CUPS Hijacking in Mobile RAN Slicing: Modeling, Prototyping, and Analysis

CUPS Hijacking in Mobile RAN Slicing: Modeling, Prototyping, and Analysis

Rupendra Nath Mitra[1], Mohamed M. Kassem[2], Jon Larrea[1], and Mahesh K. Marina[1]
The University of Edinburgh[1], Email: {rupen.mitra, S2004865, mahesh}@ed.ac.uk
University Of Surrey[2], Email: m.kassem@surrey.ac.uk

***Abstract*—In emerging mobile networks, control and user plane separation (CUPS) plays a critical role in scaling the control-plane and user-plane functions independently and enables network virtualization through network slicing. However, a CUPS hijacking attack on a mobile network slicing system and the resulting network performance degradation is yet to be studied.**

In this work, we investigate the consequences of CUPS hijacking of a radio access network (RAN) slicing system on the overall network performances. We *quantify* the impacts of CUPS hijacking by designing an Impact Factor metric *I*, *prototype* a real-world RAN slicing use case on an end-to-end mobile network test-bed, and systematically *analyze* the empirical results to reveal the impacts of CUPS hijacking on the network performance. We show a successful CUPS hijacking by a rogue slice owner in a RAN slicing system increases the RAN slice control-plane signalling delay above 2ms, the operational upper-bound of our system, to disrupt the control plane operations by injecting low rate DoS (LDoS) traffic in user-plane. The naive hijacking can degrade throughput performances of the rogue slice as well as a co-located victim slice down to 0 Mbps. We further show that a carefully crafted user-plane traffic by the attacker can regain $\sim$92% of its original user-plane packet delivery success rate while other slices are under the denial of service.

***Index Terms*—5G security, secure slicing, RAN slicing, CUPS hijacking, DoS**

I. Introduction

In modern mobile networks, control and user plane separation (CUPS) refers to the complete separation between *control-plane functions* such as user authentication, connection management, etc., and *user-plane functions* such as user data traffic forwarding [1]. The key advantage of CUPS is that it enables the mobile operators to scale the user plane (UP) and control plane (UP) independent of each other. CUPS is an integral part of virtualized mobile networks aim to support diverse services through network slicing. However, the CUPS mechanism can be vulnerable to cyberattacks like CUPS hijacking, where the adversary disrupts the CP communications from the UP, obscuring the logical separation between the two planes.

Although CUPS hijacking attack is well-studied in cloud-computing context, the impact of the CUPS hijacking on the mobile network performance is underexamined. For the first time, this paper aims to empirically show the effects of a CUPS hijacking attack on the mobile network performance. In order to demonstrate the feasibility of CUPS hijacking and study its effects on the network performance we develop a threat model, present a quantitative modelling of a RAN slicing system, and perform experiments on a mobile network testbed.

CUPS mechanism is an integral part of today's LTE and 5G mobile networks. CUPS-based network architecture was borrowed from the Software-Defined Networking (SDN) and was introduced to the telecom paradigm through the 3GPP Release-14 for the LTE core networks, and later it was adopted in the 5G service-based core network architecture by 3GPP Release-15 [2] [3].

Beyond the service-based core network functionalities, mobile network virtualization that is achieved by the network slicing technique is another key application area where CUPS plays a critical role. Network slices are independent, logically separated, virtualized set of network resources exclusively orchestrated to cater to different performance requirements of network latency, security, and throughput for various vertical industries such as eHealth, automotive, smart factories, etc. An end-to-end mobile network slicing is achieved by combining the RAN slicing, transport network slicing, and core network slicing. CUPS plays a pivotal role in a network slicing system as it allows a fully customizable CP for each slice so that a slice owner (often a mobile operator) can tailor the slice on the fly according to the service requirements (i.e., Quality of Services (QoS)).

Despite the paramount importance and timeliness of the CUPS technique in emerging mobile networking, studies on CUPS hijacking in mobile networks from a security perspective are lacking in the literature because of at least the two following challenges *(i)* most of the real-world end-to-end mobile network slicing techniques are closed-sourced and/or running on commercial networks that are not accessible to the research community; *(ii)* setting up a realistic mobile network running with modern CUPS mechanism and slicing techniques to conduct empirical studies on CUPS hijacking often demands enormous engineering efforts, human-hours, and sophisticated hardware and network configurations.

In this paper, we attempt to change the status quo through a study on CUPS hijacking on mobile network slicing systems and its impacts on network performance. To be reasonably detailed in our study, we restricted the scope of this work only to the CUPS hijacking of a RAN slicing system instead of an end-to-end slicing system. We consider a real-world scenario of a RAN slicing system running on a typical neutral host and multi-operator indoor small cell (NHMO) as a use case (see Fig. 1). We also set up OpenAirInterface[1] driven end-to-end mobile network testbed built on comercially available off-the-shelf (COTS) computing devices and peripherals in a controlled lab environment. Finally, we replicate the NHMO setup on the

[1]https://www.openairinterface.org/

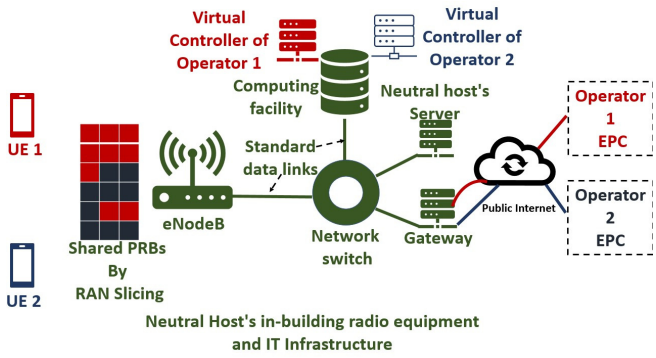

Fig. 1. A schematic diagram of a typical neutral host and multi operator small indoor cell use-case where the host's in-building radio equipment and IT infrastructures are shared between the operators through RAN slicing [4]. Here a two operator scenario is illustrated.

testbed to perform a thorough empirical analysis of the CUPS hijacking on the mobile network performance.

