On the Criticality of Integrity Protection in 5G Fronthaul Networks

Jiarong Xing*
Rice University

Sophia Yoo*
Princeton University

Xenofon Foukas
Microsoft

Daehyeok Kim
UT Austin

Michael K. Reiter
Duke University

Abstract

The modern 5G *fronthaul*, which connects the base stations to radio units in cellular networks, is designed to deliver microsecond-level performance guarantees using Ethernet-based protocols. Unfortunately, due to potential performance overheads, as well as misconceptions about the low risk and impact of possible attacks, integrity protection is not considered a mandatory feature in the 5G fronthaul standards. In this work, we show how vulnerabilities from the lack of protection can be exploited, making attacks easier and more powerful than ever. We present a novel class of powerful attacks and a set of traditional attacks, which can both be fully launched from *software* over open *packet-based* interfaces, to cause performance degradation or denial of service to users over large geographical regions. Our attacks do not require a physical radio presence or signal-based attack mechanisms, do not affect the network's operation (*e.g.,* not crashing the radios), and are highly severe (*e.g.,* impacting multiple cells). We demonstrate the impact of our attacks in an end-to-end manner on a commercial-grade, multi-cell 5G testbed, showing that adversaries can degrade performance of connected users by more than 80%, completely block a selected subset of users from ever attaching to the cell, or even generate signaling storm attacks of more than 2500 signaling messages per minute, with just two compromised cells and four mobile users. We also present an analysis of countermeasures that meet the strict performance requirements of the fronthaul.

1 Introduction

Modern 5G cellular networks are maturing in new and expanding deployments across the globe [50, 55, 57, 69]. They facilitate substantial increases in data rates, higher network capacity, ultra-low latency, and improved availability, ushering in a new era of emerging real-time applications, such as VR/AR, self-driving cars, and unmanned aerial vehicles.

* The first two authors contributed equally to this work.

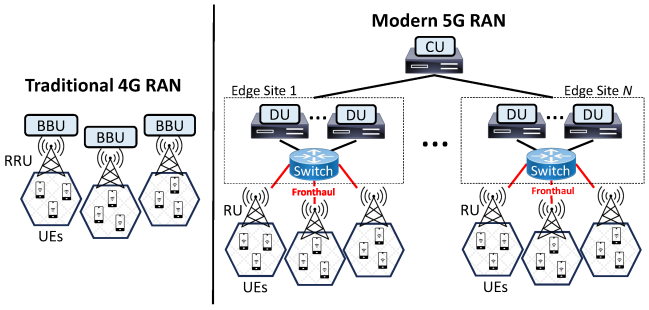


Figure 1: Disaggregated 5G RAN architecture: each CU handles multiple edge sites with racks of DUs, each DU connects to one or more RUs (over the packet-based *fronthaul* network), and each RU provides coverage for all UEs in their cell.

One major industry trend in 5G cellular networks is the *disaggregation* and *virtualization* of radio access network (RAN) functions. As shown in Figure 1, the baseband unit (BBU) and colocated remote radio unit (RRU) of traditional RANs (*e.g.,* used in 4G) are disaggregated into a Radio Unit (RU), Distributed Unit (DU), and Centralized Unit (CU) in modern 5G RANs, where one CU can serve multiple DUs. The RAN functions that previously ran on proprietary vendor-specific hardware are now virtualized, running in software on commodity off-the-shelf (COTS) servers, reducing vendor lock-in and enabling more rapid innovation [9, 34, 44].

A key part of the 5G RAN architecture is the fronthaul network, which transports user and control data between the DU and RU, to be converted into wireless signals for transmission to user equipment (UEs). Unlike traditional RANs, where fronthaul connectivity is realized using a proprietary link-layer protocol called the Common Public Radio Interface (CPRI), the 5G fronthaul uses *Ethernet-based* enhanced CPRI (eCPRI) [18], which was designed for performance and to enable emerging technologies, like Massive MIMO [27].

However, as an Ethernet-based interface, the fronthaul is vulnerable to packet manipulation attacks. Adversaries that have gained access to the physical RAN infrastructure [19, 38] can insert themselves between the DU and RU, acting as a

man-in-the-middle (MITM) adversary. From there, they can manipulate fronthaul packets to cause service degradation or connection disruption (*e.g.,* denial of service to attached UEs).

To protect against MITM attacks, integrity protection of fronthaul packets via solutions like MACsec and IPSec would be a natural approach. However, integrity protection of fronthaul traffic is currently *optional* in the protocol standards, due to concerns of increased processing delay incurred by potential security mechanisms, which could break the stringent performance requirements of the eCPRI protocol [38, §5.4.1.2, §5.5]. According to the O-RAN Security Work Group, the standardization body responsible for formulating security specifications for fronthaul [11, 12], lack of integrity protection over fronthaul is acceptable for three perceived reasons [38, §7.4: T-UPLANE-01]:

- **R1** Low likelihood for MITM attacks over fronthaul
- **R2** Costly sophistication required on the part of an adversary to launch attacks
- **R3** Low severity of potential attacks

In this work, we show for the first time that the above perspective on optional fronthaul integrity protection is flawed. Leveraging an enterprise-scale 5G testbed built on our premises with commercial-grade, standards-compliant RAN functions and RUs, we performed an extensive study, complete with both novel and traditional attacks. We make the following observations, which directly challenge the commonly accepted security stance:

- **O1** MITM fronthaul attacks are practical and feasible to launch, in a manner that bypasses the port-based network access control of IEEE 802.1X [3], on which the standards rely for their security stance (§3.2)
- **O2** Adversaries do not need to be overly sophisticated to launch meaningful attacks, and can directly manipulate vulnerable fronthaul traffic that is left unsecured by higher-layer protection mechanisms such as Packet Data Convergence Protocol (PDCP) (§3.3)
- **O3** Attacks exploiting the lack of fronthaul integrity can be severe, impacting RAN processes at a higher layer than that of the targeted DU, expanding over vast geographical regions, and affecting mobile users in cells that are not even directly under attack (§3.3)

To fully support our above observations, we introduce FRONTSTORM, a new, highly severe class of availability attacks that can impact higher layers of the RAN through signaling storms (**O3**) [25, 45]. We demonstrate that by carefully modifying and routing fronthaul packets, we can initiate higher layer processes (*e.g.,* cell reselection, handover) at a massive scale, equal to the number of the UEs attached to the cells, and at a very high rate. This leads to signaling storms at the CU that cover extensive geographical regions, impacting *many* DUs and all associated RUs and UEs. Such high-severity attacks can affect UEs not even associated with the targeted cells. Additionally, we present FRONTSTRIKE, a family of traditional attacks that breaks the fronthaul physical layer (in a similar manner to fake base station attacks, radio link jamming, and signal overshadowing) [31, 49, 71], but without requiring high levels of sophistication and hardware overheads from adversaries, who can directly modify fronthaul packets at line rate (**O2**). Unlike previous methods for launching these attacks, which require the use of transmitters (*e.g.,* a physical radio presence) and only target one cell at a time, our attacks operate at the *packet* level, making them much harder to detect and scalable to several cells at a time, since several RUs can be linked to the same affected DU.

Based on our findings, we conclude that integrity protection of the fronthaul traffic should be mandatory in the standards. Given the standardization bodies' concerns regarding the potential overhead of integrity protection on eCPRI traffic, we study the impact of the MACsec protocol to fronthaul performance. Our study demonstrates that, due to recent software and hardware advances, it is possible to achieve the necessary integrity protection at low cost and with minimal overhead, making it a practical solution. Finally, and to cover scenarios where integrity protection is absent, we present lightweight countermeasures leveraging real-time RAN analytics.

Responsible disclosure. We believe that knowledge of the vulnerabilities of an unprotected eCPRI-based fronthaul and the concrete high-impact attacks exposed in our study will be highly valuable to the broader 5G community and to the standards bodies. Thus, we have shared our report with the vendors of the equipment we worked with, and also disclosed our results to the standards bodies (ETSI [8] and O-RAN [10]), to bring awareness towards addressing these issues.

2 Background

In this section, we provide a brief background with relevant details on the 5G RAN architecture and 5G RAN fronthaul.

2.1 5G RAN Architecture

The O-RAN architecture is a widely accepted reference 5G architecture driven by the O-RAN Alliance [10] and 3rd Generation Partnership Project (3GPP) [6] standards bodies, which provide specifications for interfaces and protocols. O-RAN is globally supported by many major network operators, adopted by the European Telecommunications Standards Institute (ETSI) [8], recognized by hundreds of other operators, vendors, research and academic institutions, and is being deployed in many large-scale networks around the world today [29, 50, 55, 57, 69].

This subsection briefly provides relevant background on O-RAN principles of the 5G architecture.

<table>
<tr><th>C/U-Plane</th><th colspan="2">S-Plane</th><th>M-Plane</th></tr>
<tr><td rowspan="2">ECPRI / RoE</td><td rowspan="2">PTP</td><td rowspan="2">SyncE</td><td>NETCONF</td></tr>
<tr><td>SSH</td></tr>
<tr><td>UDP (optional)</td><td colspan="2" rowspan="2"></td><td>TCP</td></tr>
<tr><td>IP (optional)</td><td>IP</td></tr>
<tr><td>Ethernet + VLAN</td><td colspan="2">Ethernet</td><td>Ethernet + VLAN</td></tr>
<tr><td colspan="4">Physical Layer</td></tr>
</table>

Figure 2: Fronthaul protocols.

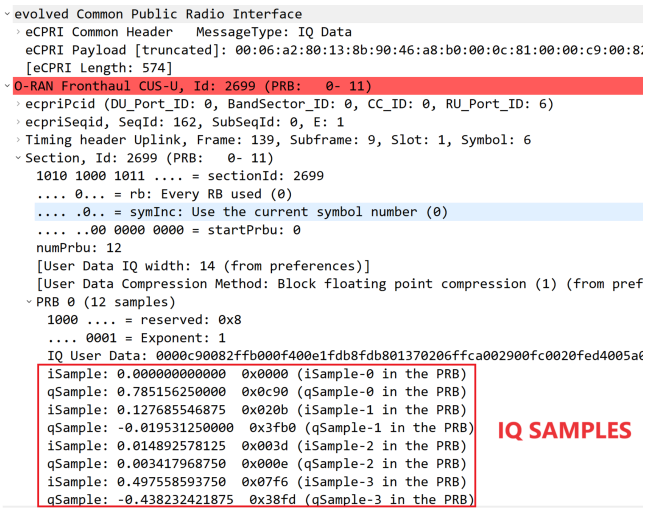


Figure 3: Wireshark capture of a representative U-plane packet, with I/Q samples as the complex number payload.

