2019 IEEE Symposium on Security and Privacy

Touching the Untouchables: Dynamic Security Analysis of the LTE Control Plane

Hongil Kim
KAIST
hongilk@kaist.ac.kr

Jiho Lee
KAIST
jiholee@kaist.ac.kr

Eunkyu Lee
KAIST
ekleez@kaist.ac.kr

Yongdae Kim
KAIST
yongdaek@kaist.ac.kr

***Abstract*—This paper presents our extensive investigation of the security aspects of control plane procedures based on dynamic testing of the control components in operational Long Term Evolution (LTE) networks. For dynamic testing in LTE networks, we implemented a semi-automated testing tool, named LTEFuzz, by using open-source LTE software over which the user has full control. We systematically generated test cases by defining three basic security properties by closely analyzing the standards. Based on the security property, LTEFuzz generates and sends the test cases to a target network, and classifies the problematic behavior by only monitoring the device-side logs. Accordingly, we uncovered 36 vulnerabilities, which have not been disclosed previously. These findings are categorized into five types: Improper handling of (1) unprotected initial procedure, (2) crafted plain requests, (3) messages with invalid integrity protection, (4) replayed messages, and (5) security procedure bypass. We confirmed those vulnerabilities by demonstrating proof-of-concept attacks against operational LTE networks. The impact of the attacks is to either deny LTE services to legitimate users, spoof SMS messages, or eavesdrop/manipulate user data traffic. Precise root cause analysis and potential countermeasures to address these problems are presented as well. Cellular carriers were partially involved to maintain ethical standards as well as verify our findings in commercial LTE networks.**

I. Introduction

Long Term Evolution (LTE) is the most advanced telecommunication technology thus far. It not only provides faster data transmission with low latency but also ensures high reliability and robustness against unexpected failures compared to older generation networks such as Global System for Mobile communication (GSM), and Universal Mobile Telecommunication System (UMTS). Mobile network operators are aggressively deploying LTE infrastructure; as of 2018, 600 carriers in 200 countries have deployed LTE networks, having more than 3.2 billion subscribers worldwide [1], [2].

In addition to facilitating traditional telecommunication services such as data and voice call, LTE is considered a key enabler for providing always-on mobile connectivity in both the emerging industries (e.g., autonomous vehicles and the Internet of Things) and nation-wide communication infrastructure (e.g., Public Safety LTE and LTE-R for railway communication). Typically, these applications are safety-critical and require high availability and robustness. This means that users could face a significant threat to their safety if these applications were to malfunction by the accidental disconnection of LTE services. It is therefore pivotal to investigate the potential threats to the LTE service procedures that can cause unexpected failures as a result of accidents or adversaries.

The 3rd Generation Partnership Project (3GPP), a de facto standard for LTE, defines the behavior of all network components including security features, which have been significantly improved compared to earlier networks (e.g., stronger encryption and integrity protection algorithms, mandatory use of integrity protection in control plane protocols) [3]. Moreover, it provides conformance test suites for commercial LTE chipsets to ensure compliance with the specification.

In spite of these efforts to eliminate the risks of unexpected errors, recent studies have uncovered various security vulnerabilities in the control plane procedures of LTE networks. For example, an active adversary can relay an LTE communication between a mobile device and a network to hijack the location of the device [4] or redirect the DNS request of the device [5]. Attacks using rogue base stations can either track the location of a user device or deny the LTE service by exploiting both device side design flaws and implementation bugs [6]–[8]. However, none of these studies focused on analyzing network-side problems in operational LTE networks although vulnerabilities of this nature can influence a number of their subscribers once exploited.

Motivated by the fact that the control plane components in LTE are still under-explored, we investigated potential problems of the control plane procedures in operational LTE networks by dynamically analyzing the core network responses resulting from carefully crafted malicious inputs. In general, dynamically testing network behavior is challenging because: ❶ Exploiting the control plane protocols with a commercial smartphone is quite difficult. This is because commercial devices implement the control plane protocols on a baseband chipset, from which generation of arbitrary messages is difficult. ❷ The deployed carrier networks are closed systems. Their configurations are proprietary and the control plane logs are unavailable to devices. Thus, it is difficult to correctly determine the root causes of the identified problems on the device side. ❸ Transmitting signals to an operational network using an uncertified device may not be allowed depending on the regulations for carriers and countries.

We overcome these challenges by utilizing open source LTE implementations [9]–[11] for testing purposes. To this end, we implemented a semi-automated testing tool, named **LTEFuzz**, by using fully controllable LTE open source software, which 1) dynamically generates and sends the test cases to a target network or a device [9], and further, 2) classifies problematic behavior by only inspecting the responses in the tester and victim device from the target. Second, we collaborated with carriers to avoid ethical issues. For each of the

DOI 10.1109/SP.2019.00038

critical test cases, we interviewed carriers to establish whether it would interfere with the LTE network. Suspicious cases that could interfere with other users were handled independently. Regarding the regulations on using licensed bands, our tester device acts as if it were a single LTE device and executes test messages individually. Therefore, nearby legitimate users do not experience any interference, performance degradation, and failure of their LTE services. In addition, we sent spoofed messages by using the identity of our LTE phone. Thus, only our phone might experience the intended failure in case our test case is accepted by the target networks. Lastly, we reason out the root causes based on 1) a review of 3GPP standards and 2) interview with the carriers to confirm our findings.

To generate the test cases systematically, we first created three security properties by extensively analyzing the correct behavior of network components and their security requirements mandated in the 3GPP specifications. Using these security properties, we determined the scope of the target messages and generated rules for specific test cases. Then, the test case generator mutated the random inputs among the datasets of the control plane logs of a commercial network, which we collected globally for approximately one year. The reason for only considering the inputs in the commercial logs is to prevent unexpected crashes in the receiving nodes due to parsing errors. Total test cases cover 13 messages for inspecting the behavior of network nodes and 29 messages for inspecting the behavior of commercial devices.

By conducting tests against the operational network, we found 51 vulnerabilities (36 new and 15 previously known), which are mainly caused by the improper handling of ① unprotected initial procedures, ② crafted plain requests, ③ messages with invalid integrity protection, ④ replayed messages, and ⑤ security procedure bypass. We also demonstrate all the attacks by exploiting the vulnerabilities we found in the operational network while strictly following the ethics. Our attacks can ❶ deny various LTE services to either a target user or arbitrary users, ❷ spoof control plane messages for privacy leaks, ❸ send spoofed Short Message Service (SMS), and ❹ eavesdrop and manipulate data communication. A complete list of vulnerabilities is given in the Table IV in the Appendix.

To summarize, our contributions are as follows:

- We propose a systematic approach to expose vulnerabilities in the control plane procedures of LTE networks. Our method can examine various security aspects in several control plane protocols by simply creating concrete security properties and test cases.
- To the best of our knowledge, we are the first to analyze the security problems caused by incorrect behavior in the control plane nodes in commercial networks such as *Radio Resource Control (RRC)* and *Non Access Stratum (NAS)* by executing crafted uplink test cases in both the *RRC* and the *NAS* layer.
- We demonstrated that most of our findings are exploitable by an adversary who would be able to launch critical attacks such as the denial of LTE services, spoofing of both control and data plane messages, eavesdropping user data traffic, and conducting phishing attacks on legitimate users.

The remainder of the paper is organized as follows. Section II presents an overview of LTE technology with the network architecture and key control plane procedures. In Section III, we introduce our approach to systematically find existing vulnerabilities in commercial network environments. Section IV provides a detailed implementation and experimental setup of our proposed approach. We present the results of our test and root cause analysis of each finding in Section V. The attacks that exploit the identified vulnerabilities are presented in Sections VI, VII, and VIII. Section IX discusses potential countermeasures and Section X presents related work. We conclude our paper and present future work in Section XI.

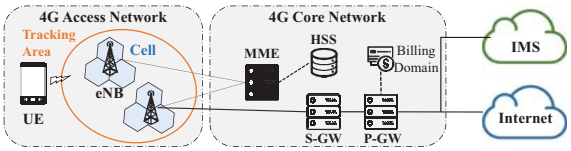

Fig. 1. LTE network architecture

II. Background

This section presents the basic concepts of standard LTE technology along with network architecture and essential control plane procedures [12]–[19].

A. LTE network architecture

Fig. 1 illustrates the LTE network architecture, which consists of User Equipment (UE), evolved Node B (eNB), and Evolved Packet Core (EPC) components.

UE refers to a mobile device, which can provide a legitimate user with subscribed services such as data and voice call by connecting to a base station. The UE is uniquely identified by an International Mobile Station Equipment Identity (IMEI). One distinctive feature of the UE is the use of a Universal Subscriber Identity Module (USIM), a smart card that can be physically inserted into the UE, which includes the subscriber identifier known as the International Mobile Subscriber Identity (IMSI), cryptographic keys, and algorithms.

An **eNB** is the Base Transceiver Station (BTS) in LTE, which enables the UE to establish a wireless connection to the LTE network. A typical eNB has a Baseband Unit (BBU) that is responsible for processing the baseband signals, and it is connected to multiple Remote Radio Units (RRUs), which directly process the transmission and reception of RF signals. Each RRU covers a sector (also known as a cell), thus, one eNB can cover multiple cell sites as shown in Fig. 1 [1]. The eNB is connected to a Mobility Management Entity (MME) for control-plane communication and 4G Gateways (GWs) for both control-plane and user-plane data transmission.

The **MME** is the key control-node in the LTE network. It authenticates the UE and manages the mobility status of subscribers and Evolved Packet System (EPS) bearers. The cryptographic keys for authenticating and protecting the UE are contained in the Home Subscriber Server (HSS). The MME performs UE authentication through an authentication protocol known as Evolved Packet System-Authentication and Key Agreement (EPS-AKA) with the key information received from the HSS. The MME is also involved in the activation/deactivation process of the EPS bearer, a logical tunnel

[1]The number of cells in one eNB can range from 1 to 256 in the specification and, in practice, this depends on the operators' cell planning.

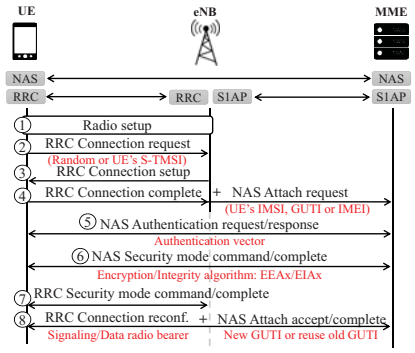

Fig. 2. UE attach procedure with our target control plane protocol stack

created during a session between the UE and GW to service the Internet connection. Thus, it manages user network access by setting up and terminating the session. User mobility is managed by tracking each user's states, stored in the MME (e.g., whether the UE is attached to the network) and it tracks the location of the UE in cell units. These cells are grouped into a Tracking Area (TA).

