UE Security Reloaded: Developing a 5G Standalone User-Side Security Testing Framework

Evangelos Bitsikas
bitsikas.e@northeastern.edu
Northeastern University

Syed Khandker
syed.khandker@nyu.edu
New York University Abu Dhabi

Ahmad Salous
a.salous@nyu.edu
New York University Abu Dhabi

Aanjhan Ranganathan
aanjhan@northeastern.edu
Northeastern University

Roger Piqueras Jover*
rogerpiqueras@google.com
Google

Christina Pöpper
christina.poepper@nyu.edu
New York University Abu Dhabi

ABSTRACT

Security flaws and vulnerabilities in cellular networks lead to severe security threats given the data-plane services that are involved, from calls to messaging and Internet access. While the 5G Standalone (SA) system is currently being deployed worldwide, practical security testing of User Equipment (UE) has only been conducted and reported publicly for 4G/LTE and earlier network generations. In this paper, we develop and present the first open-source based security testing framework for 5G SA User Equipment. To that end, we modify the functionality of open-source suites (Open5GS and srsRAN) and develop a broad set of test cases for the 5G NAS and RRC layers. We apply our testing framework in a proof-of-concept manner to 5G SA mobile phones and provide detailed insights from our experiments. While being a framework in development, the results of our experiments presented in this paper can assist other researchers in the field and have the potential to improve 5G SA security.

CCS CONCEPTS

• **Security and privacy** → **Mobile and wireless security**.

KEYWORDS

5G, Security Testing, User Equipment, srsRAN, Open5GS

ACM Reference Format:

Evangelos Bitsikas, Syed Khandker, Ahmad Salous, Aanjhan Ranganathan, Roger Piqueras Jover, and Christina Pöpper. 2023. UE Security Reloaded: Developing a 5G Standalone User-Side Security Testing Framework. In *Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '23), May 29-June 1, 2023, Guildford, United Kingdom.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3558482.3590194

*This author did not contribute any code for this project.

WiSec '23, May 29-June 1, 2023, Guildford, United Kingdom

1 INTRODUCTION

The adoption of 5G technology [31] has rapidly increased due to its significantly faster data speeds, lower latency, improved capacity, and the growing support for a wide range of new applications, ultimately leading to the milestone of 1 billion connections in 2022 [14]. 5G Standalone (SA) systems, i. e., networks built from scratch using the full 5G architecture and protocols without relying on existing 4G infrastructure, are increasingly deployed [14], along with the availability of 5G mobile equipment that supports the new radio technology. 5G's security mechanisms have been improved over previous generations, incorporating stronger authentication (5G-AKA), better privacy mechanisms (encrypted SUCI), and separate virtual networks (network slicing), each with their own security controls and policies. However, it can be challenging to determine the level of security achieved in real-world deployments solely from the specifications, and security considerations as well as potential directions for further improving 5G security are intensely discussed [5, 11, 21].

The criticality of mitigating design and implementation vulnerabilities and the need to investigate new attack vectors potentially introduced by new 5G applications motivate the need for a security testing framework specifically designed for 5G SA devices. Existing security testing frameworks [9, 16, 18, 25, 27] and fuzzing-based approaches [13, 26] primarily cater for the 4G/LTE ecosystem and do not account for changes in the control plane since 4G/LTE in terms of NAS (Non-Access Stratum) and RRC (Radio Resource Control) messages, their structure, and parameters. With the aim of obtaining insights into the security levels of 5G in real-world deployments and enabling comparisons, we designed and developed the 5G SA security testing framework presented in this paper.

Developing such a framework poses a number of challenges: i) Open-source software that would support such tests (e. g., Open5GS, srsRAN) has just evolved, is in flux, and exhibits practical issues with connecting 5G phones reliably (details in Section 6.1), ii) the software cannot be used out of the box and needs to be substantially modified, iii) comprehensive test cases for the 5G SA context do not exist yet, and iv) automation, rather than manual analysis, is preferred for evaluating a greater number of system components in terms of security.

In this paper, we take the first steps towards building such a security testing framework designed for 5G SA devices. The developed

framework is based on several key objectives, including modularity, practicality, and automation, at evolving levels of realization, in order to enable an experienced user to create the desired test cases rather than relying solely on predefined tests from specifications. In addition, we take into account the testing of 5G-capable devices by looking for implementation flaws that contradict design rules.

In short, our main contributions in this paper are:

(1) We develop the first 5G SA security testing framework for User Equipment that is built upon the open-source projects Open5GS and srsRAN. To extend these software suites to provide our desired testing functionality, we had to develop and implement our own version of handling uplink and downlink (UP/DL) traffic.
(2) We provide a broad set of 5G test cases for NAS and RRC layers for uplink and downlink traffic that we derived from specification documents and related work, consisting of 82 individual test cases. We also selected unique test cases for 5G, especially in terms of parameter violation.
(3) We apply our testing framework in a proof-of-concept manner to 5G SA mobile phones and report on our success in establishing 5G connection and testing with the existing software suites. We went through intense learning curves in setting up experimental tests in the new 5G SA context and report in detail on lessons learned.
(4) We have explored several known potential design flaws on 5G Standalone that could lead to Denial-of-Service (DoS), attachments to rogue stations, and downgrades.

Code Release. To support further research and investigations on this topic, we make the code and configurations available on GitHub at https://github.com/vaggelis-sudo/5G-UE-SecurityTesting. The framework should continue to evolve and we encourage further 5G security testing from the research community.

2 BACKGROUND AND RELATED WORK

2.1 5G Security Architecture

The 3GPP standard [2, 3] describes the security architecture of 5G networks. Figure 1 illustrates the steps that User Equipment (UE) devices follow to establish secure communication with the core network. First, the UE receives broadcast messages from the network (Step 1) in the form of a Master Information Block (MIB) and System Information Block (SIB). The UE then establishes a Radio Resource Control (RRC) connection with the gNodeB (Step 2). Next, the UE performs the Authentication and Key Agreement (AKA) procedure with the Core Network, which involves several steps. The UE communicates with the Access & Mobility Management Function (AMF)/Security Anchor Function (SEAF) (Step 3a), the AMF communicates with the Authentication Server Function (AUSF) (Step 3b), and the AUSF communicates with the Unified Data Management (UDM) (Step 3c). The AKA's main purpose is to validate the subscriber credentials, provide mutual authentication, and agree upon the keys for control- and user-plane traffic. After successful authentication, the Non-Access Stratum (NAS) and Access Stratum (AS) Security Contexts are enabled with the Core Network and the gNodeB, respectively (Steps 4 and 5). Finally, the UE establishes sessions and communicates securely.

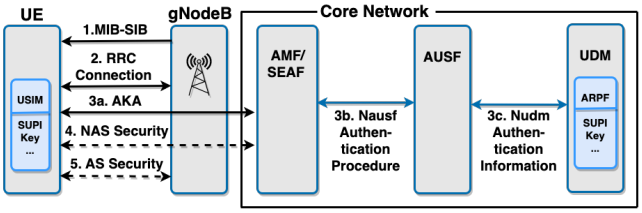

Figure 1: Security Architecture.

The **UE** includes the Universal Subscriber Identity Module (USIM) to store the necessary credentials for authentication and security establishment of the traffic. The USIM's material allows the UE to calculate and send the response as a challenge to the AMF/SEAF during the AKA.

UDM is a database containing the subscriber credentials (e. g., profile information, authentication information, and encryption keys). Specifically, the Authentication credential Repository and Processing Function (ARPF) within the UDM, which is identical to what the USIM contains, keeps the authentication credentials, used during the AKA.