Disaggregation & open interfaces. As mentioned previously, one of the key architectural changes of the 5G RAN is disaggregation, which splits the previously centralized baseband unit (BBU) and colocated remote radio unit (RRU) into three logically separate units: a software-based centralized unit (CU), software-based distributed unit (DU), and hardware-based radio unit (RU) (see Figure 1). These disaggregated RAN elements are connected by *fronthaul*, *midhaul*, and *backhaul*, bridging communication between RU to DU, DU to CU, and CU to core, respectively. O-RAN also breaks open all previously closed and proprietary interfaces, instead using open protocols built on an Ethernet-based transport.

Virtualization. In traditional architectures, RAN functions were tightly integrated with vendor-specific hardware, effectively acting as embedded devices. In 5G, these "blackbox" elements can be abstracted as native functions running on commodity off-the-shelf (COTS) servers. As shown in Figure 1, software-based DUs can be hosted in a server rack at the edge and connected to other edge infrastructure (*e.g.,* switches). The decoupling of RAN software from dedicated hardware platforms encourages less vendor lock-in, more flexible RAN provisioning, simpler management, and improved cost efficiency, attracting an increasing number of cellular operators to follow this trend [29, 51, 57, 70].

2.2 5G RAN Fronthaul Design

2.2.1 5G Fronthaul Standards

As shown in Figure 2, modern fronthaul protocols run over Ethernet, making the packet structure of the fronthaul highly accessible (*i.e.,* publicly known). In the case of the control and user planes, packets are encapsulated using either eCPRI [18] or IEEE Radio over Ethernet (RoE) [2], with the eCPRI variant having met the most widespread success.

The eCPRI specification has been a cooperative effort amongst the biggest telco vendors (*e.g.,* Ericsson, Nokia, Huawei, NEC) [18], and defines the structure of the Ethernet frame carrying the fronthaul data (*e.g.,* types of eCPRI packets). However, certain implementation details (*e.g.,* the exact contents of the payload) are left out of the specification, meaning that eCPRI is not interoperable across vendors. To fill this gap, in recent years, the O-RAN Alliance and ETSI standardization bodies have built on top of eCPRI and have provided a full specification, which enables interoperability between the RUs and the DUs of different RAN vendors [8, 53]. The popularity of eCPRI-based O-RAN is apparent by the adoption that it is starting to see in the networks of major telco operators, like AT&T [29], Deutsche Telekom [15], DISH Wireless [47], O2 Telefonica [30, 62] and Vodafone [69]. For example, AT&T and Ericsson recently announced a major deal to see the AT&T RAN powered by Ericsson network functions using the O-RAN fronthaul protocol [29]. Given the widespread adoption of the eCPRI-based O-RAN fronthaul, we focus on this variant for simplicity in this paper. However, it should be noted that all of our observations also apply to the more general eCPRI specification.

2.2.2 O-RAN Fronthaul Specification

Now, we provide some details about the O-RAN fronthaul specification that are relevant for the remainder of this paper.

Communication planes. The fronthaul network enables the communication of the DU and the RU through downlink (DU-to-RU) and uplink (RU-to-DU) transmissions. The O-RAN fronthaul specifies four different communication planes: synchronization plane (S-plane), management plane (M-plane), control plane (C-plane), and user plane (U-plane). In this paper, we focus on the U-plane, which transfers waveforms transmitted to and from the radio in the frequency domain, carrying both user and cell data. While the C-, S-, and M-plane traffic is entirely internal to the fronthaul (remaining between the DU and RU) and is thus invisible to UEs, U-plane traffic carries data to and from the UEs and can have the most immediately obvious impact on UEs.

U-plane message details. The U-plane transports baseband signals between the RU and the DU. These signals are transferred in the form of *I/Q samples* in the frequency domain,

which are complex numbers with a real (I) and an imaginary (Q) part. Figure 3 shows a representative U-plane packet captured using our O-RAN testbed (§4.1). The number of I/Q samples each U-plane packet carries depends on the RU and cell configuration (*e.g.,* cell bandwidth, number of antenna ports, *etc.*). Each U-plane packet carries a set of I/Q samples that fit into one *symbol*. All DUs and RUs must transmit U-plane messages using a rigid symbol-based schedule, characterized by a specific strict transmission window (*e.g.,* 35 µs). The exact latency tolerance of the fronthaul depends on the supported use cases, but generally, it should not exceed 100 µs for typical deployment scenarios [52].

Among other fields, each U-plane packet has a source and destination MAC address (that of the DU or the RU) carried as part of an Ethernet frame (see Figure 4 for the transmission process). U-plane packets also contain an RU (logical) port ID, as part of their eCPRI header, that designates the antenna port that the I/Q samples are being transferred to/from.

Higher-layer signaling. Baseband signals transmit data to/from the higher layers of the RAN. This includes user application data and broadcast and control messages transmitted by RUs in the downlink direction from higher layers and requested by UEs on-demand in the uplink direction. Broadcast messages are required for downlink and uplink synchronization and carry real-time control signals that allow UEs to discover the cells and provide UEs with technical instructions on attaching to cells. The loss of these broadcast messages affects the ability of UEs to successfully attach to a cell.

To provide context for the attacks we demonstrate later, we briefly introduce here two important control messages transmitted in the downlink direction over fronthaul: the Synchronization Signal Block (SSB) and System Information Block 1 (SIB1). These are the first message blocks decoded by the UE during cell search, enabling it to identify the cell, synchronize timing, discover cell uplink and downlink configurations, and determine how to decode future message blocks.[1] For uplink, we study the Physical Random Access Channel (PRACH), which allows the UE to achieve uplink synchronization and align transmission timing with the radio.

3 MITM Attacks over 5G Fronthaul

In this section, we present our threat model, discuss the feasibility of MITM attacks, and introduce practical, high-impact fronthaul attacks (both novel and traditional), contrasted against the stance of the O-RAN standards body.

3.1 Threat Model

Setting. We consider a fronthaul network with one or more software-based DUs running on commodity servers in an edge site. For a single DU server, one or more commercial RUs can be directly connected (one RU per physical NIC port). In the case of multiple DU servers, the DUs can be interconnected via physical Ethernet cables, potentially through an Ethernet switch, to one or more RUs, which broadcast radio signals to all UEs in their coverage region. The adversary's goal is to insert herself as an MITM on the link between the DU(s) and RU(s) to stealthily modify fronthaul packets and cause connection degradation or disruption for users. The adversary could be placed on any of the available links (*i.e.,* on the direct connections between the RUs and the single DU server, on the links between the RUs and the switch, or on the links between the switch and the DU servers). We assume multiple RUs for our FRONTSTORM (§5) attacks, but not for our FRONTSTRIKE (§6) attacks. We assume the fronthaul network is secured using IEEE 802.1X, and that adversaries can bypass this protection by obtaining an initial foothold for MITM attacks through on-site access to the 5G edge site, through insider threats motivated by competition or financial gain, or through supply chain vulnerabilities, particularly from untrustworthy vendors, as discussed in §3.2.

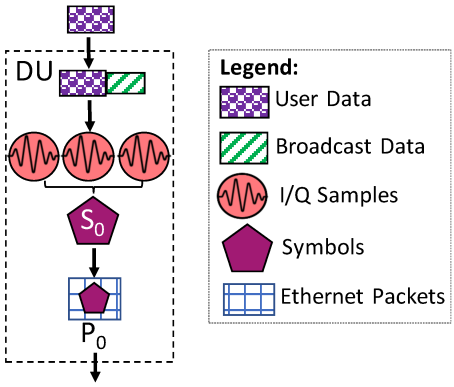


Figure 4: User data from higher layers are combined with data from lower layers (*e.g.,* broadcast data) at the DU. These are converted into I/Q samples and transmitted at structured times (*i.e.,* symbols). I/Q samples for a given transmission window are sent over fronthaul as encapsulated Ethernet packets.

Cell configuration. We assume that frequency and bandwidth configurations are common across cells, but this requirement can be relaxed for most of our attacks. Our FRONTSTRIKE attacks (§6) will work regardless of how the cell is configured with frequency or bandwidth. Additionally, one of our FRONTSTORM attacks (attack A2, §5.3) could work even with cells of different bandwidths, as long as they have overlapping frequencies. We make no assumptions about the configuration of the cells in terms of radio-related parameters.

Stealthiness. Our definition of stealthiness is that our attack methods (*e.g.,* packet modification) do not trigger incorrect behavior on the RU or UE, but directly exploit appropriate responses from the RU and UE to cause undesirable behavior.

[1] We focus on these messages for simplicity, but our attacks can also generalize to other messages.

We note that similar to existing DoS and signaling attacks, an operator can see the *effect* of our attacks (*e.g.,* service outage, cell swapping, etc.). However, our attacks remain stealthy since the *cause* of the attack is not directly apparent and could be from benign activities (*e.g.,* signaling storm due to a flash crowd event, like a large number of UEs attaching to a cell simultaneously after an airplane landing).

3.2 Feasibility of MITM Attacks

The O-RAN security standards body deemed the likelihood of MITM attacks to be low (**R1**) because of existing security requirements: port-based authentication of RAN equipment with IEEE 802.1X [38, §7.2, §7.4]. However, our investigation reveals that this assumption does not hold in various scenarios, creating vulnerabilities that could be exploited.

On-site 802.1X bypassing. O-RAN requires a device to be authenticated and authorized through IEEE 802.1X [3] before it can connect to the fronthaul network, which prevents illegal access and potential security breaches from rogue devices. However, in the case of wired networks, IEEE 802.1X is vulnerable to interceptors that introduce passive devices in the link with on-site access [24]. On-site access is a feasible and likely entry point for 5G fronthaul, where server clusters are typically co-located with base stations and spread geographically over thousands of edge sites. Due to the location and distribution of edge sites, particularly in public spaces, obtaining physical access to fronthaul infrastructure (akin to accessing outdoor IoT devices) is easier than accessing other parts of the network or traditional centralized cloud data centers [26]. Most fronthaul deployments are located on sidewalks, rooftops, or basements of buildings [20], and because of this, major vendors are strongly advocating for Zero-Trust Architectures for O-RAN security [28]. Adversaries with physical access to the 5G edge site can bypass IEEE 802.1X by inserting a rogue server with two network interfaces (*e.g.,* a mini PC [5]) into the fronthaul. Figure 5 illustrates an example scenario. As demonstrated in prior work [24], the rogue device could work as a network bridge that modifies and forwards traffic using the original connections already authenticated by IEEE 802.1X.

Insider threats. In addition to external threats, fronthaul MITM attacks can be enabled by insider threats, which originate from within the targeted organization. This can include current or former employees, contractors, or business associates with inside access to the company infrastructure. A recent survey revealed that insider threats have become more frequent, and more than 50% of surveyed organizations experienced such threats at least once in 2023 [40]. Insider threats facilitate the ease of launching MITM attacks by installing malicious devices within the fronthaul cluster. This method parallels the previously discussed 802.1X bypass but with the added advantage of legitimate on-site access. Moreover, insiders could even bypass 802.1X remotely by installing malicious software on the DU servers to enable packet interception. Example motivators for employees to engage in such attacks could include financial gains from service competitors or harbored resentment for former organizations.

