstrates that even higher-security ML-KEM parameter sets can be integrated into IKEv2/IPsec with predictable performance characteristics, albeit with a measurable increase in setup latency.

Overall, Figures 11–14 demonstrate that while both IPsec and post-quantum key exchange introduce measurable overhead, this overhead remains bounded and predictable, supporting the practical deployment of PQC-secured IPsec tunnels in O-RAN control-plane environments. As explained above, in comparison to the classic ECDH Curve 25519 as the baseline, IPSec with ML-KEM-768 and ML-KEM-1024 introduces an additional delay of approximately 2.7 ms and 4.7 ms, respectively, for the IKE initiation process. For ML-KEM-768 and ML-KEM-1024, this is an expected result. Most importantly, this provides evidence for the increased SA setup time for ML-KEM in comparison to the classical KEM (i.e., ECDH Curve 25519) baseline.

Finally, we calculate the time between the execution of the xApp command and the first xApp packet leaving the VM2 interface (i.e., RIC) using the xApp packet flow for each iteration, which was captured by tcpdump. In both cases, with and without IPSec, we got the same results, as shown in Figure 15. This means that the IPSec setup has no impact on xApp performance. This is expected, since IPSec should be operational before any data is transferred between gNB and the near real-time RIC.

```
mario@vm2:~/project-lab/test-cases_6.2.4$ ./delay.sh
xApp start time  : 1756279277.158345
First packet time: 1756279279.191873
Delay (seconds)  : 2.033528
```

Figure 15. Delay between xApp command and first packet leaving the RIC interface.

5.4. Limitations of Testbed Environment

Since the testbed was built using a virtualised environment, it introduced certain limitations when interpreting the results in the context of real-world deployments. The testbed consisted of multiple Ubuntu virtual machines deployed on a single physical host using the Oracle VM VirtualBox hypervisor. While this approach enables controlled experimentation and reproducibility, it does not fully capture the behaviour of real-world

deployments where distributed network elements operate on separate physical machines and dedicated networking hardware. One key limitation is virtualisation overhead and shared resource contention. Additionally, the network connectivity between components was implemented using host-based virtual network interfaces and host-based networking. As a result, packet transmission occurs within the host's software networking stack rather than across physical network hardware. This can lead to differences in latency behaviour, buffering, and interrupt handling compared to hardware network interface cards and specialised packet processing frameworks typically used in operational environments.

Another limitation is the absence of hardware acceleration mechanisms for cryptographic or network offloading, which may be available in production network environments. As a result, the measured computational overhead of cryptographic operations reflects software-based processing within the VMs and may differ from the performance observed on hardware platforms with acceleration support. For this study, the measurements were conducted using the standard utilities available in Linux for measuring latency, CPU, and memory utilisation. Hence, the precision of the observations is essentially limited, while others, such as CPU utilisation and energy consumption, were not feasible, as we discussed above. Despite these limitations, the virtualised testbed provides a consistent and controlled environment for comparative analysis between classical and post-quantum configurations. Since all experiments were conducted under identical conditions, the relative performance differences observed remain meaningful for understanding the computational impact of integrating PQC mechanisms into secure network interfaces. Future work will extend the evaluation to distributed physical testbeds to validate the findings under more realistic deployment conditions.

6. Conclusions

In this paper, we presented a testbed-based evaluation of integrating a NIST-aligned ML-KEM into IKEv2/IPsec, protecting the 5G O-RAN E2 interface. Using an open-source stack (srsRAN, Open5GS, and FlexRIC) and a liboqs-enabled strongSwan, we compared three configurations: no IPsec, classical ECDH-based IPsec, and ML-KEM-based IPsec, under realistic Near-RT RIC signalling workloads. Repeated, automated trials show that ML-KEM primarily affects the key-establishment phase, where tunnel setup latency increases by a small, reasonable ≈2.7~4.7 ms relative to classical IPsec, while the runtime behaviour of Near-RT RIC xApps and control-loop operation remained stable in our experiments in terms of successful E2 session establishment, uninterrupted message exchange, and absence of packet loss or control-plane interruptions during the observation period. These results indicate that ML-KEM-based IPsec on the E2 interface is practically feasible on contemporary server-class hardware and can form part of early, staged quantum-safe migration strategies for O-RAN deployments. Note that the quantitative performance results presented in this work should be interpreted as representative of ML-KEM-based PQC IPsec deployments rather than being universally applicable to all PQC KEM algorithms. Nevertheless, the integration approach and evaluation methodology described here remain applicable to other post-quantum KEMs and provide a framework for future comparative studies of PQC algorithms in O-RAN security.