AUSF has access to the UDM and retrieves the UE-related authentication material (i. e., SUPI, RAND, AUTN, and XRES*). It performs the calculation of the serving network challenge which is sent to the AMF/SEAF along with the AUTN and RAND, while it stores the obtained home network challenge. In AKA's final step, it validates the UE's response value with the stored challenge, and if successful, grants the AMF/SEAF the anchor key (i. e., K_{seaf}) for the security context.

AMF/SEAF is responsible for transmitting the authentication values (i. e., AUTN, RAND, ngKSI, and ABBA) to the UE through the Authentication Request. Once the UE provides its response to the challenge, its hashed value is generated and compared with the serving network's challenge. If they coincide, the procedure is successful and the response is sent to the AUSF for the home network validation. The AMF is the primary entity of control-plane communication with the UE even after the AKA procedure.

2.2 NAS & RRC Structures

NAS (Non-Access Stratum) and RRC (Radio Resource Control) messages play a crucial role in the control-plane traffic of cellular communication. NAS messages are used for high-level signaling between the UE and the network, while RRC messages are used for low-level radio resource control between the UE and the base station. Together, NAS and RRC messages form the control-plane traffic of cellular communication and are crucial for establishing and maintaining the communication sessions between the UE and the network. Both types of messages have a well-defined structure.

According to 3GPP [1], NAS messages can either be plain or security protected. The type and functionality of the message determine the format. A *plain message* consists of the Extended Protocol Discriminator, Security Header Type, Procedure Transaction Identity, Message Type, and other Information Elements (IE). A *protected message* comprises the Extended Protocol Discriminator, Security

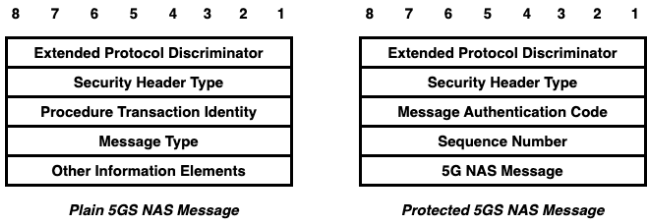

Figure 2: Message Organization Example for Plain and Protected 5G NAS Messages, based on the specifications [1].

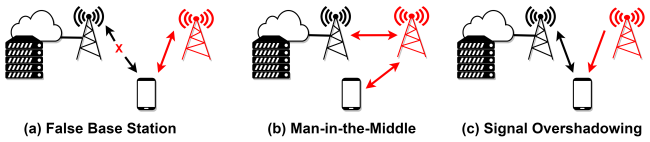

Figure 3: Typical 5G Attack Scenarios: an adversary may execute FBS, MitM, and Signal Overshadowing attacks.

Header Type, Message Authentication Code (MAC), Sequence Number (SN), and the 5GS NAS message. Figure 2 illustrates the NAS structures. Furthermore, the uplink and downlink NAS messages are protected (i. e., ciphered and integrity protected) once the EPS/NAS security context exists, containing the MAC, the Security Header Type, and SN.

RRC messages follow a similar format by incorporating the message type field, various IEs, and potential protection [2]. RRC messages, which are Signalling Radio Bearer (SRB) type 1-4 are integrity protected and ciphered by Packet Data Convergence Protocol (PDCP) once the AS security context is activated, including those carrying NAS messages. Nonetheless, SRB0 messages are transmitted without protection.

2.3 State-of-art Security Testing Frameworks

In recent years, researchers have focused on uncovering potential attacks and threats within the 5G ecosystem [6, 7, 10, 17, 18, 30]. They have investigated security issues related to User Equipment capabilities and paging procedures [17, 30] in both 4G and 5G networks, as well as demonstrated the feasibility of targeted 5G SUCI catchers [10], which pose a significant threat to user privacy. Additionally, attacks on the 5G handover [6] and 5G warning/emergency systems have been demonstrated [7].

Despite these efforts, a comprehensive and efficient approach to UE security testing with broad test cases for the 5G Standalone system has not yet been developed. While Hussain et al. created the 5GReasoner [18] framework for formal verification of the 5G control plane, it lacks practical applicability for commercial off-the-shelf (COTS) UEs and exploration of implementation flaws. Similarly, AutoFuzz [13] focuses on malformed and out-of-order packets, while Berserker [26] only applies ASN.1-based fuzzing to srsLTE and OpenLTE setups.

In contrast, LTE has seen numerous works concentrated on creating UE testing frameworks using various techniques at different levels. For example, protocol verification [20] leverages a semantic model as a finite-state machine to verify the tested properties, and LTE-Inspector [16] uses a symbolic model checker and a cryptographic protocol verifier to identify design flaws and improper practises. Non-compliance checker [19], on the other hand, adopts a property-agnostic and black-box approach for control-plane testing against implementations in COTS UEs. Natural language processing and machine-learning techniques have also been explored [8] to scan a large number of specifications and generate test cases, hence further enhancing an LTE testing environment.

Other works focus on implementation flaws within devices, such as those developed by Rupprecht et al. [27] which target the discovery of LTE implementation flaws in security functions like data encryption and network authentication. UE dynamic testing [12, 22] has also been used to promote automation and uncover vulnerabilities in the LTE control plane that cannot be discovered with protocol verification alone. Tools like DoLTEst [25] improve the detection of implementation flaws by concentrating on negative testing with a deterministic oracle derived from specification analysis. BaseSAFE [24] and FIRMWIRE [15] use fuzzing against LTE firmware to discover vulnerabilities, such as buffer overflows, which can then be verified through over-the-air testing.

2.4 5G UE Attack Scenarios

In the cellular environment, there are three major categories of active attacks that an adversary can perform (Figure 3): (a) False Base Station (FBS), (b) Man-in-the-Middle (MitM), and (c) Signal Overshadowing. FBS attacks involve maliciously attaching a UE to the attacker's rogue base station for a limited duration, during which the attacker attempts to interact with the user. However, the attacker does not have the cryptographic keys to establish a full connection with the victim and instead relies on the pre-authentication traffic, as well as on the unprotected RRC and NAS messages to compromise the victim. A more powerful attacker leverages the MitM approach between the UE and the legitimate network. This not only allows the adversary to capture and control the traffic even after the AKA, but also modify it in certain cases [28, 29]. Finally, signal overshadowing [32] specifically exploits the physical layer to inject malicious sub-frames stealthily, without the need of malicious attachments.

Various works [6, 7, 10, 23] have already demonstrated all three attacks and have even included them in the threat model, e. g., DoLTEst [25]. We follow the same convention in this work. Although our intention is not to perform attacks, our testing framework and experimental setup use equipment and setups that attackers may use to conduct attacks. Specifically, we use a 5G Core and RAN network software along with a transmission device (i. e., USRP) to send and modify NAS and RRC messages, as well as capture responses from the testing device (i. e., UE).

3 TESTING FRAMEWORK DESIGN

In this section, we clarify the goals for our framework, describe its design, and explain our evaluation process.

Figure 4: Testing Flow. Our testing approach focuses on modifying downlink messages towards the UE (Step 3).

