CellDAM: User-Space, Rootless Detection and Mitigation for 5G Data Plane

Zhaowei Tan, Jinghao Zhao, Boyan Ding, and Songwu Lu, *University of California, Los Angeles*

https://www.usenix.org/conference/nsdi23/presentation/tan

This paper is included in the Proceedings of the 20th USENIX Symposium on Networked Systems Design and Implementation.

April 17–19, 2023 • Boston, MA, USA

978-1-939133-33-5

Open access to the Proceedings of the 20th USENIX Symposium on Networked Systems Design and Implementation is sponsored by

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

CellDAM: User-Space, Rootless Detection and Mitigation for 5G Data Plane

Zhaowei Tan, Jinghao Zhao, Boyan Ding, Songwu Lu
University of California, Los Angeles

Abstract

Despite all deployed security fences in 5G, attacks against its data plane are still feasible. A smart attacker can fabricate data packets or intelligently forge/drop/modify data-plane signaling messages between the 5G infrastructure and the device to inflict damage. In this work, we propose CellDAM, a new solution that is used at the device without any infrastructure upgrades or standard changes. CellDAM exploits the key finding that such data-plane attacks by the adversary would trigger unexpected data signaling operations. It thus detects all known and even currently unreported attacks via verifying data signaling correctness with novel state-dependent model checking. CellDAM could work with or without firmware access at the device using inference on low-level 5G signaling and configurations. It mitigates the damage upon detection by inducing frequency band switches at the device via the existing handover procedure. The prototype and empirical evaluation in our testbed confirm the viability of CellDAM.

1 Introduction

The current 5G and its legacy 4G cellular networks provide anywhere, anytime Internet access for billions of users. Security is an important design goal for 5G systems. Multiple security fences are thus deployed or enhanced [7], e.g., device authentication, enforced data encryption and integrity check. They aim to defend against recent attacks against the control plane [23, 41], as well as the data plane [47, 48, 60].

The current security fences are mostly *proactive protection* on data packets. However, this is insufficient for 5G data plane. Certain categories of attacks still cannot be protected. For example, a smart attacker can selectively drop a few data-plane signaling to incur cascading effect. Moreover, proactive protection is sometimes of high cost, such as protecting low-layer, cleartext, data signaling messages. Furthermore, proactive protection could be turned off with certain attacks [43], or unavailable to legacy devices in developing countries [1, 52].

In this paper, we explore a *reactive* solution approach called CellDAM towards 5G security. Instead of proactively preventing attacks, CellDAM complements the existing efforts by detecting whether a potential attack is underway and mitigating its damage. In addition to identifying the above-mentioned attacks that cannot be handled by proactive solutions, it offers two more benefits. First, the solution needs no standard change or hardware upgrade, thus being immediately deployable. Second, by verifying the correct operations, a reactive solution can find any attacks that do not follow the standard procedure, including both known and unreported ones.

A well-known challenge for data-plane detection is the high data throughput by 5G. It is thus considered impractical to inspect data packets at Gbps on a mobile device without consuming excessive processing or energy resources. CellDAM addresses the issue with a novel approach. We do not check each data packet directly. Instead, we inspect the data-plane signaling messages, which are standardized to facilitate data delivery but incur 1-2 orders of magnitude less overhead compared with monitoring all data packets. Our approach can detect attacks against both signaling messages and data packets. This is because every data delivery must exchange signaling messages for resource grant over the licensed 5G wireless channels. Therefore, *undesired data-plane signaling operations* might be triggered at the device by data-plane attacks.

We "verify what is right" when inspecting data-plane signaling. We thus model signaling operations for 5G data delivery; any operation that deviates from the standardized model is considered as a potential attack. It turns out that, a single-protocol, static checking scheme cannot detect all attacks. We thus devise a novel cross-layer, state-dependent model checking to validate data-plane signaling operations. At each state, we perform context-dependent validation to spot unexpected messages. Our experiments show that we have discovered three unreported attacks in addition to the known ones.

CellDAM is designed to work with various levels of privilege. Note that it requires access to signaling messages for inspection. The messages can be easily obtained with firmware access. However, for a user application, low-level 5G signaling cannot be accessed without root privilege. To overcome

this limitation, CellDAM utilizes SecHub, a separate companion node placed near the device. It uses a new inference technique to capture the signaling messages of interest from/to the protected device, but filter out all others.

Upon detection of an attack, CellDAM mitigates impact by triggering a 5G-standard handover procedure. This switches the device to a new cell. The attacker cannot track the device due to the encrypted handover messages and the dense deployment of 5G cells. Meanwhile, the handover procedure only incurs a disruption that lasts less than 100ms. With firmware access or network assistance, CellDAM could trigger the procedure using standardized commands. If they are unavailable, CellDAM leverages the SecHub to impact the channel measurement results to trigger a handover procedure on the victim device, but does not affect other devices.

We show that, CellDAM offers a practical solution to detecting and mitigating data-plane attacks with high accuracy. We prototype a device-side, user-level solution in C++ and Python based on open-source srsRAN [51]. SecHub implements components to infer the parameters, detect by verifying data-plane signaling, and mitigate with frequency band switching; all components do not require root privilege on the protected device. It achieves the detection accuracy of 0.989∼1.0, recall of 0.705∼1.0, and the F1 score of 0.823∼1.0. It can detect known and new attacks within 28ms on average. For mitigation, the handover can be triggered at 100% success rate with a latency of 1.85s on average. This procedure only incurs an average of 72.3ms disruption on applications. SecHub can successfully find the protected COTS devices with the accuracy of 97.2%∼100% under various traffic types.

2 Background

2.1 5G Primer

5G Architecture 5G system has 3 major components (Figure 1): User Equipment (UE), Base stations (gNB), and 5G Core network (5GC). The UE is a 5G user device. The gNB powers up the 5G network and provides wireless access in its coverage area for UE. It also forwards data to and from 5GC. 5GC includes the control plane for authentication and mobility support, and the data plane for data packet delivery.

5G Protocol Stack The 5G protocol stack consists of multiple protocols for both control-plane and data-plane functions. Non-access Stratum (NAS) and Radio Resource Control (RRC) are in charge of control-plane signaling. NAS facilitates control-plane signaling message exchange between the UE and 5GC. RRC carries the control messages and data-plane parameters for setup, power management, and handover behavior between the UE and the gNB. The data-plane enables IP packet delivery. We describe 5 protocols involved in data-plane operations: Service Data Adaption Protocol (SDAP), Packet Data Convergence (PDCP), Radio Link Control (RLC), Medium Access Control (MAC), and Physical Layer (PHY). SDAP manages the quality of service for data delivery. PDCP takes charge of encryption and integrity protection for control-plane and data-plane packet. RLC performs data concatenation and reorganization to ensure reliable, in-order data transfer. MAC controls radio access in licensed bands, and PHY performs wireless signal processing.

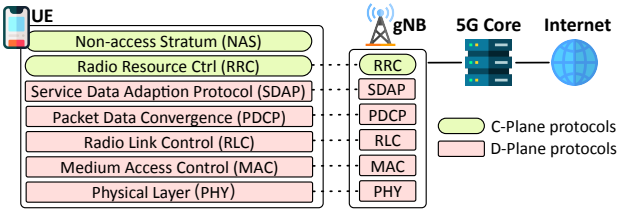


Figure 1: Architecture of 5G and its protocols.

Data-Plane Signaling In addition to RRC and NAS signaling for the control plane operations, 5G data-plane also has signaling between UE and gNB to facilitate packet delivery. The data-plane signaling exhibits in multiple forms. Various flags at the MAC layer (grant assignment DCI, scheduling request, HARQ acknowledgements, etc.), MAC control elements [3] that convey power control, time alignment or buffer status, and RLC control [5] for reliable transfer are all instances of data-plane signaling.

2.2 Protecting Data-Plane in 5G

Mutual Authentication Mutual authentication is a critical security measure in 5G, inherited from 4G with little change. The UE and network perform a secure Authentication and Key Agreement (AKA) procedure during connection for authentication and session key set-up. The session key is used to generate keystream for every packet with several parameters such as sequence number.

Protection on Data Packets The keystream is generated to encrypt data packets without key reuse for both control-plane and data-plane packets [7]. The sender also updates another keystream to generate an integrity code attached to the message for integrity check at the receiver. While integrity protection for control messages is enforced in 4G, it was optional for data plane due to high overhead. The vulnerability allows attacks that fabricate data packets [48, 49]. 5G aims to enforce integrity protection on all data packets. Although its usage is still optional, both 5G UE and gNB should support integrity protection at the full speed starting from release 16 [7]. With the increasing capacity of the hardware, it is expected such integrity protection will be mandatory.

3 A Case for Detection and Mitigation for 5G Data-Plane Attacks

3.1 Threat Model

In this paper, we consider an adversary who seeks to incur various damages on the target victim 5G device on its *data*

plane. The attacker has the following capabilities: 1) Connect to the same cell as the victim device, which is feasible with fingerprinting [32] or social engineering [36]; 2) Eavesdrop on, transmit data, or send noises on physical channels; 3) The adversary may exploit fake base station (FBS) [48, 49] or overshadowing attacks [60]. Although FBS is mitigated in 5G [4], it is still possible to launch certain variants of FBS, such as relay FBS as man-in-the-middle. We do not limit the message that can be forged by the adversary, which could be user packets or signaling messages.

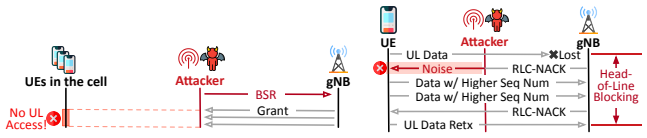
We do *not* consider attacks that threaten control-plane, such as an IMSI catcher. We also do *not* consider an insider attack, where the adversary can steal security keys stored in SIM/networks or even compromise a 5G base station. Protecting such attacks is beyond the scope of this work.

3.2 Proactive Protection is Insufficient

State-of-the-art: Per-message proactive protection The authenticity of 5G user packets is protected by end-to-end (such as TLS [29]) and 5G-specific integrity protection (as introduced in §2.2). However, data-plane attacks are still possible despite the protection.

Issue 1: Not every message could be protected at low cost Unlike end-to-end packets, the small-sized data-plane signaling messages are not protected by proactive solutions [2], due to high overhead or impracticability being in the below-PDCP layer. Fabricating these messages can incur serious damages as shown in recent studies [54]. An attacker only needs to forge a few messages to incur damages consistently. One example attack is shown in Figure 2(a). An attacker forges a Buffer Status Report (BSR) that requests uplink grant from the gNB. The uplink grant in a time period will be assigned in accordance with the request. A malicious BSR could contain a large request, wasting licensed resource and blocking uplink delivery of any UE in the cell for hundreds of milliseconds. The attacker is capable of repeatedly forging such messages, which can continuously drain the resource. We note that, gNB cannot distinguish this attack message from a legitimate one.

