Bookworm Game: Automatic Discovery of LTE Vulnerabilities Through Documentation Analysis

Yi Chen[1], Yepeng Yao[2,3,*], XiaoFeng Wang[1,*],
Dandan Xu[2,3], Chang Yue[2,3], Xiaozhong Liu[1], Kai Chen[2,3], Haixu Tang[1], Baoxu Liu[2,3]
[1]Indiana University Bloomington
[2]{CAS-KLONAT[†], BKLONSPT[‡], SKLOIS[ǂ]},Institute of Information Engineering, CAS
[3]School of Cyber Security, University of Chinese Academy of Sciences
chenyi.yy1@gmail.com, {xw7, liu237, hatang}@indiana.edu, {yaoyepeng, xudandan, yuechang, chenkai, liubaoxu}@iie.ac.cn

2021 IEEE Symposium on Security and Privacy (SP) | 978-1-7281-8934-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/SP40001.2021.00104

***Abstract*—In the past decade, the security of cellular networks has been increasingly under scrutiny, leading to the discovery of numerous vulnerabilities that expose the network and its users to a wide range of security risks, from denial of service to information leak. However, most of these findings have been made through ad-hoc manual analysis, which is inadequate for fundamentally enhancing the security assurance of a system as complex as the cellular network. An important observation is that the massive amount of technical documentation of cellular network can provide key insights into the protection it puts in place and help identify potential security flaws. Particularly, we found that such documentation often contains *hazard indicators* (*HIs*) – the statement that describes a risky operation (e.g., abort an ongoing procedure) when a certain event happens at a state, which can guide a test on the system to find out whether the operation can indeed be triggered by an unauthorized party to cause harm to the cellular core or legitimate users' equipment. Based upon this observation, we present in this paper a new framework that makes the first step toward intelligent and systematic security analysis of cellular networks. Our approach, called *Atomic*, utilizes natural-language processing and machine learning techniques to scan a large amount of LTE documentation for HIs. The HIs discovered are further parsed and analyzed to recover state and event information for generating test cases. These test cases are further utilized to automatically construct tests in an LTE simulation environment, which runs the tests to detect the vulnerabilities in the LTE that allow the risky operations to happen without proper protection. In our research, we implemented Atomic and ran it on the LTE NAS specification, including 549 pages with 13,598 sentences and 283,850 words. In less than 5 hours, our prototype reported 42 vulnerabilities from 192 HIs discovered, including 10 never reported before, under two threat models. All these vulnerabilities have been confirmed through end-to-end attacks, which lead to unauthorized disruption of the LTE service a legitimate user's equipment receives. We reported our findings to authorized parties and received their confirmation that these vulnerabilities indeed exist in major commercial carriers and $2,000 USD reward from Google.**

***Index Terms*—Cellular Network, 4G, LTE, Vulnerability, Attack, Documentation Analysis, NLP**

I. Introduction

The past four decades have witnessed the phenomenal development of wireless cellular networks, with new technologies emerging every ten years, from the first generation of analog system (1G) in early 80s, to the digital 2G, 3G broadband, today's 4G and tomorrow's 5G. These technologies profoundly changed our life, enabling truly ubiquitous computing across billions of users world-wide, and laying the groundwork for a vibrant mobile ecosystem with numerous applications (mobile payment, energy infrastructure, emergency services, self-driving car, etc.) that define the modern digital society and power the economy growth in the past decades. With such a pivotal role played by these telecommunication technologies, their security and privacy naturally become critically important and are expected to be thoroughly scrutinized. Unfortunately, discoveries made in recent years reveal that actually the cellular systems in use today are much less protected than thought, often involving various security or privacy risks.

Vulnerability discovery in cellular networks. Cellular networks are massive, highly complicated systems, and like other complex systems, comprehensive protection and high security assurance cannot be easily achieved. Studies in the past decade, particularly those in recent years, have brought to light numerous vulnerabilities in these networks, ranging from jamming [1]–[6], other Denial-of-Service (DoS) risks [7]–[10], [10]–[13], to authentication flaws [14]–[16], to information leak [7], [10], [11], [13], [16]–[20].

Most of these vulnerabilities have been discovered through ad-hoc manual analysis. More systematic solutions to enhance the cellular network's security quality rely on in-depth understanding about the network's design and operations, as elaborated in its documentation. A prominent example is the recent work on semi-automatic LTE fuzzing [13] in which a set of security properties are identified from the 3GPP standard to guide the selection and mutation of messages injected into a Long-Term Evolution (*LTE*) network or a device to evaluate their compliance with these properties. This approach works particularly well in finding implementation errors (e.g., invalid integrity protection) but has also led to the discovery of the problems in the specifications (e.g., BTS resource depletion and Blind DoS). Other studies focusing more on the design issues [11], [14]–[16] recover a high-level ecosystem model for the LTE or 5G protocol from the documentation, for the purpose of model checking and protocol verification [21],

*Corresponding Authors
[†]Key Laboratory of Network Assessment Technology, CAS.
[‡]Beijing Key Laboratory of Network Security and Protection Technology
[ǂ]State Key Laboratory of Information Security, IIE, CAS

DOI 10.1109/SP40001.2021.00104

[22]. All these approaches require significant human effort in document inspection, which is expensive and error-prone. For example, the work on the LTE protocol only includes the messages for security operations (e.g., authentication, key exchange, etc.), while the interactions not involved in a cryptographic protocol but still with security implications, like exception handling, fall through the cracks. Given the massive amount of documentation (e.g., 549 pages for LTE 24.301 [23]), manual recovery of its semantics is inadequate and needs to be complemented by automated analysis.

Also a key observation made in our study is that in addition to such knowledge as the ecosystem model, a cellular network's documentation often contains explicit or implicit *indicators* for potential security risks, such as warning about disruptive operations. For example, the sentence "If the network receives a DETACH REQUEST message before the ongoing identification procedure has been completed, the network shall abort the identification procedure and shall progress the detach procedure." describes a potential risky situation where a UE can be denied access to the core network by a DETACH REQUEST message; in the absence of authentication, the adversary can block the UE by issuing the message with the victim's IMSI. Once recovered from the documentation, such indicators can be leveraged to identify potentially hazardous states in a cellular network, leading to the detection of security flaws, which however has never been done before.