Figure 5: An adversary can intercept and manipulate fronthaul traffic without violating IEEE 802.1X by introducing a rogue server with two network interfaces into the fronthaul network. To stay stealthy, the rogue server can be a mini PC (*e.g.,* GoWin R86S Pro [5]) with low power consumption. It works as a network bridge, enabling the adversary to view and modify the fronthaul traffic. Solid arrows indicate normal traffic, while dashed arrows represent manipulated traffic.

Supply chain vulnerabilities. With 5G infrastructure being built by multiple global vendors, supply chain security becomes a major concern. A recent government report [4] identified 5G supply chain attacks as a significant threat vector. Using fronthaul hardware and software from untrusted providers (*e.g.,* adversarial countries) provides a foothold for MITM attacks. For instance, an adversary could leave a backdoor on the fronthaul switch hardware to manipulate the fronthaul traffic. Notably, such breaches will not result in detection from 802.1X protocol violations, making them particularly stealthy.

3.3 Practical, High-Impact Attacks

Recall that the standards group deemed that because of the existing security requirement for the Packet Data Convergence Protocol (PDCP), which is expected to provide integrity protection of user data at higher layers between the *CU and UE*, an adversary would need to be sophisticated to bypass this protection and launch attacks (**R2**) [38, §7.2, §7.4]. They also deemed that potential attacks on the fronthaul would only have low severity (**R3**), with the expectation that any impact on the RAN would be minor or unnoticeable and that only *one* DU/RU pair could be affected [38, §7.1, §7.4]. In contrast, we show how the PDCP security mechanism does not safeguard all fronthaul traffic between the *DU and RU*, leaving critical messages and traffic generated from layers lower than that of PDCP unsecured and vulnerable to adversaries that are not overly sophisticated. We also show several practical, high-impact attacks over fronthaul, which break the existing security assumptions, and can impact *many* DU/RU pairs.

Limitations of PDCP integrity protection. As shown in Figure 6, PDCP is an L2 protocol in the 5G protocol stack that operates between the CU and the UEs. PDCP provides several services, including integrity protection, to ensure that data

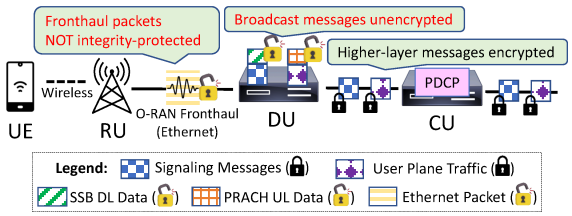


Figure 6: While higher-layer messages (*e.g.,* user data and control-plane signaling messages) are encrypted, broadcast messages (*i.e.,* lower-layer control data related to initial cell search and attachment) are unencrypted. Meanwhile, all I/Q samples encapsulated into Ethernet-based packets and sent over fronthaul are not integrity-protected.

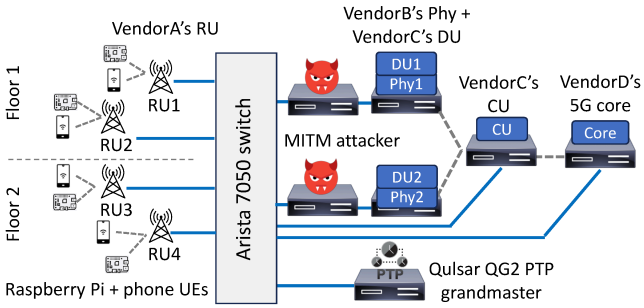


Figure 7: Testbed setup, with emulated MITM adversary.

packets are not tampered with during transmission. It achieves this by generating and validating a Message Authentication Code (MAC-I)[2] for each data packet, ensuring detection of unauthorized modifications to the contents of the data packets.

However, our study shows that PDCP is insufficient to protect the open fronthaul from MITM attacks, even from relatively unsophisticated adversaries. First, certain fronthaul traffic originates from layers lower in the 5G protocol stack than PDCP. For example, as shown in Figure 6, broadcast messages (*e.g.,* SSB data carrying cell selection information needed for UE attachment) or reference signals used for channel estimation and signal quality are generated by the MAC and PHY layers of the DU. This type of traffic falls outside the purview of PDCP, thus remaining unsecured, and exposing the system to packet modification attacks that target the intermediate RU and DU. Second, even for higher layers, the MAC-I generation and validation require the senders and receivers to use a negotiated key, which is only attainable *after* UEs attach to a cell, leaving all *pre-attachment* messages entirely unprotected. In other words, all traffic associated with the initial UE attachment procedure, particularly all broadcast messages, remains unprotected by PDCP.

This vulnerability enables relatively unsophisticated adversaries to launch high-impact MITM attacks as follows:

FRONTSTORM attacks. Leveraging the fact that each CU handles multiple edge sites, we demonstrate a novel class of high-impact attacks called FRONTSTORM. These attacks introduce signaling storms [25, 45] at the CU, significantly degrading CU performance through fronthaul routing manipulation and I/Q sample multiplexing. The attack impact extends to vast geographical regions, impacting all DUs and their associated RUs and UEs, even those that are not directly associated with the cells where the attacks are initiated.

FRONTSTRIKE attacks. We further present a family of traditional attacks we call FRONTSTRIKE attacks, which target breaking the physical layer. Once an adversary has gained the status of MITM, she can easily attack the system by directly viewing and modifying the fronthaul traffic on the fly, using only "simple" packet capture and modification techniques, without additional hardware or sophisticated signal-processing mechanisms. This can leak critical information about the cell, degrade the cell's performance, or cause denial of service to all UEs within the cell. Our FRONTSTRIKE attacks achieve similar goals and effects as traditional physical layer attacks (*e.g.,* fake base station attacks, radio link jamming, and signal overshadowing) [43, 46, 48, 59–61, 63, 64, 67]. However, unlike prior work, it does *not* require a transmitter and can operate on a much larger scale, encompassing all the cells that are under the control of the affected DU.[3]

Next, we describe our setup and steps for launching attacks (§4). We then extensively demonstrate and evaluate FRONTSTORM and FRONTSTRIKE attacks in §5 and §6.

4 Setup and Attack Preparation

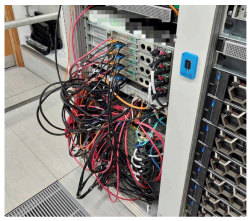
4.1 Commercial-Grade Multi-Cell Testbed

We demonstrate our attacks leveraging the infrastructure of an enterprise-scale O-RAN testbed [14] (Figure 7). Our study leverages commercial-grade O-RAN RUs from VendorA[4] with 100 MHz 4x4 MIMO operating at 3.5 GHz. Our cluster has a rack of HPE Telco DL110 servers, each featuring Intel Xeon 6338N CPUs, an Intel E810 100 GbE NIC, and an Intel ACC100 accelerator for PHY forward error correction (Figure 8a). The servers were optimized for real-time performance, running Linux kernel v6.1 with real-time patches applied. They run VendorB's PHY layer, VendorC's 5G stack for the DU and CU layers, and VendorD's 5G Core, all of which are O-RAN compliant components. The RU and servers are interconnected via a 100 GbE Arista 7050 switch. To achieve time synchronization between the RUs and DUs, we use a Qulsar QG2 PTP grandmaster clock, that feeds the synchro-

[2] We use MAC-I to denote a message authentication code scheme for integrity protection, and MAC to denote RAN's L2 medium access control.

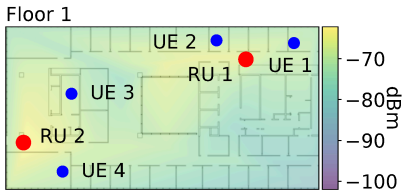
[3] Typical deployments could span a minimum of 3-6 cells per DU.

[4] Considering that the attacks we study are related to the fronthaul specifications and not to a vendor implementation, we have anonymized the vendors we used during our evaluation for fairness. All the vendors we used are amongst the O-RAN compliant vendors listed in the TIP Exchange [68].



(a) 5G server cluster

(b) Phone and Raspberry Pi



(c) RU/UE locations & signal coverage

Figure 8: Our commercial-grade multi-cell 5G O-RAN testbed. (a) is our O-RAN server cluster. (b) shows a phone UE and a Raspberry Pi UE. (c) plots the cell signal coverage and UE placement on one of the floors in our building. The rest of the RUs and UEs in our testbed are at similar locations on another floor; their images are omitted.

nization signal through the Arista switch. To emulate a realistic deployment scenario, we use four RUs and eight UEs distributed across two floors within our building. Each floor accommodates two RUs and four UEs. All cells use a subcarrier spacing of 30 KHz (*i.e.,* each slot is 500 µs) and are configured with the same central frequency of 3.46GHz. The UEs are OnePlus Nord N10 5G smartphones and Raspberry Pis equipped with Quectel RM502Q-AE modems (Figure 8b). The signal coverage and locations of RUs and UEs on one of the floors is shown in Figure 8c.

We emulate an MITM adversary by deploying an additional server with identical configurations within the rack. This server sits between the Arista switch and the DU server (as shown in Figure 7), and it has two 100G Intel E810 NIC ports. We connect the DU to NIC port one and the Arista switch to NIC port 2. It is worth noting that in practical scenarios, adversaries typically do not need to use such powerful servers and NICs. A more modest setup, such as a mini server with dual 25G NIC ports (*e.g.,* GoWin R86S Pro [5]), is often sufficient to manage most 5G O-RAN deployments.

4.2 Manipulating Fronthaul U-Plane Packets

We have implemented our attacks in approximately 1000 lines of C++ code based on DPDK for high performance. Our attacks capitalize on the lack of integrity protection of both the header and payload of U-plane messages, stealthily modifying fronthaul I/Q samples at line rate without raising an alarm. For traffic that does not need to be modified, we perform passive eavesdropping and simple forwarding, *i.e.,* each packet received from the DU is passively forwarded to the RU, and vice versa. To launch the different FRONTSTORM (§5) and FRONTSTRIKE (§6) attacks, we implemented the following low-level packet manipulation capabilities, toward the goal of causing UE misbehavior or service disruption:

`C1`: Ethernet header manipulation. We modify the source and destination MAC addresses of U-plane packets to reroute them to different RUs or to make the DU believe that the packet arrived from a different source RU than it did.

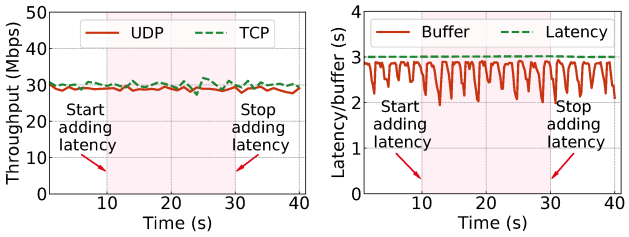
`C2`: I/Q sample multiplexing. We sum the values of I/Q samples carried in the U-plane packets for two cells. Recall that I/Q samples are represented as complex numbers, so this operation corresponds to an element-wise addition, *e.g.,* the first I/Q sample of cell 1 is added to the first I/Q sample of cell 2. This operation is equivalent to having both cells transmit and receive signals in overlapping regions, causing interference with each other.