3.1 Goals and Overall Design

Goals. The objective of our testing framework is to provide a comprehensive methodology for security testing of 5G Standalone-capable devices. In terms of security assessment, it assists the user on 1) determining if the 5G implementations have been improved since LTE/3G/2G, meaning whether vulnerabilities have been addressed on 5G, and 2) potentially identifying new vulnerabilities in the implementations and design on 5G. Moreover, it should allow the user to design and select a number of desired test cases, which can be subsequently executed to retrieve responses from an out-of-the-box smartphone device. The 5G core network permits the use of 5G NAS compared to the LTE core, thus testing and evaluation of new features and changes that occurred in the NAS protocol for messages, parameters, and values. This applies to the 5G RAN for the RRC protocol as well. Without the 5G SA components, testing of the 5G-related firmware/modem in user equipment for design or implementation flaws is unattainable.

We prioritize a modular approach, usability, and repeatability, and aim to eliminate the need for limited and hard-coded modifications in open-source software. Our goal is to assess and report a device's security posture using a security evaluation framework, with the possibility to adopt a scoring system. We *do not aim* to perform fuzzing, but rather to conduct design and implementation testing against selected security features on UEs from various manufacturers. Any newly discovered vulnerabilities can be added as a test case to the database, enabling users to reproduce them.

Framework Design. We design a framework that allows users to send 5G control-plane messages, i. e., NAS and RRC messages, to a COTS UE and record the responses for security evaluation. The user can alter the normal execution flow, and modify messages including the contained parameters. Therefore, by creating security-related test cases we can perform an extensive security evaluation of a COTS UE's behavior, based on which we can identify possible security flaws and deficiencies, as well as non-compliance with the 3GPP 5G specifications.

Testing Procedure. Figure 4 illustrates our testing procedure. A computer with a radio transmitter (USRP) loads a test case from our database and sets up the custom 5G network (Step 1). The smartphone (UE) being tested discovers and interacts with this network after (re)-initiating the cellular service (e. g., after toggling airplane mode or rebooting). Depending on the current test case, the UE sends a specific uplink message (Step 2), which triggers a change in the execution flow, and a corresponding downlink message is sent to the UE (Step 3). The UE's potential response to the downlink message is recorded (Step 4). Unresponsiveness is also registered, which, in fact, can be an indication of proper vs. erroneous behavior. This response is then evaluated for correctness (Step 5). We use the 3GPP 5G specifications as primary reference for correct behavior and to identify potential security issues in the device. By verifying the device's configuration and adherence to the specifications, we determine if its responses are appropriate or pose any security risks.

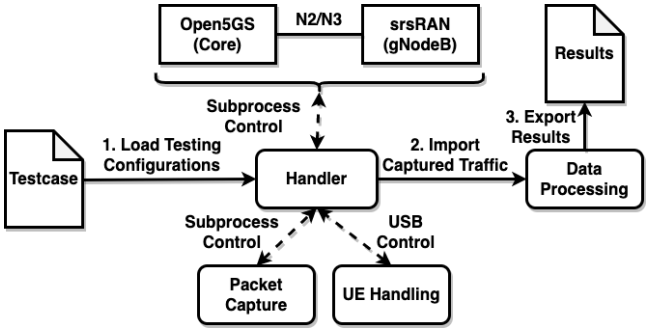

Figure 5: System Components.

3.2 Framework Components

The framework comprises multiple components for executing the testing procedure and analyzing the results, as shown in Figure 5 and described below.

5G Core & RAN. We use two software suites, Open5GS[1] and srsRAN[2] as Core Network component and gNodeB (radio access network base station), respectively. We modify Open5GS and srsRAN (as will be explained in Section 3.3) and combine them to import test cases from a user initializing them accordingly. The two software suites are configured to communicate through the typical N2 and N3 5G interfaces, which correspond to the gNodeB with AMF and with UPF connections, respectively. The Core and RAN are deployed to operate within the same computer device and configured as a testing Public Land Mobile Network (PLMN). Each UE is supplied with a programmable (or commercial) SIM card that enables the connection to the 5G SA network. Finally, the network is controlled using sub-processes of the operating system.

UE Handling. The tested smartphone is controlled by the same computer device via USB and Android Debug Bridge (ADB). This allows the framework to manage the device's access to the custom network and retrieve baseband logs if necessary. Although the debugging mode is important for ADB before the start of the testing, UE rooting is not mandatory. Nevertheless, rooting could permit access to restricted commands and files that might be useful during testing and evaluation. A potential manual interaction with the device is restricted to the refreshment after each test case, which can be achieved by various methods, e. g., rebooting or airplane-mode toggling. Otherwise, this process is automated.

Execution Flow Structure. Our automated execution of the security testing is handled by a Python script called the *Handler*. The Handler manages the test cases and ensures that every test case is executed subsequently. Before executing a new test case, the Handler refreshes the network and the UE. This action guarantees that all previous values and configurations relating to previous

[1]https://open5gs.org/
[2]https://www.srslte.com/ and https://github.com/srsran/srsRAN

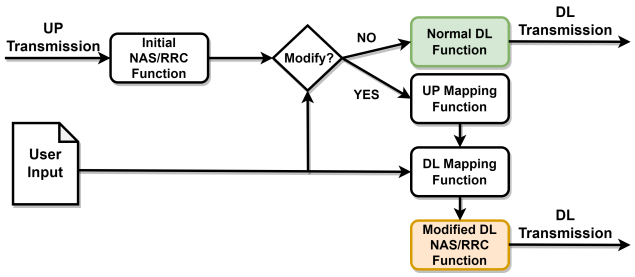

Figure 6: Hooking the Uplink Transmissions.

test cases are released. Furthermore, during the execution of each test case, the script collects the traffic captures and logs from the NAS and the RRC layers by the custom 5G network. The collected traffic is then inspected based on the objectives of each test case (as will be explained in Section 3.4). The data processing eventually determines whether the device passed or failed the test.

3.3 Execution Flow Modifications

In order to control the execution flow of the testing procedure and modify the NAS and RRC messages, we made modifications to the Open5GS NAS and srsRAN 5G RRC layers. These adjustments are categorized into two types:

(1) Uplink-downlink (UP-DL) hijacking, and
(2) Control-plane modification.

For UP-DL hijacking, we utilize hooking on the uplink communication, which corresponds to a UE message destined for the network. Figure 6 depicts our fundamental hooking concept and structure. The uplink transmission with its parameters is received by the normal NAS handler function. However, after our modifications, this function contains our hook which determines if the execution flow must be altered based on user input. If the user test case refers to a specific UP message to be hijacked, the hook transfers the execution to the UP mapping function that handles the message. Alternatively, the hook is dismissed, and the normal operations continue without interruption. If the UP mapping functions take control of the execution, the DL mapping function is leveraged to determine the DL message that the network must send back to the UE. Based on the particular test case, this function collects the necessary data (DL message type, parameters, etc.) and calls the modified NAS or RRC function. Our modified version takes into account the parameters of the user in order to initialize and construct the message accordingly. Once the message is ready, it is transmitted to the UE, and the hooking procedure relinquishes control of the execution flow to restore normal operations.

The control-plane modifications refer to changes implemented in each NAS and RRC function in the concluding part of our hijacked flow, which generates and sends the message. Each modified function allows for the assignment of selected user values in the specified parameters of the message. The construction typically starts with an abstract initialization of parameters in case the user does not require any modifications. If the requested changes are made, the Information Element (IE) values are first assigned, followed by the security header type, which determines whether the message is protected or not, and finally, the security properties (i. e., encryption and integrity protection) are defined. The exact messages that we modified are presented in Section 4.

