2025 International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM) | 979-8-3315-6873-3/25/$31.00 ©2025 IEEE | DOI: 10.1109/MSWiM67937.2025.11308758

Building a Security Canopy Over O-RAN: The SDP Paradigm

Tejas Sameer Deshmukh
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad
Hyderabad, India
cs22mtech12005@iith.ac.in

Antony Franklin A
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad
Hyderabad, India
antony.franklin@iith.ac.in

***Abstract*—The Open Radio Access Network (O-RAN) architecture provides flexibility in the deployment of RAN, but introduces significant security vulnerabilities due to its open interfaces and disaggregated nature, expanding the attack surface for threats like DoS, DDoS, MiTM, unauthorized access, etc. Current security solutions often lack comprehensive protection. This paper proposes an O-RAN-SDP framework integrating Software Defined Perimeter (SDP) principles to establish security protection in the O-RAN with Zero-Trust deployment. By deploying an SDP Controller (SDPC) as an xApp within the Near-RT RIC and embedding SDP agents in O-RAN components, our approach utilizes the Single Packet Authorization (SPA) and secure tunneling in the O-RAN-SDP architecture. This provides robust authentication, authorization, and attack surface reduction, inherently mitigating DoS/DDoS, MiTM, and zero-day vulnerability attacks across the O-RAN ecosystem.**

***Index Terms*—Open RAN, SDP, DoS/DDoS, Zero-Trust Architechture (ZTA), O-RAN Security, xApp Security.**

I. Introduction

The traditional Radio Access Network (RAN) is undergoing a fundamental transformation towards an open, disaggregated, and intelligent architecture known as Open RAN (O-RAN). This model deconstructs the monolithic RAN into distinct components namely the O-RAN Radio Unit (O-RU), Distributed Unit (O-DU), and Centralized Unit (O-CU)—interconnected via standardized, open interfaces. While this architectural paradigm fosters a multi-vendor ecosystem and accelerates service innovation, its open and disaggregated nature makes O-RAN interfaces susceptible targets for potential attackers, vastly expanding the attack surface. The O-RAN architecture is still in an active development and standardization phase, which makes it a prime target for zero-day vulnerability exploits.

As highlighted in [1], each element in the O-RAN architecture represents a potential DoS/DDoS target, including but not limited to the primary ORAN components, the internal elements of the RAN Intelligent Controller (RIC) and the xApps it hosts. The security risk is amplified by several factors inherent to the O-RAN architecture itself. The reliance on xApps and components from multiple third-party vendors increases the chances of attacks originating within the supply chain. This vulnerability is compounded by multiple contributing factors, including: the varying levels of trust between disparate O-RAN components, the potential for malicious xApps to be onboarded due to insufficient security vetting, the absence of granular access control policies to prevent components from communicating unnecessarily, and systemic deficiencies in authentication and authorization techniques. The urgency of this issue is underscored by recent industry data. A report from Cloudflare [1] observed that in the first quarter of 2025 alone, 20.5 million DDoS attacks were mitigated. This figure represents a 358% year-over-year (YoY) increase and a 198% quarter-over-quarter (QoQ) increase. Telecommunication companies are the most frequent targets, accounting for almost half of the total attack volume. This escalating threat landscape highlights the critical and immediate need for robust DDoS detection and mitigation frameworks. The O-RAN specifications[2], define several possible DoS attacks such as C-plane, U-plane, S-plane PTP, A1 Interface, etc. The specifications suggest that these attacks must be mitigated at the component level. However, current O-RAN specifications only mention that such threats may exist, but do not provide detailed threat models, descriptions of attack execution, or concrete mitigation strategies. This gap is further evidenced by a growing number of Common Vulnerabilities and Exposures (CVEs) related to DoS and other attacks in O-RAN components [3]. The affected components are numerous, leading to severe consequences, such as the crashing of the Near-RT RIC, disconnections between the RIC and the E2 Nodes, and DoS conditions on the xApps, the E2 nodes, and the internal components of the RIC.