For future work, we are pursuing two main directions. First, we will extend the experimental framework to protect O-RAN interfaces that rely on TLS (e.g., A1, O1, O2) using PQC and hybrid PQC–classical key exchange. This will evaluate the impact of ML-KEM and alternative NIST-standardised KEMs on TLS 1.3 handshake latency, session resumption, and control-plane message delay. Secondly, we will be investigating how Federated Learning (FL) can enhance PQC security within the modular O-RAN environment. FL enables collaborative learning for PQC security optimisation across distributed entities without

sharing sensitive cryptographic data or relying upon a centralised learning framework. Our initial focus will be on the following:

- Optimising PQC: Leveraging federated feedback to determine optimal algorithm selection and parameter settings that balance security and efficiency.
- Collaborative Anomaly Detection: Aggregating locally trained model insights to capture behavioural and implementation anomalies without exposing raw data.
- Post-Deployment Hardening: rapid identification of novel attacks and implementation weaknesses based on federated insights from live systems, supporting ongoing attack resilience.

Author Contributions: Conceptualization, M.P., M.M. and M.H.E.; Methodology, M.P.; Software, M.P.; Validation, M.P., M.H.E. and A.R.; Formal analysis, M.P.; Investigation, M.P., M.M. and M.H.E.; Writing—original draft, M.P.; Writing—review & editing, M.M., M.H.E., A.R., N.S. and M.K.M.; Supervision, M.M. and M.H.E.; Project administration, M.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

3GPP	3rd-Generation Partnership Project;
A1	O-RAN A1 interface (Non-RT RIC ↔ Near-RT RIC);
E2	O-RAN E2 interface (Near-RT RIC ↔ O-CU/O-DU);
ESP	Encapsulating Security Payload;
IKEv2	Internet Key Exchange version 2;
IPsec	Internet Protocol Security;
ML-DSA	Module-Lattice Digital Signature Algorithm;
ML-KEM	Module-Lattice Key Encapsulation Mechanism;
NIST	National Institute of Standards and Technology;
Near-RT RIC	Near-Real-Time RAN Intelligent Controller;
Non-RT RIC	Non-Real-Time RAN Intelligent Controller;
O-Cloud	Cloud infrastructure hosting O-RAN VNFs/CNFs;
O-CU	O-RAN Central Unit;
O-DU	O-RAN Distributed Unit;
O-FH	Open Fronthaul;
O-RAN	Open Radio Access Network;
O-RU	O-RAN Radio Unit;
PQC	Post-Quantum Cryptography;
RAN	Radio Access Network;
RIC	RAN Intelligent Controller;
RRM	Radio Resource Management;
SMO	Service Management and Orchestration;
TLS	Transport Layer Security;
UE	User Equipment;
VNF	Virtual Network Function;
VM	Virtual Machine.