In particular, our ***key contributions*** are threefold:

(i) we **model** the CP behavior of an SDN-based RAN slicing system and quantify the impact of CUPS hijacking on the network performances in terms of a novel 'Impact Factor *I*' metric;

(ii) we **prototype** a real-world use case of NHMO infrastructure, where RAN slicing plays an integral role, on the OpenAirInterface based end-to-end mobile network running on a lab test-bed to empirically show that CUPS hijacking is feasible in a mobile RAN slicing system under reasonable assumptions;

(ii) we **analyze** the empirical results that reveal the following three-way impacts of a CUPS hijacking attack on the overall mobile network performance: *(a) cross-plane impact:* CUPS hijacking increases the RAN slice control-plane signalling delay above 2ms, the operational upper-bound of our system, to completely disrupt the control plane operations by injecting low rate denial of service (LDoS) traffic in user-plane; *(b) cross-slice impact:* CUPS hijacking degrades the throughput performance of a co-located victim slice down to 0 Mbps; *(c) in-slice impact:* a naive hijacking may completely diminish the throughput of the adversary slice itself. From the metric formulation process, we further infer and demonstrate experimentally that a carefully crafted user-plane traffic by the adversary can regain $\sim$92% of its original user-plane packet delivery success rate while keeping other slices under the denial of service. Finally, we show that the impact factor metric *I* as modeled in section III is effective in quantifying the effects of CUPS hijacking by showing high correlations between *I* and network performance metrics such as throughput and latency.

II. Background

A. RAN Slicing Primer

As the mobile network is all set to advance from the LTE/4G network which is essentially a "best-effort" network to 5G, a flexible, service-oriented mobile network, the notion of a "one size fits all" network becomes unsuitable. To host multiple services with diverse quality of service (QoS) requirements, it is required to segregate the physical infrastructure into multiple virtual networks called slices. An end-to-end network slicing envisages on-demand virtualization of all three major network segments of a mobile network: the core network, the transport network, and the RAN. Thus, RAN slicing is essentially a network virtualization technique to enable dynamic allocation of radio resources and management of virtual RANs [5]–[8]. Each virtualized RAN created on top of common radio hardware and network resources can be individually customized to meet different levels of QoS requirements for different slices' service level agreements (SLAs).

Orion [9] is a state-of-the-art RAN slicing system that relies on decoupled CP from its UP, thereby complying with the SDN approach of network virtualization. From a system perspective, a hypervisor is the core component of a RAN slicing system. The hypervisor acts as the middleman between the CP and the UP of the system and is responsible for allocating radio resources, more precisely physical resource blocks (PRBs), to the slices as well as represent the available PRBs as virtualized resources to the controller. Ideally, a RAN slicing system is expected to provide a twofold assurance such as *functional isolation* between co-located slices so that no slice can affect other slices, and *separation between CP and UP functionalities* so that control-plane functions cannot be disrupted by the user-plane traffic flows and vice versa.

B. A RAN slicing use case: NHMO

In this section, we introduce NHMO, which is a highly practical use case for RAN slicing systems. The conventional RAN currently deployed worldwide assumes that the outdoor macro-cells cater "well" to the in-building consumers. However, in reality, as many as 43% of the mobile subscribers face regular coverage issues inside their offices. The exponential growth in the demand for indoor mobile data usage and adoption of the higher frequency spectrum in RAN motivate a high capacity addition to in-building mobile infrastructures. There is a wide consensus around the idea of *neutral-host* operator to achieve a faster indoor small-cells deployment. The central concept is the property owner (of commercial buildings such as urban shopping mall or offices) builds and manages the indoor radio access network, local computing, and IT facilities as a part of smart building infrastructures and offers it to multiple mobile operators to come and share for a fee [10] [11].

Fig. 1 depicts a schematic diagram of a typical NHMO setup where the radio equipment and the associated IT infrastructure (data communication links, typically Ethernet links, network switch, computing node, server (not shared, exclusive to the neutral-host), and gateway) are owned and managed by the neutral-host. Operators can rent the infrastructure, the radio resource in terms of Physical Resource Blocks (PRBs) from the host and run their *controller* to manage their virtual RANs. The controller communicates with the eNodeB (eNB) through the slicing system's CP. A similar concept called *"Bring your own controller"* is already proposed in infrastructure-as-a-service

(IaaS) clouds context to enable enterprise-level tenant sharing more flexible and affordable [12].

Sharing network resources and computing facilities has evolved as a successful business model in the IT industry in past few years. Mobile networks are going to experience a similar evolution through merging a majority of their network functions running on general-purpose IT infrastructure and sharing the physical infrastructures at a scale for the first time with the 5G deployment. For example, core network implementation in a public cloud facility is successfully achieved paving the way for further progress toward a merged IT and telecom infrastructure [13]. Techniques like SDN that ensures CUPS, and Network Function Virtualization (NFV) that offers easy network reconfiguration on the fly, have been adopted in designing mobile networks. Thus a significant fraction of the emerging mobile network becomes deployable on the general-purpose data network and IT infrastructures achieving unprecedented acceleration in network implementation, reconfiguration, and sharability among multiple operators.

However, sharing infrastructures among the close competitors with limited physical isolation opens up unforeseen security threat surfaces with significant concerns to the stakeholders. Moreover, the merging of telecom networks with IT networks makes the telecom network vulnerable to the attack vectors, such as a LDoS, which originally targets IT infrastructures.

In this paper, we choose the NHMO as a real-world use case to implement it in a lab testbed environment to conduct the CUPS hijacking attack on it because CUPS-based RAN slicing is an integral part of the NHMO, and the use case is applicable in a dense-urban commercial indoor small-cell which is realistically replicable in a lab environment. We consider the study is quite timely and motivating for further research to identify vulnerabilities in modern mobile network design, recognize potential drawbacks in operational practices by the mobile operators and vendors, and investigate the impact of unattended security loopholes on the stakeholders.

III. System Modeling

In this section, we first provide a detailed interpretation of the working principle of a state-of-the-art SDN-based RAN slicing system, Orion, deployed in the NHMO setting as depicted in Fig. 1. Then, we quantify the impact of hijacking by introducing the impact factor metric *I*. We evaluate the usefulness of *I* in revealing the severity of CUPS hijacking on the network performance in section VI.

A. A RAN Slicing System Model

An SDN-based RAN slicing system, as introduced in section II, seeks to provide strict isolation guarantee among the slices while enables an efficient network resource sharing by the mobile operators. The hypervisor that lies on top of the physical layer is the key component of a slicing system like Orion that we deploy on the testbed to prototype the infrastructure for the small-cell indoor NHMO use case. The hypervisor joins the isolated RAN slices to the PRBs and the shared physical infrastructure (i.e. eNB), by offering a virtual abstraction of the underlying PRBs and the UP states and accordingly updating any state changes in the physical UP by mapping virtual to physical resources. The hypervisor allocates the PRBs among slices after virtualizing them. In the NHMO setting, the hypervisor is part of the network infrastructure provider's software suite that facilitates RAN slicing.