The O-RAN specifications do not provide defenses against known attacks, and most identified vulnerabilities remain unresolved, with no effective solutions available in the literature. Many related works lack sufficient implementation details, hindering the ability of researchers to build up on them. One significant challenge is the gap between theoretical models and real-world applicability. For example, the authors of [4] propose an Intrusion Detection System (IDS) for the Open Fronthaul Interface (O-FHI) by training a Long Short-Term Memory (LSTM) model on a subset of 23 features from the generic CIC-DDoS2019 dataset. This contribution

[1]DDoS Threat Report for 2025 Q1 at https://blog.cloudflare.com/ddos-threat-report-for-2025-q1/.

is limited, as the approach is not validated with a Proof-of-Concept (PoC) or tested against O-FHI-specific data, such as eCPRI real-time traffic, PTP S-Plane data, or CU-Plane data. Similarly, while [5] demonstrates a CU-Plane DoS attack capable of crashing an O-DU and severely degrading O-RU traffic, the authors omit critical details for confidentiality reasons. This renders the work irreproducible by the research community and, critically, does not offer an attack prevention technique. Another category of work identifies threats, but offers insufficient mitigation. In [1], the authors show how a DDoS on a RAN's gateway severely affects QoS parameters such as latency, jitter, and bandwidth, highlighting the need for a robust defense framework, while [6] suggests an AI-based detection xApp but relies on simple blacklisting for mitigation. Such a model would likely suffer from poor recall, as many attack scenarios might be missed. The framework proposed in [7] uses a zero-trust approach for xApp-to-xApp communication, deploying access policies as smart contracts on a blockchain for authentication and authorization. Although this looks promising for MiTM or API attacks, its effectiveness against DoS, DDoS, and zero-day vulnerabilities remains a question, as the O-RAN elements themselves can still be targeted. In addition, blockchain systems are inherently slow due to their consensus protocols. Other works similar to this focus mostly on authentication and authorization, without considering the security of O-RAN's open interfaces or specific threats like DoS/DDoS attacks.

A review of the security coverage offered by these works reveals significant gaps. Most solutions focus narrowly on a single component or interface (e.g., O-FHI, CU-Plane, or gNB), often providing threat detection without robust prevention mechanisms. Furthermore, the security of the RIC and its internal xApps is often overlooked or addressed in isolation. To our knowledge, our work is the first to propose a holistic security "Canopy" that protects all O-RAN elements and their open interfaces, as well as the internal RIC components from attacks such as DoS, DDoS, port scanning, MiTM, and password cracking. Most importantly, it is designed to address zero-day vulnerabilities by providing explicit and granular authorization, component validation, integrity protection, authentication, a zero-trust and zero visibility (black infrastructure) model, and confidentiality.

II. Proposed Solution

To establish a robust Zero-Trust security posture for O-RAN, this proposal takes advantage of the principles of SDP. SDP is a security framework developed by the Cloud Security Alliance (CSA) that implements ZTA principles to provide dynamic and identity-centric network access control [8]. The framework operates on a "black cloud" approach where, network infrastructure remains invisible to unauthorized users, effectively eliminating the attack surface through zero visibility and zero connectivity until proper authentication and authorization is completed. This robust security is achieved through five integrated layers: SPA for initial device authentication before any network ports are opened; mutual Transport Layer Security (mTLS) for strong bidirectional authentication and encrypted communication; Device Validation (DV) to ensure that cryptographic keys are held only by authorized devices; Dynamic Firewalls that enforce a default-deny policy with dynamically created access rules; and Application Binding, which forces authorized applications to use the secure TLS tunnels. This multilayered approach addresses critical challenges including authentication and trust, access control, and data privacy, making it particularly effective for modern, cloud-first environments where traditional perimeter-based security models prove inadequate.

To address the challenges discussed, this section proposes a Zero-Trust framework for O-RAN and its internal RIC components that integrates the principles of SDP [9] into the O-RAN architecture, enhancing the security of the O-RAN and RIC components and their interactions. This integration is designed to be non-disruptive, ensuring no structural modifications are required to the O-RAN specifications leveraging the xApps framework.