References

1. Polese, M.; Bonati, L.; D'oro, S.; Basagni, S.; Melodia, T. Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 1376–1411. [CrossRef]
2. Liyanage, M.; Braeken, A.; Shahabuddin, S.; Ranaweera, P. Open RAN security: Challenges and Opportunities. *J. Netw. Comput. Appl.* **2023**, *214*, 103621. [CrossRef]
3. O-RAN Alliance e.V. O-RAN Security Requirements and Controls Specifications 13.0. 2025. Available online: https://specifications.o-ran.org/download?id=991 (accessed on 12 August 2025).
4. O-RAN Alliance e.V. O-RAN Security Protocols Specifications 13.0. 2025. Available online: https://specifications.o-ran.org/download?id=992 (accessed on 12 August 2025).
5. National Institute of Standards and Technology (NIST). FIPS 203—Module-Lattice-Based Key-Encapsulation Mechanism Standard. 2024. Available online: https://csrc.nist.gov/pubs/fips/203/final (accessed on 12 August 2025).
6. Groen, J.; D'Oro, S.; Demir, U.; Bonati, L.; Villa, D. Securing O-RAN Open Interfaces. *arXiv* **2024**, arXiv:2404.15076. [CrossRef]
7. Djuitcheu, H.; Kakani, P.K.; Brunke, D.; Fraunholz, D.; Schotten, H.D. Exploring the Implications and Methodologies of Securing the E2 Interface. In Proceedings of the IEEE Future Networks World Forum (FNWF), Dubai, United Arab Emirates, 15–17 October 2024.
8. Scalise, P.; Garcia, R.; Boeding, M.; Hempel, M.; Sharif, H. An Applied Analysis of Securing 5G/6G Core Networks with Post-Quantum Key Encapsulation Methods. *Electronics* **2024**, *13*, 4258. [CrossRef]
9. Software Radio Systems. srsRAN Enterprise 5G | Software Radio Systems. Available online: https://srs.io/srsran-enterprise-5g/ (accessed on 22 August 2025).
10. Open5GS. Building Open5GS from Sources. Available online: https://open5gs.org/open5gs/docs/guide/02-building-open5gs-from-sources/ (accessed on 22 August 2025).
11. Mosaic5g. Flexible RAN Intelligent Controller (FlexRIC) and E2 Agent. Available online: https://gitlab.eurecom.fr/mosaic5g/flexric (accessed on 22 August 2025).
12. The strongSwan Team. strongSwan. Available online: https://strongswan.org/ (accessed on 22 August 2025).
13. Salvat, J.; Ayala-Romero, J.; Zanzi, L.; Garcia-Saavedra, A.; Costa-Perez, X. Open Radio Access Networks (O-RAN) Experimentation Platform: Design and Datasets. *IEEE Commun. Mag.* **2023**, *61*, 138–144. [CrossRef]
14. Azariah, W.; Bimo, F.; Lin, C.-W.; Cheng, R.-G.; Nikaein, N.; Jana, R. A Survey on Open Radio Access Networks: Challenges, Research Directions, and Open Source Approaches. *Sensors* **2024**, *24*, 1038. [CrossRef] [PubMed]
15. O-RAN Alliance e.V. O-RAN Architecture Description 15.0. 2025. Available online: https://specifications.o-ran.org/download?id=932 (accessed on 22 August 2025).
16. Wani, M.S.; Kretschmer, M.; Schröder, B.; Grebe, A.; Rademacher, M. Open RAN: A Concise Overview. *IEEE Open J. Commun. Soc.* **2025**, *6*, 13–28. [CrossRef]
17. Kim, J.; Park, J.; Lee, J.-H. Integration of the Layer 2 Protection Extended Application for Open RAN: A Security by Design Approach. *Comput. Electr. Eng.* **2025**, *126*, 110479. [CrossRef]
18. Aizikovich, E.; Mimran, D.; Grolman, E.; Elovici, Y.; Shabtai, A. Rogue Cell: Adversarial Attack and Defense in Untrusted O-RAN Setup Exploiting the Traffic Steering xApp. *arXiv* **2025**, arXiv:2505.01816. [CrossRef]
19. Mimran, D.; Bitton, R.; Kfir, Y.; Klevansky, E.; Brodt, O.; Lehmann, H.; Elovici, Y.; Shabtai, A. Security of Open Radio Access Networks. *Comput. Secur.* **2022**, *122*, 102890. [CrossRef]
20. O-RAN Next Generation Research Group (nGRG). Platform Security in Next-Gen Mobile Networks. 2025. Available online: https://www.o-ran.org/research-reports/platform-security-in-next-gen-mobile-networks (accessed on 12 August 2025).
21. Alimohammadi, H.; Mayhoub, S.; Chatzimiltis, S.; Shojafar, M.; Bhutta, M.N.M. Toward a Multi-Layer Defence Framework for Securing Near-Real-Time Operations in Open RAN. *IEEE Open J. Commun. Soc.* **2026**, *7*, 480–497. [CrossRef]
22. O-RAN Alliance e.V. The O-RAN ALLIANCE Security Working Group Continues to Advance O-RAN Security. 2024. Available online: https://www.o-ran.org/blog/the-o-ran-alliance-security-working-group-continues-to-advance-o-ran-security (accessed on 12 August 2025).
23. Eiza, M.H.; Akwirry, B.; Raschella, A.; Mackay, M.; Maheshwari, M.K. A Hybrid Zero Trust Deployment Model for Securing O-RAN Architecture in 6G Networks. *Future Internet* **2025**, *17*, 372. [CrossRef]
24. O-RAN Next Generation Research Group (nGRG). Research Report on Quantum Security. 2023. Available online: https://mediastorage.o-ran.org/ngrg-rr/nGRG-RR-2023-04-Quantum-Security-v1_1.pdf (accessed on 12 August 2025).
25. National Institute of Standards and Technology (NIST). NIST Releases First 3 Finalized Post-Quantum Encryption Standards. 13 August 2024. Available online: https://www.nist.gov/news-events/news/2024/08/nist-releases-first-3-finalized-post-quantum-encryption-standards (accessed on 12 December 2025).
26. Joseph, D.; Misoczki, R.; Manzano, M.; Tricot, J.; Pinuaga, F.D.; Lacombe, O.; Leichenauer, S.; Hidary, J.; Venables, P.; Hansen, R. Transitioning organizations to post-quantum cryptography. *Nature* **2022**, *605*, 237–243. [CrossRef] [PubMed]