Issue 2: Certain attacks cannot be proactively protected The attacker can also intelligently corrupt/drop certain messages to incur serious damages. Such attacks *cannot* be protected by any integrity check. One example attack is shown in Figure 2(b). In this attack, the adversary corrupts an RLC control message NACK, which is used to request retransmission for certain packets. When this NACK is lost, the retransmission is delayed. Due to the RLC mechanism, all subsequent data packets will be blocked. The effect will last until the next RLC message, which could be hundreds of milliseconds based on common RLC configurations. Moreover, the attacker can repeatedly send the message to cause persistent damages. The attacker does not require fake base stations or channel jamming. Instead, a lightweight attacker only needs to send a single signaling every hundreds of milliseconds.



(a) Fabricate data-plane signaling. (b) Selective corruption.

Figure 2: Example data-plane attacks that cannot be protected with current proactive approaches.

Issue 3: Proactive protection could be turned off Forging user data packets is still possible despite the data packet integrity check in 5G. The usage of integrity protection is still negotiation-based [7]. Hence, the attacker can disable integrity protection by leveraging certain vulnerabilities, such as those from firmware [43]. Furthermore, legacy 4G devices or 5G devices on earlier versions, which are still a considerable number [42, 52, 56], do not implement mandatory integrity check [1]. For these devices, known attacks can incur serious damage when end-to-end protection is not used. An attacker can manipulate DNS requests to a malicious Web server [48]. The malicious server could then send any forged content to the victim. Beyond Web, the attacker could forge arbitrary encrypted data through a man-in-the-middle [49].

Insight: Reactive detection and mitigation can complement proactive protection Given all the existing threats which cannot be protected by proactive approaches, we shall also develop proper *reactive* solutions to complement the proactive protection. They should include both detection and mitigation. The detection methods will spot any suspicious activities, while the mitigation will help the victim recover from the damages. They can complement existing solutions, while not requiring any 3GPP framework change.

4 Overview of `CellDAM`

In this section, we present the design goal and challenges of designing a reactive protection.

4.1 Security Goals

Following our insights, we seek to *detect and mitigate* 5G data-plane attacks *without standard changes*. As discussed in §3.2, this solution approach offers an immediate remedy for certain attacks that are not protected by current proactive approaches. We argue that, such detection is essential on the device side. First, it is more scalable compared to network-centric solutions. Second, some attacks are only detectable on the device side. Take the signaling attack (Issue 1) in §3.2 as an example. In this attack, the UE will receive uplink grant that it did not request. However, the network cannot distinguish whether the request is malicious or legitimate. Therefore, device-side detection is required to detect and mitigate the attacks.

Meanwhile, our reactive detection and mitigation scheme should achieve the following goals:

Verify what is right for attack detection We design and implement approaches that verify whether the runtime, data-plane operations follow the correct procedures stipulated by the standards, and treat any undesired behaviors, rather than specific attacks, as suspicious. This ensures the detection of a category of attacks that yield improper data-plane operations. Even if an attack has not been reported yet, it can be detected by our approach as long as it triggers undesired behavior. We admit that certain attacks that adhere to the standardized approach might not be detected by our approach. Our solution prioritizes soundness over completeness.

Detection and mitigation need to be practical We aim to design a reactive method without hardware update or extra privilege. It can thus benefit legacy devices.

•***No infrastructure upgrade*** We will design our solution *on the device side*. Unlike network-centric protection approaches [24, 44, 46, 57], our device-side scheme needs no expensive infrastructure or hardware upgrades. Moreover, as we discussed earlier, certain attacks can only be observed by the device.

•***1-2 orders of magnitude lower overhead compared to verifying each data packet*** It is nontrivial to monitor the 5G data plane. Data throughput in 5G is expected to reach a few Gbps. The traffic volume of data packets is several orders of magnitude higher than the control-plane signaling messages. We aim to design a lightweight security solution that can work under heavy data traffic. The overhead should be 1-2 orders of magnitude smaller compared to monitoring the entire traffic.

Applicable to different defense models The solution should work given different levels of privilege. We consider three defense models in this work: (1) Defense with firmware access; (2) A user-space application that can communicate with the operator; and (3) A user-space application without any privilege. All three models have their own usage scenarios. For (1), a device vendor implements the solution to enhance the security of its 5G devices. For (2), we consider an operator who tries to protect its users with additional security requirements. However, the operator cannot directly access the firmware. For (3), we consider a normal user who tries to protect the device. All three models complement each other under different scenarios to protect 5G data plane.

For the Defense model (1), the solution could be implemented inside the firmware, as it has access to all cellular-specific info and OS-level privilege. For both (2) and (3), mobile OS only provides control-plane basic info [10], such as connection state. The application cannot access device-side cellular-specific info unless extra privilege (e.g., Diag port) is granted for tools like MobileInsight [37] or QXDM [45]. However, such extra privilege (e.g., root access) exposes new vulnerabilities [15, 21, 22, 50] and is unavailable to most devices due to technical or legal concerns [34]. Our solution aims to function despite the limitations.

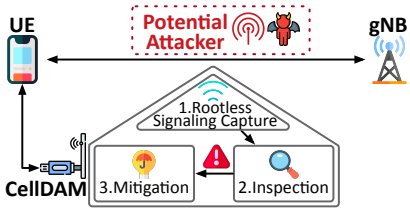


Figure 3: Envisioned procedure of `CellDAM`.

State-of-the-Art: Existing *reactive* approaches cannot achieve these goals To the best of our knowledge, all previous works on 5G/4G attack detection focus on undesired behavior on the control plane [26, 27]. This is insufficient, as recent studies [54, 60] show that an advanced adversary can bypass control plane and directly attack the data plane.

4.2 Solution Overview

We design `CellDAM`, a 5G data-plane inspection scheme without root privilege, as illustrated in Figure 3. It satisfies all design goals. `CellDAM` first captures all data-plane signaling messages from/to the protected UE. It could do so in a separated node called `SecHub` and bypass the requirement of in-device extra privilege. `CellDAM` then inspects the *lightweight* data-plane signaling messages to spot undesired behavior. Finally, `CellDAM` uses `SecHub` and in-device operation to mitigate the attack damage via handover-based cell switching.

Inspecting lightweight signaling messages with state-dependent checking (§5) We first show that, inspecting data-plane signaling offers an effective way to detect data-plane attacks, while it has magnitude lower overhead due to its low volume. We further propose a *cross-layer, state-dependent* model checking for attack detection. If the device spots an incoming signaling message that is undesired in the current state, it detects a potential attack.

Rootless signaling inference (§6) Signaling verification requires access to the messages. For the defense model with firmware access, such privilege is granted. To serve the majority of rootless devices, `CellDAM` incorporates a separate, companion node named `SecHub` for the protected device. The goal is to share a consistent view of the physical channels. The node can continuously capture over-the-air signaling messages by inferring the device ID and traffic pattern. It needs no extra privilege on the protected device.

Device-side reaction without infrastructure upgrade (§7) Once a suspicious operation is detected, we activate the mitigation module that switches to other, potentially attack-free 5G channels. This could be done by leveraging the standardized 5G handover procedure and dense cell deployment. Handover incurs small disruptions in the applications, while being resilient to attacks. `CellDAM` can initiate such a procedure via two approaches. It can directly create a standard-compliant message for handover. It can also leverage `SecHub` to affect the device to trigger a handover, without affecting other user devices.

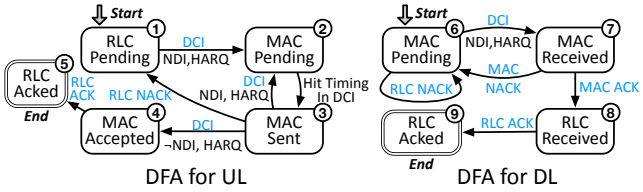


Figure 4: DFA for tracking states with data-plane signaling. See Appendix C for an extended version.

5 Cross-layer, State-dependent Checking on Data Plane Signaling

We now present how CellDAM inspects the data-plane signaling messages to detect undesired behavior. We follow our guideline of "verifying what is right" to model the device-side protocols and detect any undesired behavior. To this end, we devise validation schemes using the standardized data-plane delivery procedure to check each incoming/outgoing data-plane signaling message.

The solution has two highlighted components. First, instead of verifying data packets of large throughput, CellDAM inspects lightweight, yet critical data-plane signaling messages for attack detection. Second, we present a state-dependent model checking method to find suspicious behaviors.

5.1 Monitor and Inspect Data Plane Signaling

A straightforward solution can verify each data packet. This can be done by applying Deep Packet Inspection (DPI) or detecting anomalies in the wireless signal for each data packet. However, they cannot detect all attacks, such as the attacks that target signaling messages. Meanwhile, this will incur huge overhead, especially given the large 5G throughput.

Why inspect data-plane signaling We show that, monitoring data-plane signaling offers an effective alternative method for detecting data-plane attacks, including attacks on both data-plane signaling and data packets. First, delivering each data has a standardized sequence of signaling messages due to 5G infrastructure-controlled data access. For any type of attack, the attacker needs to tamper with the signaling messages. Therefore, this solution approach will cover a wide range of attacks. Second, the frequency of data-plane signaling is smaller than data packets due to 5G aggregation scheduling.

Data-plane attacks might manipulate data-plane signaling in standardized data delivery For data delivery, each device must follow a standardized approach, as 5G uses gNB to mandate the radio resources. Therefore, we can use the signaling messages to model the state of each packet delivery. To model and track the state, we use Deterministic Finite Automata (DFA), a common technique for state tracking in attack detection [19, 26]. We study 3GPP standards across all 5G-specific data-plane sublayers and manually create DFA based on *mandated, standardized data delivery procedure*. We form cross-layer DFA for each RLC data packet with its necessary state transition at the MAC layer. We do not include PDCP, as it does not maintain state or buffer packets.

The constructed DFA is shown in Figure 4. For uplink, data transmission follows a scheduling-based feedback loop. The device first sends requests (Buffer Status Reports or BSR) to ask for resource grants until the packet is delivered by MAC. The MAC fast retransmission is notified by a new DCI with the same HARQ ID and NDI. The packet is considered delivered when the RLC ACK is received. For downlink, the data transmission follows the same procedure but without the request-grant loop, as the gNB initiates the transmission. The device sends the MAC feedback of ACK/NACK in PUCCH.

Inspecting data-plane signaling is lightweight Compared with data-plane packets, inspecting data-plane signaling is of much lower overhead. First, the size of each signaling message is much smaller than the actual data packet. The signaling messages (such as DCI, RLC Control) are at most several bytes long. Some PHY messages are merely 1-bit indicators. Second, 5G data delivery will transmit multiple IP data packets in a single data block (aggregated and segmented by the 5G RLC protocol). Therefore, for signaling messages that facilitate data delivery (e.g., DCI), only one such message is needed for the large block.

We validate the hypothesis that control traffic is significantly lower than data traffic by comparing their traffic volume. We show results from operational traces in a commercial network and in our SDR-based testbed. Our testbed runs standard-compliant srsRAN 5G [51] and the details will be shown in §10. Since our testbed does not support features such as MIMO or carrier aggregation for higher throughput, we also collect traces from commercial operators. As the current open-source 5G decoding tools (e.g., MobileInsight 5.0 [37, 38]) have not supported 5G data plane, we collect and analyze 4G data plane as a reference, considering that data-plane signaling design remains largely unchanged in the current 5G NSA [3,5][1]. As shown in Figure 5, the processing of data-plane signaling is 1∼2 orders of magnitude lower than that of data packets.