`C3`: I/Q sample scaling. We multiply the values of I/Q samples carried in a U-plane packet by a scaling factor. The amplitude of the signal vector in the complex plane captures the signal's power. Thus, scaling the I/Q samples by a factor is equivalent to scaling the signal's power (*e.g.,* attenuating) by the square of that factor.

`C4`: I/Q sample replacement. We replace the payload of U-plane packets with a different set of I/Q samples. The I/Q samples could be completely random, actual I/Q sample traces captured and replayed from the same or a different cell, or I/Q samples manipulated through `C2` and `C3`.

Compression. I/Q samples carried in U-plane packets are typically compressed using standardized lightweight compression techniques (*e.g.,* Block Floating Point (BFP) and μ-law compression). Thus, to perform operations on I/Q samples, we must first decompress them before modification, and then recompress them before forwarding. We implement BFP-based (de)compression, which our RU supports. To reduce latency overhead, we accelerate I/Q sample manipulation with Intel Advanced Vector Extensions 512 (AVX-512) instructions.

Timing sensitivity. We note that the operations `C1`-`C4` used by our attacks are lightweight and thus do not violate the stringent latency constraints of the fronthaul. Specifically, for all the operations above, the processing overhead introduced in the fronthaul ranges from approximately 80 ns up to 20 µs, including the compression/decompression step. Such a delay is invisible to the higher-layer protocols, allowing us to launch our attacks without breaking the RU to DU connection. To demonstrate this, we measured the downlink TCP and UDP throughput, as experienced by an attached Raspberry Pi UE, while introducing 20 µs of extra latency in the U-plane packets. As shown in Figure 9a, both TCP and UDP throughput remain unaffected throughout the experiment. To further assess the impact of this latency on real-world applica-



(a) TCP and UDP throughput (b) DASH latency and buffer

Figure 9: Processing overheads of 20 μs incur no observable performance disruption for TCP, UDP, and DASH streaming.

tions, we used our phone UE to measure the performance of DASH.js [35], a live video streaming application that plays low-latency streams and reports streaming quality metrics including live latency, video bitrates, and video buffer sizes. We set a target latency of three seconds, which requires the DASH player to stream the video with a delay of at most three seconds behind the live event. For other settings, we adhere to the values recommended by the framework. Figure 9b presents the buffer size and lag latency during the experiment, demonstrating that the overhead we introduced had no visible impact on the application.

5 FRONTSTORM: Amplification Attacks

In this section, we first describe a passive information retrieval attack (A0), which an adversary could use to support all of our other attacks. We then present FRONTSTORM, a new family of amplification attacks leveraging capabilities C1–C4. FRONTSTORM attacks can cause signaling storms that affect the availability of the CU. The high-level idea behind these attacks is to force UEs to generate a high rate of signaling messages toward the CU. Consequently, the CU can become unresponsive, negatively impacting users in a wide region beyond the cells that are directly under attack.

5.1 A0: Cell Information Collection

In our simple forwarding state (before any packet manipulation), an adversary can perform passive eavesdropping on fronthaul traffic, obtaining cell information that could be used in other attacks. For example, by collecting sufficient downlink traffic samples in a DU-RU fronthaul connection, adversaries can capture periodically transmitted and unencrypted Master Information Block (MIB), SSB, and SIB1 messages within these samples. Using the same signal processing techniques as those employed by the base station, which are openly standardized and easily accessible through open-source implementations like OpenAirInterface and srsRAN [13, 65], adversaries can infer important cell configurations such as the cell numerology, physical cell ID

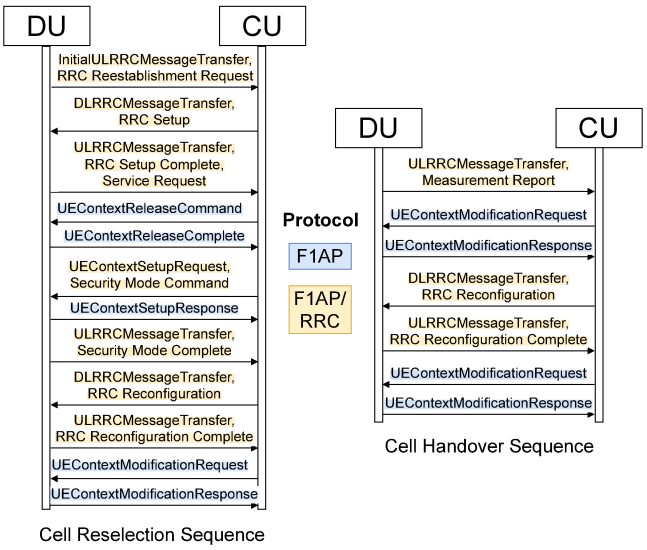


Figure 10: Message sequences for a single UE during cell re-selection and cell handover, taken from a Wireshark trace captured in our 5G testbed.

(PCI), Public Land Mobile Network (PLMN) ID, I/Q sample compression scheme, and symbol positions for PRACH data. Moreover, this information enables adversaries to launch more targeted attacks. Specifically, the above information allows a motivated adversary to create a complete map of all the cells hosted by the DU and then designate which cells to attack based on a range of cell IDs.

5.2 A1: Signaling Storm via Cell Reselection

In this attack, we utilize a radio event called Radio Link Failure (RLF) [32, §5.3.10], which occurs when the radio link between the UE and the DU is lost, *e.g.,* due to interference, bad coverage, and failed handovers. When this happens, the UE will seek to recover the signal and sets an RLF timer for this process. If the recovery fails and the timer expires (typically in a few hundred milliseconds), the UE will trigger a process called *cell re-selection*. In this re-selection process, the UE measures the signal strength of available cells within its range and selects the one with the best signal for attachment. Then, the UE establishes a connection with the chosen cell, triggering message exchanges between the DU and the CU. This process can be seen in the left part of Figure 10 for a natural cell-reselection trace captured in our testbed, spanning 12 messages (7 DU-to-CU and 5 CU-to-DU). At a high level, the UE triggers the process by sending an *RRC Reestablishment Request* message to the CU via the DU, followed by a sequence of messages by which the CU tears down the context of the UE from the old cell and creates a new context at the new cell.

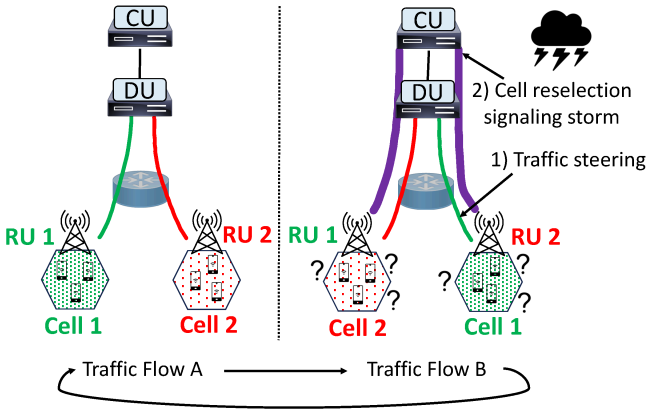
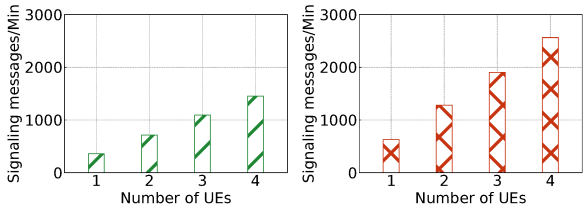


Figure 11: Signaling storm attack via cell re-selection. *Traffic Flow A* shows the DU transmitting fronthaul packets destined for Cell 1 and Cell 2 through RU 1 and RU 2, respectively.[5] *Traffic Flow B* shows our traffic steering, which causes traffic destined for *Cell 1* to go through *RU 2*, while traffic for *Cell 2* goes through *RU 1*. UEs attached to the respective cells believe they have lost their connection since they begin to receive traffic inconsistent with their existing cell configuration. All UEs then initiate the cell re-selection process, generating dozens of messages *per device*, and placing the CU under a signaling storm. After two seconds (allowing time for all UEs to reassociate to the new cell), *Traffic Flow A* is again adopted. While not shown in this figure for simplicity, returning to this traffic steering pattern would again cause cell re-selection and continue the signaling storm.

The cell re-selection process and RLFs have been used in the past for attacks trying to determine the precise location of users [63]. In contrast, in this work, we leverage the fact that the cell re-selection process generates a long sequence of messages to/from the CU, and a single cell can have hundreds of UEs attached or in an active state [56], forcing all UEs to perform cell re-selection to cause a signaling storm to the CU. We achieve this using capability `C1`, which periodically steers the fronthaul packets of cells from one RU to another by swapping the MAC addresses. This action makes the RUs transmit the signal of a different cell every few seconds, making the UEs experience an RLF and periodically trigger the cell re-selection process. This attack is illustrated in Figure 11, for the case of two cells. During time window t1, we simply forward the traffic between the RUs and the DU, with RU 1 communicating with cell 1 and RU 2 with cell 2. However, in time window t2, we swap the MAC addresses of fronthaul packets, steering all the traffic of cell 1 to/from RU 2 and all traffic of cell 2 to/from RU 1. As a result, the UEs lose the signal of their original cell and are forced to trigger the cell re-selection process and attach to the new cell that is visible to them. In time window t3, we return to the same

[5]For visual clarity, flows for each cell are shown as separate streams, but in practice, all flows traverse a single port on the switch link.



(a) Via cell re-selection (`A1`) (b) Via handover (`A2`)

Figure 12: The effectiveness of FRONTSTORM attacks (generated signaling messages per minute) linearly increases with the number of UEs.

MAC address assignment as in t1, making the RUs transmit their original signal and triggering another cell re-selection. This process is repeated periodically, every few seconds, generating a constant volume of control plane traffic to/from the CU proportional to the number of active UEs in the affected cells.

Validation. We validated the feasibility of this attack using two cells (Figure 7 and Figure 8c show the setup), initially mapping cell 1 to RU 1 and cell 2 to RU 2. We swapped the signals of the two cells every two seconds and captured all the messages exchanged between the DU and the CU during the attack. First, we verified that the attack works by analyzing the captured trace and observing that, for each UE, the message sequence of Figure 10 was indeed generated approximately every two seconds. For higher confidence in our results, we analyzed UE-level traces using QUALCOMM eXtensible Diagnostic Monitor (QXDM) logs and correlated them with the CU-side traces to validate that an RLF triggered the message sequence. It should be noted that the choice of two seconds for traffic swapping was driven by the observation that for lower values (*e.g.,* swapping every one second), cell re-selection would not be triggered consistently for all UEs, leading to the generation of a smaller number of signaling messages. We believe this is due to the RLF timer's value, which might not be set high enough to trigger the cell re-selection process before the traffic is swapped again. Different UE and cell configurations could cause the most effective timing interval for traffic swapping to vary, but this can be tuned quickly.