27. García, C.R.; Rommel, S.; Takarabt, S.; Vegas Olmos, J.J.; Guilley, S.; Nguyen, P.; Monroy, I.T. Quantum-resistant Transport Layer Security. *Comput. Commun.* **2024**, *213*, 345–358. [CrossRef]
28. Rathi, V.; Chopra, L.; Agarwal, M.; Rajput, N.; Sharma, K.; Mundepi, S.; Gangwar, S.; Rawal, R. Q-RAN: Quantum-Resilient O-RAN Architecture. *arXiv* **2025**, arXiv:2510.19968v1.
29. Montenegro, J.A.; Rios, R.; Lopez-Cerezo, J. A performance evaluation framework for post-quantum TLS. *Future Gener. Comput. Syst.* **2026**, *175*, 108062. [CrossRef]
30. Bae, S.; Chang, Y.; Park, H.; Kim, M.; Shin, Y. A Performance Evaluation of IPsec with Post-Quantum Cryptography. In *Information Security and Cryptology—ICISC 2022, Proceedings of the 25th International Conference, ICISC 2022, Seoul, Republic of Korea, 30 November–2 December 2022*; Springer: Cham, Switzerland, 2023.
31. Otero-García, P.; Fernández-Vilas, A.; Fernández-Veiga, M. Introducing Post-Quantum algorithms in Open RAN interfaces. *arXiv* **2025**, arXiv:2501.10060v1. [CrossRef]
32. Patero. Patero and Eridan Pioneer Quantum-Secure Open Radio Area Network. 31 July 2025. Available online: https://www.patero.io/news/how-technology-can-help-curb-attention-disorders (accessed on 9 January 2026).
33. Nokia. Nokia and Turkcell Demonstrate Industry Leading Quantum-Safe Protection for Mobile Subscribers. 20 December 2024. Available online: https://www.nokia.com/newsroom/nokia-and-turkcell-demonstrate-industry-leading-quantum-safe-protection-for-mobile-subscribers/ (accessed on 9 January 2026).
34. Samsung Research. Quantum Security for Future Communication Networks: Standards Perspective. 2 July 2025. Available online: https://research.samsung.com/blog/Quantum-Security-for-Future-Communication-Networks-Standards-Perspective (accessed on 9 January 2026).
35. Open5GS. Open5GS—Open Source Implementation for 5G Core. Available online: https://open5gs.org/ (accessed on 22 August 2025).
36. Northeastern University. Colosseum: The Open RAN Digital Twin. Available online: https://colosseum.sites.northeastern.edu/ (accessed on 22 August 2025).
37. free5GC. free5GC—Open Source Core Network Implementation. Available online: https://free5gc.org/ (accessed on 22 August 2025).
38. O-RAN Project. Installation Guides. Available online: https://docs.o-ran-sc.org/projects/o-ran-sc-ric-plt-ric-dep/en/latest/installation-guides.html (accessed on 22 August 2025).
39. Maheshwari, M.K.; Raschella, A.; Mackay, M.; Eiza, M.H.; Wetherall, J.; Laing, J. 5G High Density Demand (HDD) Dataset in Liverpool City Region, UK. *Sci. Data* **2025**, *12*, 1992. [CrossRef] [PubMed]
40. Software Radio Systems. srsRAN gNB with srsUE. Available online: https://docs.srsran.com/projects/project/en/latest/tutorials/source/srsUE/source/index.html (accessed on 22 August 2025).
41. The ZeroMQ Authors. ZeroMQ. Available online: https://zeromq.org/ (accessed on 22 August 2025).
42. Schmidt, R.; Irazabal, M.; Nikaein, N. FlexRIC: An SDK for next-generation SD-RANs. In Proceedings of the 17th International Conference on Emerging Networking EXperiments and Technologies (CoNEXT '21), Virtual, 7–10 December 2021.
43. Mellanox Technologies. Mellanox/sockperf: Network Benchmarking Utility. 21 September 2022. Available online: https://github.com/Mellanox/sockperf (accessed on 12 August 2025).