3.4 Evaluation Process

Security & Specifications. Our security tests determine if a smartphone device has passed or failed. Depending on the test case we can uncover the exact security issue, e. g., message or parameter, that leads to the associated behavior. We evaluate the UE's response to the modified DL message and categorize the outcome in terms of specification compliance and security:

(1) Security violation with specification compliance.
(2) Security violation without specification compliance.

Security violation means that the UE misbehaved, triggered by the modified DL message. This behavior may be included in the specifications as a legitimate response (Design Flaw) or the UE may have deviated from the standard (Implementation Flaw). In addition, the flaw can be exploited by an adversary to launch an over-the-air attack. Our setup and security tests can target both categories of devices, those with and without specification compliance.

We use the captured traffic to determine whether the exchange of messages between the UE and the network results in a security violation. It is important to understand what the specifications state about the specific exchange of messages, and then determine if there is a violation. The Pass-Fail evaluation may eventually allow a taxonomy or classification of the tested devices as well.

UE's Reaction to Modified Messages. The UE behavior can be characterized by three categories based on the UL-DL interaction:

(a) *Response Status.* A UE can respond back with a message or not respond at all, thus (mis)handling the message internally. Our initial objective is to discover whether the UE responds within a reasonable time-frame.
(b) *Specification Compliance.* We explore if the UE's behavior is defined by the specifications, which may explicitly mention the UE's response based on its state and network's DL message, or not specify it at all.
(c) *Security Violation.* The UE's reaction to the DL message is evaluated in terms of security. If a response sets the device in a vulnerable state, an adversary can exploit the involved security flaw to launch an attack. Attacks could target users' privacy, cause DoS, or expose other sensitive information.

For example, if a UE receives a malicious `Security Mode Command` with NIA0 (null integrity protection), it should respond with a `Security Mode Reject` (rejection) for its behavior to be considered compliant and secure. If a UE does not respond, it may have discarded the DL message silently (handled correctly) or encountered an error (handled incorrectly).

Specification Ambiguities. During our evaluation process, we discovered that the specifications do not provide explicit descriptions or guidance for complex test cases. This makes the analysis arduous and obscure. For instance, malformed or erroneous DL messages may force the UE to process the message internally without a response. Given that the 3GPP specifications are a large collection of documents with various procedures being split into different files,

complete comprehension of the specifications may be unattainable. Thus, implementing and deploying a thoroughly automated process to perform security evaluations could be infeasible. In such cases, manual inspection of the captured traffic and device logs is necessary.

4 5G SECURITY TESTING

In this section, we will explain the testing categories and provide details on the format used in our framework. We will also describe our selection of 5G NAS and RRC test cases.

4.1 Message Protection in the Specifications

The 3GPP specifications [1, in 4.4.4.2-3] define the NAS messages that can be sent without (integrity) protection. RRC [2, in Annex B.1] also encompasses multiple types of messages that can be accepted without protection by the UE and AMF, at least before the AS activation. We focus on those messages as a starting point to help us develop the testing categories, and eventually the test cases. Table 1 highlights in italics and red those NAS and RRC messages in our 5G deployment.

We note that `Identity Request` and `Identity Response` are allowed unprotected only for SUCI, and `Service Reject` and `Registration Reject` only when GMM causes no. 76 and 78 are not included. `RRCReconfiguration` and `RRCReconfiguration-Complete` can be sent unprotected as long as they are unrelated to handover procedures and SRB2, SRB4, multi-cast MRB, and DRB establishments. Additionally, `RRCRelease` must not include the *DeprioritizationReq*, *SuspendConfig*, *RedirectedCarrierInfo*, and *CellReselectionPriorities* information fields when unprotected.

4.2 Security Testing Categories

In our framework, we do not leverage brute-forcing, oracles, or machine-learning techniques to generate a large number of tests, as was, e. g., applied for negative testing using invalid or prohibited messages [25]. Instead, we allow the user to evaluate particular security features of interest by creating and using test cases, supporting both design and implementation testing. As a starting point, we have collected 82 test cases for NAS and RRC communication. Our principal goal is to identify test cases where the UE becomes susceptible to active over-the-air attacks, possibly leading to DoS, location tracking, sensitive information extraction, downgrading, and user- or control-plane compromise. We selected the test cases uniquely for 5G SA, targeting new messages and parameters, while emphasizing five security categories. At the same time, we also cover tests that have similarities with previous LTE works [22, 25], but we apply them on the 5G domain.

I. Misuse of Normal Messages. There are messages in normal operations that can be sent unprotected according to the specifications. The concept behind this category is that an adversary can leverage such messages to exploit the victim, even with specification compliance (Design Flaws). For instance, in RRC, we include messages such as `RRCReject` and `RRCRelease`, and in NAS, messages such as `Authentication Reject` and `Registration Reject`. We also test secure versions of messages (i. e., encrypted and integrity protected), e. g., `Deregistration Request`, in order to compare them with potentially insecure versions.

II. Parameter Violations. Violations of parameters intend to force the UE to accept invalid or malicious parameters in specific DL messages. These DL messages diverge from their legitimate versions and may not be congruent with the specifications. In cases where the UE accepts such messages and responds back, critical security exposure may occur. One fundamental example, which was mentioned in Section 3.4, is the use of `Security Mode Command` with NIA0 cipher (null integrity) in 5G in order to compel the smartphone device to accept unprotected messages. By approving this message, the user-plane traffic can get compromised, permitting an attacker to modify messages.

III. Security Header Mismatches. The *security header type* of protected messages can be altered from "Integrity Protected and Ciphered" to "Plain NAS" or to "Integrity Protected and Ciphered with New Context" illegitimately and without compliance. Headers with "Plain NAS" may be used to either identify security errors or force the UE to accept unprotected messages. Consequently, this can benefit an attacker who does not possess the cryptographic keys. Moreover, "Integrity Protected and Ciphered with New Context" type attempts to trick the UE into believing that a new security context has been established. This can be used in NAS-protected messages, such as `Registration Accept` and `Service Accept`.

IV. Wrongly Accepted Messages after Security Enforcement. The specifications define the set of messages that must be protected after the security activation on NAS and RRC (e. g., `GMM Status` and `RRCReestablishment`). The UE is not permitted to accept those messages without protection. As a consequence, in this category, our goal is to determine if there are messages (which may or may not be compliant) that can be accepted by the UE without protection after the security mode activation. We not only design the message to be sent unprotected but also use plain security header wherever applicable in order to look more legitimate.

V. Wrongly Accepted Messages before Security Enforcement. Similar to the previous category, our objective is to identify wrongly accepted messages. However, we target messages that must not be sent without protection before the security activation. For instance, this may include the `Configuration Update Command` in NAS and `RRCResume` in RRC. This test set is important since an adversary could interact with the UE without the need for keys, leading to potential security issues. The design of the message is identical to the aforementioned category.

4.3 Test Case Format

In our framework, we separate the test case format into three authentication-related parts; **pre-AKA**, **AKA**, and **post-AKA**. Pre-AKA refers to the traffic with messages prior to the UE sending the `Registration Request` message such as the `RRCSetup` message. AKA traffic consists of the messages that begin with `Registration Request` and ends with either `Registration Accept` or `Registration Complete`. An example of an AKA message is the `Security`

Table 1: The 5G NAS and RRC messages are the focus of our testing framework. Uplink messages are hijacked to alter the execution flow. Downlink messages are modified according to user input. Messages in *red italic* are those that can be sent unprotected at least before the security activation between the UE and AMF.