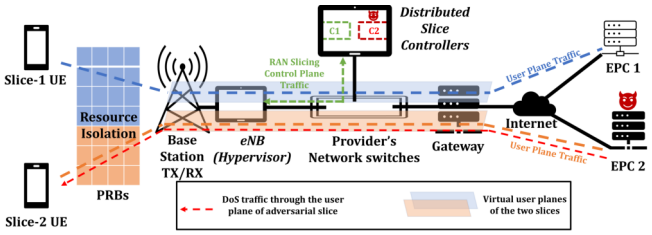

Fig. 2. Virtual user-planes and the control-plane traffic-flows through the shared network links of an NHMO infrastructure considered as a RAN slicing use case in the study.

Fig. 2 shows the two RAN slices with user-planes and the control-plane traffic flows through the network switches, the eNB (where the hypervisor is running), the network server systems (where two RAN slicing controllers are running), and the gateway through which the user-plane traffic flows to and from the EPCs (evolved packet core networks) of the two operators. In our experimental setup on the test-bed, as described in section V, we considered a similar RAN slicing architecture with two RAN slices running on the NHMO infrastructure.

In our system, the mobile operators (e.g., mobile virtual network operators (MVNOs) or business verticals) can realize their RAN slices by instantiating virtual eNBs on top of the hypervisor. For each of the virtual eNB, a virtual control plane is created to manage the user-plane state which is virtually exposed to the eNB by the hypervisor. The virtual control plane of a RAN slice is essentially the RAN slice controller running separately on a remote computing facility (as shown in Fig. 2) and is capable to tailor the functionality as if the slice was operating on its own dedicated infrastructure.

The slice controller communicates with the hypervisor in a near real-time through logically independent communication channels. Any significant time delay introduced in the channel has critical implications on the performance of the hypervisor because it can delay the resource allocation which is expected to happen in near real-time following the traffic demand of the slice.

B. Formulation of Impact Factor

In an NHMO network, there are ***N*** RAN slicing controllers each owned by a mobile operator. A single controller may tailor slice for a distinct QoS requirement. For instance, in Fig. 2, we have a set of two, (***N***=2), distributed virtual RAN slicing controllers, C_i, where i $\in \{1,..., N\}$. Each virtual RAN slicing controller has a control-plane signalling delay, δ_i, that is the latency control signals experience through the network links in between the hypervisor and the controller C_i.

To define the average controllers' response time (CRT), we consider the cumulative control-plane request traffic $\Sigma\phi$, that

flows in between the hypervisor and a RAN slicing controller. The average CRT of the *i*th slice controller is denoted by $\sigma_{ci}(t)$, where i $\in \{1, ..., N\}$. Let's assume the *i*th slice controller's maximum capacity of processing requests from the hypervisor at a time is $\gamma_{i,max}$ and the cumulative request traffic flows to a particular controller is $\Sigma\phi$. We model the distributed controllers as M/M/1, and assume that the flow requests obey the Poisson distribution. Therefore, the average CRT $\sigma_{ci}(t)$ of C_i, given its maximum capacity $\gamma_{i,max}$ and load in terms of the cumulative flows $\Sigma\phi$, can be defined using the Little's theory [14] as follows

$$\sigma_{ci}(t) = \frac{1}{\gamma_{i,max} - \Sigma\phi} \tag{1}$$

On the other hand, the average control-plane signalling delay Δ_{avg} in between the hypervisor and the set of ***N*** distributed controllers, can be formulated as

$$\Delta_{avg} = \frac{1}{N}\sum_{i=1}^{N}\delta_i \tag{2}$$

In an ideal RAN slicing system, average CRT σ_{ci} is negligible because, usually in a small-cell setup, the maximum serving capacity $\gamma_{i,max}$ is designed larger than the peak cumulative flows of requests $\Sigma\phi$. However, for a distributed deployment of the RAN slice controllers over a complex network may incur wide-spread control-plane signalling delays, δ_i, that in turn, can give rise to the average control-plane signalling delay Δ_{avg}.

In our setup two controllers are co-located to each other are deployed away from the hypervisor (eNB). We empirically found Δ_{avg} is 0.3 ms in the experimental setup during normal operation (no attack) of the network. We also empirically validate the claim made in [9] that if the control-plane signalling delay δ_i of a controller C_i becomes 2ms or more, the Orion hypervisor and the controllers fall out of sync impacting the network services drastically. We denote this upper limit of the the control-plane signalling delay as δ_{break}.

From the above formulation, we can infer if the user-plane traffic flow can introduce a delay of more than δ_{break} in the control-plane flows then the control-plane functions of the slicing systems are disrupted leading to a CUPS hijacking that violates the CP and UP separation.

We define Control Plane Functional Window (CPFW) of a controller C_i as the total time taken by a control-plane packet to arrive in the controller from the hypervisor, to be processed by the controller, and the reply from the controller to reach to the hypervisor. Then from equation (1) and (2), CPFW (under no attack) can be defined as follows

$$CPFW = \delta_i + \sigma_{ci} \tag{3}$$

where δ_i denotes the control-plane signalling delay between the hypervisor and the controller C_i. In our setup the two controllers are co-located to each other and placed at equal distance from the hypervisor. Hence, Δ_{avg}=δ_i. Clearly, CPFW (under attack) can be defined as $CPFW_{under\_attack} = \delta_{i,under\_attack} + \sigma_{ci,under\_attack}$. The maximum value of the CPFW of a controller C_i, $CPFW_{max}$, represents a special case of $CPFW_{under\_attack}$. $CPFW_{max}$, is the target benchmark to achieve for successfully launch a CUPS hijacking attack. When defining $CPFW_{max}$, we assume that σ_{ci} as a low-magnitude constant for an NHMO setup because of the design consideration of the high capacity $\gamma_{i,max}$. In this particular work, we assume that $\gamma_{i,max}$ cannot be compromised by an attacker, albeit, that might not be the case in a different threat model than ours which is explained in the next section. Thus, the upper bound of CPFW, $CPFW_{max}$, can be defined in terms δ_{break} as follows

$$CPFW_{max} = \delta_{break} + \sigma_{ci,under\_attack} \tag{4}$$

From equation (3) and (4), we define the impact factor of the CUPS hijacking on a RAN slicing controller C_i, I_{ci}, as a function of CPFW as follows