Atomic. In our research, we made the first step towards *automated discovery of vulnerabilities in cellular networks using the guidance recovered from their documentation*. Our approach, dubbed *Atomic* (automated LTE documentation to vulnerability discovery), is a framework designed to semantically analyze LTE documents using Natural Language Processing (NLP) to recover a set of *hazard indicators* (*HIs*) for generating test cases based upon a given threat model; these test cases are then automatically played to an LTE simulation environment at certain system states to detect the presence of security flaws. More specifically, an HI is a description in the document indicating that a risky operation (e.g., "abort a procedure") could take place once an event is triggered at a certain system state, so it can guide the construction of test messages to be issues at the state, under the assumption of the issuer's capability (the threat model), to find out whether such an operation can indeed be triggered (that is, protection is indeed absent) in the target system. To discover such HIs from the document, the user just needs to provide a sample description for the operation as a seed, like "abort procedure", which could come from the sentence in the document related to a reported attack, and Atomic then automatically expands the seed from the document to find related descriptions, such as "deactivate context". These descriptions are used to examine the sentences inside the document, to detect those implying the risk operation, through a *Textual Entailment* (*TE*) model. Such a sentence forms an HI.

Each HI further undergoes a semantic analysis to identify the state and the event it describes, which is used to parameterize templates for creating test messages. Then, based upon the specification of the adversary (e.g., a phone, a fake base station) and its capability (e.g., knowledge about the user's IMSI) in the threat model, these messages are issued to a hooked simulator, in the target state, to trigger the desired event. The outcome of the event, as observed from system logs (e.g., disconnecting the victim from the core network), is then analyzed to determine whether damage has been inflicted, and if so, the test case is output as a Proof-of-Concept (PoC) exploit together with the description of the vulnerability discovered. We further show that these PoC exploits can be executed end-to-end, since their target states can be realistically observed or inferred by the adversary.

Findings. In our research, we implemented the Atomic framework and evaluated it on two threat models: user-equipment (UE) DoS through core network and UE DoS through fake base station, by automatic analysis of the LTE NAS (Non-Access Stratum) control-plane document. Our evaluation shows that Atomic effectively recovers HIs from the document (very likely without any false negative) and accurately detects vulnerabilities (without false positives) through running test cases generated from the HIs on related systems and devices. Altogether, it reported 42 vulnerabilities, 10 in the LTE core network and 32 in the LTE components on the UE side in less than 5 hours. Particularly, all core-network flaws have never been reported before, enabling a malicious UE to realistically block a legitimate device's attempt to attach to the network. After manually analyzing these vulnerabilities, we found that 15 of them are design weaknesses, which fail to consider the situation that the adversary can impersonate a victim UE or the network before the security context establishment or mutual authentication. Even the some implementation flaws indicate the potential design complexity: for instance, to enable the transfer of an attached UE across different service areas, strict integrity check (e.g., ATTACH REQUEST) may not always be enforced (e.g., before the secure exchange is established), opening an avenue to an impersonation attack.

Most importantly, many of these vulnerabilities are discovered from the HIs appearing at the part of the NAS document not directly related to security and privacy (e.g., in the context of different procedure workflows), which therefore has not been inspected before for security protection. As an example, an HI "upon reception of a paging for EPS services using IMSI...the UE shall deactivate any EPS bearer context..." appears in the specification of the paging procedure. This demonstrates that automated HI analysis indeed enables more comprehensive evaluation of the cellular network's security, complementing existing manual, semi-automatic approaches. We have reported our findings to device manufacturers and other authorized parties, and received confirmation that these vulnerabilities indeed exist in real world.

Contributions. The contributions are outlined as follows:

• *New technique*. We designed the first intelligent technique that automatically discovers vulnerabilities using the guidance from the documentation of carrier networks and further gen-

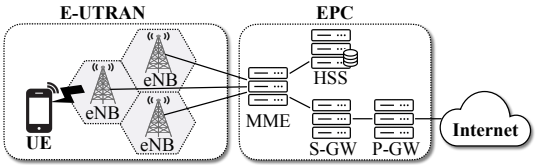


Fig. 1: Architecture of the LTE network.

erates PoC exploits. The Atomic framework has the potential to be extended to cover various types of vulnerabilities in different layers, for both LTE and 5G. This new technique presents an important step towards more systematic and automatic vulnerability discovery in cellular networks, which is crucial for enhancing their security quality.

• *Implementation and new findings*. We implemented our design and evaluated it on the LTE document. This preliminary effort already results in the discovery of 42 vulnerabilities, with 10 never reported before. A close look at these new flaws indicates the possible dilemma LTE faces in the presence of the contention between rigorous security protection and reliable user services, which together with all other problems found by the prior research, can help build a more effective and securer next-generation cellular network.

II. Background

A. LTE Network Architecture

As illustrated in Figure 1, the LTE network architecture contains three main components: the User Equipment (UE), the Evolved UMTS Terrestrial Radio Access Network (E-UTRAN), and the Evolved Packet Core (EPC). Here, the EPC communicates with packet data networks such as the Internet.

UE. The user equipment is any device directly used by an end-user to communicate through the LTE network, such as the mobile phone, the IoT device, etc. The UE carries the Universal Integrated Circuit Card known as the SIM card, with a Universal Subscriber Identity Module (USIM) that stores user-specific data such as her phone number, authentication credentials, an International Mobile Subscriber Identity (IMSI), and a Globally Unique Temporary ID (GUTI). Both the IMSI and GUTI are for unique identification of an LTE subscriber, and their disclosure could expose the UE to an impersonation risk [11]–[13].

E-UTRAN. The E-UTRAN is a radio access network, built on top of evolved base stations (*aka. Evolved Node B (eNB)*), for handling the radio communication with the UE. The E-UTRAN acts as an intermediary to facilitate the connection between the UE and the EPC. More specifically, under the LTE, a geographical area is partitioned into hexagonal cells, and each cell is served by at least one eNB. When a UE detects signals from several eNBs, it often chooses the one with the strongest signal.

EPC. EPC is a framework based upon an all-IP mobile core network, which is responsible for providing mobile core functionalities such as authentication, data-packet routing, etc. Most relevant to our study is its Mobility Management Entity (MME) that handles the signaling messages related to mobility and security for E-UTRAN access. For example, it tracks a UE's location to send downlink data once needed, checks the UE's authentication to determine whether it can receive the service, and performs the key management for secure communication between the UE and the EPC [24]. The set of protocols between the UE and the MME are known as the Non-Access Stratum (NAS).