Therefore, detecting attacks with data-plane signaling is of much smaller overhead than monitoring the entire data-plane packets. Prior work [20] has already shown an SDR-based system is capable of monitoring DCI messages for *all devices in a cell*. It is thus feasible for CellDAM to capture all data-plane signaling *for a single device* while performing inspections in real-time (detailed in §9). This is important considering 5G's high data rate.

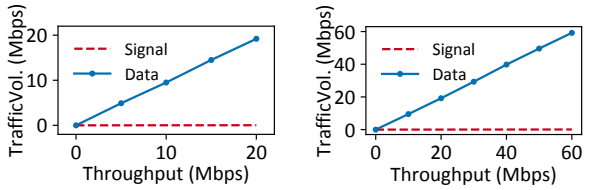
5.2 Cross-layer, State-Dependent Checking

Stateless checking cannot detect certain attacks The detector could perform certain checks within each protocol

[1] One major difference is that 5G cancels PHICH which is used for uplink data retransmission. Therefore, we do not include it for calculation.

Check	State	Message	Validation Details
c_1	All	Any Message	The data-plane signaling shall be in the accepted list for each state. (Appendix C)
c_2	s_3, s_6	RLC NACK	It shall not be received after RLC timer and MAC retransmission timer expire or after receiving an RLC ACK with higher sequence number.
c_3	s_4, s_8	RLC ACK	An RLC ACK shall not be received before the packet is acknowledged in MAC.
c_4	s_7	MAC ACK/NACK	The ACK/NACK in PUCCH should be delivered at correct timing after DCI; If not, this is an indicator that the previous grant/data is received during DRX OFF.
c_5	s_1	DCI for UL Grant	There should not be large "free" grant when no request is sent or no data in buffer.

Table 1: Validation checks performed by CellDAM based on state and message.



(a) Under Testbed (5G) (b) Under Commercial (LTE)

Figure 5: The comparison of traffic volume per second for data packets vs. data-plane signaling messages.

regarding whether the incoming signaling message conforms to the standards. However, this solution will fail to spot certain attacks against 5G data plane.

We showcase a concrete example. When an RLC signaling message ACK is received, it is possible if MAC layer has accepted this message. Otherwise, this message is impossible and a potential attack is detected. Therefore, the checking must rely on the current state across different protocols.

Cross-layer, state-dependent model checking with DFA Therefore, we propose a state-dependent checking for 5G data plane security. Instead of proposing a few checks statically in each protocol, we leverage the cross-layer DFAs we built for data delivery, whose inputs for transition are *data-plane signaling messages* or their derived events. If the next captured signaling message m passes all validation checks, the DFA moves to a new state; Otherwise, a potential attack is detected and we initiate the mitigation procedure.

Formally, we maintain n deterministic finite state machines $M = \{M_1, M_2, ..., M_n\}$. For each DFA $M_i, i = 1, 2, ..., n$, we denote it as a 5-tuple $(S_i, S_i^0, S_i^1, \Sigma, T_i)$, where S_i is a finite set of states with $S_i^0 \in S_i$ being the initial state and $S_i^1 \subseteq S_i$ being the accepted states, Σ is a finite set of input messages[2], $T_i : S_i \times \Sigma \rightarrow S_i$ is a transition function mapping the pairs of a state and a received message to the next state.

We build validation checks $V = \{V_1, V_2, ..., V_k\}$. Each V_i is associated with a DFA M_j, a state $S \in S_j$, and a message $m \in \Sigma_j$. Every time the DFA M_j with state S inspects a new message m, CellDAM runs the corresponding check(s). They map the signaling message m to 0 (fail) or 1 (succeed) given the current context, which is derived from past records or other DFAs. A potential attack is identified if one of the validation checks fails; Otherwise, M_j accepts message m and updates its state accordingly.

[2]To make the problem tractable, we only consider the discussed data-plane signaling messages. We prioritize soundness over completeness.

Based on the state, we perform a few validations on each incoming/outgoing signaling message. We show the list of validations in Table 1. First, all states will have a list of accepted messages. CellDAM checks whether the next message is in the list. The detailed list for this check is shown in Appendix C. For c_2 to c_3, we are checking whether the RLC operations are consistent with the MAC layer for both uplink and downlink. For example, upon receiving RLC NACK, we check whether an early RLC ACK that has already acknowledged a packet is received. For c_4, we use the indicator of no ACK/NACK to detect a possible forged message received in DRX OFF. For c_5, CellDAM detects abnormal grants from gNB without any request. Note that a gNB can freely grant the device with small grants. However, they are usually 100-200 Bytes long. Any larger grant incurs a waste of resources. Therefore, it could be the outcome of a forged BSR or grant signal. If all checks pass, the DFAs are switched to the new state.

6 Access Signaling Messages for Detection

With CellDAM's checking techniques, an end device can inspect signaling messages for attack detection. We now discuss how to access them under three defense models in §4.

With direct firmware access For defense solution that has firmware access, it could directly capture the signaling messages. In fact, these messages are already processed by firmware to realize the functionalities.

No direct firmware access If CellDAM is deployed on the application layer, it cannot directly access the signaling messages. Although messages are available in tools such as MobileInsight [37] or QXDM [45], using these tools and accessing the low-level messages require root, which exposes new vulnerabilities [15, 21, 22, 50] and is unavailable to most users [31]. Even the application is allowed to communicate with the network (i.e., defense model 2), it cannot access the *device-side* signaling for detection.

Idea: Infer signaling messages without root privilege in a separate node To address the concern, we design and deploy a separate gadget (e.g., extended hardware with wireless capability), SecHub. The node is placed close to the protected device within a meter; it passively receives and decodes the data-plane signaling over the air. It can also communicate with the user-space security manager application at the protected device. The device and SecHub connect each other,

either with wire (a gadget that is attached to the device via USB) or wirelessly (i.e., Bluetooth).

However, it is not trivial to infer such info in the user space with SecHub. We address this issue in the next section.

Challenge: Unknown configurations Although 5G data-plane signaling messages are not encrypted, SecHub must identify which ones belong to the protected devices. Several configurations are needed: (1) The carrier frequencies; (2) The physical cell ID (PCI) that indicates the physical-layer identity of the cell; (3) The Cell Radio Network Temporary Identifier (C-RNTI) that distinguishes the target device from other devices connected to the same cell.

Directly infer the parameters will not work Unfortunately, not all these configurations could be acquired from the protected device without root access. Android provides APIs for applications to obtain the current band and PCI [10]. In contrast, C-RNTI can only be extracted from the victim device with root privilege. Without C-RNTI, SecHub cannot recognize which traffic is for the protected device. Therefore, we need a solution that can recognize which configurations are assigned to the protected device over the air.

Idea: Use high-layer traffic pattern that is visible to SecHub Since the user space has no access to lower-layer C-RNTI, we must identify the configurations with higher layer features. We note that, data traffic pattern is *visible to both device and SecHub*. Therefore, it is an ideal "channel" to insert fingerprint and notify the identity to SecHub.

Henceforth, we generate specific traffic patterns on the target device, with SecHub being aware of the pre-agreed pattern in advance. It can thus observe the channel and identify the target device's C-RNTI by analyzing low-layer signaling. Our approach takes three concrete steps.

Step 1: Traffic Pattern Coordination First, SecHub randomly generates a traffic interval and sends it to the target device through the wired or wireless channel. This interval is used as the fingerprint for the target device. The traffic interval triggers a unique pattern for the data-plane signalings for fast inference. SecHub leverages this shared traffic pattern to recognize the target device in later analysis.

Step 2: Trace Collection Second, the device creates traffic (e.g., small UDP packets) with the acquired interval. The traffic generation is performed by the application and does not require root privilege. gNB will assign grants for the device to deliver data. At the same time, SecHub monitors all the subcarriers in the target cell and tries to decode the C-RNTI from all grants with all possible positions in the band. This is necessary as grants do not always locate on the same subcarrier in the band (for reducing inter-cell interference). SecHub records the decoded C-RNTIs with corresponding time slots for inference.

Step 3: C-RNTI Inference Finally, SecHub aggregates the grants for each C-RNTI decoded from the collected trace. SecHub first ranks the C-RNTIs by grant numbers and filters the top 10% C-RNTIs as the candidates. The time intervals between consecutive grants are calculated and compared with the negotiated interval for all the candidates. The grants for the fingerprinting traffic will show the same interval. Although there may be background traffic from the target device, the grants triggered by the fingerprinting traffic still show the periodic pattern and could be filtered from the device's grant traces. By ranking the ratio of matched intervals with the total interval number, the top C-RNTI will be selected as the target's C-RNTI. To ensure robustness, SecHub performs the procedures twice and checks if the inferred C-RNTI values match. If the candidates do not match, SecHub will perform the inference again until a candidate is selected.

Combining with the frequency and PCI of the device from OS API and the C-RNTI inferred from the collaborated traffic fingerprinting, the SecHub could successfully camp on the cell and capture the downlink/uplink messages. The entire procedure does not require root access at the device.

Tracking configuration change We also note that, the C-RNTI configuration could be updated within an encrypted message in both static and mobility scenarios. CellDAM incorporates a new solution to prevent such change and enable continuous tracking. We detail this solution in Appendix D.

7 Device-Centric Mitigation

Although 5G standard updates could fundamentally defend against forgery attacks, they require months or even years to be deployed in practice. To this end, we design device-centric mitigation to provide a quick remedy for existing devices. We leverage the existing, dense 5G cell deployment to help the victim dodge the attacker.

7.1 "Quick Dodge" with Handover

Band switching to avoid the attacker on a specific cell We observe that, the attacker must camp on the cell that serves the victim device and forge messages in the current band to launch the attack, regardless of the attack methodology (§3.2). Therefore, the victim could quickly escape from attackers by switching to another frequency band (i.e., a different 5G cell).

With the insight, CellDAM thus aims to switch the band (i.e., cell) that serves the victim device to avoid the attacker.

Use handover procedure to trigger band switching In CellDAM, we leverage the 5G handover procedure to realize band switching. In 5G, a UE measures the signal quality by metrics of Reference Signals Received Power (RSRP) and Reference Signal Received Quality (RSRQ). When the experienced signal quality of the serving cell is worse than the thresholds configured by gNB, the UE sends reports to gNB, which makes the handover decision and sends a handover command to the device.

Although rare, it is possible that there are no other cells available. In such cases, CellDAM could opt to generate a warning instead of triggering the handover.

Why is band switching via handover effective against attackers? First, the attacker could not control or know which cell the device is switching to, as the handover command is encrypted. Given that, the attacker needs to enumerate all nearby cells and use fingerprinting on each to find whether the device is in the cell. It takes prolonged time and effort for an attacker. This could take minutes, before which the device might already move to a new cell.

Application is resilient to handover Handover incurs little overhead on applications. UE does not go through the slow cell search or connection setup procedure. Instead, it only has to update its PHY parameters for the new cell. Therefore, the disruption to the applications is minimum.

How to trigger a handover? To initiate a handover procedure, the most straightforward way is by requesting the gNB to send a handover command to the device. This is possible if CellDAM can communicate with the operator (Defense model 2). On the other hand, the device can send a 5G measurement report to the base station. It indicates that the measurement result from another cell is better than the current one by an offset configured by gNB (i.e., a measurement event in 5G). This will subsequently trigger a handover procedure. Defense model 1 could take this approach. However, for Defense model 3, neither approach is available. In the next section, we describe how it could still initiate a handover.