NAS Messages		RRC Messages	
Uplink	**Downlink**	**Uplink**	**Downlink**
Registration Request	*Registration Reject*	*RRCSetupRequest*	*RRCRelease*
Registration Complete	*Registration Accept*	RRCReestablishmentRequest	RRCReestablishment
Deregistration Request	Deregistration Request	*RRCSetupComplete*	*RRCSetup*
Service Request	*Service Reject*	Security Mode Complete	Security Mode Command
Security Mode Reject	Service Accept	*UE Capability Information*	*UE Capability Enquiry*
Authentication Response	*Authentication Request*	*RRCReconfigurationComplete*	*RRCReconfiguration*
Authentication Failure	*Authentication Result*	RRCReestablishmentComplete	RRCResume
UL Information Transfer	*Authentication Reject*	*UL Information Transfer*	*RRCReject*
Deregistration Accept	*Deregistration Accept*		CounterCheck
Configuration Update Complete	Configuration Update Command		MobilityFromNR
GMM Status	GMM Status		
Security Mode Complete	Security Mode Command		
Identity Response	*Identity Request*		
Timers			

`Mode Command`. Finally, post-AKA involves messages after the `Registration Accept` or `Registration Complete`, such as the `Configuration Update Command`, where the security context exists.

Each part of the test case defines four parameters: i) *ue_ul_handle* which is the uplink message to be hijacked, ii) *dl_reply* which represents the downlink message to be transmitted to the UE, iii) *command_mode* indicating whether to *send* or *replay* the downlink message, and iv) *dl_params* which are the parameters for the downlink message. A typical example of a test case format is presented in Listing 1 (Appendix) which shows an altered execution flow when `Security Mode Complete` is received.

4.4 NAS & RRC Messages

We utilized the 5G NAS and RRC messages provided by Open5GS and srsRAN in the uplink and downlink as listed in Table 1. These messages are used in the AKA-related parts of the test case format and cover scenarios before and after the AS and NAS activation.

For NAS messages, we modified the execution of the UE-initiated messages in the uplink which are handled by the network. We also hijacked the flow of the timers: t3570, t3560, t3550, t3555, t3513, and t3522, in case the user attempts to deliver a DL NAS message upon their expiration. For the downlink messages, we implemented our modified version which allows the user to alter and/or import parameters and values in them. This includes the parameters and IEs associated with each message, the ciphering and integrity-protection activation or elimination, and the security header type. We have excluded, however, modifications of the message type and extended protocol discriminator (Fig. 2), since they produce malformed messages which the device will discard.

We modify the uplink communications and in the DL RRC message parameterization (belonging to the Dedicated Control Channel and Common Control Channel). However, for similar reasons as above, we exclude the message definition and transaction ID alterations.

5 EXPERIMENTS

We describe our experimental setup and comment on the results and insights we have obtained so far.

5.1 Experimental Setup

Our setup consists of a Lenovo Thinkpad laptop running Ubuntu 20.04 and connected to a USRP B210 to conduct the experiments. The modified versions of the 5G Core Network (i. e., Open5GS) and gNodeB (i. e., srsRAN) run on the laptop, as described in Figure 5. All tests run over the air, but we connected each smartphone to the laptop using USB for monitoring and handling. In addition, the smartphones were equipped with a custom 5G-capable USIM. The USIM was programmed to work with service 124[3] supporting SUCI and all the necessary 5G features. The experiments were conducted in a secluded environment without interfering with legitimate cellular operations. Table 3 (Appendix) shows the 5G capable UEs that we successfully connected to this testing setup.

Network Configurations The network was configured as a testing PLMN (i. e., 00101) with the purpose of demonstrating how the tool operates. For more comprehensive analyses, one should use a non-test/commercial PLMN for accurate security results, as some modems behave differently when they enter into a test mode, triggered by the test PLMN.[4] Furthermore, we operated the 5G core (AMF, SMF, etc.) in localhost along with the gNodeB. The AMF was configured with a Network Slice Selection Assistance Information (NSSAI) equal to 1 (including the Slice/Service type, SST, and Slice Differentiator, SD). Additionally, the Tracking Area Identity (TAI) was set to 1. For the gNodeB, we configured a RAN cell with cell ID, Tracking Area Code (TAC), and Root Sequence Number equal

[3]Service 124 refers to the Short Message Service (SMS) Point-to-Point (PP) service. The USIM includes various service codes used to activate or deactivate specific services. Service 124 is one of these codes and is used to activate/deactivate the SMS PP service.
[4]For tests with SIM cards set with PLMN=01001, modems are likely to go into a test/debug mode potentially altering security operations.

to 1. We used mainly Band 3 for the communication (1710–1785 MHz for the downlink and 1805–1880 MHz for the uplink).

5.2 Results

We report our results by ordering them according to the categories outlined in Section 4.2. Our findings consist of design (Sec. 5.2.1) and implementation flaw explorations thereafter. Table 2 summarizes our experimental results per category and protocol.

5.2.1 Misuse of Normal Messages. In this category, we use legitimate messages and parameters to discover design issues that can lead to attacks. Overall, the 4G/LTE versions of unprotected NAS and RRC messages (Section 4.1) with known vulnerabilities were investigated and confirmed, and remain fairly similar for 5G.

NAS Tests. Our first test cases aimed at verifying that the unprotected messages can still impact UEs on 5G SA. Our experiments show that `Authentication Reject` was maliciously accepted by both devices only when they transmit the `Authentication Request` before the AKA takes place. Similarly, `Service Reject` was effectively used against `Service Request` which occurs even when the NAS security context is available. This message deprived the devices of access to network resources. `Registration Reject` had a more lasting effect as it forced them into a Deregistered state, as a response to the `Registration Request` before the AKA. Additionally, the message allowed us to force the UE to store rejected NSSAI. In summary, Huawei and OnePlus devices failed against the above security tests rendering them susceptible to malicious attachments, DoS, and potentially downgrade attacks. It is important to note that these are known OEM- and OS-agnostic LTE protocol flaws. We validate that they apply to 5G-SA, too.

We have also investigated the use of GMM causes included in the rejection messages. While most of them can be used by an attacker to assist the DoS attempts, e. g., *Illegal UE*, *Tracking Area not allowed*, and *PLMN not allowed*, particularly *Redirection to EPC required* and *5GS services not allowed* showed a tendency for downgrades, affecting both phones. `N1 mode not allowed` affects the N1 interface connection of the UE and AMF, which can lead to 5GMM-NULL state (does not exist on LTE). This disables the 5GS services in the UE for 3GPP access when unprotected, and additionally for non-3GPP access when integrity is protected. We noticed that both devices did not recover the 5G connection after the demonstration.

Finally, both devices revealed their SUPIs (null-scheme) in the *Registration Request*, when service 124 was disabled in the custom SIM card. This indicates that old commercial SIM cards may face compatibility issues when devices are forced to connect to a 5G network. However, further experimentation with more commercial configurations is needed to confirm the issue.

RRC Tests. We selected the messages which are allowed to be sent unprotected by the network (Sec. 4.1) and have an actual impact on the UE. We noticed that `RRCReject` and `RRCRelease` were processed by the devices leading to disconnections. Specifically, `RRCReject` was accepted even after the security establishment. This known protocol flaw, which can lead to DoS and attachments to rogue stations, has not improved since LTE. `UE Capability Enquiry` is normally allowed before the security activation, however specifications [2, Annex B.1] specify that *'The network should retrieve UE capabilities only after AS security activation'* as in LTE [4]. However, our experiments with testing configurations on Huawei and OnePlus showed that an attacker can indeed retrieve the UE capabilities before the security is activated (e. g., after `RRCSetupComplete`), rendering this security measure still redundant on 5G.