B. EPS Mobility Management Protocol

The EPS Mobility Management (EMM) protocol is a sub-protocol of the NAS (see 3GPP TS 24.301 [23]), which provides procedures for mobility and security control when the UE is using the E-UTRAN. Depending on their functionalities, most EMM procedures fall into two categories, mobility-related or security-related.

Mobility-related. These procedures include the ones for `attach`, `detach`, `tracking area updating` and `service request`, which are all initiated by the UE to request or stop using the service from the E-UTRAN. Among them is the `paging` procedure launched by the MME for establishing an NAS signaling connection or prompting the UE to re-attach.

Security-related. The `GUTI reallocation`, `identification`, `authentication` and `security mode control` procedures are initiated by the MME in the presence of an NAS signalling connection. Specifically, the MME runs the identification procedure to get the UE's IMSI; the authentication procedure establishes trust between the UE and the MME; the security mode control enables key exchange for data encryption and integrity protection. These procedures could be performed during attach, tracking area updating, etc., based upon the request received.

C. Natural Language Processing

Textual entailment. Textual entailment (TE) takes a pair of sentences (or sub-sentence) and predicts whether the facts in one of them (called a premise) necessarily imply the facts in the other (called a hypothesis). The "implication" here follows a relatively relaxed definition: if a premise entails a hypothesis, then a human believing the premise would typically be able to conclude that the hypothesis is most likely true [25]. An example of a positive TE is that the hypothesis *"Giving money to the poor has good consequences"* can be inferred from the premise *"If you help the needy, God will reward you"*. The state-of-the-art textual entailment (e.g., the RoBERTa model [26]) can achieve a 92% accuracy. In our study, we leverage the AllenNLP library [27], which integrates state-of-the-art textual entailment NLP models for discovering the descriptions from the LTE documentation that present the security-critical operations (e.g., disrupting a procedure). Note that such TE models are trained to identify grammatical variations (e.g., passive tense), synonyms and other semantic-preserving transformation, which is particularly useful for determining the implication of risky operation hypotheses.

Dependency parsing. Dependency parsing analyzes the syntactic structure of a sentence or several neighboring sentences

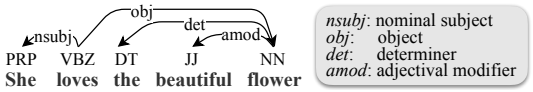


Fig. 2: Example of the dependency parsing.

to identify the grammatical relations between terms. In the structure, verb takes the central role in the clause structure, with all other words either directly or indirectly modifying the verb. Figure 2 shows a simple example. From the figure, we can see that for the verb "love", the word "she" is its subject through the relation *nsubj*, "flower" is its object through *obj*, and "beautiful" adjectivally modifies "flower" through *amod*. The state-of-the-art dependency parser (e.g., Stanford parser [28]) can achieve a 92.2% accuracy. In our study, we utilized the parsing results of the HI sentences discovered from the LTE specification to find out the object receiving a message, the ongoing procedure, etc., so as to determine the current state and the event message.

D. Threat Model

Prior research [10] shows that a cellular network is subject to the threats from adversaries with various objectives (DoS, impersonation, data collection, etc.), knowledge (the victim UE's IMSI, GUTI, etc.) and capabilities (control of a fake eNB, generation of strong signals, etc.). The Atomic framework developed in our research is meant to discover different kinds of security risks and related design or implementation weaknesses under these models. For this purpose, our approach takes a well-defined threat model as its input. Such a model can be described as a 4-tuple: $T = \{A, O, I, G\}$, where A is a set of *actors*, including adversary a, victim(s) v and others, and O is a set of *operations* the adversary can perform, I is the information at his disposal and G includes his goal(s). In our research, we analyzed the LTE NAS documentation on two threat models, T_1 and T_2. Under T_1, inside A are a malicious UE controlled by the adversary, a victim UE and the MME of the LTE network; the adversary's goal G is to disrupt the service the victim receives using the malicious UE, which is capable of generating, issuing all UE side messages and capturing other parties' communication through a sniffer; also his information set I contains the victim's IMSI and GUTI, the mobile identity known to be obtainable through various channels, such as from paging message (Section IV-B), but not other victim data, e.g., its security key that may not easily come by. Under T_2, A contains a fake eNB under the control of the adversary, who intends to block a specific victim UE's service and can produce all messages issued to the UE.

III. Atomic: Design and Implementation

A. Overview

As mentioned earlier, our approach is based upon the observation that some risky operations (such as disrupting an ongoing procedure, disclosing user data), as described in the cellular-network documentation, could be triggered in an insecure way, so identification of these operations and their conditions (the system state and events) can help determine whether they have been properly protected and whether a vulnerability is present and can be exploited. Our research shows that this process can be automated, with a threat model and a risky operation description (*ROD*) as the input.

More specifically, Atomic runs a suite of NLP techniques to first "extend" the ROD, finding other related descriptions about the risky operation, then discover all hazard indicators that imply any of the descriptions and messaging events, and further parse the HIs, recovering their states and events that cause the operation. The states and events are utilized to automatically generate test cases, which are played to an LTE simulation environment configured according to the threat model, in an attempt to determine whether the adversary's goal can indeed be achieved by these test cases, as identified from the simulator's log files.

Architecture. Figure 3 illustrates Atomic's architecture, including *Hazard Indicator Detector* (HID), *LTE test case generator* (LTCG), and *PoC Identifier* (PI). The HID extracts conditional statements, extends RODs and runs a Text Entailment model to find HIs from all located conditional sentences, those implying any of the the events and RODs. The LTCG analyzes the semantics of each HI to discover the state and events described and based upon such information, constructs the test cases – a message to be issued by the adversary (according to the threat model) at the state through the templates retrieved from a database. The PI then executes the test case on a configured simulation environment and automatically analyzes its logs to determine whether the attack succeeds. Again, the input of Atomic includes a seed ROD and a threat model, and its output is an exploit Proof-of-Concept (PoC) if a vulnerability is discovered.