7.2 Trigger Handover with SecHub

When the handover-related messages cannot be directly created, CellDAM takes a different path by affecting the measurement result. If the measured RSRQ on the serving band is bad, a legitimate report will be triggered by the device. Although either bad RSRQ or RSRP can result in handover, we focus on RSRQ, because RSRP is measured based on the reference signal power and is hard to be affected by SecHub.

Solution: Precise reference signal downgrade We design adaptive signal degradation to ensure low-overhead band switching. Instead of the entire channel, SecHub only copes with the reference signal in 5G. The device measures the reference signal to monitor the signal quality regardless of PHY techniques (e.g., MIMO, carrier aggregation, dual connectivity, etc). The reference signal only exists in specific subcarriers and time slots. SecHub calculates the positions of the reference signal based on the current physical band according to the 5G standards [6]. By morphing the reference signals only, SecHub downgrades the victim's measurements of the current frequency band without much overhead.

The solution should not affect other devices. SecHub adaptively controls its power upon triggering the handover. More details on the power control are shown in Appendix B.

8 Security Analysis

8.1 Attacks Covered by CellDAM

In this section, we discuss what attacks can be detected by CellDAM. With our design, CellDAM can detect multiple attacks that target both data plane packets and signaling message, as shown in Table 2. The details for each attack and how CellDAM detects it are elaborated in Appendix A.

Attacks against data packets We consider three attack actions that target data packets: injection, deletion, and manipulation. For injection, the attacker attempts to insert a new data packet. For deletion, the attacker tries to remove a packet from being received by the device or the network. For manipulation, the attacker seeks to change certain bits in a data packet. Any bit in the IP packet (application, transport, or IP headers) could be changed.

We first show that, both injection and manipulation attacks can be detected by CellDAM. We note that, neither attack can be directly launched over the air. Flipping data bits over the air will fail the CRC check, while directly injecting a new packet will fail to be decrypted due to mandatory encryption. Therefore, the possible methodology for injection or manipulation is the Man-in-the-Middle (MitM) approach. The adversary intercepts the packet, flips the bits, re-encodes it, and injects the altered packet.

This could be detected by CellDAM. as a MitM will incur undesired signaling messages. There are two possible ways to launch MitM. For the scheme using relay FBS (A1 in Table 2), an attacker cannot directly learn the critical data-plane configurations, which are transmitted over the *encrypted* RRC messages [60]. Consequently, the forgery could be sent in an impossible context, e.g., in the time slots when the device is in its Discontinuous Reception (DRX) OFF state (i.e., sleep mode). This behavior will be detected by CellDAM. It applies to both uplink and downlink forgery, as the attacker needs to send DCI to the victim device for notification of both uplink and downlink scheduling. For the attack that corrupts the transmission and injects retransmission (A2 in Table 2), it needs to forge DCI and data packets. The next DCI from the attacker could reach the device before the acknowledgment of the forged data, thus incurring an undesired behavior.

Unlike manipulation and injection, deleting data packets cannot be detected by CellDAM. The attacker can send noises and corrupt the data packets. CellDAM cannot distinguish it from a corruption caused by environmental noises. There is no readily available scheme to defend it without changing the 5G PHY; changing PHY is beyond the scope of this work.

Attacks against data plane signaling We show that, all the detection, manipulation, and deletion of signaling messages can be detected by CellDAM. This is relatively straightforward compared with data packet forgery. Since the forged or missing signaling is not anticipated, some messages will be received in wrong or unexpected context. Three examples

#	Attack	Target Message	New?	CellDAM	Undesired Behavior	Check
A1	DL Data Manipulation w/ Relay FBS	Data packet	Adapted from [48, 49]	✓	The forged packet received during DRX OFF.	c_4
A2	DL Data Forgery w/ Retransmission	Data packet	Inspired by [54, 60]	✓	Forged DCI for forged data received in wrong context.	c_1
A3	Packet Delivery Blocking	RLC Control NACK	Adapted from [54]	✓	The RLC Control is not received when the timer expires.	c_1, c_2
A4	Prolonged Packet Delivery	DRX Command	Adapted from [54]	✓	Message received with pending transmission; DCI during DRX OFF.	c_1, c_4
A5	Radio Resource Draining	Buffer Status Report	Adapted from [54]	✓	Grant is received when no data is pending.	c_5
A6	Break Reliable Transfer	RLC Control ACK	Yes	✓	RLC packet is NACKed after being already ACKed.	c_3
A7	Data Collision	DCI UL Grant	Yes	✓	Data sent in unauthorized resource blocks will not be acknowledged.	c_5
A8	Delayed Transfer	MAC ACK	Yes	✓	Sender MAC falsely regards ACK and triggers RLC retransmission.	c_1

Table 2: List of known (A1–A5) and unreported (A6–A8) data-plane attacks and how they trigger undesired messages. See Appendix A for details of each attack.

are listed in Table 2 (A3-A5). They are adapted from those reported in 4G or Cellular IoT and include all three types of attack. Each attack violates a certain context in packet delivery and fails to pass all checks. For A3, the attack corrupts an RLC NACK signaling. This incurs head-of-line blocking, stopping the delivery for more than 100ms. The attack can be detected by the device, as the device observes no RLC NACK after it requests one in the uplink packet. For A4, the attacker injects a DRX command signaling message, which forces the device into DRX OFF even in the presence of new data. The device thus receives a DCI during DRX OFF, detected by its lack of ACK/NACK. For A5, the device manipulates the amount in the BSR request to drain the wireless resource and block access. The device will observe unsolicited grant without pending UL data.

Unreported attacks CellDAM also detects unreported data-plane attacks, since it *verifies what is right* and detects any potential attack that breaks the delivery procedure. For each DFA state, signaling message, and validation, we check if a forged message that fails the validation can be from the adversary to incur damages. Consequently, we illustrate three unreported attacks and how CellDAM can detect them in Table 2 (A6-A8) with details in Appendix A.

8.2 Attacks against CellDAM

We next consider an attacker who is aware of the existence of CellDAM and tries to break it under our threat model. We specifically focus on the security of SecHub. This is because, if the inference or mitigation is implemented within the firmware, it is considered difficult to break it without an insider attacker. This is beyond the scope of our threat model.

Attacker attempts to break CellDAM's inference We first consider an attacker who tricks SecHub and prevents CellDAM from recognizing the traffic from the protected device. With CellDAM design, this is not possible. The traffic is generated from the in-device application with the pre-defined pattern. The application is also secure as it needs no root or other privileges. Therefore, the pattern is unknown to the attacker, who cannot insert malicious packet to break the inference. In addition, CellDAM is stable by repeating inference three times to confirm the target, avoiding a malicious attacker to inject noises and break the inference.

Attacker compromises or breaks SecHub The attacker can either gain access to SecHub, or break the communication between SecHub and the device. However, neither is possible. First, SecHub is available to end users without exposing any unnecessary interfaces, such as Internet access or wireless control. A user or an application will be unable to add/change/delete the SecHub software. SecHub is solely used for CellDAM detection and mitigation purposes. A passcode is set to access its functionalities. Second, SecHub communicates securely with the protected device. The user installs an application in the protected device using the certificate that comes with the SecHub. The device thus mutually authenticates with SecHub and encrypts all traffic between them. Only nonsensitive information is exchanged over this channel.

Attacker leverages SecHub to force handover The attacker could force the device to switch band by deteriorating the signal strength sensed by SecHub. However, such an attack is very limited in its impact. First, the device has immediate data access after moving to a new cell. The handover disruption is short. Meanwhile, the attacker cannot control which new cell the device moves to. For this purpose, the attacker needs to launch a time-consuming identification procedure on all local cells. The attacker cannot control which cell/base station the victim switches to, either. Instead, the band switching in 5G will prioritize the cells from the same (legitimate) base station. Forcing the device to an illegitimate base station is thus unlikely. Therefore, the attacker gains little from forcing

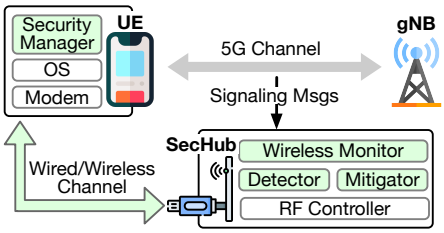


Figure 6: Implementation of CellDAM. Green blocks are CellDAM modules.

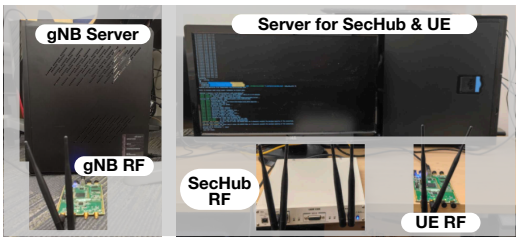


Figure 7: Testbed Setup for CellDAM.

a handover with SecHub. Defending against it is out of the scope of CellDAM.

9 Implementation

We implement CellDAM as shown in Figure 6. We implement the defense model with the least privilege (Defence model 3). Based on our discussion, we implement SecHub to perform attack detection and mitigation. A security manager app on UE facilitates SecHub for rootless signaling capture. We next elaborate on each component. It could be adapted to the other two defense models with little change. For detection, our detector module could be directly used by the other two models. For mitigation, they only require to trigger one extra command after detection.

Wireless Monitor A wireless monitor is deployed based on srsRAN [51] to perform the rootless signaling capture through the RF controller with the SDR devices. We implement it with 2,794 lines of C++ code. The monitor collaborates with the UE to camp on the target cell, infer the UE's C-RNTI, and simultaneously capture real-time messages on both the uplink and downlink channels. After the capture, it decodes the signaling messages accordingly and passes them to the attack detector.

Detector with state-dependent model checking We implement the state-dependent model checking attack detector with 1,252 lines of Python. The real-time traces from the wireless monitor will be fed into the detector for undesired behavior and potential attacks. If any consecutive signaling messages violate the cross-layer model checking, potential attacks will be reported by the detector. The detector will further notify the mitigator module to trigger the mitigation.

Mitigator We deploy the mitigator with 860 lines of Python code. After detecting any attacks, the mitigator triggers the victim handover. With the current signal conditions acquired from the security manager application from the victim, the mitigator calculates the minimum transmit power to trigger the UE handover with the SDR devices. It then notifies the RF controller to initiate signals targeting the victim UE. This will trigger handover to escape from the attacker's cell.

Security Manager App On the phone side, we deploy a security manager application with 1,264 lines of Java. The application monitors the current band, PCI, and signal conditions (RSRP and RSRQ) with the Android Telephony API [10] to facilitate the rootless signaling capture. It also generates a corresponding UDP traffic after receiving the traffic interval from SecHub for the collaborated traffic fingerprinting. It supports exchanging the data with SecHub through a wired (USB) or wireless (Bluetooth in the current implementation) connection leveraging Android APIs [9]. It supports the X.509 certificate to facilitate the mutual authentication and encryption between the app and SecHub. We also implement an equivalent application for srsUE running on user-space. The same set of information is extracted from the srsUE by hooking the current srsUE functions. Then the information is shared with the SecHub with the socket API.