Using the above setup, we measured the impact of this attack on the CU by counting the number of messages exchanged between the DU and the CU for a varying number of UEs for one minute. The results can be seen in Figure 12a for up to four UEs (total number of UEs for both cells). As we can observe, the number of signaling messages increases linearly with the number of UEs. We could generate more than 1400 signaling messages with just four UEs in just one minute. To put these numbers into perspective, according to an Ericsson report, just 500 IoT devices generating more than 100 signaling events per hour could lead to network congestion [25]. Considering that, according to recent measurement

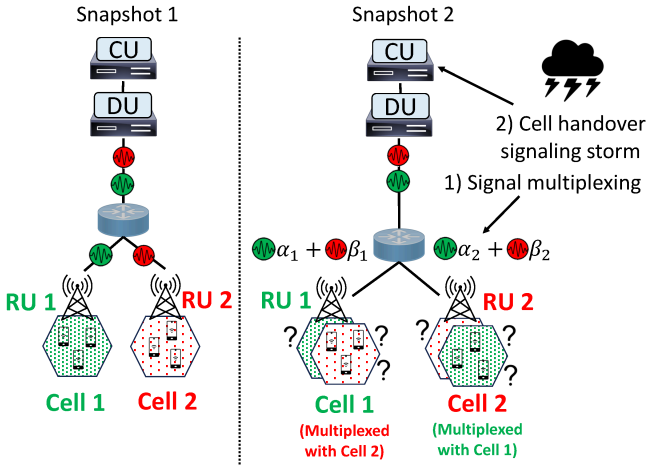


Figure 13: Signaling storm attack via cell handover. Snapshot 1 shows the normal behavior of DUs transmitting I/Q samples (in packet payloads) over fronthaul. The I/Q samples carry signals uniquely associated with respective RUs and associated cells. In Snapshot 2, we perform signal multiplexing, scaling the I/Q samples by α_1 and β_1 for Cell 1 and by α_2 and β_2 for Cell 2, and transmitting the sum of the two signals to the cells. The α and β factors are smaller than 1, thus effectively attenuating the individual signals and causing an overlapped signal to be delivered to the UEs. In this example, by choosing $\alpha_1 < \beta_1$, the signal for Cell 1 can be made to appear weaker than that of Cell 2, causing all the UEs in Cell 1 to initiate a cell handover process, generating many messages to the CU and causing a high-severity signaling storm. Simultaneously, by making $\alpha_2 > \beta_2$ for the signal transmitted to Cell 2, the signal for Cell 2 can be made to appear weaker than that of the signal for Cell 1, causing all the UEs attached to Cell 2 to initiate another handover process.

studies, real cells typically have more than 40 UEs active at any point in time [56], our attack could generate up to 30K signaling messages per minute (1.8M per hour) for just the two cells of the validation experiment.

5.3 A2: Signaling Storm via Handover

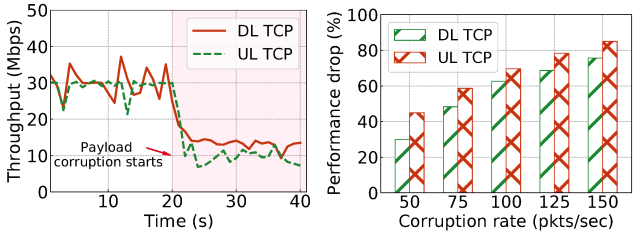
As noted in A1, while the attack was successful, we were limited in the number of times we could trigger the cell re-selection (*i.e.,* approximately once every two seconds in our setup). Here, we demonstrate how we can place even more stress on the CU by leveraging a different 5G process, namely intra-DU handovers (*i.e.,* UE handovers between cells of the same DU). As shown in the following, this enables us to generate nearly twice the number of signaling messages compared to attack A1, all within the same time frame.

The idea behind this attack is to give UEs the "illusion" that the signal quality of their attached cell goes below some threshold compared to a neighboring cell. This event triggers the handover process illustrated in the *Cell Handover Sequence* trace captured in our testbed (shown in the right part of Figure 10). Once the UE detects that the signal quality of the current cell is below some threshold compared to a neighboring cell, it sends a measurement report to the CU (first message of the *Cell Handover Sequence* in Figure 10). The CU then triggers a handover (last six messages of the *Cell Handover Sequence* in Figure 10), which involves the CU notifying the DU about the cell change and re-configuring the context of the UE.

While the handover process requires fewer CU-DU signaling messages than the cell re-selection process (seven messages for handover vs. twelve messages for cell re-selection), it can be performed more frequently, leading to a larger total number of CU signaling messages. The handover frequency is (among others) driven by three CU-configured parameters called *Hysteresis*, *TimeToTrigger* and *reportInterval* [32]. *Hysteresis* indicates a condition that must be met for the UE to move to a new cell (*e.g.,* new cell has *N* dB higher signal quality). *TimeToTrigger* indicates how long this condition has to be met for the handover to be triggered (*e.g., M* ms), to avoid ping-pong effects, while *reportInterval* specifies how frequently the measurements should be reported. The *TimeToTrigger* value is typically configured to be a few tens or hundreds of milliseconds, and the *reportInterval* hundreds of milliseconds up to a few minutes. Thus, a handover could be triggered several times per second, assuming the *Hysteresis* condition is met.

To launch this attack, we multiplex the I/Q samples of fronthaul traffic between multiple cells, by leveraging capabilities C2-C4. Specifically, as illustrated in Figure 13, we combine the payload of U-plane packets targeting the same slot and antenna port of two cells by scaling their I/Q samples by some factor (capability C3), summing up the results (capability C2), and then using the result as the new payload of the packet (capability C4). Capability C2 allows us to overlap the coverage area of two cells on the same RU, allowing all UEs in the range of the RU under attack to detect both cells. By frequently re-adjusting the signal quality of each cell through the scaling factor of capability C3, we can manipulate the signal quality of the cells and, consequently, the observed *Hysteresis*, making one cell appear to have a stronger signal quality than the other and triggering a handover. A few hundred milliseconds later, we re-adjust the scaling factors of the two cells, forcing all the UEs to move back to the original cell through another round of handovers.

Validation. As with A1, we verified the feasibility of this attack in our testbed using two cells. In our setup, the *Hysteresis* parameter is set to 0.5dB, the *TimeToTrigger* to 40ms, and the *reportInterval* to 480 milliseconds. We used a scaling factor of 1 for cell 1 (no signal attenuation) and 0.8 for cell 2, swapping the scaling factors of the two cells every 700ms. Similar to A1, we verified this approach by capturing the Cell Handover Sequence shown in Figure 10 and QXDM logs at



(a) TCP throughput (b) Performance degradation

Figure 14: The performance impact of attack A3. Data is measured on a Raspberry Pi UE. DL: downlink; UL: uplink.

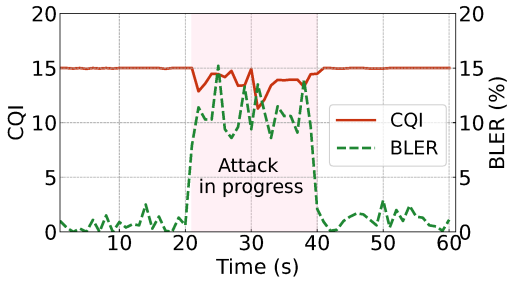


Figure 15: Payload corruption leads to lower Channel Quality Information (CQI) and higher Block Error Rate (BLER).

the UE side and by observing that a handover was triggered approximately every 700 ms due to a UE measurement report, once the scaling factors of the cells under attack were swapped. We then measured the impact of the attack on the CU by counting the number of messages exchanged between the DU and the CU for a varying number of UEs (up to four) within one minute. As illustrated in Figure 12b, similarly to A1, the number of signaling messages increased linearly with the number of UEs, but in this case, we were able to generate more than 2500 signaling messages for four UEs (approximately three handovers every two seconds), a number almost double that of attack A1.

6 FRONTSTRIKE: Signal Modification Attacks

FRONTSTRIKE attacks leverage capabilities C3 and C4 to modify radio signals (at the packet level) and break the connection and/or performance of UEs in the cell region. We introduce three attacks (A3-A5) targeting distinct aspects of the fronthaul traffic, *e.g.,* user packet payload (A3), downlink SSB symbols (A4), and uplink PRACH symbols (A5). While achieving similar effects as physical layer attacks in the literature (*e.g.,* fake base station attacks, signal overshadowing [31, 49, 71]), these attacks possess the capability to extend their impact to all cells controlled by the victim DU.

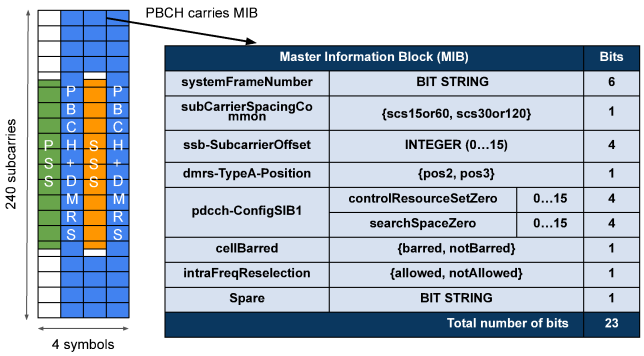
6.1 A3: Payload Corruption

The most straightforward FRONTSTRIKE attack is to corrupt regular packets by replacing I/Q samples with invalid user payloads using capability C4 (*e.g.,* with random or all-zero payloads). The rationale behind this attack is that the modified I/Q samples could result in decoding failures and packet drops during the signal processing at the physical layer. From the perspective of UEs, the effects of the modified packets manifest as packet loss at the RAN L2 layer or as noise into their communication channels, which forces the base station to perform retransmissions and to use more robust and less efficient modulation and coding schemes. Both effects eventually lead to the reduction of the UEs' traffic rates.

Validation. We implemented this attack in both directions by corrupting randomly selected symbols that carry user data and/or signal quality reference signals. Figure 14a shows the `iperf` TCP throughput during the attack, with the adversary corrupting packets at a rate of 75 packets/sec. As shown, the attack significantly reduced connection throughput by 57% on average for downlink and by 71% on average for uplink. Figure 14b plots the percentage of throughput drop at various modification rates. It shows that the attack's effectiveness increased with the corruption rate, leading to a throughput drop of up to 76% for downlink and 85% for uplink with a corruption rate of 150 packets/sec. To further understand why performance dropped, we measured the Block Error Rate (BLER) and Channel Quality Information (CQI) of the UE under study. As shown in Figure 15, during the attack, the CQI dropped from 15 to 13.5 on average. This means the quality of the channel, as perceived by the UE, degraded, forcing the radio resource scheduler to choose more robust, but less efficient, modulation and coding schemes. Meanwhile, the BLER increased from almost 0% to more than 10% when the attack started, meaning a higher fraction of erroneous blocks were received and had to be retransmitted.