$$I_{ci} = \frac{CPFW_{Under\_Attack}}{CPFW_{No\_Attack}} \tag{5}$$

Since, in our setup $\sigma_{ci} \rightarrow 0$, I_{ci}=($\delta_{i,Under\_Attack}$/δ_{i,No_Attack}). Similarly, $I_{ci,max}$= (δ_{break}/δ_{i,No_Attack}). As mentioned, we empirically find $\delta_{break} = 2ms$ and $\delta_{i,No\_Attack} = 0.3ms$, in our setup $I_{ci,max} = 6.66$. When calculating the value of *I*, if found ($\delta_{i,Under\_Attack} \geq \delta_{break}$, then ($\delta_{i,Under\_Attack}$ can be replaced by δ_{break} because, beyond δ_{break} the slicing system control plane is impacted the most and lost the CP and UP separations denoting the upper bound of the impacts. It is a good idea to represent the impact factor *I* in a normalized form. In this work, we use the widely adopted min-max normalization method and scaled the values of *I* so that $I \in [0, 1]$.

CPFW and *I* provide important insights into CUPS hijacking. For instance, CUPS hijacking can be achieved when $\delta_i \rightarrow \delta_{break}$. On the other hand, if σ_{ci} is sufficiently large then the control-plane functionality of C_i would also be disrupted. However, in a mobile network, σ_{ci} is sufficiently low to cater to the peak volume of the control request.

The formulation of *CPFW* and *I* helps us to infer that, CUPS hijacking of RAN slicing system can be possible by at least two possible ways (i) by $\delta_i \rightarrow \delta_{break}$, for instance a possible way to achieve this is to inject large volume of DoS traffic in user plane to sufficiently congest the shared physical link so that the control-plane traffic suffers high delay (greater than δ_{break}); (ii) by affecting the σ_{ci}, for instance, a possible way to achieve this is a side-channel attack from a co-located adversary slice controller that generates impersonated control requests in a sufficiently larger volume than the victim controller is designed for to handle at a time, i.e., $\gamma_{i,max}$. However, in order to demonstrate CUPS hijacking, in this work, we do not consider a side channel attack that compromises the capacity, $\gamma_{i,max}$, but a DoS attack that elevates $\delta_i \rightarrow \delta_{break}$ satisfying the pre-condition of CUPS hijacking. In the following

section IV, we design an attack model to achieve the pre-condition of CUPS hijacking.

A non-obvious insight is drawn from the CPFW is that an adversarial controller, C_k can achieve an intelligent CUPS hijacking if it can keep its own $\delta_k < \delta_{break}$ but can sufficiently increase the δ_is of the rest of the *(N-1)* slice controllers beyond the δ_{break}. In section VI, we leverage this insight to improve the primary threat model and in section VI we empirically validate this notion to achieve the CUPS hijacking.

We also elaborate on the impact factor in section VI. We show that the impact factor is correlated with the two key network performance metrics - throughput and latency, proving *I* as a suitable metric to measure the severity of the CUPS hijacking on the network performance.

IV. Threat Model and CUPS Hijacking Attack

Based on our modeling, we conjecture that under certain conditions the RAN slicing system becomes vulnerable to CUPS hijacking attack. To test our hypothesis we carry out the empirical study of CUPS hijacking attack in an SDN-based RAN slicing system by designing the following threat model with a reasonable set of assumptions.

Attack Scenario: We propose that in an NHMO setting, an operator with a legit tenancy can turn himself into an adversary and launch a CUPS hijacking attack by increasing the control-plane signalling delay through injecting malicious traffic in the user plane. Precisely, the adversary exploits the lack of physical isolation in the SDN-based RAN slicing system's CP and UP traffic flows to disrupt the network performance.

In the context of NHMO, RAN slicing is an integral part of the mobile network implementation enabling adaptive sharing of the common infrastructure among multiple operators. Due to shared physical links among operators and between CP and UP traffics, an abrupt increase of a slice's UP traffic has impacts on the CP traffic of the same slice and the other slices as well. An adversary slice controller can craft its CP traffic that causes cross-plane, cross-slice, and in-slice impacts because of the shared physical links. Thus an adversary can compromise the logical separation between control and user plane as modeled in the section III.

In the experimental setup of this work, we consider the neutral host as having two operators sharing the radio equipment and the neutral-host's in-building IT infrastructure (Fig. 2). The shared physical data links, switches, and the gateway cater to both operators.

Goal of the Attacker: The goals of the attacker, a malicious operator among the mobile operators, are threefold: (i) Fingerprint the shared links and have an estimate of how much UP traffic has a significant impact in control-plane signalling delay, δ_i, in between the eNB/hypervisor and the controller. (ii) Launch an LDoS attack through the UP to bring network functionalities under complete disruption. (iii) Keep its own traffic flowing while putting other slices under the denial of service through CUPS hijacking.

Challenges for the Attacker: The adversary aims to launch an LDoS to choke the shared links in the small cell data network. Thus it needs to estimate how much attack traffic can choke the link capacity. Therefore, the attacker needs to (1) learn the shared link capacity (2) since a continuous DoS kills traffic of the adversary slice as well, the adversary needs to let its own traffic flowing while hindering others. Our chosen method of attack addresses both the challenges to successfully launch a CUPS hijacking attack exploiting the vulnerability that emerges from the multi-operator infrastructure sharing.

Assumptions: We assume that the attacker is a legal tenant in the small cell NHMO setup and can control its RAN slice through its controller. The controllers of the tenant mobile operators are deployed in a distributed computing facility connected to the hypervisor (eNB) through shared communication links. However, we do NOT assume that the attacker has compromised or has privileged root access to any of the neutral host's servers or equipment. The threat model also assumes the CP and UP traffics share common communication links in a distributed deployment of SDN-based RAN slicing systems on NHMO infrastructure which are reasonably realistic assumptions in the context of the NHMO use case.

Attack Approach: The attacker achieves the goals by the following three-step attack approach. The attacker first tries to estimate an approximate capacity of the shared physical links (especially the link between the eNB and the slice-controllers) by injecting LDoS flows with an increasing rate in its user plane until it observes disruptive control-plane signalling delay. Once the attacker learns the shared link capacity, it launches the LDoS attack with the learned data rate to completely choke the network bandwidth. The attacker injects LDoS traffic in its UP to disrupt the CP functionalities of the overall RAN slicing system by leveraging the condition of CUPS hijacking modeled in section III. Finally, the attacker carefully controls the attack traffic flow and its legit user traffic flow to regain its network performance while keeping other slices under a denial of service as pointed out during the formulation of the impact factor *I*, in section III.