10 Evaluation

10.1 Evaluation Setup

Testbed Setup We construct a testbed for experimental validation, as shown in Figure 7. The gNB and UE are built upon srsRAN [51] 5G protocols. The physical layer encoding is still kept with 4G due to current hardware limitations. CellDAM does not rely on any 5G-specific PHY feature. The gNB software is run on an i7-9700K PC with Ubuntu 20.04. The UE runs on an Intel Xeon Silver 4214 server running Ubuntu 20.04. Both use USRP B210 as their RF frontend, with the frequency set to an unlicensed 2.4GHz ISM band. SecHub is co-located on the same server as the UE and uses USRP X300 as the RF frontend. The security manager application runs on the same server as a user-space process and shares the information extracted from srsUE with SecHub. The mobile version of security manager application is tested on a Pixel 4a with Snapdragon 730 running Android 12.

Attack Reproduction All attacks listed in Table 2 are recreated within the testbed in order to evaluate CellDAM's ability of attack detection and mitigation. We realize attacks on both data packets and signaling. We simplify the attacks with partial software emulation on our testbed for controllability and reproducibility In attacks with relay FBS, we set up both the FBS and the real gNB in the testbed. We emulate the radio link between relay FBS and real gNB in software with ZeroMQ [61] to avoid interference with the link between UE and relay FBS, which uses physical link with USRP.

On the other hand, the attacks without relay FBS rely on manipulation of the underlying data channel. We emulate the attacker using a separate thread within the gNB and UE program, which has access to the transmitting and receiving signal buffer. Eavesdropping is realized with inspection of the receiving buffer, while the forging or corruption attacks are

Attack	A1	A2	A3	A4	A5	A6	A7	A8
Precision	0.989	1	0.999	1	0.996	1	0.996	0.999
Recall	0.705	0.976	1	0.989	1	1	1	1
F1	0.823	0.988	0.999	0.994	0.998	1	0.998	0.999

Table 3: Effectiveness of attack detection with CellDAM.

emulated by injecting encoded attack messages or noise to the transmitting buffer.

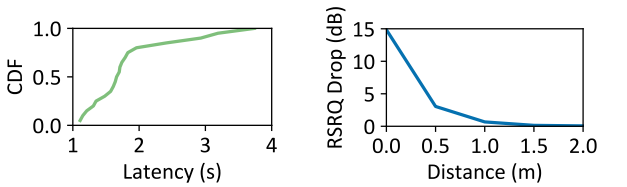
Ethical Considerations This work does not raise any ethical issues. Our testbed is carefully controlled for experiments, operating on an unlicensed 2.4GHz ISM band. The experimentation is conducted within a 5MHz channel centered at 2.49GHz. We ensure no nearby device is using the frequency band. The radio signal emitted by the testbed only reaches a few meters, ensuring that no other device is affected or attempts to communicate with our testbed. Meanwhile, we are working with collaborating mobile operators about the discovered solutions and will open source CellDAM.

10.2 Evaluation Results

Evaluation of CellDAM Attack Detection In this subsection, we answer the question of whether CellDAM can detect all attacks displayed in Table 2. For this purpose, we inject attack messages according to the attacker procedure, and observe if the detector can initiate warnings as expected. We test the attack detection under two different traffic types: A lightweight traffic with ping and a heavy traffic with iPerf3 [30] to saturate the channel. For each attack, we repeat 1,000 times under each scenario. We evaluate the results with three metrics: precision, recall, and F1 score. The ground truth can be easily obtained, as whether there is an ongoing attack is controlled by us.

Table 3 summarizes the precision, recall, and F1 score for the CellDAM detection. As shown in the table, the detection reaches a precision of 98.9%∼100% for different attack types. The high precision is achieved by the correctness of the DFA and verification, as the normal operations in legitimate 5G delivery will follow the correct procedure. Meanwhile, the recall is 70.5%∼100% for different attack detection. Note that, the relatively low recall for A1 is because a heavy-traffic scenario will extend the ON state for a device. The device can use other FBS detection methods [8, 65] to complement CellDAM. The recall is high for other attacks, as they will incur undesired data-plane signaling. This results in a high F1 score of 0.823∼0.999. The detection works well for both light and heavy traffic. This is because CellDAM only requires inspection on lightweight data-plane signaling messages. We also measure the average detection latency for attack detection. The signaling messages could be captured and fed into the detector for real-time detection. The detection achieves an average latency of 28ms. Therefore, CellDAM can quickly spot potential adversaries and take action.

Evaluation of CellDAM Mitigation We evaluate the success rate for mitigation. When SecHub detects any attack, it will trigger the target device handover to escape from the attack. We enable the handover-related RRC messages and config in gNB and let the UE monitor the handover events. Note that our testbed does not support real handover to a different cell; if UE receives a handover command, we assume a successful handover. We evaluate the ratio of successful handover and the corresponding latency for each mitigation. In all 40 rounds of experiments, SecHub successfully triggers the UE handover. Figure 8a shows the CDF of the latency. The results show that the average mitigation latency is 1.85s. 90% of them could successfully handover within 3 seconds, and all handovers are triggered within 4 seconds.



(a) CDF of the mitigation latency for "quick dodge".

(b) Relationship between the distance and RSRQ drop.

Figure 8: Mitigation Performance.

In our design, signals from SecHub will have minimum impact on other devices. We show this point by measuring the perceived RSRQ drops at the UE at different distances from the SecHub. Figure 8b shows that the RSRQ drop at 1m and 2m are only 0.67dB and 0.06dB, respectively. With the controlled power of SecHub, our mitigation will only trigger handover on the close-by protected device, while not affecting other users or devices.

Impact of CellDAM mitigation on applications We discussed in §7 that handover-based cell switching will incur little overhead on applications. We now evaluate the disruption time on the iPerf3 application during the band switch. We decode the logs and calculate the interval of application packets before and after the handover. The average disruption is 72.3 ms, with a 95th percentile of 83.7ms. We further note that, the packets during 5G handover will be kept and delivered by the new cell afterwards. Therefore, an application would only experience a small delay caused by CellDAM, without triggering any data loss or TCP connection reset.

Evaluation of SecHub Rootless Capture We then present our microbenchmarks for inferring signaling messages. We measure the ratio of correctly captured signaling in both uplink and downlink to show the effectiveness of our rootless signaling capture. We record the traces of PUCCH/PDCCH (SR and DCI) and PDSCH/PUSCH (MAC CE and RLC Control) at the UE as the ground truth. We capture the traces through the SecHub under different traffic scenarios and mobility. We use ping and iPerf3 for the light and heavy traffic scenarios, respectively. The UE and SecHub are kept static or moving at a speed of around 5km/h for mobility. The ratio of correctly decoded signaling is calculated by comparing the traces captured on the SecHub and the ground truth.

Table 4 shows the success ratio for the rootless capture. For

Traffic	Control		Data	
	PUCCH	PDCCH	PUSCH	PDSCH
Light	99.1%	100%	100%	99.7%
Heavy	98.7%	99.9%	98.1%	97.2%
Light-M	98.8%	99.9%	99.1%	98.3%
Heavy-M	98.1%	99.8%	98.0%	97.3%

Table 4: Ratio of success rootless signaling message capture under different traffic scenarios and mobility (-M: Mobile).

the control messages, 99.1% of PUCCH and 100% of PDCCH are successfully captured and decoded from the SecHub with the light traffic. For the heavy traffic, the SecHub still achieves a high success rate with 98.7% for PUCCH and 99.9% for the PDCCH. For data messages, SecHub successfully decodes all the PUSCH messages and 99.7% of PDSCH data under the light traffic scenario. 98.1% of the PUSCH and 97.2% of the PDSCH traffic is successfully captured and decoded for the heavy traffic scenario. We observe similar numbers (97.3-99.9%) in mobility cases (Light-M) and (Heavy-M). The ratio is not impacted by mobility, because the UE and SecHub experience similar channel conditions. Whether one message is decoded on UE or not, SecHub will produce similar results.

The high success rate is possible with accurate C-RNTI inference. We quantify the success rate with the collaborated traffic fingerprinting. Since the inference can be done without actively sending signals over the air, we perform the verification on both the commercial network and our testbed. The ground truth of C-RNTI can be acquired in gNB (by checking logs) and in COTS UE (by using MobileInsight [37]). Every time SecHub collects 5s of traces for the inference after the traffic pattern coordination with the target UE. We perform 120 rounds of experiments with 60 rounds on the testbed and 60 rounds on the commercial network.

On the testbed, SecHub correctly infers the C-RNTI in all 60 rounds, achieving a 100% success rate. On the commercial network, with the increased device number, SecHub successfully infers 98.3% of the C-RNTI. We further measure the overheads caused by the background ping in the continuous tracking. The result shows that CellDAM involves 0.14 KBps traffic overhead, which is marginal on the target device. The results show that CellDAM could continuously perform the monitoring during the mobility, and protect the victim without any root privilege.

11 Related Work

As new attacks on the 5G/4G have drawn increasing attention, detecting potential attacks is a popular research topic in recent years. Due to the high overhead and long cycle for network-side detection [47], current detection methods are mainly on the device side. PHOENIX [19] proposes a solution for control-plane monitoring. [53] studies the device-side attack detection for core network attacks. [16, 18, 25, 38, 63] detect cellular attacks by analyzing on-device application traces. [8, 65] detect the existence of FBS with physical characteristics of FBS such as power or signal signatures. No prior work studies attack detection for data-plane protocols. CellDAM provides the first detection scheme for attacks on data-plane packets/signaling in 5G. Unlike other works that require root access and expose additional risks [62], CellDAM detects the attack without extra privilege.

Existing mitigation for attacks on 4G/5G protocols either needs root access [33] or requires protocol changes [48, 49, 54, 64]. To our knowledge, CellDAM is the first solution that provides rootless mitigation for data-plane attacks without firmware/standard changes. Other mitigation methods for attacks on mobile apps [11, 13, 39] or cellular-based services [35, 40, 55, 58] are orthogonal to our work.

Prior studies have leveraged model checking to verify the cellular standards and discover new vulnerabilities in the protocols. [26, 28] exposed attacks in 4G LTE by adversarial model-based testing. Previous work also formally analyzed the 5G protocol components including the 5G-AKA procedures [12, 17] and NAS/RRC signalings [27]. However, the existing cellular protocol verification runs offline on the control plane and does not have runtime detection. CellDAM performs the runtime verification to discover the potential attacks timely. It targets the relatively more difficult problem of verifying the data plane, whose traffic is heavier.

12 Conclusion

Detection of data-plane attacks at the mobile device has not been viewed favorably to date. This is due to the excessive processing and energy overhead associated with 5G high data rate. In this work, we show how to use data-plane signaling messages to devise a detection solution that yields one or two orders of magnitude lower overhead. We leverage the fact that data-plane attacks would exhibit certain data signaling misbehavior. Our reactive solution may defend against certain attacks that the current proactive schemes cannot. It can work on normal user devices without root privilege or infrastructure upgrade. Once CellDAM detects an attack, it further mitigates attack damages via handover to another available channel.