Our proposal embeds SDP components within existing O-RAN elements to enforce the Zero Trust principles. Central to this design is the placement of the SDPC functionality within the Near-RT RIC as an xApp. This strategic placement leverages the inherent position of the RIC as a control and intelligence hub, allowing the SDP control plane to benefit from the real-time data access and management interfaces (e.g., E2) of the RIC, while minimizing the need for new infrastructure. The architecture operates on two distinct planes: the SDP-Control Plane (SDP-CP) and the SDP-Data Plane (SDP-DP). The SDP-CP is governed by the SDPC xApp, which serves as the central policy decision point and trust anchor, with responsibilities including identity on-boarding, policy distribution, credential management, and secure session orchestration. For logical separation, the SDPC functionality for the A1 interface can reside in the SMO, while control for all other interfaces can be managed by the SDPC xApp in the Near-RT RIC. SDP-DP is made up of SDP agents embedded within each O-RAN component. Each functional block (O-RU, O-DU, O-CU, SMO, etc.) incorporates two logical SDP roles, namely SDP Initiating Host (IH) (responsible for initiating secure connection requests to other components) and the SDP Accepting Host (AH)/Gateway (acts as the policy enforcement point, protecting the services hosted by the component from unauthorized requests). For internal RIC, the IH can be embedded in xApps and the AH in RIC's terminal nodes like ETerm (ORAN-SC)[2] or iApps (FlexRIC)[3]. Following the CSA SDP specifications [10], we adopt an embedded gateway model for communication between O-RAN components and a Client-to-Gateway model for internal RIC interactions between xApps. The fundamental principle of the architecture is that any "Consumer" O-RAN component must first interact with the SDPC xApp to obtain the authorization and credentials

[2]ORAN-Software Community is a Linux Foundation-hosted open-source reference implementation: https://github.com/srsran/oran-sc-ric.

[3]FlexRIC is a open-source nearRT RIC by Eurecom: https://github.com/openaicellular/flexric.

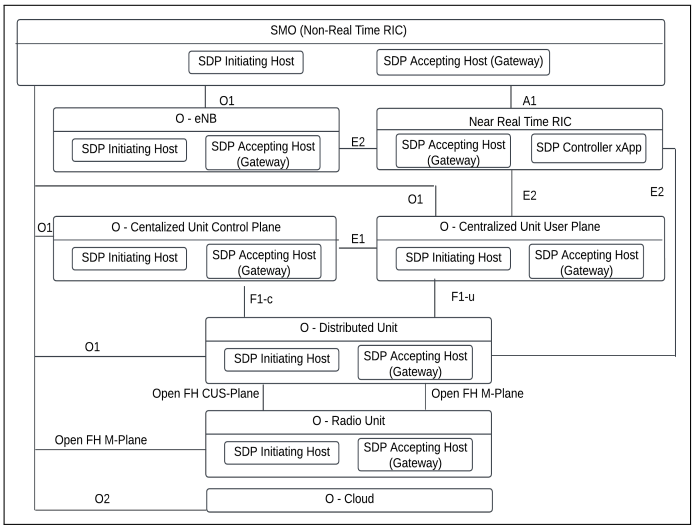

Fig. 1: Proposed O-RAN Architecture for ORAN Components

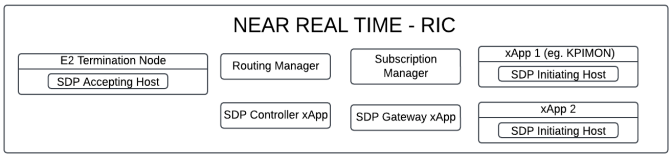

Fig. 2: Proposed O-RAN Architecture for RIC Components

before communicating with a "Producer" component. To illustrate this operational flow, we detail the scenario of an O-DU (Consumer) establishing a secure F1 interface connection with an O-CU (Producer). The sequence, depicted in Fig. 3, proceeds as follows. It is assumed that the O-DU (IH) and O-CU (AH/Gateway) are already on-boarded and registered with SDPC xApp using the AH-Controller SDP sequence, thereby establishing identities and initial trust credentials.