While designing an LDoS flow the attacker can customize its burst duration, burst magnitude, and the inter-burst gap period. We choose the burst duration as 200 ms, inter-burst period as 300 ms. and the maximum rate found 100 Mbps that completely chokes the link in our set up. In a commercial deployment, the maximum speed may go higher than 1Gbps, given high-speed Ethernet connections between the small cell network nodes.

(i) Link capacity estimation with increasing rate of LDoS: In order to estimate the share link capacity, the attacker injects streams of an increasing rate of LDoS traffic in its user plane (in our experiment we used 10 Mbps, 30Mbps, 60Mbps, and 100Mbps) each of duration of 2 seconds in the network and measures its control-plane signalling delay. To get a finer estimation of the link capacity, the attacker can choose an off-peak time of the day when all the user traffic is expected to be very low or negligible. In an urban commercial smart building, it can be midnight. With increasing LDoS traffic through the UP, the RAN slicing CP packets experience an elevated delay due to shared links between CP and UP traffics. At a certain

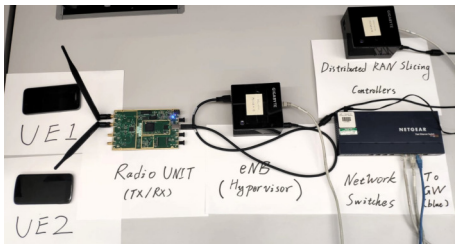

Fig. 3. A photograph of the RAN segment of the lab testbed.

LDoS rate the attacker observes that its controller has fallen out of sync with the hypervisor ($\delta_i \geq \delta_{break}$) as discussed in section III. The attacker may estimate this LDoS rate (in our case it is 100Mbps) as the link capacity and prepare for aggressive CUPS hijacking.

(ii) CUPS hijacking with constant rate LDoS: The attacker now launches the LDoS flow through its user plane at the learned rate roughly equals the shared link capacity to completely disrupt the CP functionality. However, the increased CP signalling delay, as well as the congested shared user planes, lead to a denial of services for all the RAN slices including the attacker's own.

(iii) Regaining attacker's slice-performance under LDoS: The constant rate LDoS attack kills the UP and CP traffics of all the slices indiscriminately. So, the attacker should craft an intelligent UP traffic flow scheme to regain the network performance for itself but still keeping other slices under the DoS attack. As outlined in section III, we try to realize the notion of an intelligent CUPS hijacking such that the adversary keeps its own $\delta_k < \delta_{break}$ but sufficiently elevate the δ_i of the rest of the ***(N-1)*** slice controllers above δ_{break}. To achieve the intelligent CUPS hijacking, the attacker now sends its legit user traffic only during the interburst gap period of the LDoS injection and holds it off during the burst duration.

A CUPS hijacking on the RAN slicing system may be achieved by a completely different or more robust threat models, but the above described threat model sufficiently achieves the objective of enabling understanding of the three-way impacts of a CUPS hijacking attack on a RAN slicing system by allowing to empirically answer the following set of research questions (RQs):

RQ1. How much time does the adversary need to launch the CUPS hijacking attack?

RQ2. What are the impacts of a successful CUPS hijacking launched by a malicious slice owner on the network performance?

RQ3. How effective the attack is in achieving the adversary's objective of retaining its own traffic intact while diminishing traffic from other co-located slices?

We analyze the empirical results in section VI to answer the above RQs. Thus, we demonstrate the viability of CUPS hijacking by quantitative modelling, threat model design, and performing experiments on a mobile network testbed.

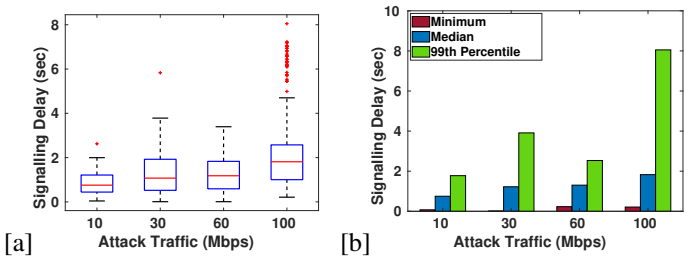

Fig. 4. ***Cross-plane impact:*** [a] Control-plane signalling delay, δ_i, experienced by the adversary RAN slicing controller during link capacity estimation using LDoS with increasing rate [b] 99% of the control-signalling packets experience 8 seconds of delay under 100Mbps attack traffic leading to a CUPS hijacking because the controllers fall out-of-sync with the hypervisor.

V. Testbed Implementation

The testbed consists of two GIGABYTE Intel-based small form factor PCs (i7-4770R CPU @ 3.20GHz, 4GB of RAM) running Ubuntu 18.04 with a low-latency kernel. The two PCs are running the RAN slicing system, Orion, and its components (hypervisor and slice controllers). The core part of the network is deployed as virtual machines (VMs) on an additional Intel-based machine (i5-3230M CPU @ 2.60GHz, 16GB of RAM). We leverage OpenAirInterface open source EPC implementation where two mobile network operators are deployed on two different virtual machines. The two VMs are running Ubuntu 16.08 with Linux 4.7.7 kernel optimized for real-time operation i.e., disabled CPU C-states, low-latency Linux kernel, and with disabled CPU frequency scaling. For the front-end radio unit, we use Ettus USRP B210 Software-Defined Radio (SDR) board equipped with two omnidirectional 2.45GHz antennas. We use Samsung Galaxy Note 4 and Huawei E3372 LTE dongle as UEs. We set the default bandwidth of the 5MHz spectrum to be shared between the two slices in LTE band 7. The Orion base station is configured to use single input single output (SISO) transmission mode, which for 5MHz spectrum can provide up to a maximum throughput of 16Mbps. Fig. 3 shows a real picture of the RAN segment of the testbed implementation of the NHMO use case in the lab environment.

VI. Results

In this section we report the experimental results and make an attempt to answer the **RQs**, framed at the end of section IV, by analyzing the empirical evidences.

RQ1: Exploitation time by the attacker. We demonstrate with a limited number of attempts an adversary can guess the bandwidth of the shared network links and subsequently launch the LDoS attack leading to CUPS hijacking. We used 100 Mbps ethernet cable as the shared link capacity that an attacker can roughly estimate in a period of 8 seconds in our threat model. Given the fact that prediction of a 100Mbps link capacity is achieved under 10 seconds, the capacity for higher speed links (1 Gbps or higher) within a few minutes.