4G GWs consist of two types of LTE gateways: a Serving Gateway (S-GW) and a PDN Gateway (P-GW). These GWs ensure mobility of the UE and provide Internet service. The S-GW becomes an anchor point when the UE moves from one base station to another, which is known as handover, and the P-GW allocates the IP addresses and manages the accounting data of the UE. It also connects the UE to the Internet.

B. LTE service procedures

1) UE attach procedure: The initial procedure whereby the UE attaches to the network (as shown in Fig. 2) is related with both connecting the RAN (eNB) and the EPC network (MME). The UE has to establish a radio connection with the eNB first, after which it attaches to the EPC network for a full connection to the Internet through the LTE network.

Connection between the UE and eNB. When the UE is turned on, it first finds a suitable cell by listening to broadcasting messages from nearby base stations according to the settings (e.g., operator specific codes, LTE bands, and channels) configured in the USIM card and the device modem. Once the UE finds a suitable cell, (1) it initiates the Random Access (RA) procedure to obtain uplink resource allocation and timing information. Then, it saves the assigned identity known as a Temporary Cell Radio Network Temporary Identifier (Temporary C-RNTI) from the selected cell. With this Temporary C-RNTI and the uplink assignments, (2) the UE attempts to establish a *Radio Resource Control (RRC) Connection* by transmitting an `RRC Connection request`. (3) On receiving the request, the eNB replies with the `RRC Connection setup` including information about the dedicated radio resource allocation and the C-RNTI value to be used to distinguish the UE for subsequent radio communication. (4) Lastly, the UE completes the connection setup by sending an `RRC Connection setup complete` message to the cell. Once all these steps have been performed, the UE is synchronized with the cell in both the uplink and downlink. In addition, both the UE and eNB change their *RRC* state from *IDLE* to *CONNECTED* as defined in the 3GPP standard [19]. In the *RRC CONNECTED* state, the UE communicates with the connected cell using the *RRC* protocol for control-plane procedures. In Section V-A, we show that no security measures are typically adopted in these initial procedures, thereby causing serious problems in target eNBs.

Connection between UE and MME. Once the UE is successfully connected to a nearby eNB, it has to attach to the EPC network to obtain LTE services. First, (4) the UE piggybacks the `NAS Attach request` to the `RRC Connection complete` and sends it to an MME. Upon receiving this message, (5) the MME starts the Authentication and Key Agreement (AKA) procedure by replying to the `NAS Authentication request` from the UE with an *Authentication vector* generated from the HSS. The UE then authenticates the MME through the contents in the `NAS Authentication request` and replies by sending an `NAS Authentication response` to the MME. At this stage, both the UE and the MME are mutually authenticated. Next, in terms of key agreement, (6) the MME selects the encryption and integrity protection algorithms to use, and sends a `NAS Security mode command` to the UE to inform it about the selected algorithms. Upon the receipt of this message, the UE generates security keys for the *NAS* layer. Likewise, (7) the eNB proceeds `RRC Security mode command` to inform the security algorithms for *RRC* layer and user plane data security. After these steps, all the control plane messages that should be protected according to the standards [15], [19] are encrypted and integrity protected using the negotiated security algorithms. Lastly, after optional configurations are exchanged in the *NAS* and *RRC* layer, (8) the MME allocates a Globally Unique Temporary Identity (GUTI)[2] to substitute the permanent identity (IMSI in LTE) and sends the GUTI and other connection information by transmitting an `Attach accept` to the UE. The attach procedure is finally completed when the UE sends an `Attach complete` message to the MME.

2) UE mobility management: As shown in Fig. 1, each MME manages its own Tracking Areas (TAs), each of which is identified by a Tracking Area Code (TAC). Each TA contains several eNBs, which operate several cells to efficiently cover the geographical regions with no signal interference according to the operating policy of the carrier. Upon arrival of an incoming service request to a specific UE, the MME first checks whether the UE is *RRC CONNECTED* or *RRC IDLE*. If the UE is in the *RRC IDLE* state, the MME has to wake up the UE to make the *RRC Connection* and the other radio bearers of data traffic. This procedure is known as *Paging* in LTE terminology. As the MME only has information about the TA that previously served the UE, the *Paging* message is broadcast to all the eNBs in that TA. If the UE is not found in the particular TA, the MME can broadcast the *Paging* to the other TAs. Note that the specific *Paging* policy to find UEs might be different across carriers depending on the relative priority allocated to QoS, the signaling load, and other operating issues.

III. The Proposed Approach

This section presents LTEFuzz, the proposed approach to systematically conduct dynamic security analysis in operational LTE networks in a semi-automated way. LTEFuzz consists of three main steps (as shown in Fig. 3):

(1) Extracting security properties: First, we extensively analyze the LTE standards of the control plane procedures by

[2]The GUTI value consists of MME group id and S-TMSI.

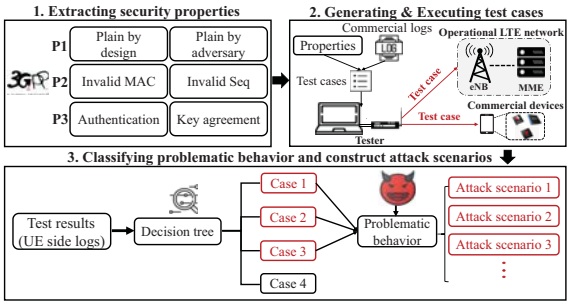

Fig. 3. Overview of LTEFuzz

focusing on the security aspects. Based on the analysis, we create three security properties the network and the mobile devices need to follow to ensure they are protected against unknown security threats.

(2) Generating test cases: Next, we generate test cases to identify situations in which the target control plane component violates the security properties. The test cases are generated based on the specified rules of target protocol messages and its fields for each property.

(3) Classifying problematic behavior: When executing the test cases, we need to determine which responses and state changes on the side of the UE are considered as problematic behavior. To this end, we build a simple decision tree logic to classify the problematic cases. Our model only considers the control plane logs and states on the UE side when the test case is executed. This model, therefore, enables LTEFuzz to identify problematic cases in an automatic way.

The remainder of this section considers each of these steps.

A. *Extracting security properties from standards*

A thorough analysis of the specifications regarding the control plane procedures and security requirements enabled us to identify potential security holes that might cause the confidentiality and integrity protection of control plane procedures be circumvented depending on the implementation and configuration policies of carriers. First, initial procedures before establishing a security context might be exploited by adversaries who are able to eavesdrop and manipulate LTE signals. Second, quite a few exceptional situations exist in which the receiving entity would accept the received control plane message without integrity protection (Section 4.4.4 in [15] and Appendix 6 in [19]). Third, although the specification adopts counters for the control plane protocol (such as NAS and RRC), it specifies the use of a sequence number (partial bits of the counter) in the received message when verifying the message integrity. Therefore, message replay might be allowed. From these observations, we created the three basic security properties listed in Table I, each of which focuses on the ability of the responsible entities to correctly respond to malicious behavior by an adversary. We assume that the adversary has minimal privilege: the adversary neither owns valid keys to register with the LTE network nor do they have information about other legitimate users' keys. In addition, as each property focuses on different security aspects (i.e., incorrect handling of unprotected procedures, invalid security protected messages, and mandatory AKA procedures), the test scenarios and the rules of selecting target messages vary from one property to another. Note that, when considering the security properties, we only targeted the *NAS* and the *RRC* protocol among the various control plane protocols because (1) these protocols are used to perform critical control plane procedures between the UE and the network (e.g., UE attach procedure, mobility management, and authentication), (2) we were able to capture and analyze these procedures at the UE, and (3) the identified vulnerabilities in these protocols directly affect both the UE and the network.

Each of these properties is explained in more detail below.

1) Property 1: It confirms whether the receiving entity (the eNB or MME for an uplink and the UE for a downlink) appropriately handles unexpected inputs when an adversary sends crafted plain messages during the initial procedures. To validate this property, we consider two situations when selecting the target messages: (1) crafted plain messages that can be sent before security is activated, and (2) messages that should not be sent unprotected after security activation according to the standard. For the first case, we mainly inspect the potential threats of initial plain messages that cannot be protected by the nature of the symmetric key cryptography of LTE. For these unprotected messages, it is hard to distinguish whether the received message is sent from the adversary or a benign user. On the other hand, the purpose of the second situation is to inspect whether the deployed cellular components are correctly implemented to reject or discard invalid plain messages that do not comply with the standard. Messages transmitted after security activation usually adhere to security critical procedures. Thus, an adversary might affect the connection state of a UE or expose the private information of the UE if the receiving entities incorrectly handle these security-protected messages. Here we assume that the adversary neither subscribes to the particular mobile phone service nor do they have the security keys of the other UE. Therefore, the adversary would be able to create and send plain messages with arbitrary contents, but these messages are not valid ones. During the test, the adversary acts as a *malicious UE* when investigating the behavior of the eNB and MME (uplink direction) and as a *rogue LTE network* when investigating UE behavior (downlink direction).

Example cases. An example of a message representing the first situation is an `RRC Connection request`. Because the initial *RRC Connection* procedure is not protected by design, an adversary can spoof any contents while establishing the *RRC Connection*. Victim eNBs without proper security measures would accept these fabricated messages. Another example to illustrate the second situation could be a plain `NAS Attach request` spoofed with the victim's GUTI. In normal cases, when UE attempts to perform re-registration with their previous cryptographic key information (known as the *security context* in 3GPP) and GUTI, an integrity protected `NAS Attach request` is sent. Upon receiving the message, the MME allows UE registration without performing an AKA procedure because the UE is already authenticated by its valid integrity protected message. Thus, if the MME does not correctly check whether the received message has to be security protected, an adversary may disconnect the victim's existing connection by sending a plain `NAS Attach request` spoofed with the victim's GUTI. A detailed analysis of the consequences of these two situations is provided in Section V-A and V-B, respectively.

2) Property 2: It validates whether the receiver appropriately handles unexpected messages that are incorrectly

TABLE I. SECURITY PROPERTIES FOR SYSTEMATIC SECURITY TESTING

	Security property	Target procedures/messages	Example
P1	Invalid plain messages should be properly handled	Messages that are allowed to be sent in plaintext	RRC Connection request, IMSI Attach request
		Messages that are not allowed to be sent in plaintext	GUTI Attach request, Uplink NAS transport
P2	Invalid security protected messages should be properly handled	Messages with invalid integrity protection	PDN disconnect request, Service request
		Messages with invalid sequence number	
P3	Mandatory security procedures should not be bypassed	Mutual authentication procedure	Authentication request
		Key agreement procedure	NAS/RRC Security mode command

encapsulated with a security header. According to the specification [15], all *NAS* messages after the AKA procedure should be both encrypted and integrity protected except some messages, such as an `Attach request`, a `TAU request`, and a `Security mode command`, all of which are only integrity protected. To this end, when a UE sends an *NAS* message after the AKA procedure, it encrypts the plain *NAS* messages first and then calculates the Message Authentication Code (MAC) for integrity protection (shown in Fig. 10). We demonstrate this property by investigating two specific cases to determine whether the receiving entity appropriately verifies (1) the integrity of the security protected messages and (2) the sequence number, which consists of the eight least significant bits of the 32-bit counter value synchronized between the sending and receiving entities. Intuitively, if the receiving entity does not verify the integrity of the message, an adversary can spoof any unencrypted messages. Further, if the sequence number is not thoroughly verified, the adversary can launch a replay attack using security-protected messages that were previously captured from victim UE. Similar to property 1, because the adversary does not have any cryptographic keys to generate valid messages, they send invalid messages after establishing a connection to the eNB. The target messages for each of the cases constitute every possible message that should be protected after security activation.