Example. Here we use an example to explain how our approach works. From the seed ROD "abort procedure", the HID recovers a set of related descriptions, such as "deactivate context". Then it evaluates these RODs' and the messaging event's relations with all conditional sentences in an LTE document through the entailment model, finding all HIs. One of them is "If the network receives a DETACH REQUEST message before the ongoing identification procedure has been completed, the network shall abort the identification procedure and shall progress the detach procedure." (see 5.4.4.6 of 3GPP 24.301 specification [23]). This HI is then parsed by the LTCG into a dependency tree, according to its grammatical structure, for state and event discovery. Since the LTE protocol is event-driven, at each state (except the last one) of every procedure (attach, detach, etc.), a party (MME, UE, etc.) is either ready to send a message or wait for a message (see Figure 4). So the LTCG locates from the dependency tree the verb phrase starting with "send", "receive" or their equivalents to recover the noun describing the message "DETACH REQUEST" that triggers the event, and further looks into the clause led by "before" to get a state-related noun phrase "identification procedure", which is used to find the state indicating MME waiting for the IDENTITY RESPONSE message from a simple finite-state machine (FSM) summarized from the protocol

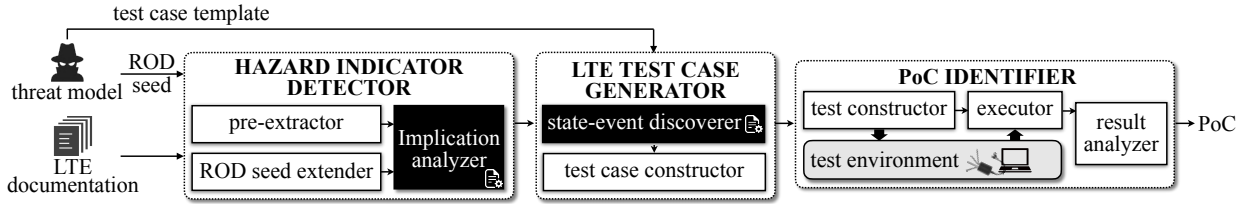


Fig. 3: Architecture of the Atomic.

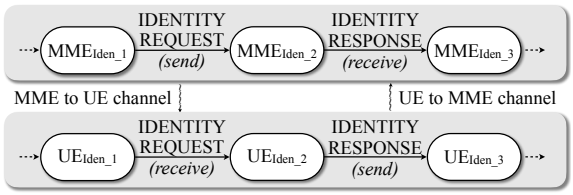


Fig. 4: The identification procedure's FSM.

actors: MME, UE_v, UE_m
state of MME: **MME_{Iden_2}**
event:
message: **DETACH REQUEST**
direction: UE_m to MME
parameter: set mobile identity to UE_v's IMSI/GUTI

Fig. 5: Example of a test case.

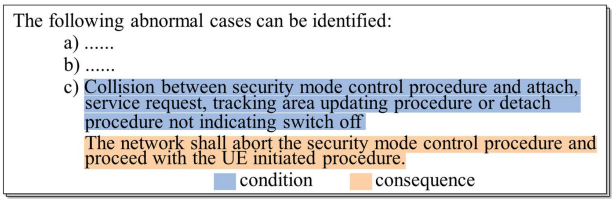


Fig. 6: An example of a conditional statement cross sentences in the unique structure.

diagram for each procedure (Section III-C).

Using the message identified, under the threat model T_1, the LTCG constructs a test case by parameterizing a template, as illustrated in Figure 5. In the model, a malicious UE sends a DETACH REQUEST message with the victim UE's IMSI to the MME when it is waiting for an IDENTITY RESPONSE from the victim UE. This test case is then handed over to the PI, which utilizes a hooked LTE simulator to locate the state for message injection: for the test case, once the PI detects that the MME enters the specified state, it informs the malicious UE to issue the DETACH REQUEST message to the EPC. The outcome of this test is then analyzed through inspecting the simulator's logs. If the PI finds that the MME has sent a message named UEContextReleaseCommond to an eNB after executing the test case, it reports that the victim UE's attach attempt is disrupted and the test case is then output as a PoC for pinpointing the DoS vulnerability.

Following we elaborate the design and implementation of individual components.

B. Hazard Indicator Discovery

A Hazard indicator is a short description documents an operation, which we consider to be risky, to be triggered by a specific message issued under a certain state. Such an operation, once takes place, could cause a damage to a party involved in the LTE protocol, like losing its connection, exposing its private data, etc., and therefore could be exploited by the adversary under a given threat model should proper protection not be in place. To find the HIs, our approach (the HID) is designed to search for all conditional sentences implying the semantics of messaging events and a set of RODs, which are derived through expanding a given seed over an LTE document. The conditional sentence here is the one with conditional clauses, with "if", "upon", "when" and others, which we consider could carry an HI. In addition to such a single sentence, our approach also looks for a unique multi-sentence structure from the LTE documentation for HIs, which lays out itemized statements (including potential conditions) followed by one sentence describing related operations (see Figure 6 for an example). These sentences are analyzed using a Textual Entailment (TE) model to identify those that semantically entail events and RODs. In our research, the HID running on the LTE 24.301 specification [23] discovered 5,652 conditional sentences from 13,598 in the document and further reported 192 HIs.

ROD seed extension. As mentioned earlier, a ROD describes a potentially risky operation, such as aborting a procedure. Such a ROD is represented as a verb phrase like "abort procedure". The idea of our approach is to utilize a seed ROD as an input, to find out other related RODs in the documentation. These RODs are expected to describe different activities than the operation stated in the seed ROD: for example, starting a different procedure, which implies the termination of the current procedure. Note that at this step, we do not attempt to find variations of the same statement such as "terminate a procedure" or "the procedure should be stopped", since they can be automatically discovered by the TE model.

The HID finds a new verb phrase related to a known ROD through assessing their pointwise mutual information (PMI), a standard approach in computational linguistics. The unique part of our approach is to iteratively extend a seed to a set of RODs in this way, and each round select new phrases from both the current and its prior and posterior sentences when these phrases all share the same subject and are under the same condition. Specifically, given a ROD, our approach search throughout the whole documentation for all sentences containing the verb phrase (using an approximate match, allowing no more than 8 words between the verb and its object), and then semantically parses each of the sentences and its neighboring two sentences to identify other co-occurring verb phrases with the same subject and condition. Each of these new phrases y is then evaluated against the ROD x for

the PMI: $pmi(x, y) = \log \frac{p(x,y)}{p(x)p(y)}$. When $pmi(x, y)$ is above a threshold, which was conservatively set to 3.97 in our study (with the estimated p-value of 0.01), y is determined to be strongly associated with the ROD x, and therefore considered as a new ROD and utilized to find other RODs. Note that this approach could introduce false positive, which however can be handled by the testing step (Section III-D).