RQ2: Impacts of CUPS hijacking. We present a comprehensive ***threefold impacts*** of a CUPS hijacking attack on the network behavior as follows.

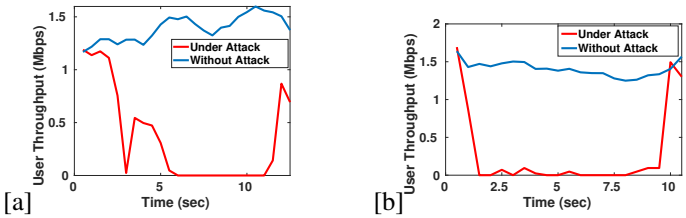

Fig. 5. ***Cross-slice impact:*** [a] User throughput of the UE attached to the co-located victim slice under increasing rate of LDoS and under no attack. [b] User throughput of the UE attached to the co-located victim slice under 100 Mbps LDoS and under no attack.

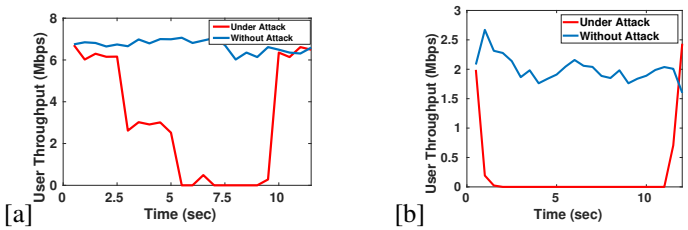

Fig. 6. ***In-slice impact:*** [a] User throughput of the UE attached to the adversary slice under increasing rate of LDoS and under no attack. [b] User throughput of the UE attached to the adversary slice under 100Mbps LDoS and under no attack.

Cross-plane impact: Fig. 4[a] shows that data traffic in the user-plane can affect the control-plane signalling delay, δ_i, resulting in a complete disruption of the slicing system's control operations when the condition, $\delta_i \rightarrow \delta_{break}$, is met. The attack introduces a delay in the RAN slice CP, above the operational upper-bound of our system δ_{break}=2ms, to disrupt the control plane operations by injecting LDoS traffic in the UP. Fig. 4[b], shows that 99% of the control-signal packets experience $\sim$8 seconds (far greater than $\delta_{break} = 2ms$) of delay under 100Mbps LDoS leading the RAN slicing controllers fall out-of-sync with the hypervisor. This empirical evidence proves the feasibility of CUPS hijacking as modeled in section III-B.

Cross-slice impact: Isolation between any two RAN slices is a critical condition of a robust deployment of a slicing system so that no adversary can impact cross-slice network performance and encroach on virtualized network resources that is not allocated to it [15]. However Fig. 5[a] shows, a UE attached to the co-located victim slice starts experiencing poor user-throughput performances with an increasing rate of LDoS injection by the adversary in the user plane of the adversary slice. In this study, the LDoS lasts for 8 seconds and the LDoS rate increases from 10Mbps to 30 Mbps, to 60Mbps, and to 100Mbps in every 2 seconds. As shown in Fig. 5[b], under a 8 second-long continuous 100Mbps (the estimated maximum link capacity) LDoS injection form the adversary in its user plane, the same UE experiences a near-zero user-throughput during the entire attack period. Figs. 5[a] and 5[b] prove that not only cross-plane but also the cross-slice functional isolation that a RAN slicing system guarantees can be voided by a successful CUPS hijacking.

However, the characteristics of gradual degradation perceived by the UE of a co-located slice can be used for an early detection of presence of an adversary in the pool of controllers.

in-slice impact: As shown in Fig. 6[a], the user throughput of a UE attached to the adversary slice gradually decreases with time as the shared communication link accumulates LDoS traffic. The network performance significantly aggravates when the attacker injects 100Mbps LDoS through its UP leading to the CUPS hijacking. Fig. 6[b] shows the user throughput of the same UE, completely diminished under a continuous 100Mbps LDoS injection. Figs. 6[a] and 6[b] show the throughput performance degradation perceived by a UE attached to the adversary slice under the increasing rate of LDoS and a continuous 100Mbps LDoS.

From Figs. 5 and 6 we infer that a naive CUPS hijacking attack has *similar diminishing impacts* on the network throughput in the both co-located and the adversary slices motivating the adversary to regain its own slice performance.

RQ3. Attacker's Gain: The attacker can now leverage the insights learned in the section III to exclusively regain the performance of its own slice. To achieve the efficient CUPS hijacking, as proposed in the threat model in section IV, the attacker sends its legit user traffic only during the inter burst gap period of the LDoS injection and holds it off during the burst duration. As the Fig. 7[a] shows, instead of sending the user traffic under the influence of LDoS in a naive manner, if the adversary carefully crafts the user traffic to be sent only during the inter-burst gap period of the LDoS, it can regain his user-plane packet delivery success ratio upto $\sim$92% from 0% under a CUPS hijacking. We name this scheme as intelligent DoS, reflecting an efficient CUPS hijacking.

Impact on the user-plane latency: Fig. 8[a] shows the UP round-trip time (RTT) latencies percieved by a UE attached to the network under the attack and under no attack. The x-axis is the RTT in seconds computed from the TCP acknowledgments in the Wireshark traces, and the y-axis is the cumulative distribution of the RTT. We observe that under no attack, the RTT of a UE always remains below 0.25 seconds. However, under the CUPS hijacking attack the RTT exceeds 0.25 seconds with a high probability of $\sim$75%.

Impact on UE attachment procedure: We performed another experiment to investigate if the CUPS hijacking has any impact on mobile network control plane and we found that under the 100Mbps LDoS injection a UE could not attach to the network. The UE, in our set up, keep sending attach request to the EPC every 10.611 seconds. On the other hand, under no attack a UE can get connected to the network on an average within 0.63 second.

eNB port utilization: Fig. 7[b] elucidates the network port utilization of the eNB, under attack and without attack. Although, under attack the eNB port is under-utilized still the user experiences zero throughput due to the TCP retransmission time out (RTO) mechanism that further delays the packet sending because of too many acknowledgements have been missed due to the ongoing network congestion.