1) **SPA Request:** The O-DU, acting as an SDP IH, constructs a SPA packet using SPA and HMAC keys. This cryptographically signed packet with Rijnadel keys contains the O-DU's identity and its request to access a specific service on the O-CU (e.g., F1-c interface, identified by protocol/port like SCTP/38472). The SPA packet is sent to the known address of the SDPC xApp, which should initially be configured on each ORAN component.
2) **Controller Validation and Policy Check:** The SDPC xApp receives the SPA packet and authenticates the O-DU based on SPA and HMAC keys. Then, it validates the request against its configured access control policies. If the O-DU is authorized to access the requested O-CU service, the SDPC xApp proceeds with the credential and policy distribution. Otherwise, the packet is simply dropped.
3) **Credential and Policy Distribution:** Upon successful validation, SDPC xApp generates ephemeral session-specific credentials such as SPA key, HMAC key, and IPSec/mTLS keys and certificates to establish a secure data channel. The use of ephemeral credentials significantly limits the time window for possible misuse if compromised. Then, it securely transmits them to O-DU (IH) along with any additional authentication parameters by which O-CU can authenticate and validate O-DU if necessary, as well as specific access instructions (e.g., "Allow connection from O-DU's source IP on SCTP/38472" for "x" seconds) to O-CU's AH/Gateway. The AH/Gateway receives this policy before the IH attempts connection, preparing it to accept the authorized flow.
4) **Gateway Authentication (SPA Knock):** O-DU (IH) constructs a new SPA packet using the ephemeral SPA and HMAC key provided by SDPC xApp and sends it directly to AH/Gateway of O-CU.
5) **Gateway Validation and Dynamic Firewall Opening:** AH/Gateway of O-CU validates the SPA packet using the ephemeral SPA and HMAC key it received from the SDPC xApp. It also cross references the request details (source identity, requested service, etc) against the access instructions provided by the SDPC xApp. If everything matches, the AH/Gateway dynamically modifies its local firewall rules to permit the impending connection only from the authorized O-DU for the specified service for a configurable specified time.
6) **Secure Tunnel Establishment:** With the path cleared by the successful SPA knock, the O-DU (IH) initiates a secure connection setup with the O-CU's AH/Gateway. In this example, we use the ephemeral IPsec keys/certificates to perform mutual authentication and establish an IPsec tunnel. This ensures the confidentiality, integrity, and authenticity of the subsequent communication channel.
7) **Application Data Exchange:** Once the IPsec tunnel is established, the actual O-RAN application traffic (e.g., SCTP connection and the subsequent F1 interface setup messages and traffic) flows securely between the O-DU and the O-CU, encrypted and protected by the IPsec tunnel (e.g., using ESP). The underlying O-CU service receives the decrypted raw packets from the AH/Gateway. If needed, this traffic can also be encrypted and is implementation specific as pointed out by CSA. This step ensures Application Binding.
8) **Connection Closure:** Upon completion of the required communication, the connection and the IPsec tunnel are securely closed. The dynamic firewall rule in the AH/Gateway will be automatically removed or timed out after some specified time. This timeout is configurable and does not affect the data transfer when backed by the persistent connection rule. The rule is explicitly removed at closure if it has not already expired by timeout.

The proposed call flow here in Scenario 1 (O-DU to O-CU) remains consistent across various O-RAN interaction scenarios. For example, in Scenario 2 (xApp-RIC), prior to an xApp subscribing to and receiving services from the RIC or from another xApp, it must first comply with the access control policies enforced by the SDP framework. The only key distinction in Scenario 3 (Consumer xApp-Producer xApp) lies in its client-to-gateway model, where the gateway can

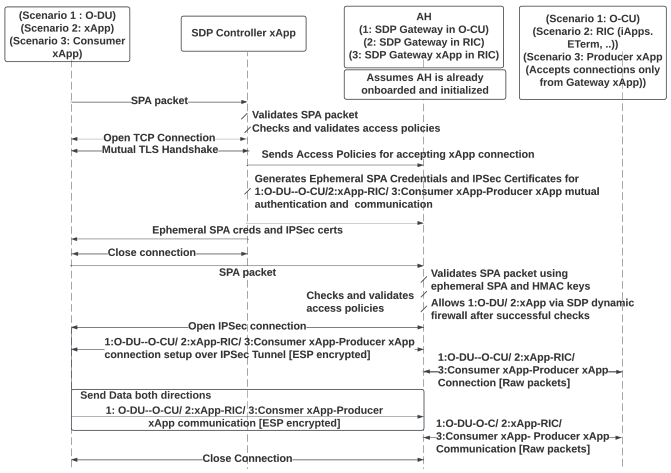

Fig. 3: Proposed SDP-Secured Call Flow between O-DU and O-CU

be deployed as an independent xApp, serving as a security mediator that protects the services offered by producer xApps. This decouples service provisioning from direct exposure, thereby ensuring zero visibility.