Example cases. An example relating to the first case could be `NAS Uplink NAS transport`, which is used for SMS within the carriers providing *SMS over NAS*. If an MME does not properly verify the integrity of this message, the adversary can exploit it for an SMS phishing attack by spoofing the contents of the `NAS Uplink NAS transport` message. Another example representative of the second case is an `NAS PDN disconnect request`. The purpose of this message is to release the established *Packet Data Network (PDN) Connection*; in particular, when the user turns off their device or switches off the data service. An adversary could send this message to the network by pretending to be the victim UE, and the network would accept this replayed message if it does not correctly verify the sequence number specified in the message. This could lead to selective denial of service of legitimate users.

3) Property 3: It confirms whether the security procedures specified in the 3GPP standards [15], [19] can be bypassed by malicious UE or a network. The LTE standard adopts EPS-AKA for mutual authentication between the UE and the network for protection of both the control and data planes. This includes the *NAS Authentication* procedure based on the challenge-response mechanism and the *Security mode command* in both the *RRC* and *NAS* layers, which are session key agreement procedures for control plane and data plane confidentiality and integrity. Three types of approaches are available to inspect whether these security procedures could be bypassed. The first is to perform a security analysis of the cryptographic algorithms adopted in the LTE standard. However, this situation is out of the scope of this paper. Second, one could consider situations in which an adversary manipulates the encryption and integrity protection algorithms selected in the `RRC/NAS Security mode command` and security header type in `NAS protected messages`. This was previously studied [20], but the authors found vulnerabilities in only one commercial modem. The last situation involves omitting parts of the mandatory security procedures. If the device allows this to occur, a *rogue LTE network* without cryptographic keys for the legitimate devices could even provide manipulated services without confidentiality and integrity protection. Despite the possibility of these situations causing serious threats if exploited, the consequences have not yet been investigated and publicly disclosed. Thus, we limit the scope of this security property to validate whether the UE correctly handles situations in which a malicious LTE network omits the mandatory security procedures such as an `Authentication request` and a `Security mode command` in both the *RRC* and *NAS* layers.

Example case. An example could be to disregard the `NAS Authentication request` to enable an adversary to continue following service procedures without authentication and key agreement, both of which are mandatory procedures [15], [19].

B. Generating test cases for each property

Although we chose the target messages for each property, several test cases exist if we consider the inputs for all possible field values in each message. For example, the generation of test cases for an invalid plain `NAS Attach request` to verify security property 1 would have to consider that it has 24 fields including an optional one and the available length in this message could be at least 16 bytes. Obviously, testing all these possibilities is expensive. To reduce the number of test cases, yet carry out a sufficient number to investigate the behavior of the target entity, LTEFuzz utilizes commercial control plane message logs. In this regard, we collected various control plane messages by triggering many functionalities in the baseband chipset by sending AT commands [21]. We then used this log to build a database in which to store all the values in the collected logs for each field, separated by carriers. Thus, when the generator creates test cases for carrier A, it selects one of the possible values marked as carrier A. When generating the test cases to check sequence number verification (second case in property 2), we generated all the test cases by capturing packets on the side of the victim UE. When generating test cases for initial plain messages (property 1) and messages with invalid MAC (first case in property 2), only the mandatory fields are considered as they are security critical for correct LTE operation. Note that when the test case message contains the identity field of the UE, the current identity of the UE such as the GUTI or IMSI is included to inspect whether the receiving entity changes the state of the victim UE.

C. Classifying problematic behavior

When each case is tested, LTEFuzz has to identify which of these cause problematic behavior in the receiving entity. This

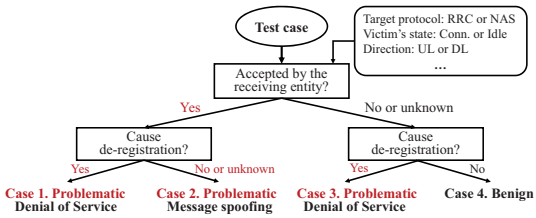

Fig. 4. Decision tree based logic for classifying problematic behavior

can easily be achieved if the operation logs of the receiving entity are available to monitor. However, obtaining the operation logs of cellular networks is not possible for researchers without support from carriers or equipment vendors. To overcome this limitation, LTEFuzz classifies problematic behavior by monitoring only the logs in the UE based on a simple decision tree as shown in Fig. 4. This logic has two decision phases: (1) whether the test case (invalid message) is accepted, and (2) whether the test case causes disconnection of the victim UE. For the first decision phase, we define the expected response when the receiving entity accepts each test case based on the 3GPP standard. Then, LTEFuzz checks whether the expected response is received by the UE when each test case is sent. For instance, if a test case is an invalid `NAS Identity request`, the expected response should be an `NAS Identity response` with the desired message contents. Once this expected response is received, LTEFuzz considers the test case to be accepted and classifies the test case as being abnormal because a test case with invalid input should not be accepted. At the second decision phase, it further examines whether the victim UE is disconnected from the network in response to a test case. If yes, it is classified as problematic, which can result in denial of service to the victim UE (case 1). If no or unknown, it is also classified as problematic because this behavior could be exploited to conduct a spoofing attack (case 2). When the test case is not accepted in the first phase, the result is divided into two different cases in the second phase. If the victim UE is disconnected from the network although the test case is not accepted in the receiving entity, it is also classified as problematic behavior, which might be rooted in misbehavior of the receiving entity when it recognizes that the received message is not valid (case 3). Otherwise, the test case is classified as benign, which means the receiving entity correctly handled the invalid message (case 4). Based on this classification, we can easily identify the problems and even obtain attack scenarios. For example, the crafted messages classified as case 1 and 3 can be exploited by an adversary to perform a DoS attack against the victim UE.

D. Ethical considerations

Testing against operational networks. The purpose of our study was not to identify failures causing crashes or memory leaks. Instead, we focused on finding semantic failures in LTE operations. To this end, we generated all possible test cases that would have been correctly parsed in the receiving entities as the field values are created based on the control plane logs from the operational networks. For instance, our test cases for carrier A are only created using the field values found in the control plane messages of this carrier. In addition, we only carried out the tests against our subscribed mobile phone to ensure that the tests did not affect the connection state of other legitimate users. It might only change the state of our target device into an unexpected one if our test cases are accepted by the receiving entity. Control plane overhead was negligible considering the overhead of a normal situation [22]. The only test case that may affect other UEs in the victim cell was conducted by 1) first testing against a femtocell utilizing the frequency bands that are not used in operational networks, and 2) testing on the testbed of the carrier.

Testing against commercial phones. A testbed operating in an LTE licensed band might influence legitimate users who are not participating in our experiments. To prevent normal users from connecting to our testbed network, we only utilized frequency bands that are not used by operational networks. In addition, we set the transmission power of our eNB to a minimum such that only our target UE within a distance of 20 cm was connected to our testbed. As a result, we confirmed that no legitimate users were attempted to connect to our testbed network during the experiments.

Legal restriction. Many countries have legal restrictions that forbid unauthorized signals to be sent to commercial network systems for the purpose of disrupting their stable operation. Thus, dynamic testing of a commercial LTE network is also strictly prohibited without permission. Fortunately, for the purpose of inspecting and validating the vulnerabilities we uncovered, two carriers gave us a permission to conduct dynamic security testing. In addition, similar to carefully generating the test cases to avoid disrupting the normal LTE services, we also confirmed that our test cases were not problematic in terms of the availability and reliability of network components by cooperating with the carriers. After conducting the tests, we also responsibly disclosed our findings to the carriers and vendors to address any problems immediately. With regard to vulnerabilities attributed to specification defects, we are planning to contact the standard bodies soon.

IV. Implementation

Although we explained above that we reduced the number of test cases, manually conducting this number of test cases is time-consuming; furthermore, a manual approach might increase the possibility of introducing mistakes that could affect the consistency and reliability of the experiments. To this end, we tried to automate test operations as much as possible with the help of fully controllable open source LTE stacks [9], [10] and a control plane logging tool known as SCAT [23]. The experimental setup and implementation of LTEFuzz to execute the test cases are divided into two types: 1) inspecting operational network components for uplink test cases, and 2) inspecting commercial mobile devices for downlink test cases.

1) Inspecting operational networks. In this setup, the tester acts as malicious UE and sends a crafted message with the test case input to a target operational network. The tester is implemented on open source standard UE stack known as srsLTE [9]. To confirm whether each test case is executed or triggers a failure on the network side, we utilized a decision tree to classify the problematic case by only monitoring the logs on the UE side as described in III-C. To this end, our tester sends test case messages by pretending to be a victim UE by spoofing the identity of the victim UE in both the

RRC and *NAS* layers[3]. Then, we observed the behavior of the victim UE by utilizing SCAT whenever our tester executed a test case. We automated this procedure by establishing a communication channel between the tester and the victim UE. ① When the victim UE is ready, the tester executes a test case and sends a notification to the victim UE. ② Upon receiving the notification, the victim UE sends ping requests to a public website (i.e., www.google.com) and checks the ping response. If it is *Network is unreachable*, the test case is labeled as "Caused de-registration". ③ Lastly, we analyze the logs on the UE side and classify each test case based on our decision tree. Fig. 11 in the Appendix shows an actual screenshot of running our uplink test. We carried out uplink tests by considering the following three cases to validate the effectiveness of each test case. A victim and a tester are located in (1) the same *cell*, (2) different *cells* but in the same *eNB*, and (3) different *eNBs* but in the same *MME* pool.

2) Inspecting commercial mobile devices. Unlike the above setups, the experimental setup for commercial mobile devices is simpler. In this case, the tester acts as a *rogue LTE network* implemented on top of openLTE [10] and the victim UE is connected to the host PC to capture the control plane logs. The automated test operation is as follows. ① Once a test case is submitted as input to the *rogue LTE network*, it waits until the victim tries to connect to our network. ② When the victim sends an `RRC Connection request`, our *rogue LTE network* operates as specified in the test case and notifies the victim side that the test has been executed. ③ Upon receiving the notification, the victim side saves the control plane messages. In addition, in case the victim falls into an invalid state from which it cannot recover to the normal state, the host PC forces the victim UE to reboot by sending an Android Debug Bridge (ADB) command [24].