5.2.2 Parameter Violations. Our parameter violation testing focused on one parameter at a time, while retaining the rest of the message structure intact. We also explored the GMM causes further.

NAS Tests. For these experiments, we selected various parameters for demonstration: (1) ngKSI, (2) ABBA, (3) AUTN, (4) RAND, (5) Replayed UE Capabilities, (6) Null Ciphers, (7) IMEI, (8) GMM Cause.

The Key Set Identifier for Next Generation Radio Access Network (ngKSI) is used to identify the 5G NAS security context which combines the security parameters for authentication, integrity protection, and ciphering. Since this value is assigned by the AMF, we tested arbitrary and illegitimate values that deviated from normal in our setup and incorporated them in the `Authentication Request` and `Security Mode Command` to determine if they could be accepted. In all cases, the devices responded with a `Authentication Failure` including the "non-5G Authentication Unaccepted" and `Security Mode Reject` including the "Unspecified" before deregistration, thus passing the tests.

The Anti-Bidding-down Between Architectures (ABBA) is a security value enforcing that the UE does not access older mechanisms and according to the specifications, this value is zero. As 5G versions increase and improve later, this value is supposed to define a specific version, and the UE should always check it before completing the registration. Consequently, we tested both phones with non-zero values included in the `Authentication Request` and `Security Mode Command`, which responded with `5GMM Status` "Invalid mandatory information" and `Security Mode Reject` "Unspecified", meaning that they successfully passed the tests.

In the `Security Mode Command`, we experimented with the authentication parameters, AUTN and RAND, attempting to use illegitimate values, such as zeroing out the RAND. Furthermore, we modified the Replayed UE Capabilities to include null ciphers (5G's NEA0 and NIA0, and/or LTE's EIA0 and EEA0) for encryption and integrity protection and specifically changed the `Security Mode Command`'s integrity cipher to null (i. e., NIA0). Both phones passed the test by responding with `Security Mode Reject` and detaching from the network. All the above test cases verified that these specific Huawei and OnePlus devices meet the necessary cryptographic security on 5G standards.

Furthermore, we continued with the IMEI extraction from the UE illegitimately by exploiting the `Identity Request` before the AKA, which can lead to privacy issues and location tracking. According to the specification, only the SUCI is allowed to be transmitted over the air in this message. Both phones passed by not revealing their IMEI in the `Identity Response` and detaching from the network.

Finally, we evaluated the GMM Causes and we discovered that *Not authorized for this CAG or authorized for CAG cells only* was processed (based on `logcat`) by both devices in the `Registration`

Table 2: Categories and results per protocol with a testing PLMN (i. e., 00101) setup.

Security Testing Categories	OnePlus Nord 2 5G	Huawei P40 Pro 5G
Misuse of Normal Messages	NAS: ✗, RRC: ✗	NAS: ✗, RRC: ✗
Parameter Violations	NAS: ✓, RRC: √†	NAS: ✓, RRC: √†
Security Header Mismatches	NAS: ✓, RRC: –	NAS: ✓, RRC: –
Wrongly Accepted Messages After Security Enforcement	NAS: ✓, RRC: –	NAS: ✓, RRC: –
Wrongly Accepted Messages Before Security Enforcement	NAS: ✓, RRC: ✓	NAS: ✓, RRC: ✓

✗= vulnerabilities demonstrated/failed tests, ✓= no vulnerabilities detected/passed tests,
√†= some violation observed/inconclusive tests, – = not tested

`Reject` and forced them to disconnect. Specifications in [1, clause 5.5.1.3.5] state that the message should have been simply discarded when received without integrity protection indicating that either the specifications are obscure in this part or there is an implementation flaw in both devices. Similarly, *N1 mode not allowed* was processed by both smartphones and caused them to deregister completely. For the *N1 mode not allowed* though, specifications require integrity protection for non-3GPP access only. Nonetheless, in order to fully corroborate the above behavior, extra assessment is necessary with commercial configurations.

RRC Tests. For this set of test cases we concentrated on the `Security Mode Command` for null integrity and `RRCRelease`. The null integrity cipher in the `Security Mode Command` was correctly denied by the devices, exactly like the NAS cases indicating that only protected messages (enforced in the PDCP layer) are accepted and handled. For `RRCRelease`, we noticed that the smartphone devices were released from the RRC connection as in normal scenarios, even in the presence of the *CellReselectionPriorities* IE. The violation happens if the reselection priorities are also stored, as with the other fields specified in Section 4.1 which cannot be included in the `RRCRelease` before the security activation. Further, `RRCReconfiguration` which can be transmitted before the security activation, must not carry out handover procedures when unprotected. For all cases, this anomaly needs to be investigated further.

5.2.3 Security Header Mismatches. In this category, we evaluate NAS messages with abnormal security headers. Specifically, we targeted the security-protected messages beginning from the `Security Mode Command` which utilize the *NAS Security Header Integrity Protected and New Security Context* and *NAS Security Header Integrity Protected and Ciphered*. This includes messages such as `GMM Status`, `Registration Accept`, and `Configuration Update Command`. Instead of the normal headers, we substituted them with *NAS Security Header Plain NAS Message*, *NAS Security Header Integrity Protected and New Security Context*, and *NAS Security Header Integrity Protected and Ciphered With New Integrity Context* accordingly.

The goal is to force the smartphone devices to accept the headers leading to potential security deactivation or implications in the already activated security context. Nonetheless, both devices handled these messages without indications of critical security compromise (i. e., Pass). The devices chose to drop the messages and deregister silently from the network. In the case of `Security Mode Command`, the phones responded with a `Security Mode Reject` including the "Unspecified" reason, and then they deregistered. Nevertheless, we noticed that when they received unintended headers it took a longer time to reconnect to the testing network.

5.2.4 Wrongly Accepted Messages after Security Enforcement. The concept of this category is to transmit messages with disabled security which must never be sent in clear after security enforcement.

NAS Tests. We tested network-initiated NAS messages, e. g., *5GMM Status*, and `Registration Accept`, after the AKA. Even though the messages were received by the devices, they did not reveal any security issue, i. e., Pass. We believe that the messages were dismissed by both smartphones, but they also disconnected abruptly from the network. In most cases, they re-initiated the RRC connection to re-register and recover the communications. Out-of-order messages were also rejected by smartphone devices. In summary, these specific models were not affected during this experimentation.

5.2.5 Wrongly Accepted Messages before Security Enforcement. In this scenario, we used messages that should always be sent protected but only before the security activation.

NAS Tests. We used similar messages to the previous category for testing without protection, e. g., *5GMM Status* and *Deregistration Request*, but before the establishment of the security context. Nonetheless, we noticed identical behavior to the aforementioned test cases. Both devices denied the messages and deregistered, thus passing our security tests. Generally, we believe that the devices have shown appropriate resilience against out-of-order messages and malformed structures.

RRC Tests. For this set of experiments we emphasized on the `RRCReestablishment`, and `CounterCheck`, `RRCResume`. According to the specifications [2, Annex B.1], they must always be accepted with protection by the UE. We transmitted them without the Message Authentication Code (MAC) once the `RRCSetupComplete` was received by the network. Both testing devices dismissed the abnormal packets and re-initiated the RRC connection later, thus passing our security tests again.