In a broader scope, we believe data-plane security deserves more attention given that activating all security measures on lightweight control-plane messages is relatively straightforward. In contrast, data plane delivery involves complex interactions across protocols and the adversary has plenty playground to launch various attacks from applications, transport layer, to IP and 5G protocols. While the end-to-end approach and existing 5G data-plane security may secure application data, it fails to secure the 5G infrastructure and mobile device. To this end, this work reports our initial effort to explore a lightweight, reactive solution to 5G data-plane security.

Acknowledgments. We would like to thank the anonymous reviewers and our shepherd, Dr. Aaron Schulman, for their constructive comments and feedback.

References

[1] 3GPP. TS33.501: Security architecture and procedures for 5G System-V15.4.0, May. 2019.

[2] 3GPP. TS36.321: Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification, Sep. 2019.

[3] 3GPP. NR; Medium Access Control (MAC) protocol specification, Dec. 2020.

[4] 3GPP. TS33.809: Study on 5G security enhancements against False Base Stations (FBS), Dec. 2020.

[5] 3GPP. NR; Radio Link Control (RLC) protocol specification, Jan. 2021.

[6] 3GPP. TS38.211: NR; Physical channels and modulation, Jan. 2021.

[7] 3GPP. TS33.501: Security architecture and procedures for 5G System-V16.4.0, Mar. 2022.

[8] ALI, A., AND FISCHER, G. Enabling fake base station detection through sample-based higher order noise statistics. In *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)* (2019), IEEE, pp. 695–700.

[9] API, A. B. https://developer.android.com/guide/topics/connectivity/bluetooth.

[10] API, A. T. https://developer.android.com/reference/android/telephony/package-summary.

[11] BALAPOUR, A., NIKKHAH, H. R., AND SABHERWAL, R. Mobile application security: Role of perceived privacy as the predictor of security perceptions. *International Journal of Information Management 52* (2020), 102063.

[12] BASIN, D., DREIER, J., HIRSCHI, L., RADOMIROVIC, S., SASSE, R., AND STETTLER, V. A formal analysis of 5g authentication. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security* (2018), pp. 1383–1396.

[13] BUI, D., YAO, Y., SHIN, K. G., CHOI, J.-M., AND SHIN, J. Consistency analysis of data-usage purposes in mobile apps. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security* (2021), pp. 2824–2843.

[14] CALCULATIONS, U. W. R. https://www.electronicdesign.com/technologies/communications/article/21796484/understanding-wireless-range-calculations.

[15] CASATI, L., AND VISCONTI, A. The dangers of rooting: data leakage detection in android applications. *Mobile Information Systems 2018* (2018).

[16] CHLOSTA, M., RUPPRECHT, D., HOLZ, T., AND PÖPPER, C. Lte security disabled: misconfiguration in commercial networks. In *Proceedings of the 12th conference on security and privacy in wireless and mobile networks* (2019), pp. 261–266.

[17] CREMERS, C., AND DEHNEL-WILD, M. Component-based formal analysis of 5g-aka: Channel assumptions and session confusion.

[18] DABROWSKI, A., PIANTA, N., KLEPP, T., MULAZZANI, M., AND WEIPPL, E. Imsi-catch me if you can: Imsi-catcher-catchers. In *Proceedings of the 30th annual computer security applications Conference* (2014), pp. 246–255.

[19] ECHEVERRIA, M., AHMED, Z., WANG, B., ARIF, M. F., HUSSAIN, S. R., AND CHOWDHURY, O. Phoenix: Device-centric cellular network protocol monitoring using runtime verification. *arXiv preprint arXiv:2101.00328* (2021).

[20] FALKENBERG, R., AND WIETFELD, C. Falcon: An accurate real-time monitor for client-based mobile network data analytics. In *2019 IEEE Global Communications Conference (GLOBECOM)* (2019), IEEE, pp. 1–7.

[21] GASPARIS, I., QIAN, Z., SONG, C., AND KRISHNAMURTHY, S. V. Detecting android root exploits by learning from root providers. In *26th USENIX Security Symposium (USENIX Security 17)* (2017), pp. 1129–1144.

[22] HO, T.-H., DEAN, D., GU, X., AND ENCK, W. Prec: practical root exploit containment for android devices. In *Proceedings of the 4th ACM conference on Data and application security and privacy* (2014), pp. 187–198.

[23] HOLTMANNS, S., RAO, S. P., AND OLIVER, I. User location tracking attacks for lte networks using the interworking functionality. In *2016 IFIP Networking conference (IFIP Networking) and workshops* (2016), IEEE, pp. 315–322.

[24] HOLTMANNS, S., RAO, S. P., AND OLIVER, I. User location tracking attacks for lte networks using the interworking functionality. In *2016 IFIP Networking Conference (IFIP Networking) and Workshops* (2016), pp. 315–322.

[25] HONG, B., BAE, S., AND KIM, Y. Guti reallocation demystified: Cellular location tracking with changing temporary identifier. In *NDSS* (2018).

[26] HUSSAIN, S., CHOWDHURY, O., MEHNAZ, S., AND BERTINO, E. LTEInspector: A systematic approach for adversarial testing of 4G LTE. In *Network and Distributed Systems Security (NDSS) Symposium 2018* (2018).

[27] HUSSAIN, S. R., ECHEVERRIA, M., KARIM, I., CHOWDHURY, O., AND BERTINO, E. 5greasoner: A property-directed security and privacy analysis framework for 5g cellular network protocol. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (2019), pp. 669–684.

[28] HUSSAIN, S. R., KARIM, I., ISHTIAQ, A. A., CHOWDHURY, O., AND BERTINO, E. Noncompliance as deviant behavior: An automated black-box noncompliance checker for 4g lte cellular devices. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security* (2021), pp. 1082–1099.

[29] IETF. The Transport Layer Security (TLS) Protocol Version 1.3, August 2018. https://datatracker.ietf.org/doc/rfc8446/.

[30] IPERF3. https://github.com/esnet/iperf.

[31] KASPERSKY. Rooting your Android: Advantages, disadvantages, and snags, June 2017. https://usa.kaspersky.com/blog/android-root-faq/11581/.

[32] KOHLS, K., RUPPRECHT, D., HOLZ, T., AND PÖPPER, C. Lost traffic encryption: fingerprinting LTE/4G traffic on layer two. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks* (2019), pp. 249–260.

[33] KOTULIAK, M., ERNI, S., LEU, P., ROESCHLIN, M., AND CAPKUN, S. Ltrack: Stealthy tracking of mobile phones in lte. In *31st USENIX Security Symposium (USENIX 2022)* (2022).

[34] LAB, K. Rooting your android: Advantages, disadvantages, and snags. https://www.kaspersky.com/blog/android-root-faq/17135/, Jun. 2017.

[35] LI, C.-Y., TU, G.-H., PENG, C., YUAN, Z., LI, Y., LU, S., AND WANG, X. Insecurity of voice solution volte in lte mobile networks. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (2015), pp. 316–327.

[36] LI, M., ZHU, H., GAO, Z., CHEN, S., YU, L., HU, S., AND REN, K. All your location are belong to us: Breaking mobile social networks for automated user location tracking. In *Proceedings of the 15th ACM international symposium on Mobile ad hoc networking and computing* (2014), pp. 43–52.

[37] LI, Y., PENG, C., YUAN, Z., LI, J., DENG, H., AND WANG, T. Mobileinsight: Extracting and analyzing cellular network information on smartphones. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking* (2016), pp. 202–215.

[38] LI, Y., PENG, C., ZHANG, Z., TAN, Z., DENG, H., ZHAO, J., LI, Q., GUO, Y., LING, K., DING, B., ET AL. Experience: a five-year retrospective of mobileinsight. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking* (2021), pp. 28–41.

[39] LU, H., XING, L., XIAO, Y., ZHANG, Y., LIAO, X., WANG, X., AND WANG, X. Demystifying resource management risks in emerging mobile app-in-app ecosystems. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security* (2020), pp. 569–585.

[40] LU, Y.-H., LI, C.-Y., LI, Y.-Y., HSIAO, S. H.-Y., XIE, T., TU, G.-H., AND CHEN, W.-X. Ghost calls from operational 4g call systems: Ims vulnerability, call dos attack, and countermeasure. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking* (2020), pp. 1–14.

[41] MJØLSNES, S. F., AND OLIMID, R. F. Easy 4G/LTE IMSI catchers for non-programmers. In *International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security* (2017), Springer, pp. 235–246.

[42] OPENSIGNAL. USA 5G Experience Report, Jan 2022. https://www.opensignal.com/reports/2022/01/usa/mobile-network-experience-5g.

[43] PARK, C., BAE, S., OH, B., LEE, J., LEE, E., YUN, I., AND KIM, Y. Doltest: In-depth downlink negative testing framework for lte devices. In *USENIX Security Symposium* (2022).

[44] POSITIVE TECHNOLOGIES. Security assessment of Diameter networks, 2020.

[45] QUALCOMM. QxDM Professional - QUALCOMM eXtensible Diagnostic Monitor. http://www.qualcomm.com/media/documents/tags/qxdm.

[46] RAO, S. P., KOTTE, B. T., AND HOLTMANNS, S. Privacy in lte networks. In *Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications* (2016), pp. 176–183.

[47] RUPPRECHT, D., DABROWSKI, A., HOLZ, T., WEIPPL, E., AND PÖPPER, C. On security research towards future mobile network generations. *IEEE Communications Surveys & Tutorials 20*, 3 (2018), 2518–2542.

[48] RUPPRECHT, D., KOHLS, K., HOLZ, T., AND PÖPPER, C. Breaking LTE on layer two. In *2019 IEEE Symposium on Security and Privacy (SP)* (2019), IEEE, pp. 1121–1136.

[49] RUPPRECHT, D., KOHLS, K., HOLZ, T., AND PÖPPER, C. IMP4GT: Impersonation attacks in 4G networks. In *Symposium on Network and Distributed System Security (NDSS). ISOC* (2020).

[50] SHAO, Y., LUO, X., AND QIAN, C. Rootguard: Protecting rooted android phones. *Computer 47*, 6 (2014), 32–40.

[51] SRSRAN. https://www.srslte.com/.

[52] STATISTA. Where 5G Technology Has Been Deployed , July 2022. https://www.statista.com/chart/23194/5g-networks-deployment-world-map/.

[53] TAN, Z., DING, B., ZHANG, Z., LI, Q., GUO, Y., AND LU, S. Device-centric detection and mitigation of diameter signaling attacks against mobile core. In *2021 IEEE Conference on Communications and Network Security (CNS)* (2021), IEEE, pp. 29–37.

[54] TAN, Z., DING, B., ZHAO, J., GUO, Y., AND LU, S. Data-plane signaling in cellular iot: attacks and defense. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking* (2021), pp. 465–477.

[55] TU, G.-H., LI, C.-Y., PENG, C., LI, Y., AND LU, S. New security threats caused by ims-based sms service in 4g lte networks. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (2016), pp. 1118–1130.

[56] US, W. What is 5G? Facts Stats You Need to Know, April 2022. https://www.whistleout.com/CellPhones/Guides/5G-statistics.

[57] VIRTUALISATION, N. F. An introduction, benefits, enablers, challenges & call for action. In *White Paper, SDN and OpenFlow World Congress* (2012).

[58] WANG, S., TU, G.-H., LEI, X., XIE, T., LI, C.-Y., CHOU, P.-Y., HSIEH, F., HU, Y., XIAO, L., AND PENG, C. Insecurity of operational cellular iot service: new vulnerabilities, attacks, and countermeasures. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking* (2021), pp. 437–450.