The foundation of trust within this O-RAN-SDP framework is based on robust cryptographic mechanisms and management for which a trusted Certificate Authority (CA) is essential. This CA could potentially be integrated within the SMO / Non-RT RIC framework, using the existing O-RAN management infrastructure, or it could be an external, trusted third-party CA. This root CA issues the initial certificate to the SDPC xApp, establishing it as the primary trust anchor within the O-RAN-SDP domain. The SDPC xApp is then responsible for managing the trust lifecycle for all participating O-RAN components. This includes issuing identities and initial credentials during the on-boarding phase, generating and distributing ephemeral session credentials (SPA keys, HMAC keys, IPSec keys, IPSec certificates) as described in the workflow. The SDPC xApp also issues intermediate or short-lived certificates derived from its own trust anchor status, handling credential revocation if a component is compromised or decommissioned. All control plane communication (e.g., between SDPC and IH/AH) and authorized data plane communication (tunneled via IPsec) are protected using these managed cryptographic materials, ensuring confidentiality and integrity throughout the network.

The proposed O-RAN-SDP architecture directly implements the core tenets of ZTA like no implicit trust on any component assuming breach regardless of its operation within the network; authentication and authorization of all access requests before connection establishment based on identity and context-aware policies managed by SDPC xApp; and least privilege access wherein policies enforced by the AH/Gateway grants only minimum necessary permissions for a specific task and duration. The key security benefits derived from this approach are given below.

- **Reduced Attack Surface:** The SPA mechanism effectively cloaks the network ports of protected O-RAN services. Components do not respond to unsolicited network scans or connection attempts, making reconnaissance significantly harder for attackers. Open ports are only temporarily and selectively revealed to authenticated and authorized entities.
- **Pre-Authentication and Pre-Authorization:** Access is granted before any network-level connection (like TCP/SCTP handshake or IPsec negotiation) is permitted, filtering out unauthorized attempts early. So, no connection can be attempted unless authenticated and authorized.
- **Protection Against Network Exploits:** By preventing unauthorized connection attempts, the framework mitigates the risks associated with exploiting vulnerabilities, including potential zero-day vulnerabilities in network protocols or services before authentication can occur.
- **DoS/DDoS Mitigation:** Since AH/Gateways silently discard unauthorized SPA packets and do not respond to unsolicited connection attempts, they are inherently more resilient to resource exhaustion attacks targeting protected O-RAN services.
- **Replay Attack Resistance:** SPA packets typically incorporate nonces or timestamps to prevent attackers from capturing and replaying valid packets.
- **Enhanced Confidentiality and Integrity:** The mandatory use of secure tunnels (e.g., IPsec) for authorized data flows ensures that inter-component communication across potentially non-trusted transport networks remains confidential and tamper-proof.

O-RAN-SDP framework enforces Zero Trust security by using existing structures while adding strong authentication, dynamic authorization, and attack surface reduction through SPA and secure tunneling, ensuring that all communication is explicitly verified and secured on a per-session basis, thereby significantly enhancing the overall security posture of O-RAN.