6 DISCUSSION

6.1 Challenges & Lessons Learned

Framework-based. Although the framework is constantly evolving and improving, there are some challenges. One such challenge is that open-source software may not have implemented all cellular

features and may contain bugs that could affect the expansion of test cases. However, we have not encountered any serious restrictions so far. Another challenge is that the lack of automatic iOS UE handling (unlike ADB for Android) means that testing for iOS devices is manual. Additionally, since the framework is based on various testing approaches, it is possible that during testing, the UE may exhibit behaviors that are not clearly defined, which may require a detailed understanding of the specifications and manual analysis before making a proper decision (i. e., pass/fail).

5G Connection-based. During our experiments, we discovered that connecting a commercial off-the-shelf 5G UE to a custom 5G network is not always straightforward. In our laboratory, we were able to connect only two devices (OnePlus Nord 2 5G and Huawei P40 Pro 5G) to the test 5G network. We encountered several hurdles that made a complete 5G connection difficult in many cases. One challenge is that the UE may not identify the network, even after a manual search. This could be due to misconfigurations or improper technical calibration, including issues with frequency division duplexing (FDD) or time division duplexing (TDD) and their corresponding frequencies, required synchronization with a GPS disciplined oscillator (GPSDO), gNodeB performance issues (such as low resources, under-flows, or weak signal strength), or the carrier's modulation and coding scheme (MCS). In such cases, the tester needs to first identify the correct configuration for the COTS device which should be reflected in the setup's configurations.

In addition to technical configurations, device manufacturers may enforce carrier policies that restrict or limit the use of 5G technology based on public land mobile networks (PLMNs) and UE capabilities. For example, the tested PLMN may be excluded or disallowed, or the device may not support certain capabilities. The solution to this problem is to use commercial configurations for the device and setup in a secluded environment (e. g., Faraday cage). Additionally, the use of testing PLMN, i. e., 00101, should generally be avoided for accurate results, because modems may enter into a debug mode and behave differently.

Programming the USIM to support 5G posed another challenge. Typically, the SIM should have Service 124 enabled for Subscription Identifier Privacy Support (SUCI), the keys provisioned correctly (in the SUCI Calculation Information EF), and the Routing indicator set. However, errors can still occur, preventing a full 5G connection, such as *MAC failure* and *Unknown SUCI* in the `Authentication Failure` message. If the errors persist, disabling the concealment support altogether can alleviate the problems, but this reveals the device's SUPI in clear.

Finally, identifying technical issues and mitigating them is challenging because of the lack of debugging tools, particularly at the lower layers that are manufacturer-specific. This is especially true when testing various devices with different properties. To overcome these difficulties, we suggest employing tools like Network Signal Guru, Qualcomm Debugger, commercial equipment (e. g., Amarisoft), and ADB logs throughout the testing process.

6.2 Future Work

Framework Expansion and Automation. The framework is designed to support 5G control-plane testing. In the future, we aim to cover Packet Data Unit (PDU) session-based messages, such as the `Session Release Command` and `Session Modification Command`, as well as paging messages. Similarly, we consider expanding the test case collection and including more complicated test cases with two UP-DL pairs within a test case, which is currently not covered in our work. Although we target the NAS and RRC layers, we could expand testing to other protocols as well. We currently select to focus on specific test cases, even though a test case generator could be deployed separately to produce a large number of test cases (in JSON format). We have not covered this additional process to date, as the massive generation of test cases may lead to redundant and irrelevant tests, and may impede automatic analysis requiring manual intervention, thus leading to performance degradation.

Regarding automation, we have primarily focused on reducing manual effort for device handling, test case loading, and testing synchronization. Nonetheless, we plan to expand the automation to improve the data processing and final results reducing manual interventions for the covered test cases. We recognize, though, that due to the ambiguity of the specifications and device behavior in certain cases, the analysis of the results cannot be automated in its entirety, also depending on the complexity of the test case.

Device Testing. As described in Section 6.1, 5G security testing can be challenging, and requires proper parametrization in the network and modifications in the devices. We have currently performed security testing against two 5G SA-capable devices with our test case collection. We plan to expand the testing to include devices from various manufacturers in the future.

Responsible Disclosure. In Section 5.2.1 and Table 2 we report security issues related to the specifications. Table 1 also presents the unprotected messages on 5G. Our results have been shared with 3GPP and we are in discussion with them at the time of writing this paper. For future new and substantiated vulnerabilities, we will inform the responsible bodies.

7 CONCLUSION

In this paper, we presented the first security testing framework for 5G SA user equipment. While the development of a fully comprehensive framework is a work in progress, we reported substantial results. We developed a proof-of-concept functional testing implementation, significantly altered the open-source suites Open5GS and srsRAN, and developed tens of unique test cases for 5G NAS and RRC layers. We successfully applied our testing framework to two 5G SA mobile phones, reported identified flaws, and provided detailed lessons learned from our experiments.

ACKNOWLEDGMENTS

The authors thank David Rupprecht and Merlin Chlosta for their advice in the bootstrapping phase of this project. This work is supported by a Google 2022 Android Security and PrIvacy REsearch (ASPIRE) Award and an Abu Dhabi Award for Research Excellence (AARE) 2019 (#AARE19-236). It is also supported by the Center for Cyber Security at New York University Abu Dhabi.

REFERENCES

[1] 3GPP. 2023. *5G; Non-Access-Stratum (NAS) protocol for 5G System (5GS); Stage 3.* 3rd Generation Partnership Project (3GPP). Version 17.9.0.

[2] 3GPP. 2023. *5G; NR; Radio Resource Control (RRC); Protocol specification.* 3rd Generation Partnership Project (3GPP). Version 17.3.0.

[3] 3GPP. 2023. *5G; Security architecture and procedures for 5G System.* 3rd Generation Partnership Project (3GPP). Version 17.8.0.

[4] 3GPP. 2023. *LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification.* 3rd Generation Partnership Project (3GPP). Version 17.3.0.

[5] 5G Americas. 2020. Security Considerations for the 5G Era. https://www.5gamericas.org/wp-content/uploads/2020/07/Security-Considerations-for-the-5G-Era-2020-WP-Lossless.pdf.

[6] Evangelos Bitsikas and Christina Pöpper. 2021. Don't Hand It Over: Vulnerabilities in the Handover Procedure of Cellular Telecommunications. In *Annual Computer Security Applications Conference (ACSAC '21)*. Association for Computing Machinery, NY, USA, 900–915. https://doi.org/10.1145/3485832.3485914

[7] Evangelos Bitsikas and Christina Pöpper. 2022. You Have Been Warned: Abusing 5G's Warning and Emergency Systems. In *Annual Computer Security Applications Conference (ACSAC '22)*. Association for Computing Machinery, New York, NY, USA, 561–575. https://doi.org/10.1145/3564625.3568000

[8] Yi Chen, Yepeng Yao, XiaoFeng Wang, Dandan Xu, Chang Yue, Xiaozhong Liu, Kai Chen, Haixu Tang, and Baoxu Liu. 2021. Bookworm Game: Automatic Discovery of LTE Vulnerabilities Through Documentation Analysis. In *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*. IEEE, 1197–1214. https://doi.org/10.1109/SP40001.2021.00104

[9] Merlin Chlosta, David Rupprecht, Thorsten Holz, and Christina Pöpper. 2019. LTE security disabled: misconfiguration in commercial networks. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks, WiSec 2019, Miami, Florida, USA, May 15-17, 2019*. ACM, 261–266. https://doi.org/10.1145/3317549.3324927

[10] Merlin Chlosta, David Rupprecht, Christina Pöpper, and Thorsten Holz. 2021. 5G SUCI-Catchers: Still Catching Them All?. In *Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '21)*. Association for Computing Machinery, New York, NY, USA, 359–364. https://doi.org/10.1145/3448300.3467826

[11] ENISA. [n. d.]. Security in 5G Specifications. https://www.enisa.europa.eu/publications/security-in-5g-specifications.