As a first step, we considered the two DoS threat models in our study (see Section II), with T_1 using the seed "abort procedure" and T_2 using "consider USIM", both from the LTE NAS specification, based upon known vulnerabilities under the two threat models reported by the prior research [10], [11]. Specifically, under T_1, we leveraged a DoS vulnerability reported by the previous work [11], which can be exploited by continuously sending ATTACH REQUEST messages with different security capabilities to cause the UE and the HSS sequence numbers to lose synchronization, thereby preventing the UE from connecting to the EPC. This vulnerability is related to a description in the specification ("If one or more of the information elements in the ATTACH REQUEST message differs from the ones received within the previous ATTACH REQUEST message, the previously initiated attach procedure shall be aborted and the new attach procedure shall be executed"). From the sentence, we extracted the verb phrase that describes the risky operation "abort procedure" as the indicator and the seed for the ROD extension under T_1. For T_2, we extracted the verb phrase "consider USIM" from the sentences such as "the UE sets the LTE status to EU3 ROAMING NOT ALLOWED and considers its USIM invalid for the network until it is rebooted or the USIM is re-inserted", as reported in the prior research [10]. We utilize this phrase since it describes the UE's potential operation. Also by scanning the whole LTE NAS specification, we found that this verb phrase only appears in "consider the USIM as invalid", so it was used as the seed for T_2. Some other verb phrases were also tried in our study but these two seeds performed best: either themselves have been frequently implied by related descriptions or they lead to many more RODs ("consider USIM" in particular), which helped discover the vulnerabilities that could not be found using the seeds only .

Altogether, our approach recovered from the NAS specification 5 RODs for T_1 and 8 for T_2. Among them under T_1 are the verb phrases such as "release resources", "deactivate context" and more subtle ones like "progress procedure". This phrase appears counter-intuitive, as it talks about starting a procedure not terminating one. Actually this ROD implies that the old procedure comes to an end and was found to rarely refer to other situations in the document. Also interesting is under T_2 our discovery of 7 RODs about deleting specific information, such as list, GUTI, TAI, which is *semantically unrelated to "consider USIM"*. It turns out that such operations indicate that the UE no longer keeps the mobility information, and therefore the network connection will be disrupted. We list all the RODs in Table I in Appendix. It is important to note that under T_1, 2 extended RODs led to the successful generation of 4 (17.39%) test cases, which resulted in the detection of 2 vulnerabilities (20% of all vulnerabilities in this category); under T_2, 3 new RODs enabled the generation of 33 (73.33%) test cases and identification of 21 vulnerabilities (65.63%). This indicates that ROD extension is indeed effective in finding vulnerabilities, even though not every extended ROD is useful, as some of them might not be entailed in any HI with proper state and event information.

Implication analysis. On the RODs discovered, the HID further performs an implication analysis to capture all the conditional statements that entail messaging events and these risky operations, which are then reported as HIs. To this end, our approach utilizes Textual Entailment (Section II-C), a neural network model for predicting whether a premise implies a hypothesis. Here, to find a messaging event in TE, a premise is the conditional part of a conditional statement, that is, the clause "If the network receives a DETACH REQUEST message before..." of the sentence "If the network receives ... the network shall ...", and the hypotheses are "Send a message", "Receive a message" and "Procedures collided" (which implicitly indicates the issue of a message). As to the risky operation, the premise is the consequence part of a conditional statement, which is the clause "the network shall abort the identification procedure.." in the above example, and the hypothesis is a sentence derived from a given ROD. To extract the two premises from the sentence, in which one is the conditional part and the other is the consequence part, the HID utilizes the result of dependency parsing as mentioned earlier: on the dependency tree generated by the parser, our approach locates the verb of the conditional clause first by following the *mark* relations from the terms "if", "when" and etc., and includes all other terms on the tree directly or indirectly related to the verb as part of the conditional clause, except the one marked with the *advcl:if* relation, which is the verb for the consequence clause; then HID connects all the terms related to the verb for the consequence clause except the the one marked with the *advcl:if* relation.An example is illustrated in Figure 7: our approach automatically recovers the conditional clause "if the network receives a DETACH REQUEST message before the ongoing identification procedure has been completed" as the premise for deciding events and the consequential clause "the network shall abort the identification procedure and shall progress the detach procedure" as the premise for identifying risky operations. For the multi-sentence structure, our approach treats each itemized statement as a conditional clause and the sentence following the list as the consequential clause for the implication analysis. To build the hypothesis for risky operation identification, we simply convert a ROD verb phrase to a short sentence: e.g., "Abort a procedure". Such premise-hypothesis pairs are then analyzed by the TE model.

In our implementation of Atomic, we integrated a state-of-the-art TE model provided by AllenNLP [27], which reports a percentage value for each premise-hypothesis pair to rate the possible presence of entailment. Utilization of the model for HI detection requires a threshold, which was set in our research as follows. We first randomly sampled 1,000 of the

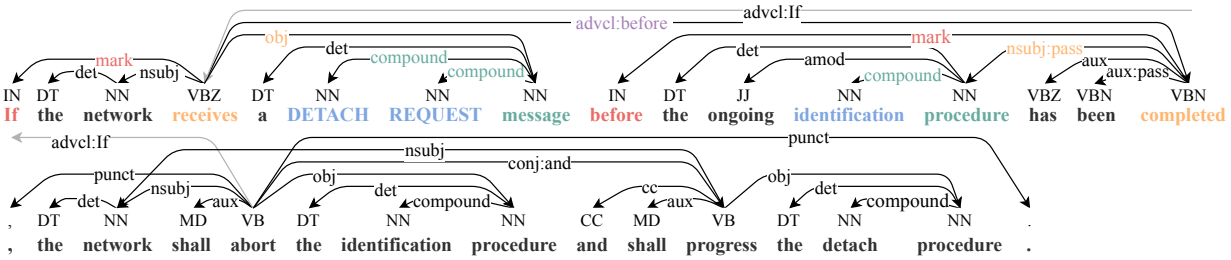


Fig. 7: Example of the dependency parsing on a hazard indicator.

premises derived from the 5,652 conditional sentences in the NAS document for a manual inspection, discovering 19 hazard indicators as the ground truth. Then, we ran the TE model on these samples and hypotheses to analyze its precision, recall and f2-score under different thresholds (see Figure 10 in Appendix). As mentioned earlier, Atomic is designed to automatically evaluate HIs to find out whether they indeed prelude vulnerabilities. So we can live with a moderate level of false positives, which will not result in false reports of vulnerabilities due to the testing step. This precision/recall balance can be struck using the f2-score. As we can see from Figure 10, the f2-score peaks at 91.21%, with a 100.00% recall and 55.88% precision, which is used as our threshold. With the threshold, Atomic reported 192 HIs from 13,598 sentences in the NAS document.