Correlation between *I* and network performances: Section III-B introduces the impact factor metric *I*, to quantify the severity of the CUPS hijacking attack. Fig. 8[b] and 8[c]

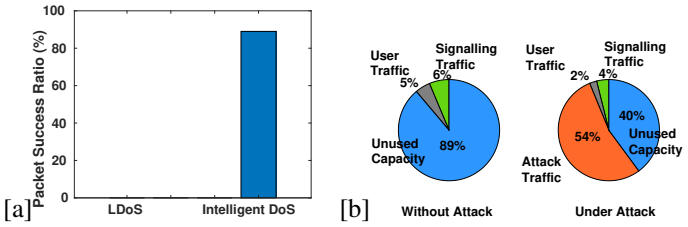

Fig. 7. [a] User packet transmission success ratios under naive LDoS and intelligent DoS [b] Port utilization of the eNB without attack and under 100Mbps LDoS

show the correlations between the impact factor I and the UP throughput and RTT of a UE attached to the network. We scaled the parameters by the min-max normalization to the interval of [0,1]. Each point in the plot represents experiments under different rate of LDoS injection. We see a positive correlation between the impact factor and the network RTT latency and a negative correlation between the impact factor and the user throughput, that proves the impact factor I is a pertinent metric to quantify the severity of a CUPS hijacking attack.

VII. Discussion

Although our work empirically investigates the effects of CUPS hijacking attack on the performances of LTE/4G mobile network equipped with an SDN-based RAN slicing systems in the context of NHMO use case, the research findings have implications on the emerging 5G network design implementations. For instance, the notion of split eNB protocol stacks of ORAN[2] supports shared links in between the Distributed Units (DU) and Centralized Unit (CU) CP and UP through F1-c and F1-u interfaces respectively, hence cross-plane impacts of a possible CUPS hijacking remains relevant in the 5G RAN sharing in O-RAN context [16] if implemented defying the security best practices.

Limitations: Despite our arduous attempt to model an SDN-based RAN slicing system and analyze the impacts of a CUPS hijacking attack on it, the work in its present form bears certain limitations due to constraints stemming from issues such as lack of access to commercial-grade mobile networks, the codebase of proprietary RAN slicing systems, and real network datasets. For instance, the results presented in this work are obtained from a prototyped mobile network and can only be taken as indicative to infer the performance of a commercial NHMO network under a similar set of assumptions rather than an exact replication. Moreover, our threat model presented in this paper exclusively targets SDN-based RAN slicing systems running on shared network infrastructure assuming no traffic shaping in action with no physical separations between CP and UP traffic flows. The threat model might not be extendable for RAN slicing systems designed with significant deviations from SDN architecture and deployed in a setting where physical isolation is insured. However, other slicing systems in the literature, like [17], are reported with a similar SDN-based architecture to that of Orion where CUPS plays a crucial role. Impacts of a CUPS hijacking on such SDN-based RAN slicing systems deployed on shared physical infrastructures can gain indicative insights from the results presented in this paper. Finally, this work focuses on RAN slicing system, and not end-to-end mobile network slicing system, the latter is a natural target for a follow-on work.

Traffic isolation techniques as countermeasures: In the commercial space, operators can consider leveraging tagged VLAN and VPNs to ensure strict isolation between CP and UP [18], [19]. For the transport network, soft-isolation through tunnelling is necessary on top of physical isolation to ensure QoS among various services. However, our demonstration of CUPS hijacking shows, in a shared IT infrastructure, where the data communication links are having limited bandwidth capacity, soft-isolation alone not be able to mitigate the impacts of the CUPS hijacking unless a VPN with strict bandwidth policing is adopted.

VIII. Related Work

SDN Security: SDN in the IT industry has matured and witnessed a wide spectrum of innovative attacks which now become relevant to the mobile networks since significant network functions of mobile networks now run on the IT infrastructure. Specifically, in the context of CUPS,Thimmaraju et al. show the virtual switches widely used in SDN cloud networks are vulnerable and authors were able to exploit the vulnerability to take control of the network via the user plane [20]. Cao et al. show how a low rate DoS attack in the user plane can disrupt the control-plane functions leaving the whole SDN paralyzed [21].

Mobile network security: Mobile network security, in general, is increasingly getting attention from the security research community after several exploitable vulnerabilities are reported in all existing mobile network standards. Basin et al. report security weaknesses in the 5G authentication and key agreement protocol and provides provable fixes [22]. Rupprecht et al. recently propose a cross-layer clone attack enabling an adversary to perform a full impersonation of the phone or the network on the user plane by exploiting the missing integrity protection on the user plane of LTE standard [23]. Li et al. successfully exploit security vulnerabilities in the voice-over LTE protocol to disrupt both data and voice [24]. However, a security vulnerability in a commercial mobile network does not always emerge due to weak standards but also due to malpractices by the operators or due to flawed network design in a practical implementation as reported in [25].

There is a research void in the area of empirical security analysis of emerging mobile networks. For instance, CUPS hijacking, side-channel attacks, (in)security of open-source software stacks on potentially untrusted hardware are a few areas need research attentions. Our work is a step towards filling the gap in the literature and secure slicing for emerging 5G networks.

[2]https://www.o-ran.org/

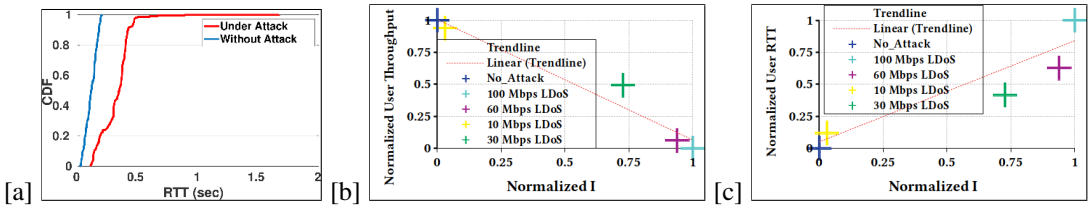

Fig. 8. [a] Distributions of the user-plane latencies (RTT) during attack and under no attack. The x-axis is the round-trip time in seconds. The CUPS hijacking increases the UP latency. [b] and [c] Correlate the normalized impact factor I with the normalized user throughput [b] and normalized user RTT [c]. Different colors represent experiments with different rates of LDoS traffic. We see a negative (positive) correlation between I and user throughput (user RTT) that proves the impact factor I, is a useful metric to quantify the network performance degradation due to CUPS hijacking.