III. Security Analysis and Threat Mitigation

To evaluate the resilience of the proposed ORAN-SDP architecture, we conducted a security analysis on a custom-built testbed. The environment was hosted on a Dell EMC server (Intel Xeon Gold 5318Y, 197GB RAM) running Proxmox VE 8.0 as a hypervisor. The testbed comprised 4 virtual machines, each provisioned with 12 host CPU cores, 11.72 GB of RAM, and running Ubuntu 22.04. The implementation consists of two parts, the ORAN setup and the SDP component setup. SDPC was installed on the FlexRIC machine as an xApp, SDP IH and SDP AH are present in all the ORAN elements, as per the proposed architecture. VM1 hosts the Near-RT RICs (FlexRIC/ ORAN-SC) and the 5G Core (OAI-5GC/ open5GS); VM2 hosts the O-CU (OAI-CU/ srsRAN); VM3 hosts the O-DU (OAI-DU); and VM4 was dedicated to hosting various FlexRIC and ORAN-SC xApps. For the use case of xApp-xApp communication, a Gateway xApp has also been deployed on the RIC machine. We also deploy ORAN-SC RIC on the RIC machine in order to demonstrate one of the attack scenarios. To facilitate our research, it was necessary to develop a custom implementation of the SDP

framework. Standard open source solutions, such as fwknop, lack the required flexibility, offering no support for custom protocols or standards like SCTP, and are confined to TCP, UDP, and ICMP. Consequently, we have implemented our own SDPC xApp, SDP-AH, SDP-IH, and SDP-Gateway xApp. The specific versions installed are: iptables v1.8.7 on all VMs, Linux Libreswan 3.32 (netkey), and OpenSSL 3.0.2 on all VMs.

To assess the security posture of the O-RAN-SC RIC, we designed and implemented a spoofed Denial-of-Service (DoS) attack, as illustrated in Fig. 4. Our methodology takes advantage of the fact that RIC indications are transmitted to the KPIMON xApp over a standard TCP connection. We implemented and deployed a malicious xApp programmed to continuously sniff the traffic directed at the KPIMON xApp. Upon observing a TCP packet, it constructs a fraudulent TCP RST packet with a sequence number incremented by one (`SEQ_RST = SEQ_OBSERVED + 1`) and sends this spoofed packet using the source IP address of the E2Term. As KPIMON xApp continuously receives these RST packets, it ceases sending TCP ACKs to the legitimate E2Term, leading to a disconnection. Although E2Term reattempts the connection, it persistently fails due to the RST packets injected by our malicious xApp. This results in an effective DoS on the KPIMON xApp, preventing it from receiving any RIC Indication messages for the duration of the attack. Fig. 4 illustrates

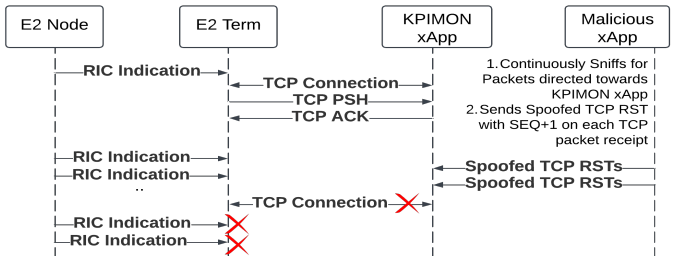

Fig. 4: Attack scenario: Spoofed DoS attack on xApp resulting in lost KPIs

how this attack is prevented. With the proposed architecture, such malicious packets are inherently discarded by the KPIMON xApp. The solution utilized is our proposed xApp-RIC callflow. In the proposed framework, for KPIMON xApp to subscribe to and receive RIC indications from E2Term, it is required to first complete the secure callflow. Upon successful completion, an IPSec or mTLS tunnel is established using ephemeral keys provided by the Controller xApp. This measure eliminates the malicious xApp's ability to sniff packets, as the communication is now protected with confidentiality. This addresses directly the root cause of the attack; without visibility into the TCP sequence numbers, a valid RST packet cannot be constructed. Even if an RST packet with a random sequence number was sent, it would be discarded by KPIMON xApp without further processing. If a malicious xApp were to attempt to send such RST packets, it would need to do so within the IPSec/mTLS tunnel. However, this would require obtaining the ephemeral credentials from the Controller xApp.

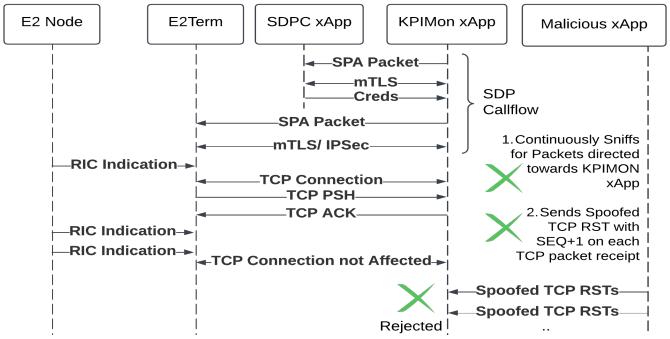

Fig. 5: Spoofed DoS attack prevention with the proposed xApp–RIC SDP callflow.