Our implementation consists of 3,470 lines of code (LoCs) identified via in-depth analysis of more than 90K lines in 537 files of open source tools [9], [10], [23]. This includes 1,937 LoCs of C++ for the test input generator, 1,390 LoCs of C++ for the uplink/downlink tester and 143 LoCs of Python for the communication channel between the tester and the victim UE.

V. Test Results

We carefully conducted a dynamic test on two Tier-1 carrier networks (with three different MMEs and three eNBs) and commercial UEs (by including three different baseband vendors). The test results for each test case (Table IV in the Appendix) indicate that we uncovered 51 vulnerabilities across different target network components and device vendors. We confirmed the validity of most of our findings in the operational network by interviewing counterparts from the carriers. For clarity purposes, we explain the results of our findings and their root cause analysis by categorizing them into five types.

A. Initial RRC procedure is not protected

Test case observation. The test on property 1 in the *RRC* layer indicated that the *RRC Connection* procedure is neither encrypted nor integrity protected; thus, all the messages that belong to the *RRC Connection* procedure are classified as case 1 or 2 as listed in Table IV. Therefore, an adversary could exploit these messages to spoof the contents or deny the connection of the victim UE during the *RRC Connection* procedure. For example, if an adversary changed the contents of the *ueIdentity* field in the `RRC Connection request` to victim's S-TMSI, she could deceive the eNB and lead it to believe that the victim UE is currently in the *RRC CONNECTED* state despite the victim being in the *RRC IDLE* state.

Root cause analysis. According to 3GPP standards [19], the *initial authentication* procedure between the UE and MME occurs via the *NAS protocol*, which is processed after the *RRC Connection* procedure. Thus, any eNB first allows the UE's `RRC Connection request` and leaves the authentication procedure to an MME. When the authentication procedure fails due to an invalid response from the UE (e.g., in the case of an unsubscribed user or illegal UE), the MME sends a `UE Context release request` message to the eNB to release the existing *RRC Connection* of the abnormal user. Therefore, by design, even illegal users who are not legitimately subscribed to a particular carrier would be able to connect to the eNBs of this carrier. However, they would be unable to maintain the *RRC Connection* longer than several seconds because their device would be unable to respond to the `NAS Authentication request` correctly. Despite this limitation, we determined that an adversary who only has the ability to connect to an eNB (but cannot proceed with the connection to achieve full registration), could still perform critical attacks such as blocking any *RRC Connections* to a target eNB (Section VI-A), disconnecting current *RRC Connections* (Section VI-B), and blocking a target user's incoming services (Section VI-B). These attacks are mainly rooted from the 3GPP standard such that the initial *RRC Connection* procedure is unprotected and can be abused by an adversary. Detailed descriptions of each type of attack are presented in Section VI.

B. Invalid uplink NAS plain messages cause failures

Test case observation. As explained in Section IV, we carried out the uplink tests for three different cases: A victim UE and a tester UE[4] are located in (1) the same *cell*, (2) different *cells* but in the same *eNB*, and (3) different *eNBs* but in the same *MME* pool. Note that tests relating to these cases are also described in Section V-C and V-D. The results indicated that an adversary could send invalid plain requests through an *RRC Connection* spoofed as the victim UE. Interestingly, three MME types have different problematic behavior upon receiving our invalid plain requests. For example, when the tester sends a crafted plain `NAS Attach request` to the MME_1 and MME_3, they removed the connection of the victim UE and sent a release command to the tester, thereby implicitly detaching the victim UE from the network (case 3). In this case, the victim UE does not receive notification of its disconnection from the service unless it initiates the transmission of uplink data by sending a `NAS Service request`. When the victim UE receives `Service reject` with the cause *Implicitly detached*, it has to proceed with an initial *Attach* procedure to reconnect to the LTE network and this results in several seconds of service disconnection. For another de-registration case, upon receiving the plain `NAS Detach request`, all three MMEs immediately de-registered the victim UE and replied `NAS Detach accept` to the tester UE (case 1). Besides, we also confirmed that

[3]Detailed spoofing techniques for each case are presented in Section V.

[4]Tester could be a UE or a network as explained in Section IV.

TABLE II. SUMMARY OF MAJOR FINDINGS

Property	Vulnerability/Attack	Implications	Detection	Root cause
P1	RRC Connection manipulation	Connection resource depletion on eNB/Cell	Case 2	Design flaw
	RRC connection spoofing	• (Blindly) Denial of incoming service when a UE is in IDLE • Disconnection of current radio connection when a UE is in CONNECTED	Case 1	Design flaw Implementation flaw
	Improper handling of uplink plain NAS messages	• Implicit de-registration of a UE • SMS phishing due to non-security protection	Case 1, 2, 3	Implementation flaw
P2	Improper handling of incorrectly integrity protected NAS messages	• Implicit de-registration of a UE • SMS phishing due to non-integrity verification • Content manipulation for security protected messages	Case 1, 2, 3	Implementation flaw
	Improper handling of replayed NAS messages	• Implicit de-registration of a UE • SMS phishing due to non-replay protection • Deactivation of selected service (e.g., data or voice) • Fake location update of a UE		
P3	Bypass RRC Security mode command	Eavesdropping & manipulation of user data	Case 2	Implementation flaw

MME_2 processed plain `NAS Uplink NAS transport` without any protection (case 2). In this case, an adversary could exploit this MME_2 to launch an SMS phishing attack to any users without being charged if they are subscribed to any carriers that have a roaming agreement with the aforementioned vulnerable carrier. In Table III, the affected plain initial messages are identified by **(P)**.

Root cause analysis. According to the 3GPP standard, the UE sends initial requests without security protection only when it has no valid security context as in certain cases such as an expired context timer or unexpected errors. In case the UE has no valid security context (i.e., a session key for encryption and integrity protection), the MME needs to create a new valid security context for further steps to achieve full registration. The first step toward creating a victim's new valid security context is an *Authentication* procedure between the UE and the MME. Therefore, upon receiving the spoofed initial requests without protection from our tester (acting as a malicious UE), the MME would be required to perform an *NAS Authentication* procedure to confirm whether this abnormal message originated from a legitimate user without processing the message or immediately disconnecting existing UE connections. Accordingly, as we assume that the adversary UE has no valid security context, it cannot perform the *Authentication* procedure correctly, whereby, an existing connection for legitimate UE would not be affected. Therefore, if this was implemented correctly in MME, the victim UE should not have been implicitly detached from the network, as observed in our test case. In conclusion, we found that all three MMEs did not handle invalid plain requests correctly.

C. Non-integrity checking makes spoofing attack possible

Test case observation. For the validation of the first case in property 2, our tester generated security protected `NAS messages` with an incorrect MAC and sent them to either the MME or the UE to inspect whether they appropriately verify the MAC. Accordingly, we observed three different kinds of inappropriate behavior (case 1, 2 and 3) in different MMEs upon receiving a message with an invalid MAC. The MME that belongs to case 1 and 2, did not verify the MAC, and simply accepted the invalid message. For example, when the tester sent an `Uplink NAS transport` message with an invalid MAC, the MME accepted this as being valid; hence, the SMS is sent to the destination UE. On the other hand, another MME that belongs to case 3 verified the MAC when it received the message. However, the receipt of a message with an incorrect MAC value resulted in the MME de-registering the existing connection of the victim UE regardless of the type of received message. In this case, the `Uplink NAS transport` with incorrect MAC caused de-registration of the victim UE and did not transmit the SMS message to the destination UE. Messages affected by an invalid MAC are marked with **(I)** in Table III. We present detailed scenarios that exploit improper handling of messages with an invalid MAC in Section VII.

TABLE III. EXPLOITED NAS MESSAGES IN TWO MME TYPES

Exploited NAS Messages	Implications		
	MME_1	MME_2	MME_3
Attach Request	DoS (**P**, **I**, **R**)	×	DoS (**P**, **I**, **R**)
TAU Request	DoS (**P**, **I**, **R**)	×	DoS (**I**), False location update (**R**)
Service Request	Spoofing (**R**)	×	Spoofing (**R**)
Uplink NAS Transport	DoS (**P**, **I**), SMS phishing (**R**)	SMS phishing (**P**, **I**, **R**)	-
PDN Connectivity Request	DoS (**I**)	×	DoS, DosS (**R**)
PDN Disconnect Request	DoS (**I**), DosS (**R**)	×	DosS (**R**)
Detach Request	DoS (**P**, **R**)	DoS (**P**, **I**, **R**)	DoS (**P**, **I**, **R**)

DosS: Denial of selective Service, **P:** Plain, **I:** Invalid MAC, **R:** Replay

Root cause analysis. According to the 3GPP standard [15], both the UE and the MME have to verify the integrity of the `NAS message` once a valid security context exists between the UE and the MME. However, we speculate that device vendors misunderstood the acceptance of `NAS messages` without integrity protection by the MME in certain exceptional situations (e.g., when the UE sends the message before security is activated for the initial message. In our test scenario, because both the victim UE and the serving MME have the valid security context, the MME should have verified the integrity of every received message. Failed verification should have resulted in the MME dropping or rejecting the received message while maintaining the existing connection of the victim UE. Therefore, these cases obviously constitute implementation mistakes for all three MMEs.

D. Replayed messages are accepted

Test case observation. While validating the second case in property 2, we also inspected whether the receiving entity (the network component or the UE) verifies the sequence number to prevent a message replay attack. The results confirmed that some crafted `NAS messages` were accepted as being valid by both the MMEs and the UEs because the tester subsequently received expected reply messages (case 1). For instance, when the tester sent a replayed `NAS PDN disconnect request` for disconnecting a specified existing data bearer, the tester received a security protected `NAS message`[5] whereas the victim UE blindly lost the connection of the data bearer (case 1).

[5]We could not exactly identify this message as it is encrypted, but it is assumed to be a `NAS Deactivate EPS bearer context request` when considering the message length.

In addition, when the tester sent replayed `NAS TAU request` to MME_3, it replied `NAS TAU accept` to the tester, which means that MME_3 falsely updated the Tracking Area (TA) of the victim UE (case 2). However, when the tester sent the replayed `TAU request` to MME_1, it immediately de-registered the existing connection of the victim UE whereas the tester received an `RRC Connection release` (case 3). The uplink messages affected by a replay attack are marked by **(R)** in Table III. For downlink `NAS messages`, we confirmed that a HiSilicon baseband accepted the replayed message, allowing an adversary to perform a message spoofing attack (case 2). The affected messages are listed in Table IV in the Appendix.

Root cause analysis. The 3GPP standard [15] requires both the MME and UE to support replay protection for security protected `NAS messages` and it provides a detailed method of replay protection for the vendors. However, the same document requires the receiving entity to use the *NAS* sequence number included in the received message for eight LSBs of the counter when verifying the integrity of the received `NAS message`. Moreover, when the integrity verification succeeds, the receiving entity should update its local counter with the value of the received sequence number. In this integrity verification method, a message replay is possible unless the 28 LSBs of the local counter is different from the counter used when calculating the message integrity. Therefore, the integrity verification method specified in the standard contradicts the security requirement that the replay protection should be supported for the `NAS messages`. As a result, all three MMEs were vulnerable to replay attack at least for one `NAS message` and one of our target basebands accepted replayed messages.