IX. Conclusions and Future Work

In this work for the first time, we empirically study the CUPS hijacking on the mobile RAN slicing system. Our work provides two key insights: (i) *need for traffic isolation:* without physical isolation or strict bandwidth restricted soft-isolation (such as VPN service with constricted traffic regulations) between CP and UP traffic and among the UP traffics of co-located slices, an SDN-based RAN slicing system running on shared infrastructure becomes vulnerable to CUPS hijacking attacks; (ii) *feasibility of CUPS hijacking on RAN slicing:* Although RAN slicing system guarantees CUPS and inter-slice functional isolation, under the CUPS hijacking attack, both promises are voided with serious implications on network performance.

A general principle of CUPS hijacking is known in cloud-computing systems with shared infrastructure, however, this work presents interesting findings like failure of UE attachment or eNB network port utilization under CUPS hijacking which are specific to RAN slicing systems. We believe, our results stimulate cybersecurity research community to take on further research endeavors envisioning secure slicing in 5G and impact RAN slicing design considerations when deployed on shared infrastructures. The results of this study drive us to initiate follow-up research in the direction of securing the emerging RAN solutions such as O-RAN and serverless RAN architecture by designing innovative and robust threat models. Our work highlights the need for systematic studies of security vulnerabilities in modern mobile network deployment on public clouds and multi-operator scenarios through designing sophisticated threat models involving side-channel attacks and CUPS hijacking to ensure a robust and trustworthy mobile network architecture that is rapidly merging with IT infrastructures.

References

[1] Cisco, "Control Plane and User Plane Separation (CUPS)," Tech. Rep., 10 2018, retrieved May, 2021 from https://tinyurl.com/c2x488xn.

[2] "3GPP;TSG Services and System Aspects; Rel 14 Description; (Release 14)," 3GPP, TS, 06 2017. [Online]. Available: https://www.3gpp.org/release-14

[3] "3GPP; TSG Services and System Aspects; Rel 15 Description; (Release 15)," 3GPP, TS, 09 2019. [Online]. Available: https://www.3gpp.org/release-15

[4] K. Mun, "Making Neutral Host a Reality with OnGo," Tech. Rep., 12 2018.

[5] X. Costa-Pérez, J. Swetina, T. Guo, R. Mahindra, and S. Rangarajan, "Radio access network virtualization for future mobile carrier networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 27–35, 2013.

[6] T. Frisanco, P. Tafertshofer, P. Lurin, and R. Ang, "Infrastructure sharing and shared operations for mobile network operators from a deployment and operations view," in *NOMS 2008-2008 IEEE Network Operations and Management Symposium*. IEEE, 2008, pp. 129–136.

[7] K. Samdanis, X. Costa-Perez, and V. Sciancalepore, "From network sharing to multi-tenancy: The 5G network slice broker," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 32–39, 2016.

[8] L. Zhao, M. Li, Y. Zaki, A. Timm-Giel, and C. Görg, "LTE virtualization: From theoretical gain to practical solution," in *2011 23rd International Teletraffic Congress (ITC)*. IEEE, 2011, pp. 71–78.

[9] X. Foukas, M. K. Marina, and K. Kontovasilis, "Orion: RAN slicing for a flexible and cost-effective multi-service mobile network architecture," in *Proceedings of the 23rd annual international conference on mobile computing and networking*, 2017, pp. 127–140.

[10] James Lapham, "Smal Cell: Neutral Hosting' is it the Future," 2016, retrieved June, 2020 from https://tinyurl.com/yawfvtgr.

[11] *Dense Air*, http://denseair.net/.

[12] H. Wang, A. Srivastava, L. Xu, S. Hong, and G. Gu, "Bring your own controller: Enabling tenant-defined SDN apps in IaaS clouds," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[13] B. Nguyen, T. Zhang, B. Radunovic, R. Stutsman, T. Karagiannis, J. Kocur, and J. Van der Merwe, "Echo: A reliable distributed cellular core network for hyper-scale public clouds," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018, pp. 163–178.

[14] H. Wang, H. Xu, L. Huang, J. Wang, and X. Yang, "Load-balancing routing in software defined networks with multiple controllers," *Computer Networks*, vol. 141, pp. 82–91, 2018.

[15] N. Alliance, "5G security recommendations package# 2: Network slicing," *Ngmn*, pp. 1–12, 2016.

[16] O-RAN Alliance, "O-RAN Use Cases and Deployment Scenarios ," 2020, retrieved June, 2021 from https://www.o-ran.org/resources.

[17] G. Garcia-Aviles, M. Gramaglia, P. Serrano, and A. Banchs, "Posens: A practical open source solution for end-to-end network slicing," *IEEE Wireless Communications*, vol. 25, no. 5, pp. 30–37, 2018.

[18] Cisco, "Slicing the transport network for 5G," 2018, retrieved June, 2020 from https://bit.ly/3eN6rEn.

[19] ZTE, "5G Security White Paper Security Makes 5G Go Further," 2019, retrieved June, 2020 from https://tinyurl.com/yy55sxsj.

[20] K. Thimmaraju, B. Shastry, T. Fiebig, F. Hetzelt, J.-P. Seifert, A. Feldmann, and S. Schmid, "Taking control of SDN-based cloud systems via the data plane," in *Proceedings of the Symposium on SDN Research*, 2018, pp. 1–15.

[21] J. Cao, Q. Li, R. Xie, K. Sun, G. Gu, M. Xu, and Y. Yang, "The crosspath attack: Disrupting the SDN control channel via shared links," in *28th NDSS*, 2019, pp. 19–36.

[22] D. Basin, J. Dreier, L. Hirschi, S. Radomirovic, R. Sasse, and V. Stettler, "A formal analysis of 5G authentication," in *Proceedings of the 2018 ACM SIGSAC CCS*, 2018, pp. 1383–1396.

[23] D. Rupprecht, K. Kohls, T. Holz, and C. Pöpper, "Imp4gt: Impersonation attacks in 4g networks," in *NDSS*, 2020.

[24] C.-Y. Li, G.-H. Tu, C. Peng, Z. Yuan, Y. Li, S. Lu, and X. Wang, "Insecurity of voice solution VoLTE in LTE mobile networks," in *Proceedings of the 22nd ACM SIGSAC CCS*, 2015, pp. 316–327.

[25] M. Chlosta, D. Rupprecht, T. Holz, and C. Pöpper, "LTE security disabled: misconfiguration in commercial networks," in *Proceedings of the 12th ACM WiSec*, 2019, pp. 261–266.