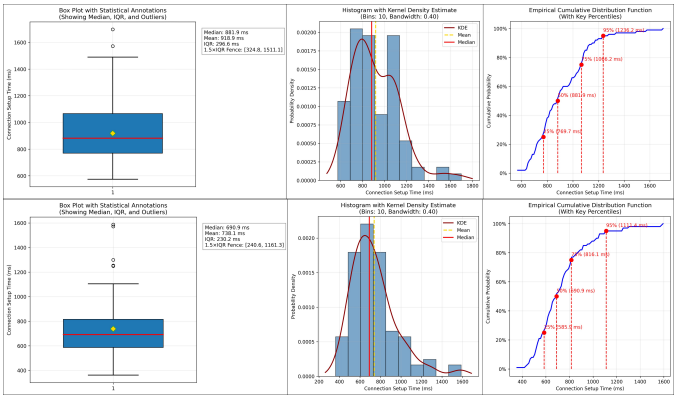

Fig. 6: SDP Connection time setup for IPSec and mTLS respectively

To acquire these credentials, the xApp must present valid authentication materials such as SPA, HMAC, mTLS key, and certificates which the unauthorized xApp does not possess. Thus, the proposed architecture effectively eliminates this attack vector. ORAN-SDP solution can extend to other use cases as well such as DoS via E2 or E42 Subscription Flood, DDoS on components like O-DU, DDoS on KPIMON as Producer xApp, etc.

IV. Experimental Results

A. Connection Setup Time

We measured the connection setup time over 100 runs with SDP integrated to O-RAN under two variations, presented in Fig. 6 (box plot, histogram, and CDF). In the first case, IPSec provided mutual authentication between the Consumer (O-DU / xApp / Consumer xApp) and the Producer / Gateway (O-CU / RIC / Producer xApp), while in the second case, mTLS was used. The connection setup time remained below 1000 ms in both cases. Although this procedure introduces some latency, it occurs only during the initial connection establishment and therefore does not impose significant overhead and is tolerable in O-RAN, where persistent sessions are commonly maintained. The average setup time with IPSec (median = 881.9 ms and mean = 918.9 ms) was higher than with mTLS (median = 690.9 ms and mean = 738.1 ms).

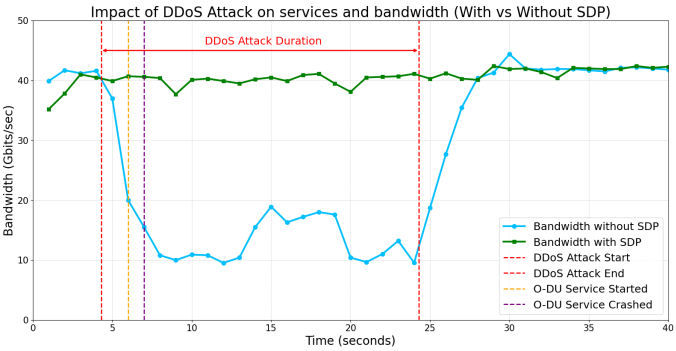

Fig. 7: Impact of DDoS Attack on services and bandwidth with and without SDP

B. Port Scanning Attack

Port scanning identifies open ports and services by sending crafted probes. We used Nmap with SYN requests (-sS stealth scan) to analyze host responses. In baseline tests without SDP, Nmap successfully detected exposed services. With SDP enabled, probes elicited no useful responses: ports appeared filtered (dropped by SDP policies) or closed (RST replies for nonexistent services). This confirmed that the SDP default-deny model and dynamic service cloaking effectively prevented reconnaissance, reducing the attack surface, and neutralizing port scanning as a viable threat in O-RAN.