C. LTE Test Case Generation

To find out whether the risky operation stated in an HI is unprotected and therefore can be activated by an unauthorized party, the Atomic framework performs a test on the HI under a given threat model. For this purpose, the LTCG automatically recovers from the HI's conditional clause the state and event (actually the message for triggering the event) of an LTE procedure and further utilizes such information to construct a test case – the message to be issued by the adversary in the threat model at the state. To enable such a "content-to-test-case translation", we utilize a simple finite state machine (FSM) defined for each LTE procedure, as recovered from the LTE documentation, to map the semantics of the HI to the LTE procedure and its state and message. In our research, from the 192 hazard indicators outputted by the HID, the LTCG automatically generated 68 test cases, including 23 cases for T_1 and 45 cases for T_2.

LTE procedure modeling. To determine from an HI the timings to issue messages to an LTE network component and the content of the messages, the LTCG needs knowledge about the stateful operations of the network. Given the fact that the LTE protocol is event-driven, with each state-transition caused by transmission of messages between different system components, we model the high-level operations of an LTE procedure (e.g., attach, identification and authentication) using a sequence of states, in which a party (e.g., the MME or a UE) is either ready to send out a specific message in the procedure to another party or wait for a specific message to come in. For instance, Figure 4 shows a pair of FSMs, one for the MME and the other for the UE; the state of each party can be decided once IDENTITY REQUEST or IDENTITY RESPONSE is seen. Our research demonstrates that such an FSM can be observed by the adversary under the threat model T_2 or estimated by the one under T_1 with the help of a sniffer.

Such a high-level FSM can be easily built from the LTE specification documentation [23] manually. Unlike the detailed state machine of the whole protocol, which is hard to recover, as the information about its states and transitions can be scattered across the whole documentation, the FSM used by the LTCG just roughly describes how a procedure operates, based upon a set of "key messages" highlighted by the document, which is used to fingerprint the time period when a test message can be issued. Actually, such messages and parties involved are already illustrated by a procedure diagram at the beginning of each procedure's specification (e.g., section 5.4.4.2 in 3GPP LTE 24.301 specification [23]). Converting them into the simple FSM is convenient. Note that such an FSM may not be complete: e.g., actions can be taken at a state in response to messages missing in the diagram (but described in other part of the documentation). Manual construction of such a complete FSM, even for each procedure, can be expensive, which our NLP-based approach is meant to avoid. Also the LTE NAS FSM becomes much more complicated if we consider the states carried through different procedures. However, our current design focuses on the vulnerabilities in a procedure that can be exploited regardless of the procedures executed before. Therefore, a simple FSM for each procedure is adequate for finding these problems. In our research, for all the procedures in LTE NAS EMM protocol as introduced early (see Section II), we constructed nine pairs of FSMs with 76 states for them, together with the templates for 26 messages. Such information forms a knowledge base for Atomic, which we release online [29].

State-event discovery. Using the knowledge base, the LTCG discovers the messages to be injected into a cellular network at the right state(s), based upon the semantics of an HI. As mentioned earlier, the HI is a short description with a conditional clause and consequence clauses that entail risky operations. The former carries the information about the state and the event (in terms of receiving or sending a message). Therefore, it goes through a semantic analysis run by the LTCG for recovering such information.

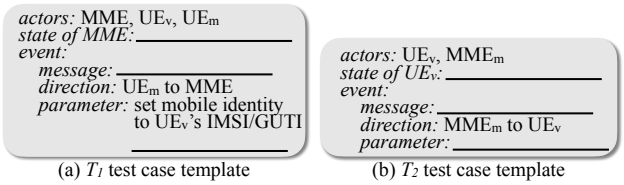
For this purpose, first we need to identify the conditional clause, a step leveraging the result of the implication analysis for HI discovery (Section III-B). The identified conditional clause is further processed to extract state and event informa-

tion. Specifically, in the event-driven LTE protocol, an event is associated with a message sent or received at a state. So, for most HIs explicitly describing the message transmission, we can leverage the unique grammatical structure of their conditional clauses to recover the message information: when the verb of the clause (pointed by "if", "when", etc. through a *mark* relation) is "send" or "receive", our approach looks for its object through the *obj* relation; if the object is "message", then the terms related to it through *compound* are considered to be the message name. For example, for the conditional clause in Figure 7 "receive a DETACH REQUEST message", the LTCG locates "receive" to find its object "message" and further identify "DETACH REQUEST", which have a *compound* relation with the object.

Besides the message name, some HIs also include information about the message's parameters: e.g., the conditional clause "whenever an ATTACH REJECT message with the EMM cause #11 is received" indicates that the parameter *EMM cause* shall be set to 11 in the ATTACH REJECT message. From the LTE documentation, we found that an HI's parameter information prefers to be presented as a parameter name (e.g., EMM cause) and a value with "#" in front of it (e.g., #11), which are grammatically connected to "message" through the preposition "with". This grammatical structure enables our approach to extract the parameter information. Specifically, the LTCG follows the relation *nmod:with* from the term "message" to identify its modifier; if it is "#", the term related to the symbol through *nummod* is considered to be parameter value (e.g., 11 in the example) and the one through *compound* (e.g., EMM cause) is labeled as parameter name.

Finding state information is more complicated, since it is provided implicitly in description, through transmission of a specific message and its timing (e.g., "before the IDENTITY RESPONSE message has been received") or through operations on a specific procedure (e.g., "before an ongoing procedure has been completed"). To address this challenge, we leverage a key observation that such information always appears in an adverb clause of time under the conditional clause, which starts with temporal adverbs like "before" and "after". Specifically, in the case that the verb at the center of the adverb clause (pointed to by "before" or "after" through *mark*) is "send" or "receive" and its object is "message", our approach automatically recover the message's name using the *compound* relation, as mentioned earlier, and then utilizes the semantics of the name, the verb and the timing adverb to infer the state(s) described. For example, from the clause "before the IDENTITY RESPONSE message has been received", the LTCG first detects the message "IDENTITY RESPONSE", which indicates the ongoing procedure – the identification procedure (see Figure 4), then it utilizes the verb "receive" to identify a state mentioned in the clause – MME_{Iden_3} ("IDENTITY RESPONSE... received" in Figure 4) and finally it concludes that the current state is MME_{Iden_2}, since it precedes MME_{Iden_3} from the time adverb "before" ("before the IDENTITY RESPONSE... received").