[59] XU, D., ZHOU, A., ZHANG, X., WANG, G., LIU, X., AN, C., SHI, Y., LIU, L., AND MA, H. Understanding operational 5g: A first measurement study on its coverage, performance and energy consumption. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication* (2020), pp. 479–494.

[60] YANG, H., BAE, S., SON, M., KIM, H., KIM, S. M., AND KIM, Y. Hiding in plain signal: Physical signal overshadowing attack on LTE. In *28th USENIX Security Symposium (USENIX Security 19)* (2019), pp. 55–72.

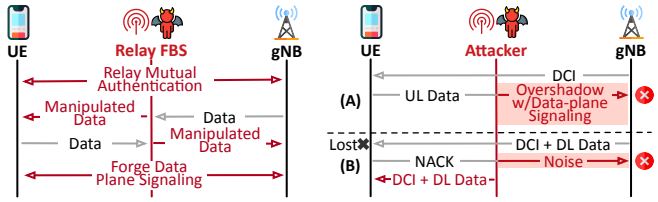
[61] ZEROMQ. ZeroMQ: An open-source universal messaging library . https://zeromq.org/, Jan 2022.

[62] ZHANG, H., SHE, D., AND QIAN, Z. Android root and its providers: A double-edged sword. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (2015), pp. 1093–1104.

[63] ZHANG, Y., LIU, B., LU, C., LI, Z., DUAN, H., HAO, S., LIU, M., LIU, Y., WANG, D., AND LI, Q. Lies in the air: Characterizing fake-base-station spam ecosystem in china. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security* (2020), pp. 521–534.

[64] ZHAO, J., DING, B., GUO, Y., TAN, Z., AND LU, S. Securesim: rethinking authentication and access control for sim/esim. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking* (2021), pp. 451–464.

[65] ZHUANG, Z., JI, X., ZHANG, T., ZHANG, J., XU, W., LI, Z., AND LIU, Y. Fbsleuth: Fake base station forensics via radio frequency fingerprinting. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security* (2018), pp. 261–272.



(a) Relay FBS (b) Overshadowing

Figure 9: Viable data-plane methodologies for data injection/manipulation.

A Details of Each Attack and Its Detection

In this section, we present the details of each attack shown in Table 2, including the attack procedure that leverages the forgery messages and the attack consequences. We also elaborate on how each attack incurs undesired behavior that can be detected by `CellDAM` validations.

A.1 Injecting Data with FBS

Attack Procedure The attacker can use a relay fake base station (FBS) to manipulate data packets [48]. The detailed attack procedure is shown in Figure 9a. The attacker sets up a fake base station (FBS). It usually runs on a different band from the real base station, and sends strong broadcast signal to lure the device into connection. The attack also sets up a fake UE that connects to the real base station. It then relays the connection setup messages without any manipulation from/to the real base station. Afterwards, although all data packets are encrypted, the attacker can flip the bits to change the content of the forgery, if data-plane integrity protection is not enforced. This is doable as the encryption is done with simple XOR. as long as the original content is known (e.g., DNS server set up by the operator), the attacker can flip bits and change the contents to target values. For instance, the attacker can manipulate the IP header of a DNS request and compensate the IP header checksum to pass the checks [48].

Undesired Behavior In this attack, we assume the adversary cannot infer the data-plane configurations encrypted in RRC messages. Therefore, when the attacker forwards the data packet, some configurations might be incorrectly set and detected on the device side. One example is DRX configuration. Without the configuration, the forwarded packet might fall in the DRX OFF period, as shown in Figure 10a. As the DRX ON period is usually very short (e.g., 10ms), most messages might be delivered outside of the period, violating check c_4. Although the FBS can repeat transmitting until the victim device acknowledges to ensure delivery, this behavior will be detected by `CellDAM`. Note that, 5G DRX includes the mechanism to stay in ON period for an extended period when a new data is received. Therefore, when the traffic is heavy, the device might keep staying in ON state. As we claimed in §10, the attack will be more detectable for light traffic. For this attacker, the PHY layer detection methods mentioned in §11 can help detect FBS that transmits abnormal signal.

A.2 Data Manipulation with Retransmission

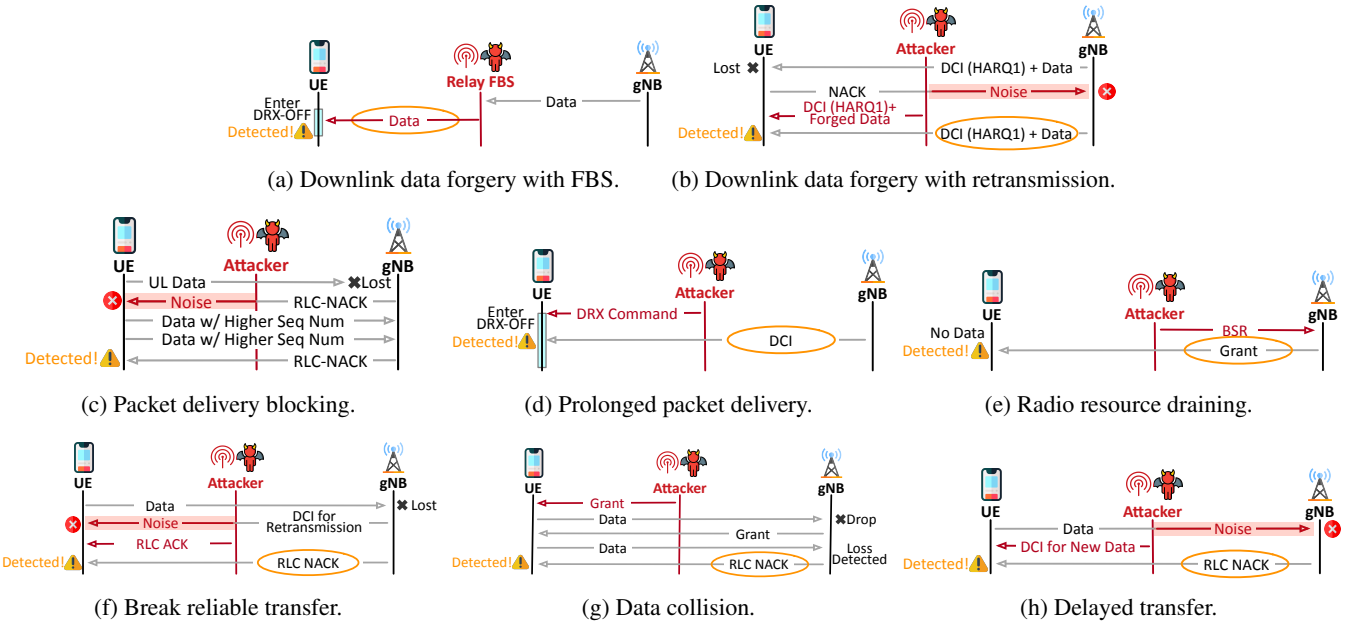
Attack Procedure We also show that, the attacker can serve man-in-the-middle to manipulate data packets without FBS. The detailed attack procedure is shown in Figure 9b. This approach can manipulate data plane packet without FBS. We consider the victim device is directly connected to the authentic base station. Note that, the attacker might not be able to forge data-plane packets directly, as they are encrypted (unlike integrity protection, which is optional). Therefore, to forge data packets, the attacker still needs to take the bit flipping approach as in A1. One viable way is to forge the data as retransmission. The attacker can eavesdrop on the channel and look for data transmission that fails on victim device (i.e., trigger NACK), while it successfully decodes the encrypted data (due to less noisy environment, etc.). The attacker can then forge the DCI and manipulated data as the retransmission.

Undesired Behavior This attack requires sending forged DCI and data to the device. This DCI can be received in an improper context. Each DCI includes a HARQ ID, which indicates the process of the transmission. Each process takes a stop-and-wait procedure. Before the last packet is acknowledged, the process will not move on to transmit the next one. Therefore, the DCI from the authentic gNB can arrive after the DCI forged by attacker and before the forged retransmission has been acknowledged. This causes an undesired behavior that can be observed on the device with c_1. This is shown in Figure 10b.

We further note two things. First, this attack method can forge *uplink* data packet without obvious undesired behavior on the device side. We do not consider such attack with pure uplink forgery in this work. However, a reasonable forgery attack needs to manipulate both uplink and downlink data. Second, an interested reader might ask whether the attacker can use the legitimate DCI to send the forged data. However, the DL DCI and the corresponding forged data are usually sent in the same time slot in 5G. It is thus hardly possible to infer DCI for data forgery in advance.

A.3 Packet Delivery Blocking

Attack Procedure We have introduced this attack in §3.2 as an example of corrupting data-plane signaling. We now present this attack in more details as shown in Figure 10c. The attacker first eavesdrops on the data channel and learns the packet sequence number that is not delivered over the air. It then selectively corrupts the RLC control that NACKs the packet. Since the packet is not acknowledged, UE will still send uplink data without retransmitting the missing one. These new data packets are blocked in the gNB, which suffers



(a) Downlink data forgery with FBS. (b) Downlink data forgery with retransmission.

(c) Packet delivery blocking. (d) Prolonged packet delivery. (e) Radio resource draining.

(f) Break reliable transfer. (g) Data collision. (h) Delayed transfer.

Figure 10: Illustration of undesired behavior caused by attacks.

head-of-line blocking for more than 100ms. No new data will be forwarded during this time period.

Undesired Behavior The attack will cause undesired behavior on RLC. From the perspective of the device, the uplink RLC will receive either ACK or NACK after $T_{reordering}$ when the timer expires. Even if the signaling is corrupted, which is rare given its small size, the MAC layer should retransmit it. Therefore, if the device finds no RLC control after timer expires and MAC retransmissions (which has a configured max count), `CellDAM` detects a potential corruption attack. We note that all MAC retransmission could fail due to extremely weak channel instead of an attacker. In this case, although no attack is present, switching to a better wireless channel with `CellDAM` is a reasonable facilitating option.

A.4 Prolonged Packet Delivery

We consider connected mode Discontinuous Reception (CDRX) in this attack. In the RRC connected state, the gNB will only deliver data during DRX ON state. A device will be in DRX ON for a small time period in a fixed periodicity. If any data is received during this period, the DRX ON state is extended for a constant amount of time. gNB will configure the ON period, DRX cycle, and the extended timer amount in encrypted RRC messages.

Attack Procedure The detailed procedure is shown in Figure 10d. In this attack, the adversary forges a DRX command to the device. DRX command terminates the DRX ON state prematurely and the device enters the sleep mode. Therefore, the device will be unable to receive all subsequent transmission in the current DRX cycle with DRX turned to OFF state. The data delivery can be delayed for hundreds of milliseconds given long DRX cycle. The attack is adapted from [54]. Note that, this attack will not stop uplink data delivery, as a UL data transmission initiated by UE can again switch the DRX state to DRX ON.