E. Security procedure can be bypassed

Test case observation. In terms of property 3, we checked three test cases: (1) skip the key agreement procedure in the *RRC* layer to nullify the security context of *RRC* and user data, (2) skip the key agreement procedure in both the *RRC* and *NAS* layer to nullify the security context of the entire control plane and data plane, and (3) skip all the security procedures in AKA in both *RRC* and *NAS*. Consequently, only the first case succeeded for our target mobile devices. If an adversary was to exploit this case, they would be able to spoof the `RRC messages` to obtain the private information of the UE, and eavesdrop the user's communication. The attack procedures and implications are described in Section VIII in detail.

Root cause analysis. As noted in the specification, the security procedure is a mandatory step for protecting the communication between the UE and the network. However, we confirmed that some commercial devices (using a Qualcomm baseband) do not follow the specifications; thus, they allow the security key negotiation in the *RRC* layer to be omitted. Therefore, this vulnerability is rooted in the implementation flaw in commercial baseband chipsets.

F. Summary and responses from the carriers

By sending carefully crafted messages, we were able to uncover 51 vulnerabilities in the *design* and *implementation* of UE, eNB, and MME. From the design perspective, unprotected initial procedures are vulnerable to spoofing attacks. We also confirmed that the behavior of operational MMEs and commercial devices in response to our test cases differs across vendors and even message types within one vendor. We reported all our findings to the corresponding carriers. Then, we validated and communicated all network side test cases with the corresponding vendors together with the carriers. As a result, we received a response from one vendor (MME_3) confirming that all the vulnerabilities we discovered are valid and that they are preparing patches for each problematic case. The following sections describe how our findings can be exploited for attacks by categorizing the target entities as eNB, MME, and UE, in Section VI, VII, and VIII, respectively.

VI. Attacks exploiting eNB

A. BTS resource depletion attack

Every commercial eNB has a maximum capacity of active user connections based on their hardware and software specifications. The purpose of the *BTS resource depletion* attack is to deplete this capacity of the active *RRC Connections*, thereby preventing other users from connecting to the target eNB.

1) Adversary model: This attack targets a commercially operating eNB. An adversary could obtain the connection information of the target eNB by passively listening to the broadcast messages similar to other normal devices.

2) Attack procedure: The adversary repeatedly performs *Random Access* and generates *RRC Connections* in order to increase the number of active *RRC Connections* as depicted in Fig. 5(a). In a normal situation, immediately after the *RRC Connection* is established, an initial *NAS Connection* procedure proceeds through either an `NAS Attach request` or `NAS Service request` piggybacked on an `RRC Connection complete` message. In our attack, the adversary sends the `NAS Attach request` with an arbitrary user IMSI. Unlike the normal procedure, once the adversary receives the `NAS Authentication request`, it restarts *Random Access* to establish a new *RRC Connection*. The reason the adversary does not reply to the `NAS Authentication request` from the MME is to sustain the established *RRC Connection* while the MME waits for a valid `NAS Authentication response`. If the adversary replies with an invalid `NAS Authentication response`, it causes immediate *RRC Connection release*. One consideration for the attack to succeed is that the number of newly established *RRC Connections* has to be greater than the number of existing *RRC Connections* that are released.

3) Implementation: We used one USRP B210 [25] for the software radio transceiver, and srsUE [9] to implement a malicious UE. To repeat the *RRC Connection* procedure continuously with different C-RNTIs, we modified the srsUE to restart another *Random Access* procedure whenever it receives an `NAS Authentication request` rather than replying with an `NAS Authentication response`. If several `RRC Connection requests` are sent with the same C-RNTI, the eNB processes this as repeated requests for the same *RRC Connection*, which is not our adversary's goal.

4) Validation: Because attacking commercially operating eNBs can affect legitimate users, we performed our *BTS resource depletion attack* against a COTS femtocell [26] connected to our testbed EPC network implemented on OpenAirInterface (OAI) [11]. We mainly attempted to determine the number of fake *RRC Connections* that could be established using one USRP device. This is accomplished by verifying active *RRC Connections* of our femtocell

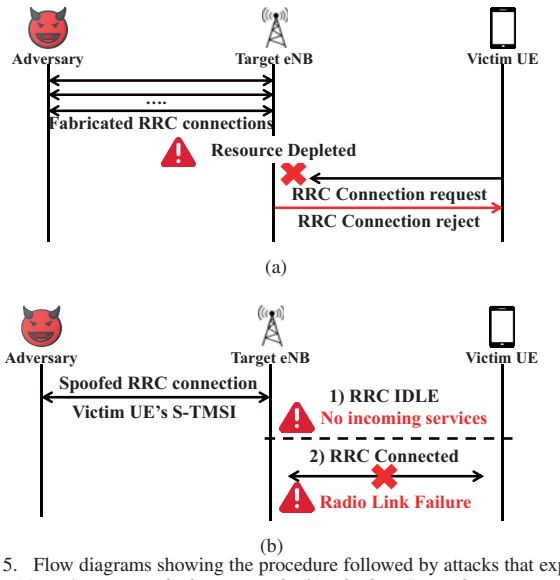

Fig. 5. Flow diagrams showing the procedure followed by attacks that exploit eNB: (a) *BTS resource depletion attack*, (b) *Blind DoS attack*

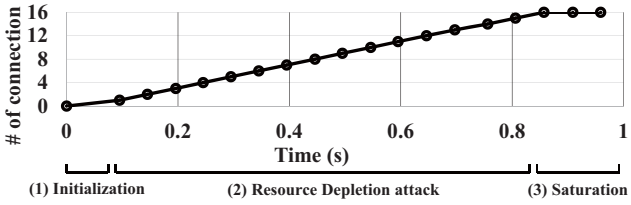

Fig. 6. Number of active *RRC Connections* in *BTS resource depletion attack*.

using an Airscope [27], which provides over-the-air user information by decoding the communication channels in the physical layer of LTE. Fig. 6 shows that the number of active *RRC Connections* increases until it reaches the maximum capacity of the femtocell, namely 16 active connections in the case of our target femtocell. Therefore, once an adversary has generated 16 *RRC Connections*, the femtocell rejects all subsequent `RRC Connection requests` either from the adversary or from the legitimate UE, as shown in Fig. 7. When demonstrating the attack, it took 0.762 s to establish 16 *RRC Connections*, and we could establish 20 *RRC Connections* per second. Therefore, an adversary would be able to create 200 *RRC Connections* in case the operational eNB was to wait 10 s for inactive *RRC Connections* to be released. We confirmed with the carrier that an attack of this nature would affect an operating eNB. In addition, the carrier suggested an even more serious scenario. If the adversary was to include "emergency" as an *establishment cause* in an `RRC Connection request`, it would even release existing *RRC Connections*, if no additional *RRC* resource was available.

B. Blind DoS attack

Unlike the aforementioned attack that denies multiple users in an eNB, the *Blind DoS attack* denies a targeted UE by establishing *RRC Connections* spoofed as the victim UE.

1) Attack model: The attacker performs the attack within the area covered by the victim's serving eNB. The attacker also

```
GSMTAP/NAS-EPS      151 Attach request, PDN connectivity request
LTE_RRC_UL_CCCH      76 RRCConnectionRequest
LTE_RRC_DL_CCCH      72 RRCConnectionReject
- LTE Radio Resource Control (RRC) protocol
  - DL-CCCH-Message
    - message: c1 (0)
      - c1: rrcConnectionReject (2)
        - rrcConnectionReject
          - criticalExtensions: c1 (0)
            - c1: rrcConnectionReject-r8 (0)
              - rrcConnectionReject-r8
                waitTime: 1s
```

Fig. 7. Victim UE receives `RRC Connection reject` during *BTS resource depletion attack*.

knows the victim's S-TMSI that can be obtained in three ways:

- An adversary who has knowledge of the victim's phone number or accounts on social media (such as Facebook and Whatsapp) could obtain the victim's S-TMSI by performing a silent *Paging* attack [7], [28].
- An adversary located in the vicinity of the target user could operate a *rogue eNB* to obtain the `NAS TAU request` of the victim UE. This request contains the S-TMSI of the victim UE. As soon as this message is received, the adversary turns off the *rogue eNB* to enable the victim UE to recover the LTE service by connecting to a carrier network.
- The adversary sniffs the *RRC Connection* procedure of the target UE to obtain the S-TMSI of the target UE as specified in the `RRC Connection setup` [5].

2) Attack procedure: The adversary carries out the attack by establishing an *RRC Connection* spoofed as the victim UE (Fig 5(b)). This can be achieved by inserting the S-TMSI of a victim UE in the *ueIdentity* field of the `RRC Connection request`. This attack can be launched with no special efforts to circumvent the deployed security measures because, by design, the *RRC Connection* procedure has no security mechanisms to conceal the content or authenticate the message sender.

3) Implementation: We used one USRP B210 [25] for the software radio transceiver, and srsUE [9] for the software LTE UE. We slightly modified the srsUE to add the S-TMSI of the target UE to the *ueIdentity* field of the `RRC Connection request`. In addition, for the same reason as for the *BTS resource depletion attack*, the attacker device does not respond to the `NAS Authentication request`.

4) Validation: We validated the attack on commercial eNBs located in the vicinity of our laboratory building. To exclude innocent victims, we only utilized the S-TMSI of our mobile phone as the identity of the victim UE. The impact of the attack was assessed by separating it into two types according to the *RRC Connection* state of the victim UE.

- The victim UE is in the *RRC IDLE* state: The UE attempts to establish an *RRC Connection* when *Paging* notifies the incoming services or the UE has outgoing service traffic. If the adversary establishes an *RRC Connection* spoofed as the victim UE, the serving eNB saves the *RRC* state of the victim as *RRC CONNECTED* and notifies the serving MME of this change. Thus, the MME does not trigger *Paging* to any eNBs, despite the existence of incoming services for the victim. In this case, the victim is blindly disconnected from the serving eNB until it attempts to establish a new *RRC Connection* for outgoing traffic from the application services. From the user's perspective, both incoming data and voice are blocked without any notifications of disconnection.
- The victim UE is in *RRC CONNECTED* state: When the

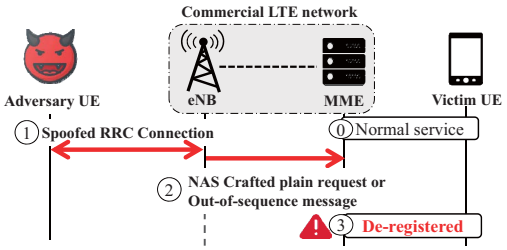

Fig. 8. *Remote de-registration attack* using either the crafted plain requests, invalid security protected messages or replayed messages.

adversary establishes a spoofed *RRC Connection*, the existing *RRC Connection* of the victim UE is released on the eNB without any notifications to the victim. In this case, the UE continues to communicate with the serving eNB but it fails because the radio bearer was already released. Once communication has failed several times, the UE falls into the Radio Link Failure (RLF) state, thus it sends an `RRC Connection reestablishment request`. However, the serving eNB rejects this request because it is already released. Upon receiving the reject message, the UE attempts to carry out the *NAS TAU* procedure and reestablishes the connection by sending an `NAS Service request`. Eventually, the UE is disconnected from the network during the re-registration procedure explained above. The time required for re-registration was approximately 0.5 s, thus if the adversary was to continuously establish the spoofed *RRC Connection* every 0.5 s, the victim would remain in the disconnected state permanently.