When the verb of the adverb clause is "complete" (or its synonyms) and its object is "procedure", our approach extracts the procedure name, again, using the *compound* relationship, and then uses it and the temporal adverbial to infer the current state. For example, from the clause "before the ongoing identification procedure has been completed" as shown in Figure 7, we discover the current procedure name and the temporal relations for locating the current state: the state is supposed to precede the completion of the identification procedure, that is, the one before MME_{Iden_3} and UE_{Iden_3}, which leaves four possible states we need to test: MME_{Iden_1}, UE_{Iden_1}, MME_{Iden_2} and UE_{Iden_2} in Figure 4.



(a) T_1 test case template (b) T_2 test case template

Fig. 8: Test case templates.

In addition to the above explicit description of messages and implicit presentation of states, there is a case in the LTE documentation where neither message nor state has been mentioned upfront. Specifically, a set of HIs report a "collision" situation in which the cellular system receives a message from one procedure when it is running another one. Oftentimes, the statement does not explicitly specify the message and instead just highlights the procedures involved: for example, "collision of the identification procedure with an attach procedure". To identify and further process such descriptions, which are often in the forms like "collision between A and B", "collision of A with B" etc., the LTCG utilizes a grammar template: from the dependency subtree of the conditional clause, our approach first captures the term "collision" (or its synonym) to find its objects through the *nmod* grammatical relation; if the objects turn out to be "procedure", we proceed to extract their names using the *compound* relation. Once the procedures in conflict are recovered, the LTCG automatically enumerates all the state-message combinations across them, for creating multiple test cases to cover all these combinations. As an example, let us look at the collision sentence above. From the sentence, our approach first discovers that the procedures in conflict are identification and attach and then retrieves from the knowledge base all states and messages in the two procedures to enumerate their combinations. Consider MME_{Iden_2} and UE_{Iden_2} in Figure 4. The former is associated with those issued in the attach procedure from the UE to the MME, while the latter is paired with the attach-related messages from the MME to the UE. All these state-message pairs are then used by the LTCG to generate test cases.

Test case construction. Once states and messages are recovered from the HIs, the LTCG continues to generate test cases automatically from the information. A test case produced by our approach is a parameterized template that describes the threat model, potentially vulnerable states and the message(s) for triggering a risky event. Figure 8 illustrates such templates.

To create a test case, our approach analyzes the discovered

state-event information. Specifically, for the threat model T_1, the LTCG first checks whether the HI's message is delivered from the UE to the MME, as the model states, and then inspects the message format retrieved from the well-summarized tables in the specification (e.g. Table 8.2.4.1 in 3GPP 24.301 [23]) to find out whether it includes an identity field that allows the malicious UE to impersonate the victim using its IMSI or GUTI. Upon passing both checks, our approach automatically fills in the MME state and message template to produce the test case. For the threat model T_2, only the message's direction (from the MME to UE) needs to be checked before the template is parameterized. Notably, in our research, we found that none of the T_2 HIs discovered describes the UE's state. So by default, the LTCG labels it with a receiver state that expects the event message described in the sentence. For example, if an event message is TAU REJECT, we mark the UE with the state right after sending out TAU REQUEST and waiting for its response. In our experiment, the LTCG automatically generated 68 test cases for the two threat models (Section II).

D. Testing and PoC Identification

For each test case generated by the LTCG, the PoC Identifier (PI) executes it to find out whether its message indeed triggers a risky operation without proper protection, achieving the attack goal stated in the threat model. To this end, the PI runs an LTE test environment built on SDR (Software-Defined Radio) boards and a set of instrumented (*hooked*) software simulators to analyze the operations of the network (MME) and the mobile (UE), and leverages the hooks implanted in the simulators to capture the state specified in the test case and command the attack module to issue the test message. The consequence of the attack is then evaluated from the network's log file. Following we elaborate the design and implementation of the test environment, and the steps to process a test case.

Test environment building. The test environment operated by the PI includes SDR boards (LimeSDR USB v1.4), each connecting to a computer that runs a simulator (Open5GS [30], OpenAirInterface (OAI) [31] or srsLTE [32]) to play the role of the UE, the network or the eNB. To avoid signal failure, we utilized RF (radio frequency) jumper cables to link the simulated MME and UE together. As illustrated in Figure 11 in Appendix, we configured two test environments for T_1 and T_2 respectively. Under T_1, two devices (each with an SDR board and a backend running a simulator) are used as a MME and a malicious UE, and a commercial phone (Samsung Galaxy S10) is used as the victim UE, which is connected to SCAT [33] through the USB bridge to monitor the UE states. Under T_2, a single device is deployed to simulate a malicious eNB and EPC that works with real-world mobile phones, Samsung Galaxy S10, Xiaomi Mix 2S, Google Pixel 2, and Nexus 6P used in our study, which also run SCAT to identify UE states and help evaluate the effectiveness of the attacks.

To capture a potentially vulnerable network or UE state and control the attack device to issue a message at the state, we implanted *hooks* in the simulators. A hook is a code fragment

```
1: procedure HOOK(state)
2:     if state in config then
3:         (entity, event, time)← config
4:         NOTIFY(entity, event)
5:         SLEEP(time)
6:     end if
7: end procedure
```

Fig. 9: The pseudocode of the HOOK function.

inserted into the simulator software where different states can be monitored and commands can be invoked at the right moment. Specifically, since each state in a procedure's FSM is defined as either sending or receiving a specific message of the procedure, we simply placed instructions to trigger the function HOOK right before the code for processing a specific inbound message and that for preparing a specific outbound message. HOOK (illustrated in Figure 9) contains a function NOTIFY to command the attack device (the attack UE under T_1 and the hooked device itself as a fake eNB under T_2) to issue a message, triggering the event leading to a risky operation, and another function SLEEP to provide a small attack window due to the delay in communicating with the attack device. Also each hook can be switched on or off by a configuration file *config*, depending on whether the corresponding state is specified in the file. Note that in our end-to-end confirmation of a discovered vulnerability (Section IV-B), all hooks were turned off, to ensure that the vulnerabilities can be realistically exploited. Also, each device runs a *listener* process to monitor commands from other devices, which upon receiving the command calls a simulator API (e.g., liblte_mme_pack_attach_request_msg) to send a message (e.g., for exploiting another device), which is parameterized based upon pre-constructed message templates.