[12] Kaiming Fang and Guanhua Yan. 2018. Emulation-Instrumented Fuzz Testing of 4G/LTE Android Mobile Devices Guided by Reinforcement Learning. In *Computer Security - 23rd European Symposium on Research in Computer Security, ESORICS 2018, Barcelona, Spain, September 3-7, 2018, Proceedings, Part II (Lecture Notes in Computer Science)*, Javier López, Jianying Zhou, and Miguel Soriano (Eds.), Vol. 11099. Springer, 20–40. https://doi.org/10.1007/978-3-319-98989-1_2

[13] Matheus E. Garbelini, Zewen Shang, Sudipta Chattopadhyay, Sumei Sun, and Ernest Kurniawan. 2022. Towards Automated Fuzzing of 4G/5G Protocol Implementations Over the Air. In *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*. 86–92. https://doi.org/10.1109/GLOBECOM48099.2022.10001673

[14] GSM Association. 2022. The Mobile Economy. https://www.gsma.com/mobileeconomy/wp-content/uploads/2022/02/280222-The-Mobile-Economy-2022.pdf.

[15] Grant Hernandez, Marius Muench, Dominik Maier, Alyssa Milburn, Shinjo Park, Tobias Scharnowski, Tyler Tucker, Patrick Traynor, and Kevin R. B. Butler. 2022. FirmWire: Transparent Dynamic Analysis for Cellular Baseband Firmware. In *Symposium on Network and Distributed System Security (NDSS)*.

[16] Syed Rafiul Hussain, Omar Chowdhury, Shagufta Mehnaz, and Elisa Bertino. 2018. LTEInspector: A Systematic Approach for Adversarial Testing of 4G LTE. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*. The Internet Society.

[17] Syed Rafiul Hussain, Mitziu Echeverria, Omar Chowdhury, Ninghui Li, and Elisa Bertino. 2019. Privacy Attacks to the 4G and 5G Cellular Paging Protocols Using Side Channel Information. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA*. The Internet Society.

[18] Syed Rafiul Hussain, Mitziu Echeverria, Imtiaz Karim, Omar Chowdhury, and Elisa Bertino. 2019. 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 (CCS '19)*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3319535.3354263

[19] Syed Rafiul Hussain, Imtiaz Karim, Abdullah Al Ishtiaq, Omar Chowdhury, and Elisa Bertino. 2021. Noncompliance as Deviant Behavior: An Automated Black-Box Noncompliance Checker for 4G LTE Cellular Devices. In *ACM SIGSAC Conference on Computer and Communications Security (CCS '21)*.

[20] Imtiaz Karim, Syed Rafiul Hussain, and Elisa Bertino. 2021. ProChecker: An Automated Security and Privacy Analysis Framework for 4G LTE Protocol Implementations. In *41st IEEE International Conference on Distributed Computing Systems, ICDCS 2021, Washington DC, USA, July 7-10, 2021*. IEEE, 773–785.

[21] Rabia Khan, Pardeep Kumar, Dushantha Nalin K. Jayakody, and Madhusanka Liyanage. 2020. A Survey on Security and Privacy of 5G Technologies: Potential Solutions, Recent Advancements, and Future Directions. *IEEE Communications Surveys & Tutorials* 22, 1 (2020), 196–248. https://doi.org/10.1109/COMST.2019.2933899

[22] Hongil Kim, Jiho Lee, Eunkyu Lee, and Yongdae Kim. 2019. Touching the Untouchables: Dynamic Security Analysis of the LTE Control Plane. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*. IEEE, 1153–1168. https://doi.org/10.1109/SP.2019.00038

[23] Norbert Ludant and Guevara Noubir. 2021. SigUnder: A Stealthy 5G Low Power Attack and Defenses. In *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '21)*. Association for Computing Machinery, NY, USA, 250–260. https://doi.org/10.1145/3448300.3467817

[24] Dominik Maier, Lukas Seidel, and Shinjo Park. 2020. BaseSAFE: Baseband Sanitized Fuzzing through Emulation. In *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '20)*. Association for Computing Machinery, New York, NY, USA, 122–132. https://doi.org/10.1145/3395351.3399360

[25] CheolJun Park, Sangwook Bae, BeomSeok Oh, Jiho Lee, Eunkyu Lee, Insu Yun, and Yongdae Kim. 2022. DoLTEst: In-depth Downlink Negative Testing Framework for LTE Devices. In *31th USENIX Security Symposium (USENIX Security 22)*.

[26] Srinath Potnuru and Prajwol Kumar Nakarmi. 2021. Berserker: ASN.1-based Fuzzing of Radio Resource Control Protocol for 4G and 5G. *17th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)* (2021), 295–300.

[27] David Rupprecht, Kai Jansen, and Christina Pöpper. 2016. Putting LTE Security Functions to the Test: A Framework to Evaluate Implementation Correctness. In *USENIX Workshop on Offensive Technologies (WOOT'16)*. USENIX Association, USA, 40–51.

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

[29] David Rupprecht, Katharina Kohls, Thorsten Holz, and Christina Pöpper. 2020. IMP4GT: IMPersonation Attacks in 4G NeTworks. In *ISOC Network and Distributed System Security Symposium (NDSS)*. ISOC.

[30] Altaf Shaik, Ravishankar Borgaonkar, Shinjo Park, and Jean-Pierre Seifert. 2019. New Vulnerabilities in 4G and 5G Cellular Access Network Protocols: Exposing Device Capabilities. In *Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '19)*. Association for Computing Machinery, New York, NY, USA, 221–231. https://doi.org/10.1145/3317549.3319728

[31] Digital Trends. 2023. What is 5G? Speeds, coverage, comparisons, and more. https://www.digitaltrends.com/mobile/what-is-5g/.

[32] Hojoon Yang, Sangwook Bae, Mincheol Son, Hongil Kim, Song Min Kim, and Yongdae Kim. 2019. Hiding in Plain Signal: Physical Signal Overshadowing Attack on LTE. In *28th USENIX Security Symposium (USENIX Security 19)*. USENIX Association, Santa Clara, CA, 55–72. https://www.usenix.org/conference/usenixsecurity19/presentation/yang-hojoon

APPENDIX

Listing 1: Test Case Example

```
[ { //PreAKA
    "ue_ul_handle": "null",
    "dl_reply": "null",
    "command_mode": "null",
    "dl_params": "null"
  },
  { //AKA
    "ue_ul_handle": "security_mode_complete",
    "dl_reply": "registration_reject",
    "command_mode": "send",
    "dl_params":{
      "gmm_cause": "PLMN_NOT_ALLOWED"
    }
  },
  { //PostAKA
    "ue_ul_handle": "null",
    "dl_reply": "null",
    "command_mode": "null",
    "dl_params": "null"
  } ]
```

Table 3: 5G-capable devices from our testing setup along with their specifications.

Device	Chipset	OS	Model	Release
OnePlus Nord 2 5G	MediaTek Dimensity 1200 5G	Android 11	DN2101	2021
Huawei P40 Pro 5G	Huawei Kirin 990 5G	Android 10	ELS-NX9	2020