6.2 A4: Downlink SSB Modification

As briefly discussed in §2.2, 5G relies on periodically broadcasted control signals for initial access, collectively called the Synchronization Signal Block (SSB). This block comprises four parts: Primary Synchronization Signal (PSS), Secondary Synchronization Signal (SSS), Physical Broadcast CHannel (PBCH), and PBCH DeModulation Reference Signal (PBCH-DMRS). When 5G UEs try to attach to a cell, they rely upon the PSS and SSS signals to attain both time and frequency synchronization and obtain the cell PCI. After that, UEs decode the Master Information Block (MIB) from PBCH, a succinct 23-bit structure containing eight parameters crucial for cell selection (Figure 16). Notably, the `pdcch-ConfigSIB1` parameter dictates which time and frequency resources UEs need to decode SIB1, encompassing additional essential information along with the MIB. The `cellBarred` determines whether UEs are permitted to select this particular cell, while



Master Information Block (MIB)			Bits
systemFrameNumber	BIT STRING		6
subCarrierSpacingCommon	{scs15or60, scs30or120}		1
ssb-SubcarrierOffset	INTEGER (0…15)		4
dmrs-TypeA-Position	{pos2, pos3}		1
pdcch-ConfigSIB1	controlResourceSetZero	0…15	4
	searchSpaceZero	0…15	4
cellBarred	{barred, notBarred}		1
intraFreqReselection	{allowed, notAllowed}		1
Spare	BIT STRING		1
	Total number of bits		23

Figure 16: The structure of SSB and MIB.

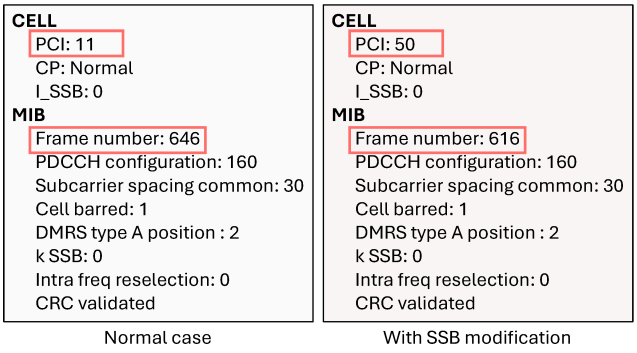


Figure 17: The cell configuration sniffed by our USRP sniffer.

the `intraFreqReselection` bit signifies if intra-frequency cell selection is allowed in the presence of cell barring.

Any failure in the receipt or decoding of SSB elements will render UEs unable to attach to the cell. As mentioned in §3.3, SSB belongs to the broadcast messages that lack protection from PDCP, so adversaries can tamper with these signals to disrupt the initial access procedures. Various attacks are conceivable. A naïve approach is to modify the I/Q samples containing the SSBs with meaningless content to obscure the normal ones, similar to attack `A3`. This disrupts the ability of UEs to decode the block, leading to failures in synchronization and subsequent steps. A more advanced adversary could achieve similar effects by selectively altering specific bits within the SSB. For instance, adversaries can introduce SSBs with incorrect PCIs or erroneous SIB1 configurations. It is particularly potent to manipulate the `cellBarred` parameter, as it enforces a 300-second restriction period before UEs can re-check the MIB. These attacks render new UEs unable to establish attachments with the cell, and they cause already connected UEs to detach from the cell.

Validation. We implemented one type of SSB modification in which the adversary replaces the I/Q samples carrying SSB packets with an incorrect PCI. This manipulation disrupts the attachment of new UEs to the cell and causes existing UEs to detach from the cell due to the PCI mismatch. For verification, we employed a USRP device with an open-sourced 5G stack [21] to capture and decode the cell's SSB signals under normal and attack conditions. Figure 17 presents the decoded results. As shown, the modified SSB led to the USRP decoding of the wrong PCI. Consequently, neither the phone nor Raspberry Pi UEs could attach. We also checked the QXDM logs on the UEs, confirming that the UEs decoded the spoofed PCI.

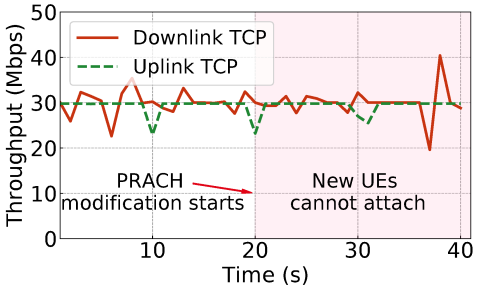


Figure 18: PRACH modification disrupts cell attachment for new UEs but keeps previously attached UEs unaffected.

6.3 A5: Uplink PRACH Modification

The uplink PRACH also plays a vital role in 5G's initial access procedure. It enables UEs to establish synchronization with the network, initiate connection requests, and efficiently identify themselves to the base station. The precise periodicity and location of PRACH data transmissions can be determined from the information captured in the SIB1 data block. UEs can begin uplink synchronization with the cell by transmitting PRACH preambles at pre-specified times, allowing the cell to detect their presence, estimate timing and frequency synchronization, and assign initial identifiers. Upon successful preamble reception, the cell responds with uplink grants, specifying resources for further communication. This procedure ensures seamless and efficient access to the 5G network, facilitating the establishment of robust connections for data transfer and communication. As a result, by manipulating uplink PRACH packets, adversaries can prevent new UEs from attaching to the cell while permitting already connected UEs to continue using the cell without interruption.

Validation. We validated the attack using a phone UE and a Raspberry Pi UE. Prior to the attack, we attached the Raspberry Pi UE to the cell and initiated a TCP/UDP connection using `iperf`. Subsequently, we started the attack by corrupting PRACH packets using capability `C4`, and then attempted to attach the phone UE to the cell. The results of our experiment are presented in Figure 18. As shown, the `iperf` throughput of the Raspberry Pi UE remained unaffected during the attack, while the phone UE failed to attach to the cell. We further examined the QXDM log on the phone UE and found that it sent out the attachment request but failed to get a response

because the adversary corrupted its PRACH symbols.

6.4 Amplifying Impact via Multi-Cell Attacks

According to the 5G O-RAN deployment principles [54], an O-RAN edge data center typically deploys a cluster of DU servers to serve a group of neighboring cells, improving energy- and cost-efficiency. As a result, FRONTSTRIKE has the advantage of amplifying the attack impact by attacking multiple cells served by the same DU together or employing multiple rogue machines to attack multiple DUs and their associated cells. This could allow FRONTSTRIKE adversaries to disrupt cellular network services for a large geographical region, such as an entire campus or small town. Moreover, adversaries could also attack multiple cells by employing different attack vectors, *e.g.,* targeting some cells using SSB modification while using PRACH modification for others. This would expand the impact area and increase the complexity of diagnosing and mitigating the attacks.

Validation. We validated the feasibility of multi-cell attacks by deploying four cells across two floors within our building, with two cells per floor and two UEs per cell. Figure 8c illustrates the positioning of the cells and UEs on one floor, and the location of FRONTSTRIKE adversaries is shown in Figure 7. We comprehensively tested FRONTSTRIKE attacks (A3-A5) in this setup, confirming they exhibit the same effectiveness, but with their impact area extending to all four cells. In addition, we conducted testing of targeted attacks A3-A5 to subsets of the cells and verified that only the UEs of the cells under attack were affected, while the rest of the cells operated normally.

7 Countermeasures

As discussed at the outset, the analysis and attacks presented in §3–§6 demonstrate that MITM fronthaul attacks are practical, require low levels of sophistication, and can introduce severe availability issues, impacting higher layers of the RAN and large regions. In this section, we discuss countermeasures that could address and combat the demonstrated attacks.

7.1 Protection via MACsec

The most obvious and effective way of securing the fronthaul network from our proposed attacks is to make integrity protection of the fronthaul packets mandatory in the standards. To this end, Media Access Control Security (MACsec) [1] is a widely adopted security protocol that provides frame data integrity and data origin authenticity at the data link layer. MACsec computes an Integrity Check Value (ICV) and attaches it to each packet, enabling devices to identify and discard modified packets with incorrect ICVs.

As explained in §1, the standardization bodies have raised several concerns regarding the implications of MACsec performance, which has led to the decision to make integrity protection an optional feature. Motivated by this claim, we performed benchmarks for MACsec encoding/decoding on jumbo-sized Ethernet frames that resemble the fronthaul packets of the 5G RAN. For our benchmarks, we considered two options: (i) basic software-based encoding/decoding on an Intel Xeon 6338N CPU, and (ii) encoding after enabling Advanced Encryption Standard Instructions (AES-NI), which accelerate the AES algorithm (*i.e.,* AES128-GCM) execution on Intel CPUs [58].

Our results show that the MACsec computation can take up to 80 µs on average for the basic software-based encoding/decoding, making integrity protection prohibitively costly for the fronthaul traffic, since it cannot meet the real-time requirements of the 5G standards, as discussed in §2.2.2. However, in the case of AES-NI, the MACsec computation time drops down to approximately 2.4 µs, which, according to the experiments we performed in our testbed, is acceptable.

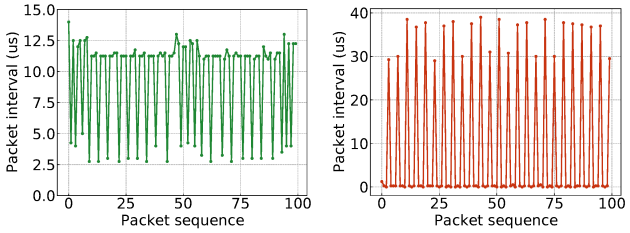
A more effective way to overcome the aforementioned performance concerns with even lower overhead could be to add MACsec protection to selected parts of the fronthaul packets, which would be enough to mitigate the most critical attacks discussed in §5, but with much lower overhead, which would be important for larger scale scenarios (*e.g.,* Massive MIMO cells with 32 or 64 antennas). For example, by adding MACsec protection to the parts of the packets that are not integrity protected by the higher layers (*e.g.,* eCPRI headers, the MIB/SIB and the radio resources carrying signal quality measurements), one could mitigate the most severe attacks, like A1 and A2, by integrity-protecting a few tens of bytes per fronthaul packet. According to our measurements, this can reduce the latency overhead per integrity-protection operation to less than 0.3 µs on average, which makes integrity protection practical for all scenarios of interest.

Furthermore, MACsec provides two protection modes: integrity only, and integrity with confidentiality. Our measurements were conducted using the latter mode to assess the maximum overhead. Using the integrity-only mode could further reduce computation time. These observations align with recent works [23, 37], which discuss the potential use of MACsec to protect fronthaul.

7.2 Real-time RAN Anomaly Detection

While MACsec is an effective proactive countermeasure, mandating its inclusion in the standards and upgrading O-RAN's software and hardware for its implementation will take time. In the meantime, more focus can be placed on reactive countermeasures (*i.e.,* detection and mitigation of MITM attacks).