Undesired Behavior It is not possible to receive downlink DCI or data during DRX OFF. If such an event happens, the previous DRX command could be forged. As `CellDAM` does not have root access, it is difficult for `SecHub` to monitor internal DRX state in real-time or infer the state with encrypted DRX configurations in RRC messages. However, there is another indication for a message during DRX OFF: If the device is in DRX OFF, it will not respond to any downlink data with ACK or NACK in PUCCH. Therefore, `SecHub` can detect such attack by checking the downlink data without any acknowledgment, after an incoming DRX command. This violates the validation check c_4. In addition, gNB will not send DRX command when the previous DL transmission has not finished, which means there will be more transmission in the DRX cycle. This violated c_1.

A.5 Radio Resource Draining

Attack Procedure We have introduced this attack in §3.2 as an example of manipulating data-plane signaling. Here we present more details. 5G/4G adopts scheduling-based data delivery. To transmit uplink data, the device needs to send the buffer status report (BSR) to the gNB for asking grants. An attacker can forge a BSR to the gNB. In the forged BSR, the attacker falsely indicates the victim device has a large amount of data to send. The gNB subsequently assigns excessive resource blocks to the device, wasting wireless resource and

blocking other users' access. The detailed procedure is shown in Figure 10e. The attack is adapted from [54].

Undesired Behavior Although the attacker forges an uplink message, it will incur observable undesired behavior on the device side as well. This is because the DCI (for UL grant) will be triggered by the forged BSR message. If the device receives grant when there is no prior request, the grant can be caused by a forged request by the attacker. We notice that, a gNB can send a device "free" small grants in case the device has something to send. However, these grants are small for the device to sufficiently deliver BSR. Therefore, to reduce false positive, we set a threshold (120 bytes) in the verification check c_5 to find unwanted resources.

A.6 Break Reliable Transfer

Attack Procedure The detailed procedure of the attack is shown in Figure 10f. The attacker can corrupt the data retransmission on MAC layer. It then forges an RLC ACK to UE. Receiving it, the victim device RLC protocol wrongly thinks the packet has been delivered, discarding it in the buffer. Therefore, this packet cannot be reliably delivered in 5G. This might further trigger TCP retransmission, which can cause more serious damage.

Undesired Behavior From the perspective of the base station, it will detect a packet gap in RLC protocol when it receives a later packet from UE. Therefore, it will still attempt to recover it by sending an RLC NACK. The NACK timing is unknown for the attacker, thus cannot be targeted for corruption. The device side will thus receive an RLC NACK first and then an ACK. This behavior is undesired in 5G and can be detected by validation check c_3.

A.7 Data Collision

Attack Procedure The detailed procedure of the attack is shown in Figure 10g. The attacker forges grant to the victim device. The device will send data using the forged grant. However, any data using these non-authorized grants will not be correctly accepted by the gNB. In addition, the legitimate transmission in the same time and frequency by another user will be corrupted by the victim's false transmission.

Undesired Behavior The attack will trigger undesired behavior on the device side. Since the UL data using false grants will not be accepted, gNB will not ACK or NACK the message delivery and consequently trigger an RLC NACK. `CellDAM` detects the undesired behavior with RLC control NACK and lack of MAC layer feedback with validation c_5.

A.8 Delayed Transfer

Attack Procedure The detailed procedure of the attack is shown in Figure 10h. In this attack, the adversary sends noises and corrupts the uplink data. The attacker can learn the time and frequency of the delivery by eavesdropping on DCI for UL grants. It consequently forges DCI for new data (i.e., indicator for ACK) to stop the UE from retransmitting the corrupted data. Consequently, the UE will start sending new data on MAC. This will later trigger retransmission on the RLC layer, which can take up to hundreds of milliseconds compared to a MAC fast retransmission.

Undesired Behavior The forged DCI from the attacker can be sent in a wrong context, where two consecutive DCIs are received but the data using the first one has not been acked yet. This can also be detected with c_1.

B Deriving Minimal Power for Targeted Switching

The solution should not affect other devices. `SecHub` adaptively controls its power upon triggering the handover. Previous 5G measurements show that -20dB RSRQ is enough to trigger the handover in more than 98% cases [59]. `SecHub` adaptively derives the minimal power so that RSRQ drops to -20dB, thus triggering handover. The current RSRQ is derived by:

$$RSRQ = \frac{N \times RSRP}{RSSI}$$

To reduce the RSRQ to -20dB, the minimal power (P_m) needed by `SecHub` follows:

$$\frac{N \times RSRP}{RSSI + P_m} = -20dB = \frac{1}{100}$$

Thus, the P_m could be derived by:

$$P_m = N \times RSRP \times (100 - \frac{1}{RSRQ})$$

All needed information can be acquired from the victim by the OS API (e.g., Android [10]) without root privilege. Furthermore, the power density is inversely proportional to the square of the distance from the antenna [14]. Assume `SecHub` is located close to the device (<0.1m). The RSRQ drop at the 1m distance is smaller than 1dB, which could be neglected by other devices. Only the victim device perceives a notable RSRQ drop and triggers its handover.

We admit that, even theoretically not affecting any other device, sending weak signals might require licensing from the operational networks or government. We envision `SecHub` can acquire such permission from mobile operators, or even manufactured by the operators themselves. If this privilege is not available, `CellDAM` could fallback to using airplane switching. Even if the cell is not changed after toggling, the device will be reassigned a physical ID C-RNTI, which makes it more difficult for the attacker to track the victim.

C List of Accepted Messages

We elaborate on c_1 by enumerating each state and the list of accepted message for each one. The results are shown in Table 5. If a message is in the list, we show the next state if all other validations are passed. Otherwise, we mark an $\times$ in the table which means an undesired behavior that fails c_1. As we mentioned, our method prioritizes soundness. Therefore, for messages that are not explicitly considered, `CellDAM` will ignore them and stay in the current state.

D Continuous Inference

We describe how `SecHub` could keep tracking the C-RNTI used by the victim device.

Challenge: Dynamic configurations Upon user mobility, the configurations could be updated within an encrypted message after the device connects to a new gNB. It is also possible that gNB updates the C-RNTI for the device upon RRC state changes without user mobility. Tracking the up-to-date configurations for the target device is critical.

Can we track the config change? One solution idea is to track the configuration change once it happens and launch the inference again. This is possible when a handover happens. `CellDAM` develops an application on the target device to track the possible configuration changes due to mobility. The application leverages the existing API to detect the PCI/band change due to handover. It requires no root access. The application could track the updates with OS-level API and notify the `SecHub` to start a new round of C-RNTI inference.

However, the same method cannot be used to infer the config change within the same cell. `SecHub` or OS APIs cannot report the change of configurations from the base station.

Idea: Prevent config change in a cell Since change detection is hard, we approach it differently by keeping the configuration constant. Indeed, this is possible. `CellDAM` leverages the operational C-RNTI update logic in the cellular deployment to retain the same C-RNTI when the device stays on the same gNB. Our study on commercial devices and operators shows that, the current gNB updates C-RNTI when the device transits from the RRC-Idle state to the RRC-Connected state. The gNB recycles the C-RNTI from the idle devices and reuses them for other devices. Therefore, we aim to keep the device in RRC-Connected to avoid configuration change.

Preventing configuration change for continuous inference We trigger a lightweight ping in the background inside the application. Our experiments show that the ping traffic with 2s interval could keep the C-RNTI unchanged. We validate it on 216 cells from three major US carriers. In all tests, the C-RNTI remains unchanged for at least 30 minutes with our light background traffic. To tolerate unexpected updates, `SecHub` also triggers the C-RNTI inference (in §6) every 10 minutes to validate that the current configuration is up-to-date.

Marginal energy consumption The ping messages incur low traffic volume to keep the device in RRC-Connected. We further note that, this has small impact on 5G energy saving. This is because, the device could still go to sleep mode for energy reservation in logical RRC-Connected state. A 5G device saves power by entering Discontinuous Reception or DRX OFF mode. However, given the device will quickly re-enter DRX OFF (within 100ms) after a data transmission, the infrequent messages (every 2s) would only incur small energy overhead. Besides, regular user traffic will also wake up the device, during which the extra ping does not further reduce sleep period. In addition, the impact on the energy from the sleep mode has a smaller impact as compared to other factors, such as the screen brightness. We run the application for 30 minutes while normally using the device (with mixed heavy and infrequent messages). The additional energy only incurs 0.5% extra energy on average.

E ACRONYMS

5GC 5G Core Network
AKA Authentication and Key Agreement
BSR Buffer Status Report
C-RNTI Cell Radio Network Temporary Identifier
CE Control Element
COTS Commercial Off-the-shelf
DCI Downlink Control Information
DFA Deterministic Finite Automata
DL Downlink
DRX Discontinuous Reception
DTLS Datagram Transport Layer Security
FSM Finite State Machine
gNB gNodeB, 5G Base Station
HARQ Hybrid Automatic Repeat Request
LTE Long Term Evolution
MAC Medium Access Control
NAS Non Access Stratum
NSA Non Standalone
PCI Physical Cell ID
PDCCH Physical Downlink Control Channel
PDCP Packet Data Convergence Protocol
PDSCH Physical Downlink Shared Channel
PHY Physical Layer
PUCCH Physical Uplink Control Channel
PUSCH Physical Uplink Shared Channel
RLC Radio Link Control
RRC Radio Resource Control
RSRP Reference Signals Received Power
RSRQ Reference Signal Received Quality
SN Sequence Number
SR Scheduling Request
TLS Transport Layer Security
UE User Equipment
UL Uplink

Table 5: List of the accepted data-plane signaling for each DFA state and their state transition. We do not include s_5 and s_9 in the table, as they are accept states. $\times$ means that the message in this state is not allowed and cannot pass c_1, $-$ means that the state is unchanged with this message. "Different" or "same" means the receiving DCI compared with the first DCI for an RLC data packet, i.e., RLC retransmission will reset HARQ and NDI with the first DCI after.

Data-Plane Signaling Message	Current State						
	s_1	s_2	s_3	s_4	s_6	s_7	s_8
MAC DCI for DL grant with same HARQ, same NDI	$-$	$-$	$-$	$-$	s_7	$\times$	$\times$
MAC DCI for DL grant with same HARQ, flipped NDI	$-$	$-$	$-$	$-$	$\times$	$\times$	$-$
MAC DCI for DL grant with different HARQ	$-$	$-$	$-$	$-$	$-$	$-$	$-$
MAC DCI for UL grant with same HARQ, same NDI	s_2	$\times$	s_2	$\times$	$-$	$-$	$-$
MAC DCI for UL grant with same HARQ, flipped NDI	$\times$	$\times$	s_4	$\times$	$-$	$-$	$-$
MAC DCI for UL grant with different HARQ	$-$	$-$	$-$	$-$	$-$	$-$	$-$
PUCCH ACK for the previous DCI	$-$	$-$	$-$	$-$	$\times$	s_8	$\times$
PUCCH NACK for the previous DCI	$-$	$-$	$-$	$-$	$\times$	s_6	$\times$
RLC Control with ACK for this packet	$\times$	$\times$	$\times$	s_5	$\times$	$\times$	s_9
RLC Control with NACK for this packet	$\times$	s_1	s_1	$\times$	s_6	s_6	$\times$
DRX Command	$-$	$\times$	$\times$	$\times$	$-$	$\times$	$\times$
BSR	$-$	$-$	$-$	$-$	$-$	$-$	$-$
Any other messages	$-$	$-$	$-$	$-$	$-$	$-$	$-$