Note that we validated this attack on three different eNB vendors. When the victim UE is in *RRC IDLE*, the attack succeeded for all eNBs. However, when the victim UE is in *RRC CONNECTED*, two of our target eNBs were affected by the attack whereas the other eNB was not. To summarize, a *Blind DoS attack* could block incoming services of a victim UE in *RRC IDLE* state by deceiving a serving eNB, which believes that the UE is in *RRC CONNECTED* state. In addition, the victim UE was permanently prevented from using the LTE service by two vendors because those eNBs only maintain a single *RRC Connection* for a single S-TMSI of a UE.

VII. Attacks Exploiting MME

In this section, we explain the way in which an adversary could exploit the vulnerabilities in operational MMEs. This includes remote de-registration of the legitimate UEs, denying a selective service, and SMS phishing without subscribing to the service. Note that all attacks described in this section occur as a consequence of the adversary exploiting a spoofed *RRC Connection* when sending `NAS messages`.

A. Remote de-registration attack

During our experiments, we discovered that operational MMEs have several implementation flaws that cause them to unnecessarily de-register the victim UE without notification. The detailed attack scenario is as below.

1) Adversary model: An adversary should be able to send malicious `NAS messages` to the MME in which the victim UE is registered. Typically, an MME manages a number of eNBs which are distributed throughout large geographical regions. The adversary also knows the S-TMSI of the victim UE. Especially, for an attack that exploits message replay, the adversary would have to capture the corresponding message before launching the attack. There are two ways to obtain a control plane message of the victim UE.

- An adversary could operate a *rogue LTE network* to capture the control plane messages of the victim UE while relaying these messages between the UE and the network [4], [5].
- An adversary could install a malicious app with control plane message logging functionality [29] on the UE.

We implemented the attack by utilizing a *rogue LTE network* to capture the control plane messages of the victim UE. In this case, the adversary cannot decrypt the messages. However, we could correctly identify the type of encrypted messages by only checking the order and length of the messages.

2) Attack procedure: As shown in Fig. 8, ① an adversary first establishes an *RRC Connection* spoofed as the victim UE (using the UE's S-TMSI). ② The adversary sends a crafted initial plain request, invalid security protected message, or replayed message to the MME serving the victim[6]. In this case, once the adversary sends the message through the spoofed *RRC Connection*, the serving eNB forwards the message to the MME serving the victim by checking the S-TMSI. ③ The MME processes the message it receives from the adversary inappropriately. Consequently, the MME de-registers the connection of the victim UE without any notification to them.

3) Implementation: We implemented the adversary using the srsLTE UE stack [9]. She sends the vulnerable `NAS messages` as soon as the spoofed *RRC Connection* is established.

4) Validation: We demonstrated the *Remote de-register attack* against an operational LTE network by exploiting either invalid plain messages, security protected messages or replayed messages. We confirmed that an adversary could perform this attack by connecting to any eNBs able to communicate with the same MME serving the victim UE. An interview with a counterpart in the carrier revealed that an eNB might communicate with any MMEs regardless of the geographical regions and that this depended on the operational policy of the particular carrier. In this case, an adversary would be able to remotely de-register arbitrary users subscribed to the carrier regardless of the user's location only if the adversary succeeded in obtaining the valid GUTIs. Note that obtaining a valid GUTI is not difficult as discussed previously. The `NAS messages` that could be used to carry out this attack are listed in Table II. A notable case for message replay is that, once the MME accepts replayed a `NAS PDN disconnect request`, the adversary can selectively deny the user's service (e.g., the adversary blindly disconnects the data service of the victim UE whereas the voice service continues to be available).

B. SMS phishing attack

1) Adversary model: In this scenario, the adversary sends an SMS message to victim UE_1 by spoofing the message sender using the phone number of victim UE_2. To this end, the adversary knows the S-TMSI of UE_2 to spoof the sender. The phone number of UE_1, to which the actual SMS message is sent, is also known. In addition, we assume that the target LTE network provides the SMS through the *NAS* layer.

2) Attack procedure: ① The adversary starts by establishing

[6]The adversary chooses the messages depending on the vulnerabilities found for each carrier.

a spoofed *RRC Connection* using the S-TMSI of UE_2. Then, ② SMS content is generated and included on an `NAS Uplink NAS transport`. ③ Immediately after the *RRC Connection* is established, the adversary sends the generated `NAS Uplink NAS transport` to the serving MME. ④ Upon receiving the message, the MME transmits this manipulated SMS to UE_1.

3) Implementation: We implemented this attack by modifying the srsLTE implementation. In particular, we simply added the functionality to support *SMS over NAS*.

4) Validation: Our test results confirmed that we successfully carried out this attack on the carrier as MME_1 does not verify the sequence number of the `NAS Uplink NAS transport` message, whereas MME_2 accepts all invalid messages (plain, invalid MAC, and replay).

VIII. Attacks Exploiting UE: AKA Bypass Attack

1) Adversary model: The adversary is located sufficiently close to the victim UE to trigger handover from an existing eNB to the adversary's *rogue LTE network*. To this end, the *rogue LTE network* transmits an LTE signal with higher transmission power than commercial eNBs. Additionally, the adversary would have to know the list of Tracking Areas (TAs) to masquerade the *rogue LTE network* as a commercial one. A valid TA Code (TAC) can easily be captured in two ways:

- If the adversary is subscribed to the same carrier as the victim, the list of TAs can be obtained by checking control plane messages such as `Attach Accept`.
- If the adversary only owns a *rogue LTE network*, she first chooses a TA randomly. Once the target UE connects to the *rogue LTE network*, it sends a `TAU request` as the TA of the connecting network is not on its list of TAs. Upon receiving the `TAU request` from the UE, the adversary can obtain the previous TAC of the UE by parsing the request. Note that the `TAU request` is only integrity protected.

2) Attack procedure: As shown in Fig. 9, the adversary builds the *rogue LTE network* and configures its operating parameters such that they are identical to the victim's operational network. ① If the transmitting power of the *rogue eNB* is higher than the serving eNBs, the victim in the *RRC IDLE* state resynchronizes to the adversary's eNB. In this case, the UE does not trigger the *NAS TAU* procedure as the TAC of the *rogue eNB* is contained in the TA list of the victim UE[7]. Thus, ② when the UE is transmitting outgoing data (i.e., by calling someone or browsing the Web) or is receiving *Paging* from the *rogue LTE network*, it establishes an *RRC Connection* and sends an `NAS Service request`. Upon receiving a valid integrity protected `Service request` from the UE, the normal eNBs perform an *RRC Security mode* procedure to regenerate the cryptographic keys for the *RRC* layer and user data. However, ③ our *rogue LTE network* omits this procedure and immediately prepares to create a radio tunnel (also known as a Data Radio Bearer (DRB)) by sending a plain `RRC Connection reconfiguration`. Upon receiving this request, ④ the UE creates the DRB and also replies with a plain `RRC Connection reconfiguration complete` message. Finally, ⑤ the UE transmits and receives unprotected user data through this tunnel with the *rogue LTE network* without receiving any notification.

[7]In LTE, the UE receives a TA list that contains adjacent TACs when it attaches to the network. The UE performs a *Tracking Area Update* procedure when it moves to a new TAC that is not included in its TA list.

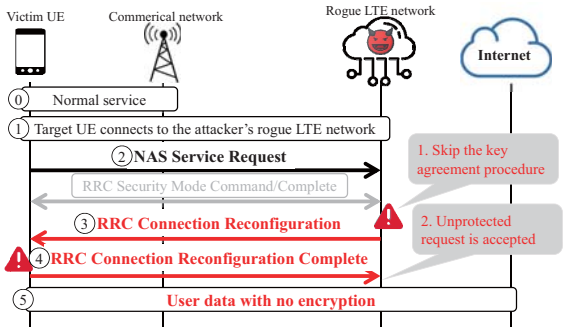

Fig. 9. Procedure of *AKA Bypass attack*.

3) Implementation: We used one USRP B210 for the radio transceiver, and openLTE [10] for the *rogue LTE network*. The adversary's *rogue LTE network* does not negotiate the security algorithm for the *RRC* layer and user data in response to a connection request from the victim UE. Further, the *RRC Reconfiguration* procedure is performed without security protection, which is against the security guidelines noted in the standard [19].

4) Validation: We validated that the *AKA Bypass attack* can nullify the existing encryption of the user data of an existing UE on multiple smartphone models (e.g., the LG G2 and Samsung Galaxy S4/S5, all of which use Qualcomm basebands). Because our *rogue LTE network* configured the TAC as in the TA list of the victim UE, this UE did not trigger a `TAU request` when it first synchronized with our eNB. Interestingly, some models frequently initiated the `NAS TAU request` during the attack period. However, if the *rogue LTE network* does not reply upon receiving the request from the UE, the victim UE reconnects by sending the `NAS Service request`. This proves that our attack is still effective even in that situation.

IX. Countermeasures

Attacks exploiting eNB. In the case of a *BTS resource depletion attack*, it is impossible for an eNB to distinguish the adversary's `RRC Connection requests` from benign `RRC connection requests`. A possible mitigation to this attack could be to reduce the *inactivity timer* value to allow an *RRC Connection* that is unresponsive to the `Authentication request` to expire. Although it does not constitute a fundamental solution, it can weaken the impact of this attack as it minimizes the number of fake *RRC Connections* the adversary can establish. However, if the carrier configures the *inactivity timer* inappropriately short, the UEs may perform frequent *RRC Connection* procedures. Accordingly, this would increase the signaling load on both the eNB and MME sides. On the other hand, a possible mitigation for a *Blind DoS attack* might be to re-assign the S-TMSI when a number of `RRC Connection requests` using the same S-TMSI are received. According to the 3GPP standard, an MME can trigger reallocation of the S-TMSI in two ways. The first is to directly send a security protected `NAS GUTI reallocation command` to the UE. However, this would not prevent a *Blind DoS attack* because the message would not be received by the UE during the attack. Another approach would be to broadcast *Paging* with the IMSI of the UE. As the *Paging* is broadcast over the entire area covered by the cell, the UE would receive it and

initiate the *Attach* procedure with the IMSI upon receiving the *Paging* message, which would increase signaling overhead.