With the recent advances in RAN telemetry tools, real-time RAN anomaly detection has become a compelling, cost-efficient alternative [41, 42, 66]. For instance, the state-of-



(a) Uplink (b) Downlink

Figure 19: The inter-packet delay pattern of U-plane traffic.

the-art RAN telemetry tool, Janus [36], provides a fully programmable monitoring and control system for 5G RANs. It allows inserting userspace eBPF programs at specified hookpoints in the RAN functions, providing a safe way to collect user-defined telemetry. Utilizing this approach, MITM attacks could be identified with continuous monitoring of real-time RAN metrics, and subsequent detection of anomalies in these measurements. Once suspicious behavior is detected, alerts could be raised and manual or automated remediation actions could be taken (*e.g.,* block all the traffic to/from the culprit RU, dispatch support to the DU site for visual inspection). Below, we describe a few representative examples of anomaly detection methods that could be employed for this purpose:

Deviations in fronthaul traffic characteristics. C-plane and U-plane packets present highly predictable traffic patterns, in terms of packet inter-arrival delays. As discussed in §2.2.2, this predictability is due to the way in which the symbols are generated in alignment to the fronthaul specifications. This is illustrated in Figure 19, for U-plane fronthaul traffic, using measurements we collected from our testbed deployment. As shown, the inter-packet delay between fronthaul U-plane packets is predictable. In the setup under study, and in the case of the uplink traffic, the inter-packet delay always ranges between 1 and 14 µs. Similarly, in the downlink, the traffic also follows a pattern, where there is a higher inter-packet delay for every fourth packet arrival. Any significant changes introduced in these patterns could be identified, by monitoring the traffic at the RU and DU side, and could raise alerts. Note that these patterns can vary for different deployments, depending on the RU and cell configuration (*e.g.,* TDD pattern, cell bandwidth, *etc.*). However, the patterns of each deployment could be easily learned through simple statistical analysis.

Correlation with higher-layer KPIs. In several cases, the aforementioned deviations could be attributed to benign root causes unrelated to security issues, such as faulty NICs, network contention with other traffic sources, or CPU contention at the cores generating the packets. By correlating fronthaul traffic deviations with higher-layer Key Performance Indicators (KPIs), one could increase the confidence in the detection of security-related anomalies. For instance, FRONTSTORM attacks (*i.e.,* `A1` and `A2`) can be detected more accurately by correlating the abnormal increase of signaling messages in the system with spikes in the inter-packet delays of the fronthaul traffic. Similarly, the payload corruption attack (`A3`) can be detected by correlating the deviations of the fronthaul traffic with significant changes of CQI and BLER (Figure 15).

8 Discussion

Additional attacks. This paper focuses on highlighting representative attacks, instead of an exhaustive list of all possible attacks. Thus, while we presented examples of SSB and PRACH modification attacks, alternative attacks via tampering of other messages (*e.g.,* RLC and MAC control elements, like in [67]) can also be conducted. We leave this as future work. Additionally, note that more straightforward attacks, like selective packet dropping DoS attacks, are also possible in our setting. However, for this study, we chose to limit our scope to attacks specifically related to integrity protection.

Vendor response. We shared our report with telco vendors, who have acknowledged the feasibility of our attacks and felt that our work could put pressure on the standards bodies to take action. Some vendors also shared concerns regarding the feasibility of making integrity protection mandatory in the standards, due to the additional compute overhead and its associated cost, which can be prohibitive (*e.g.,* for smaller RU vendors, enterprise 5G deployments, *etc.*). We believe that these concerns can be alleviated to a large extent with the countermeasures that we propose (*i.e.,* selective integrity protection and real-time RAN anomaly detection).

9 Related Work

Physical signal injection attacks. Traditional attacks against cellular networks required direct injection of physical wireless signals [31, 43, 46, 48, 49, 59–61, 63, 64, 67, 71] to impact performance, reliability, availability, and even privacy. These can be accomplished by either setting up a fraudulent base station to attract victim UEs to connect [43, 46, 59, 63, 64] or utilizing capable devices to inject spoofed radio signals [31, 49, 71]. These existing attacks on 4G and traditional 5G suffer from several limitations: they are not stealthy (requiring high-power attack signals), have high overhead (demanding specialized devices and a physical radio presence, which are costly in both compute and power), and have limited scalability (targeting only a single cell). In contrast, our attacks execute remote fronthaul packet modification entirely in software on compromised machines, with no extra overhead to the adversary.

RAN security landscape. 3GPP standardizes general security requirements for 5G architectures [33], but these are not specific to O-RAN architectures and fronthaul. Government agencies such as NIST, FCC, ENISA, NTIA, ITU, and the DoD also provide security guidelines and reports. Similarly, however, these groups put out security specifications for generic

3GPP 5G architectures and interfaces. They do not discuss the fronthaul protocol specifically. The O-RAN Security Working Group is actively defining the security requirements [39] for O-RAN, including the fronthaul. However, the current security mechanisms are insufficient to defend against our attacks. Meanwhile, several recent works, including those of major vendors, have outlined the security landscape of the fronthaul [7, 16, 17, 22, 23, 66], and have theorized the possibility of attacks exploiting the lack of fronthaul integrity protection. However, these works have remained at a high level and have not studied the details, implications, and severity of any potential attack. Our work is the first to exploit concrete vulnerabilities, present practical high-impact attacks, and propose specific countermeasures for O-RAN fronthaul.

10 Conclusion

The fronthaul network of modern 5G RANs suffers from insufficient protection of critical messages. In this work, for the first time, we study the vulnerabilities of the lack of mandatory integrity protection and present two classes of attacks (FRONTSTORM and FRONTSTRIKE). Our attacks can be launched remotely from software, do not require a physical radio presence, and can impact vast regions. We evaluate our attacks using a commercial-grade 5G O-RAN testbed, showing that our attacks can significantly degrade the network performance or cause denial of service to UEs with negligible latency added to the fronthaul traffic. We shared our results and recommendations for mandatory integrity protection with the relevant vendors and security standardization bodies, encouraging a reassessment of the criticality of fronthaul integrity protection. We believe that this work is the first step in the emerging and important, but as yet fully unexplored, space of modern RAN security.

Acknowledgments

We thank our anonymous reviewers and shepherd for their insightful feedback. Additionally, we thank Jennifer Rexford, Ang Chen, Stefan Saroiu, and Bozidar Radunovic for their valuable comments on the earlier draft. This work was supported in part by NSF grants CNS-2214272, CNS-1942219, ITE-2226339, GRFP DGE-2039656, the Open Networks Programme within the UK Department for Science, Innovation and Technology, and a Google PhD Fellowship.

References

[1] IEEE Standard for Local and Metropolitan Area Networks-Media Access Control (MAC) Security. *IEEE Std 802.1AE-2018 (Revision of IEEE Std 802.1AE-2006)*, pages 1–239, 2018.

[2] IEEE Standard for Radio over Ethernet Encapsulations and Mappings. *IEEE Std 1914.3-2018*, pages 1–77, 2018.

[3] IEEE Standard for Local and Metropolitan Area Networks–Port-Based Network Access Control. *IEEE Std 802.1X-2020 (Revision of IEEE Std 802.1X-2010 Incorporating IEEE Std 802.1Xbx-2014 and IEEE Std 802.1Xck-2018)*, pages 1–289, 2020.

[4] Potential Threat Vectors tO 5G Infrastructure. https://www.cisa.gov/sites/default/files/publications/potential-threat-vectors-5G-infrastructure_508_v2_0%2520%25281%2529.pdf, 2021.

[5] GoWin R86S Pro. Accessed 2023. https://www.servethehome.com/this-gowin-r86s-pro-is-an-everything-revolution-with-25gbe-and-2-5gbe/.

[6] 3GPP. About 3GPP. https://www.3gpp.org/about-3gpp, 2023.

[7] Aly Sabri Abdalla and Vuk Marojevic. End-to-End O-RAN Security Architecture, Threat Surface, Coverage, and the Case of the Open Fronthaul. *arXiv preprint arXiv:2304.05513*, 2023.

[8] ETSI & O-RAN ALLIANCE. ETSI RELEASES FIRST O-RAN SPECIFICATION . https://www.etsi.org/newsroom/press-releases/2120-2022-09-etsi-releases-first-o-ran-specification, 2022.

[9] O-RAN Alliance. O-RAN: Towards an Open and Smart RAN. https://assets-global.website-files.com/60b4ffd4ca081979751b5ed2/60e5afb502810a0947b3b9d0_O-RAN%2BWP%2BFInal%2B181017.pdf, October 2018.

[10] O-RAN Alliance. O-RAN Alliance Who We Are. https://www.o-ran.org/who-we-are, 2023.

[11] O-RAN Alliance. O-RAN WG11 - Security Work Group. https://public.o-ran.org/display/SFG/Introduction, 2023.

[12] O-RAN Alliance. The O-RAN ALLIANCE Security Work Group Continues Defining O-RAN Security Solutions. https://www.o-ran.org/blog/the-o-ran-alliance-security-work-group-continues-defining-o-ran-security-solutions, 2023.

[13] OpenAirInterface Alliance. OpenAirInterface5G. https://gitlab.eurecom.fr/oai/openairinterface5g, December 2023.

[14] Paramvir Bahl, Matthew Balkwill, Xenofon Foukas, Anuj Kalia, Daehyeok Kim, Manikanta Kotaru, Zhihua Lai, Sanjeev Mehrotra, Bozidar Radunovic, Stefan Saroiu, et al. Accelerating Open RAN Research Through an Enterprise-scale 5G Testbed. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, pages 1–3, 2023.

[15] Stephan Broszio. First Commercial Deployment of Open RAN with Multiple Partners . https://www.telekom.com/en/media/media-information/archive/first-commercial-open-ran-in-2023-1027618, February 2023.

[16] Joo Yeon Cho and Andrew Sergeev. Secure Open Fronthaul Interface for 5G Networks. In *Proceedings of the 16th International Conference on Availability, Reliability and Security*, 2021.

[17] Joo Yeon Cho, Andrew Sergeev, and Jim Zou. Securing Ethernet-Based Optical Fronthaul for 5G Network. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, 2019.

[18] CPRI Cooperation. Common Public Radio Interface: eCPRI Interface Specification V2.0. May 2019.

[19] Cybersecurity and Infrastructure Security Agency. Potential Threat Vectors to 5G Infrastructure. https://media.defense.gov/2021/May/10/2002637751/-1/-1/0/POTENTIAL%20THREAT%20VECTORS%20TO%205G%20INFRASTRUCTURE.PDF, October 2023.

[20] Dano. The Time I Visited a Dish 5G Cell Site. https://www.lightreading.com/the-edge-network/the-time-i-visited-a-dish-5g-cell-site, 2022.

[21] Aymeric de JAVEL. free5GRAN. https://github.com/free5G/free5GRAN#usrp-configuration, October 2023.