C. DoS/DDoS Attack

We carried out a DDoS attack on the O-DU machine and measured its impact on bandwidth, with and without SDP. The observations are reflected in Fig 7. We found that the OAI-DU exposed a service on port 4043 for UE connections, which was also targeted along with other services running on the O-DU machine. The bandwidth was measured during the DDoS attack when trying to establish the O-DU to O-CU connection, with and without SDP. Without SDP, when the O-DU service was started, we were able to crash it using the DDoS attack. Even if DDoS was initiated after the O-DU to O-CU connection had been established, the O-DU service still crashed, resulting in loss of connection to the O-CU. A significant drop in bandwidth was also observed during the attack. However, with SDP enabled, the bandwidth remained unaffected. The O-DU successfully completed the SDP call flow, connected to the O-CU without any issues, and maintained the connection throughout. The O-DU could not be crashed, even when the DDoS attack was attempted after the connection had been established. We conducted similar experiments on other O-RAN components and observed consistent results. This further demonstrates that SDP effectively secures O-RAN components and provides robust protection against DoS and DDoS attacks.

V. Conclusion and Future Work

Validated against various attack scenarios, our proposed framework provides robust Zero Trust security for the entire O-RAN architecture such as the Near-RT RIC and internal xApps. It mitigates internal and external threats ranging from DoS, DDoS, port scanning, and MiTM to zero-day vulnerabilities by establishing zero visibility ("black infrastructure"). This is achieved through a combination of explicit and granular authorization, strong authentication, component validation, integrity protection, and confidentiality, ensuring that services remain invisible until explicitly authorized.

Future research can build upon this work by integrating anomaly detection into the SDPC to identify malicious xApps and dynamically adjust access policies, decentralizing the SDPC into a distributed model to improve resilience and scalability, and enabling periodic key rotation for SPA and HMAC keys to strengthen security.

Acknowledgement

This work was supported by the Center for Cryptography and Cyber Security at the Indian Institute of Technology Hyderabad and the Information Security Education and Awareness (ISEA) project funded by the Ministry of Electronics & Information Technology.

References

[1] H. Djuitcheu, T. Shah, M. Tammen, and H. D. Schotten, "Ddos impact assessment on 5g system performance," in *2023 IEEE Future Networks World Forum (FNWF)*, 2023, pp. 1–6.

[2] O-RAN Alliance, "O-RAN End-to-End Test Specification, Version 3," Tech. Rep.

[3] "Cve details - o-ran sc vulnerabilities," https://www.cvedetails.com/vulnerability-list/vendor_id-32690/O-ran-sc.html, accessed: Sep. 13, 2025.

[4] J.-E. Chang, Y.-C. Chiu, Y.-W. Ma, Z.-X. Li, and C.-L. Shao, "Packet continuity ddos attack detection for open fronthaul in oran system," in *NOMS 2024-2024 IEEE Network Operations and Management Symposium*, 2024, pp. 1–5.

[5] F. Feliana, T. Hung, B. Chen, and R. Cheng, "Evaluation of control/user-plane denial-of-service (dos) attack on o-ran fronthaul interface," in *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2024, pp. 01–06.

[6] S. A. Soleymani, M. Eslamnejad, H. Alimohammadi, A. Akbas, C. H. Foh, and M. Shojafar, "Ddos detection and mitigation using d/xapp in o-ran," in *2024 IEEE Future Networks World Forum (FNWF)*, 2024, pp. 283–290.

[7] Z. A. El Houda, H. Moudoud, and L. Khoukhi, "Blockchain meets o-ran: A decentralized zero-trust framework for secure and resilient o-ran in 6g and beyond," in *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2024, pp. 1–6.

[8] A. Moubayed, A. Refaey, and A. Shami, "Software-defined perimeter (sdp): State of the art secure solution for modern networks," *IEEE Network*, vol. 33, no. 5, pp. 226–233, 2019.

[9] Cloud Security Alliance, "Software-defined perimeter and zero trust," Cloud Security Alliance, Tech. Rep., May 2020, Available online (accessed June 22, 2025).

[10] ——, "Software-defined perimeter (sdp) architecture guide v2," Technical Architecture Guide, Cloud Security Alliance, Tech. Rep. v2, May 2019, Available online (accessed June 22, 2025).