Attacks exploiting MME and UE. As discussed in Section V, both the *Remote de-register attack* and *SMS phishing attack* are rooted from incorrect implementation of the operational MMEs. Thus, these MMEs should be carefully implemented by strictly following the 3GPP standard. The *AKA bypass attack* is also rooted in the UE handling the mandatory security procedure incorrectly. Therefore, the UE should not proceed with any control plane procedures before completing the mandatory security procedure successfully.

X. Related work

A. Identifying unexpected behavior

Several previous studies were conducted to identify unexpected failures and performance degradations caused by either design flaws or implementation bugs [30]–[32]. Tu et al. [30] designed a model with which to analyze logical problems in inter-layer communication between LTE and 3G. Hong et al. [31] developed a control plane analysis tool to identify implementation/configuration flaws by conducting comparative analysis of commercial logs. Jia et al. [32] analyzed performance problems in Voice over LTE (VoLTE). However, all these studies attempted to identify performance problems by utilizing passively collected commercial logs. In contrast, we focused on finding security bugs by utilizing malicious input.

B. Security problems

MitM attack. Many previous studies, [7], [33]–[40] employed a *rogue BTS* in a 2G/3G network. However, the Man in the Middle (MitM) attack in LTE networks received less attention [4], [5], [20]. Rupprecht et al. [20] showed that an LTE dongle could be used for eavesdropping and tampering if the dongle incorrectly allows *null integrity* to both the control and data plane. Hussain et al. [4] demonstrated an *Authentication relay attack* to eavesdrop a victim UE's data communication if the carrier uses *null encryption* to the data plane. In addition, Rupprecht et al. [5] showed that the IP address the DNS server includes in a packet could be manipulated when the counter mode was used for user data encryption in LTE. The former can be used for eavesdropping only if a carrier allows *null encryption*, whereas the latter enables DNS hijacking. Unlike the above studies, omitting the *Security mode command* enables the user data to be communicated in plain text and can even be manipulated regardless of the integrity/encryption policy of the carrier.

DoS attack. Previous studies introduced DoS attacks that exploit vulnerabilities in LTE control plane procedures [4], [7], [41]–[44]. Shaik et al. [7] presented DoS attacks using plain reject messages (`NAS TAU reject`, `Service reject` and `Attach reject`). Raza et al. [41] demonstrated two types of DoS attacks that were able to detach a user from the network: the first uses a plain `NAS Detach request` message and the other uses *Paging* with the user's IMSI. Both studies showed that certain unprotected plain messages may cause denial of service to users. In this paper, we defined the desired security properties in an LTE network and *systematically* crafted the messages that can pose a threat to each property. Consequently, we tested (almost) all *RRC* and *NAS* uplink/downlink messages and showed that DoS attacks are possible with other message types that were previously unknown. Unlike these two studies, Hussain et al. [4] formally modeled the control plane procedures based on the LTE standard and inspected the model to identify LTE design problems. Among the vulnerabilities introduced, they presented several DoS attacks using plain `NAS messages`. However, as they inspected the LTE standard (focusing on *NAS*), their model cannot disclose the vulnerabilities resulting from design bugs in the *RRC* layer or incorrect implementations in operational networks. In contrast, our approach can be used to uncover the vulnerabilities resulting from both the design flaws and incorrect implementations in the control plane protocols. Prior to these studies, investigations of DoS attacks in 2G and 3G networks were reported [45]–[50].

XI. Concluding remarks and future work

In this study, we investigated potential security problems by dynamically testing the control plane components in an operational LTE network. The procedure of semi-automated dynamic testing consists of three steps: creating security properties based on specification analysis, generating and conducting test cases that violate the security properties, and classifying a problematic case. As a result, LTEFuzz successfully identified 15 previously disclosed vulnerabilities and 36 new vulnerabilities in *design* and *implementation* among the different carriers and device vendors. The findings were categorized into five vulnerability types. We also demonstrated several attacks that can be used for denying various LTE services, sending phishing messages, and eavesdropping/manipulating data traffic. We performed root cause analysis of the identified problems by reviewing the related standard and interviewing collaborators of the carriers.

In conclusion, LTEFuzz is an effective tool to discover design and implementation vulnerabilities caused by carriers and device vendors. Our findings were interesting in two respects: 1) even within a single carrier, two MMEs (possibly from different vendors) have different vulnerabilities, and 2) two MMEs (in two carriers) manufactured by a single device vendor have different vulnerabilities. This shows that neither the device vendors nor the carriers have checked the security of their network components carefully. In addition, LTEFuzz was able to uncover vulnerabilities in baseband chipsets from Qualcomm and HiSilicon, who ranked number 1 and 4 in market share in 2017 [51]. We plan to privately release LTEFuzz to these carriers and vendors in the near future. A public release is not planned as LTEFuzz can be used for malicious purposes. Because of space constraints, we present the limitations of LTEFuzz and future work in the Appendix.

Acknowledgment

We would like to thank the anonymous reviewers and our shepherd, Thorsten Holz, for their insightful comments and suggestions to improve the paper. This research was supported by the MSIT (Ministry of Science, ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2018-2015-0-00403) supervised by the IITP (Institute for Information & communications Technology Promotion).

REFERENCES

[1] GSA, "LTE subscriptions to 1Q 2018," https://gsacom.com/paper/lte-subscriptions-to-1q-2018-yoy-growth.

[2] 5gamericas, "The number of LTE network operators," https://www2.telegeography.com/globalcomms-database-service.

[3] 3GPP, "LTE," http://www.3gpp.org/technologies/keywords-acronyms/98-lte, 2017.

[4] S. R. Hussain, O. Chowdhury, S. Mehnaz, and E. Bertino, "LTEInspector: A Systematic Approach for Adversarial Testing of 4G LTE," in *Proceedings of the Network and Distributed Systems Security (NDSS)*, 2018.

[5] D. Rupprecht, K. Kohls, T. Holz, and C. Pöpper, "Breaking LTE on Layer Two," in *IEEE Symposium on Security & Privacy (SP)*. IEEE, 2019.

[6] S. F. Mjølsnes and R. F. Olimid, "Easy 4G/LTE IMSI Catchers for Non-Programmers," in *International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security*. Springer, 2017.

[7] A. Shaik, R. Borgaonkar, N. Asokan, V. Niemi, and J.-P. Seifert, "Practical Attacks Against Privacy and Availability in 4G/LTE Mobile Communication Systems," *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2016.

[8] H. Lin, "LTE REDIRECTION: Forcing Targeted LTE Cellphone into Unsafe Network," in *Hack In The Box Security Conference (HITBSecConf)*, 2016.

[9] "srsLTE." [Online]. Available: https://github.com/srsLTE/srsLTE

[10] "openLTE." [Online]. Available: http://openlte.sourceforge.net

[11] "OpenAirInterface." [Online]. Available: http://www.openairinterface.org

[12] 3GPP. TS 23.003, "Numbering, addressing and identification," 2017.

[13] 3GPP. ETSI TS 43.020, "Technical Specification Group Services and system Aspects; Security related network functions," 2017. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/43020.htm

[14] 3GPP. TS 24.008, "Mobile radio interface Layer 3 specification; Core network protocols; Stage 3," 2017.

[15] 3GPP. TS 24.301, "Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3," 2017.

[16] 3GPP. TS 33.402, "System Architecture Evolution (SAE); Security aspects of non-3GPP accesses," 2017.

[17] 3GPP. TS 33.102, "3G security; Security architecture," 2017.

[18] 3GPP. ETSI TS 26.300, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2," 2017. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/36300.htm

[19] 3GPP. TS 36.331, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification," 2017.

[20] D. Rupprecht, K. Jansen, and C. Pöpper, "Putting LTE Security Functions to the Test: A Framework to Evaluate Implementation Correctness," in *10th USENIX Workshop on Offensive Technologies (WOOT)*, 2016.

[21] MultiTech Systems, "AT Commands For CDMA Wireless Modems." [Online]. Available: http://www.canarysystems.com/nsupport/CDMA_AT_Commands.pdf

[22] D. Nowoswiat, "Managing LTE Core Network Signaling Traffic." [Online]. Available: https://www.nokia.com/en_int/blog/managing-lte-core-network-signaling-traffic

[23] "Signaling Collection and Analysis Tool (SCAT)." [Online]. Available: https://github.com/fgsect/scat

[24] Android Developers, "Android Debug Bridge (adb)." [Online]. Available: https://developer.android.com/studio/command-line/adb?hl=en

[25] "USRP B210." [Online]. Available: https://www.ettus.com/product/details/UB210-KIT

[26] "S60Z 4G/3G Small Cell from ip.access." [Online]. Available: https://marksquared.co.uk/s60z/en

[27] "Airscope." [Online]. Available: http://www.softwareradiosystems.com/tag/airscope

[28] D. F. Kune, J. Koelndorfer, N. Hopper, and Y. Kim, "Location leaks on the GSM Air Interface," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2012.

[29] Y. Li, C. Peng, Z. Yuan, J. Li, H. Deng, and T. Wang, "Mobileinsight: Extracting and Analyzing Cellular Network Information on Smartphones." in *Proceedings of the ACM Annual International Conference on Mobile Computing & Networking (MobiCom)*, 2016.

[30] G.-H. Tu, Y. Li, C. Peng, C.-Y. Li, H. Wang, and S. Lu, "Control-plane Protocol Interactions in Cellular Networks," in *Proceedings of the 2014 ACM conference on SIGCOMM*. ACM, 2014, pp. 223–234.

[31] B. Hong, S. Park, H. Kim, D. Kim, H. Hong, H. Choi, J. P. Seifert, S.-J. Lee, and Y. Kim, "Peeking over the Cellular Walled Gardens-A Method for Closed Network Diagnosis," *IEEE Transactions on Mobile Computing*, 2018.

[32] Y. J. Jia, Q. A. Chen, Z. M. Mao, J. Hui, K. Sontinei, A. Yoon, S. Kwong, and K. Lau, "Performance Characterization and Call Reliability Diagnosis Support for Voice over LTE," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 2015.

[33] E. Barkan, E. Biham, and N. Keller, "Instant Ciphertext-only Cryptanalysis of GSM Encrypted Communication," in *Annual International Cryptology Conference*. Springer, 2003, pp. 600–616.

[34] U. Meyer and S. Wetzel, "On the Impact of GSM Encryption and Man-in-the-Middle Attacks on the Security of Interoperating GSM/UMTS Networks," in *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on*, vol. 4. IEEE, 2004.

[35] C. Mitchell, "The Security of the GSM Air Interface Protocol," *Univ. of London, Royal Holloway, RHUL-MA-2001-3*, 2001.

[36] D. Strobel, "IMSI catcher," *Chair for Communication Security, Ruhr-Universität Bochum*, vol. 14, 2007.

[37] U. Meyer and S. Wetzel, "A Man-in-the-Middle Attack on UMTS," in *Proceedings of the 3rd ACM workshop on Wireless security*. ACM, 2004.