[22] Daniel Dik and Michael Stübert Berger. Transport Security Considerations for the Open-RAN Fronthaul. In *2021 IEEE 4th 5G World Forum (5GWF)*. IEEE, 2021.

[23] Daniel Dik and Michael Stübert Berger. Open-RAN Fronthaul Transport Security Architecture and Implementation. *IEEE Access*, 2023.

[24] Alva Duckwall. A Bridge Too Far: Defeating Wired 802.1x with a Transparent Bridge Using Linux. https://av.tib.eu/media/40535, 2013.

[25] Ericsson. Analyzing IoT Device Performance. https://www.ericsson.com/en/reports-and-papers/mobility-report/articles/analyzing-iot-device-performance, 2016.

[26] Ericsson. How to Deliver a Secure and Resilient 5G RAN? https://www.ericsson.com/en/blog/2021/4/delivering-a-secure-and-resilient-5g-radio-access-network, 2021.

[27] Ericsson. Packet Fronthaul—Design Choices Towards Versatile RAN Deployment. https://www.ericsson.com/en/reports-and-papers/white-papers/packet-fronthaul-design-choices, 2021.

[28] Ericsson. ZTA—Future of Open RAN Development and Security. https://www.ericsson.com/en/blog/2023/12/zta-future-of-open-ran-security, 2023.

[29] Ericsson. AT&T to Accelerate Open and Interoperable Radio Access Networks (RAN) in the United States through New Collaboration with Ericsson. https://www.ericsson.com/en/press-releases/2023/12/att-to-accelerate-open-and-interoperable-radio-access-networks-ran-in-the-united-states-through-new-collaboration-with-ericsson, December 2023.

[30] Ericsson. Ericsson and O2 Telefónica Begin Open RAN Journey with First Cloud RAN Deployment in Europe . https://www.ericsson.com/en/press-releases/3/2024/ericsson-and-o2-telefonica-begin-open-ran-journey-with-first-cloud-ran-deployment-in-europe, February 2024.

[31] Simon Erni, Martin Kotuliak, Patrick Leu, Marc Roeschlin, and Srdjan Capkun. AdaptOver: Adaptive Overshadowing Attacks in Cellular NAetworks. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, 2022.

[32] ETSI. 5G; NR; Radio Resource Control (RRC); Protocol Specification (3GPP TS 38.331 version 15.6.0 Release 15). 2019.

[33] ETSI. 5G; Security Architecture and Procedures for 5G System. 2022.

[34] Peter Fetterolf. The Economic Benefits of Open RAN Technology (ACG Research). https://www.delltechnologies.com/asset/en-us/solutions/service-provider-solutions/industry-market/acg-the-economic-benefits-of-open-ran-technology.pdf, 2021.

[35] Dash Industry Forum. Low Latency Streaming Powered by DASH.js. https://reference.dashif.org/dash.js/latest/samples/low-latency/testplayer/testplayer.html, October 2023.

[36] Xenofon Foukas, Bozidar Radunovic, Matthew Balkwill, and Zhihua Lai. Taking 5G RAN Analytics and Control to a New Level. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, 2023.

[37] Joshua Groen, Salvatore D'Oro, Utku Demir, Leonardo Bonati, Davide Villa, Michele Polese, Tommaso Melodia, and Kaushik Chowdhury. Securing O-RAN Open Interfaces. *IEEE Transactions on Mobile Computing*, pages 1–13, 2024.

[38] O-RAN Working Group 11 (Security Working Group). O-RAN Security Threat Modeling and Remediation Analysis. June 2023.

[39] O-RAN Working Group 11 (Security Working Group). Security Requirements Specifications. June 2023.

[40] GURUCUL. 2023 Insider Threat Report. https://gurucul.com/2023-insider-threat-report, 2013.

[41] Marcin Hoffmann, Salim Janji, Adam Samorzewski, Łukasz Kułacz, Cezary Adamczyk, Marcin Dryjański, Pawel Kryszkiewicz, Adrian Kliks, and Hanna Bogucka. Open RAN xApps Design and Evaluation: Lessons Learnt and Identified Challenges. *IEEE Journal on Selected Areas in Communications*, 2023.

[42] Marcin Hoffmann and Pawel Kryszkiewicz. Signaling Storm Detection in IIoT Network Based on the Open RAN Architecture. In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1–2. IEEE, 2023.

[43] Syed Hussain, Omar Chowdhury, Shagufta Mehnaz, and Elisa Bertino. LTEInspector: A Systematic Approach for Adversarial Testing of 4G LTE. 2018.

[44] Sachin Katti. O-RAN Alliance Who We Are. https://www.intel.com/content/www/us/en/newsroom/opinion/future-ran-virtualized-open.html#gs.70iw18, 2023.

[45] Muhammad Qasim Khan. Signaling Storm Problems in 3GPP Mobile Broadband Networks, Causes and Possible Solutions: A Review. In *2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*, 2018.

[46] Hongil Kim, Jiho Lee, Eunkyu Lee, and Yongdae Kim. Touching the Untouchables: Dynamic Security Analysis of the LTE Control Plane. In *2019 IEEE Symposium on Security and Privacy (SP)*, 2019.

[47] Junehee Lee. Breaking the Mold: Samsung and DISH Wireless Rewrite the 5G Network Deployment Handbook . https://www.samsung.com/global/business/networks/insights/blog/0616-breaking-the-mold-samsung-and-dish-wireless-rewrite-the-5g-network-deployment-handbook/#:~:text=Since%20Samsung%20and%20DISH%20Wireless,private%20and%20public%20cloud%20platforms., June 2023.

[48] Marc Lichtman, Raghunandan Rao, Vuk Marojevic, Jeffrey Reed, and Roger Piqueras Jover. 5G NR Jamming, Spoofing, and Sniffing: Threat Assessment and Mitigation. In *2018 IEEE international conference on communications workshops (ICC Workshops)*. IEEE, 2018.

[49] Norbert Ludant and Guevara Noubir. SigUnder: A Atealthy 5G Low Power Attack and Defenses. In *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2021.

[50] Mavenir. DISH Wireless Expands Cloud-Native Open RAN Network With Mavenir Open vRAN Software Solutions. https://www.mavenir.com/press-releases/dish-wireless-expands-cloud-native-open-ran-network-with-mavenir-open-vran-software-solutions/, February 2023.

[51] Mavenir. World's First 5G SA Network Using Open vRAN On a Public Cloud. https://www.mavenir.com/case-studies/mavenir-and-dish/, June 2023.

[52] Esteban Municio, Gines Garcia-Aviles, Andres Garcia-Saavedra, and Xavier Costa-Pérez. O-RAN: Analysis of Latency-Critical Interfaces and Overview of Time Sensitive Networking Solutions. *IEEE Communications Standards Magazine*, 7(3):82–89, 2023.

[53] O-RAN Working Group 4 (Open Fronthaul Interfaces WG). Control, User and Synchronization Plane Specification. June 2023.

[54] O-RAN Working Group 6 (Cloudification and Orchestration). Cloud Architecture and Deployment Scenarios for O-RAN Virtualized RAN v04.00. October 2023.

[55] Emeka Obiodu, CC Chong, and Mark Noda. Enabling the World's First GPU-Accelerated 5G Open RAN for NTT DOCOMO with NVIDIA Aerial. https://developer.nvidia.com/blog/enabling-the-worlds-first-gpu-accelerated-5g-open-ran-for-ntt-docomo-with-nvidia-aerial/, September 2023.

[56] Pablo Fernández Pérez, Claudio Fiandrino, and Joerg Widmer. Characterizing and Modeling Mobile Networks User Traffic at Millisecond Level. In *Proceedings of the 17th ACM Workshop on Wireless Network Testbeds, Experimental evaluation & Characterization*, pages 64–71, 2023.

[57] Rakuten. Rakuten Symphony's Open RAN Advantage. https://symphony.rakuten.com/open-ran, 2023.

[58] Jeffrey Keith Rott. Intel Advanced Encryption Standard Instructions (AES-NI). Accessed 2023. https://www.intel.com/content/www/us/en/developer/articles/technical/advanced-encryption-standard-instructions-aes-ni.html.

[59] David Rupprecht, Kai Jansen, and Christina Pöpper. Putting LTE Security Functions to the Test: A Framework to Evaluate Implementation Correctness. In *10th USENIX Workshop on Offensive Technologies (WOOT 16)*. USENIX Association, 2016.

[60] David Rupprecht, Katharina Kohls, Thorsten Holz, and Christina Pöpper. Breaking LTE on Layer Two. In *2019 IEEE Symposium on Security and Privacy (SP)*, 2019.

[61] David Rupprecht, Katharina Kohls, Thorsten Holz, and Christina Pöpper. IMP4GT: IMPersonation Attacks in 4G NeTworks. In *NDSS*, 2020.

[62] Samsung. Samsung and O2 Telefónica Launch vRAN and Open RAN Network in Germany. https://news.samsung.com/global/o2-telefonica-and-samsung-launch-vran-and-open-ran-network-in-germany, May 2024.

[63] Altaf Shaik, Ravishankar Borgaonkar, N. Asokan, Valtteri Niemi, and Jean-Pierre Seifert. Practical Attacks Against Privacy and Availability in 4G/LTE Mobile Communication Systems. 2016.

[64] Altaf Shaik, Ravishankar Borgaonkar, Shinjo Park, and Jean-Pierre Seifert. On the Impact of Rogue Base Stations in 4G/LTE Self Organizing Networks. 2018.

[65] srsRAN Project. Open Source O-RAN 5G CU/DU Solution from Software Radio Systems (SRS). https://github.com/srsran/srsran_project, October 2023.

[66] Azadeh Tabiban, Hyame Assem Alameddine, Mohammad A Salahuddin, and Raouf Boutaba. Signaling Storm in O-RAN: Challenges and Research Opportunities. *IEEE Communications Magazine*, 2023.

[67] Zhaowei Tan, Boyan Ding, Jinghao Zhao, Yunqi Guo, and Songwu Lu. Data-Plane Signaling in Cellular IoT: Attacks and Defense. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021.

[68] TIP. TIP Exchange. https://exchange.telecominfraproject.com/marketplace, December 2023.

[69] Juan Pedro Tomás. Vodafone UK Starts O-RAN Deployment to Replace Huawei Gear. https://www.rcrwireless.com/20230901/open_ran/vodafone-uk-starts-oran-deployment-replace-huawei-gear, September 2023.

[70] Verizon. Virtualization: What It Is and How It's Shaping Verizon's 5G Network. https://www.verizon.com/about/news/virtualization-positioning-our-5g-network-for-the-future, 2023.

[71] Hojoon Yang, Sangwook Bae, Mincheol Son, Hongil Kim, Song Min Kim, and Yongdae Kim. Hiding in Plain Signal: Physical Signal Overshadowing Attack on LTE. In *28th USENIX Security Symposium (USENIX Security 19)*, 2019.