[38] Z. Ahmadian, S. Salimi, and A. Salahi, "New Attacks on UMTS Network Access," in *Wireless Telecommunications Symposium, 2009. WTS 2009*. IEEE, 2009.

[39] M. Zhang and Y. Fang, "Security Analysis and Enhancements of 3GPP Authentication and Key Agreement Protocol," *IEEE Transactions on wireless communications*, vol. 4, no. 2, pp. 734–742, 2005.

[40] N. Golde, K. Redon, and R. Borgaonkar, "Weaponizing Femtocells: The Effect of Rogue Devices on Mobile Telecommunications." in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2012.

[41] M. T. Raza, F. M. Anwar, and S. Lu, "Exposing LTE Security Weaknesses at Protocol Inter-Layer, and Inter-Radio Interactions," in *International Conference on Security and Privacy in Communication Systems*. Springer, 2017, pp. 312–338.

[42] H. Kim, D. Kim, M. Kwon, H. Han, Y. Jang, D. Han, T. Kim, and Y. Kim, "Breaking and Fixing VoLTE: Exploiting Hidden Data Channels and Mis-implementations," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015.

[43] 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 Conference on Computer and Communications Security*. ACM, 2015.

[44] B. Michau and C. Devine, "How to not break LTE crypto," in *ANSSI Symposium sur la sècuritè des technologies de l' information et des communications (SSTIC)*, 2016.

[45] W. Enck, P. Traynor, P. McDaniel, and T. La Porta, "Exploiting Open Functionality in SMS-capable Cellular Networks," in *Proceedings of the 12th ACM conference on Computer and communications security*. ACM, 2005, pp. 393–404.

[46] P. Traynor, P. McDaniel, and T. La Porta, "On Attack Causality in Internet-connected Cellular Networks," in *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, vol. 21. USENIX Association, 2007, pp. 1–21.

[47] C. Mulliner, N. Golde, and J.-P. Seifert, "SMS of Death: From Analyz-

ing to Attacking Mobile Phones on a Large Scale." in *USENIX Security Symposium*, 2011, p. 99.

[48] P. P. Lee, T. Bu, and T. Woo, "On the Detection of Signaling DoS Attacks on 3G Wireless Networks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, 2007, pp. 1289–1297.

[49] P. Traynor, M. Lin, M. Ongtang, V. Rao, T. Jaeger, P. McDaniel, and T. La Porta, "On Cellular Botnets: Measuring the Impact of Malicious Devices on a Cellular Network Core," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2009.

[50] N. Golde, K. Redon, and J.-P. Seifert, "Let Me Answer That for You: Exploiting Broadcast Information in Cellular Networks." in *USENIX Security Symposium*, 2013.

[51] Business Wire, "Baseband Market Share." [Online]. Available: https://www.businesswire.com/news/home/20180604005896/en/Strategy-Analytics-2017-Baseband-Market-Share-Intel

[52] "The deployment plan for 5G NR Non-Standalone (NSA)." [Online]. Available: https://www.qualcomm.com/news/onq/2017/03/09/3gpp-agrees-plan-accelerate-5g-nr-global-5g-standard-2019-deployments

[53] "5G Standalone standard timeline." [Online]. Available: http://www.3gpp.org/release-15

[54] S. Park, A. Shaik, R. Borgaonkar, and J.-P. Seifert, "White Rabbit in Mobile: Effect of Unsecured Clock Source in Smartphones," in *Proceedings of the 6th Workshop on Security and Privacy in Smartphones and Mobile Devices*. ACM, 2016.

APPENDIX A
LIMITATIONS AND FUTURE WORK

Stateless Fuzzer: LTEFuzz can be classified as a stateless fuzzer that does not intend to diagnose memory bugs. Although we uncovered 51 vulnerabilities using LTEFuzz, potential vulnerabilities LTEFuzz would be unable to cover do exist. The attack model of LTEFuzz assumes that an adversary does not have access to the cryptographic keys of the victim. Therefore, the adversary sends all crafted messages while being deregistered. In other words, multi-stage vulnerabilities[8] are currently beyond the scope of LTEFuzz. To diagnose multi-stage attacks, the tester first transitions to the target state by relaying the valid control plane messages between the UE and the network. When it reaches the target state, it may send unwanted messages. Cases such as these were previously demonstrated [5], [20]. The number of possible states as well as decision tree construction for all possible combinations of states seems to be the most significant challenge for stateful fuzzing. The extension of LTEFuzz to stateful fuzzing remains a future task. Note that, as discussed in Section III, the identification of failures responsible for causing crashes or memory leaks is not currently of concern.

Carriers and Vendors: Because of ethical and legal restrictions, we were able to only include two carriers in our tests, who are collaborating with us. However, as we have shown in the paper, different vendors and carriers have different vulnerabilities. We plan to collaborate with other carriers and vendors to check their security using LTEFuzz. Furthermore, we could apply our testing method to other control plane protocols such as the S1 Application Protocol (S1AP) and X2 Application Protocol (X2AP) which carry control plane traffic among core network components. However, we need access to the core network (i.e., eNBs, MMEs, or gateways) to dynamically inspect the communication of those protocols. We would be able to conduct dynamic security tests against those protocols in future only if the carriers were to grant us access permission to their core network.

Consideration for 5G: According to the 5G standard development and deployment plan, 5G Non-Standalone (NSA)[9] will be deployed as early as 2019 [52]. Subsequently, the 5G Standalone (SA) standard will be developed by 2020 [53]. In other words, LTEFuzz would remain useful for 5G NSA as long as *open source LTE implementations such as srsLTE support 5G in radio communication*. Additional development would be required to support 5G SA, as the core network is likely to change. Therefore, ensuring that LTEFuzz supports both 5G NSA and SA remains a future task.

APPENDIX B
ADDITIONAL FIGURES

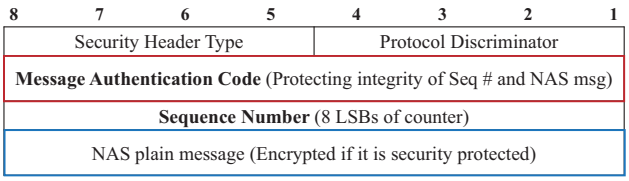

Fig. 10. Format of NAS security protected message

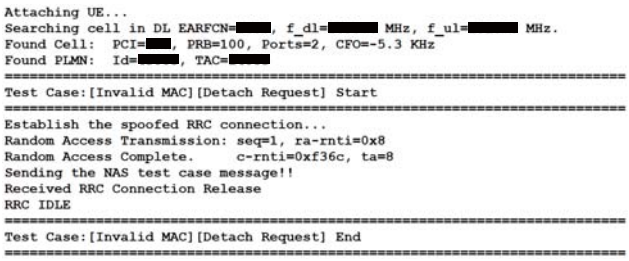

Fig. 11. Part of the output when running LTEFuzz (tester side)

[8]Multi-stage refers to multiple state changes.

[9]Initially, 5G will utilize the LTE core network (EPC).

APPENDIX C
COMPLETE RESULTS OF LTEFUZZ

TABLE IV. THE LIST OF TARGET CONTROL PLANE MESSAGES AND THE RESULTS OF DYNAMIC TESTING (PROBLEMATIC CASES IN **BOLD**)

Classification of behavior upon receiving test cases: **Problematic (1, 2, 3)**, Benign (4)
UL: Uplink, **DL:** Downlink, **P:** Plain, **I:** Invalid MAC, **R:** Replay
[#]: Reference to problematic cases previously discovered in **[#]**. All other vulnerabilities are new.

Test messages	Direction	Property 1-1	Property 1-2 (P)	Property 2-1 (I)	Property 2-2 (R)	Property 3	Implications
NAS							
Attach request (IMSI/GUTI)	UL	4	**1**	**1**	**1**	-	DoS
Detach request (UE originating detach)	UL	-	**1** [41]	**1**	**1**	-	DoS
Service request	UL	-	-	4	**2**	-	Spoofing
Tracking area update request	UL	-	**3**	**3**	**2** and **3**	-	DoS, False location update
Uplink NAS transport	UL	-	**2** and **3**	**2** and **3**	**2**	-	DoS, Spoofing
PDN connectivity request	UL	4	4	**3**	**3**	-	DoS
PDN disconnect request	UL	-	4	**3**	**1**	-	(selective) DoS
EMM status	Both	-	4	4	4	-	
ESM status	Both	-	4	4	-	-	
Attach reject	DL	**1** [8]	**1** [7]	-	-	-	DoS
Authentication reject	DL	**1** [4]	-	-	-	-	DoS
Authentication request	DL	4	-	-	-	4	
Detach request (UE terminated detach)	DL	-	**1** [4]	-	-	-	DoS
Downlink NAS transport	DL	-	4	4	4	-	
EMM information	DL	-	**1** [54]	-	-	-	Spoofing
GUTI reallocation command	DL	-	4	4	**2**	-	Spoofing
Identity request	DL	**2** [44]	4	4	**2**	-	Information leak
Security mode command	DL	-	4	4	**2** [4]	-	Location tracking
Service reject	DL	-	**1** [7]	-	-	-	DoS
Tracking area update reject	DL	-	**1** [7]	-	-	-	DoS
RRC							
MeasurementReport	UL	-	4	4	4	-	
RRCConnectionReestablishmentRequest	UL	-	-	4	4	-	
RRCConnectionRequest	UL	**1** and **2**	-	-	-	-	DoS, Spoofing
RRCConnectionSetupComplete	UL	**2**	-	-	-	-	Spoofing
CounterCheck	DL	-	4	-	-	-	
LoggedMeasurementsConfiguration	DL	-	4	-	-	-	
MasterInformationBlock	DL	**2**	-	-	-	-	Spoofing
Paging	DL	**1** [4] and **2**	-	-	-	-	DoS, Spoofing
RRCConnectionReconfiguration	DL	-	**2**	**3**	4	-	Spoofing, Eavesdropping
RRCConnectionReestablishment	DL	-	**2**	-	-	-	Spoofing
RRCConnectionReestablishmentReject	DL	-	**1**	-	-	-	DoS
RRCConnectionReject	DL	**1**	-	-	-	-	DoS
RRCConnectionRelease	DL	**1** [8]	-	-	-	-	DoS, Spoofing
RRCConnectionSetup	DL	**2**	-	-	-	-	Spoofing
SecurityModeCommand	DL	-	4	4	4	**2**	Eavesdropping
SystemInformationBlockType1	DL	**2** [4]	-	-	-	-	Spoofing
SystemInformationBlockType10 / 11	DL	**2** [4]	-	-	-	-	Spoofing (Public warning)
SystemInformationBlockType12	DL	**2** [4]	-	-	-	-	Spoofing (Public warning)
UECapabilityEnquiry	DL	**2**	-	**2**	**2**	-	Information leak
UEInformationRequest	DL	-	4	-	-	-	