Dynamic test construction. For each test case generated by the LTCG, the PI automatically builds its test by activating the hook and setting its parameters for the state specified in the test case. Specifically, our approach parses the test case description to identify the state, and then specifies it in the *config* file, together with the parameters for HOOK (Figure 9): the *entity* field is set to the message sender (according to the threat model), *event* (for NOTIFY) to the message name, direction and parameters as described in the test case (Figure 5), and *time* to a value (3.5s in our implementation, below the 4s minimal timer used by the LTE NAS protocol to avoid connection disruption).

A challenge in running a test case is how to guide our system (both the UE and the MME) to arrive at a target state. Such states are used by an FSM to model a procedure as mentioned earlier, which will be reached during a normal execution of the procedure. Therefore, all we need to do is to ensure that the procedure with the target state will be activated during the test. For this purpose, we looked into all the NAS EMM procedures (see Section II) evaluated in our study. Among them, the attach procedure runs each time when the system starts up, which will be followed by the authentication and the security mode control procedure. The identification procedure will be triggered by the UE

message carrying GUTI instead of IMSI. So invocation of these four procedures requires just setting the identity to IMSI or GUTI in *config*, when parameterizing the message to be sent. Other procedures, however, are more difficult to trigger, in the absence of specific events issued by the right party. For example, the service request procedure will only be invoked by the UE when it moves out of the idle mode and intends to upload data to the core network, while the paging procedure is called by the MME when it tries to wake up an idle UE for downloading data from the network. All these procedures will *not* execute until their trigger events take place. So in our research, we implanted three additional hooks in the system, one on the UE side and two on the MME side to activate the procedure carrying a target state. Specifically, the simulator for the UE device is instrumented right after the ATTACH ACCEPT message is delivered, indicating the completion of the attach procedure. Based upon the settings in *config*, the hook placed there either issues TAU REQUEST for launching tracking area updating procedure or SERVICE REQUEST for service request. A hook on the MME side is inserted right after the ATTACH REQUEST message is received, for running the detach or GUTI reallocation procedure, and the other is placed after the reception of the ATTACH COMPLETE message, for invoking the paging procedure.

In this way, the PI builds *config* according to the specification of each test case, to turn on proper hooks in the system and configure parameters. Once this has been done, it starts running the system to execute the test case.

Test execution and result analysis. After configuration of our system (the test environment), the PI starts running the system to execute the test case. For simplicity, our current design sets a new *config* and reboots the system for each test case, which introduces a moderate overhead, about 10 seconds for preparing a test. A further improvement could enable evaluation of multiple test cases without restarting the system, as long as these cases do not interfere with each other's execution. As mentioned earlier, following the configuration settings, devices issue messages to drive the system to the target state, in which the adversary issues a message in an attempt to trigger the risky event. Note that under both threat models, the attach procedure will always be executed. However, its setting may determine which procedure will follow, as mentioned earlier.

During the execution of each test case, the PI keeps track of all communication between the eNB and the EPC (running on the same device in our test environment, see Figure 11 in Appendix), from the launch of the system to 15 seconds after the attack message has been issued. Then, our approach inspects the network log file for the occurrence of a DoS event, as indicated by the appearance of either UEContextReleaseCommand or UEContextReleaseRequest. The first message is to ask an eNB to release its context, which is issued whenever the MME stops serving the UE. The second implies that the MME has already deleted the victim's context, which is sent by the eNB when it sees no traffic from the MME to the UE when the UE waiting for response. Further under T_2, once the victim UE was found to stop making connection attempt on the MME, indicating the presence of a vulnerability, we further analyzed the UE to understand how serious the problem is. Specifically, each time we adjusted the MME simulator's parameters to change its tracking area identity for the eNB, PLMN identity for the eNB, and then utilized Android Debug Bridge (ADB) to restart the victim UE, in an attempt to find out what it takes to let the UE reconnect to the MME, which can be observed from the presence of ATTACH REQUEST in the network traffic. In the end, for each vulnerability confirmed, the PI output its test case, together with its *config* as the PoC.

Ethics discussion. We conducted all the experiments in a responsible way: the whole test environment ran in a radio isolated shield box to prevent the exposure of signals emitted by the simulated LTE network and malicious UE.

IV. Evaluation and Discovery

A. Evaluation

We evaluated the effectiveness and performance of Atomic on the LTE NAS specification (3GPP TS 24.301 v16.6.0 [23]). Following we report the experimental results.

Settings. All experiments were conducted on two devices and four mobiles, each device including a SDR board (LimeSDR USB v1.4) and a backend host running simulators (Open5GS v2.0.0, OAI v0.5.0 and srsLTE v19.12.0) on the Ubuntu 18.04 with 2.10GHz CPU, 8GB memory and 512GB hard drive, and the four mobiles are Samsumg Galaxy S10, Xiaomi Mix 2S, Google Pixel 2 and Google Nexus 6P. Also we used a server for the semantic analysis (the HID and the LTCG), which runs Ubuntu 16.04 on a 2.10GHz CPU, 128GB memory and 3TB hard drive, and CUDA 10.0 on two GPUs (12GB GeForce GTX TITAN Z).

Effectiveness. We evaluated effectiveness of Atomic, based upon its individual components – the HID, the LTCG and the PI. Its end-to-end results are demonstrated by the precision of the vulnerabilities discovered, which is reported as the outcome of the PI.

For the HID, we analyzed the precision and coverage of the hazard indicators it reported. To this end, we first went through all detected HIs and found that HID reported 192 HIs with 87 true positives (42 under T_1 and 45 under T_2) and 105 false positives. 89 of them resulted from the limitations of the TE technique, which tends to have a low precision but a high recall for the simple hypothesis such as a ROD, while the rest 16 were caused by the extended RODs unrelated to risky operations. It is important to note that HI discovery is just one step of the Atomic pipeline and the false positives it produces will be *automatically eliminated* in the test-case generation and PoC identification steps. So for this component, we care more about false negatives than false positives. Then, to understand the recall of HID, we assume that all conditional sentences in the specification follow a Bernoulli distribution, either an HI or not, and estimate the probability of missing at least one HI by HID in the whole specification.

Specifically, we randomly sampled 4,000 sentences from the documentation and manually found 25 HIs, which were all captured by our HID. Based upon the assumption about the Bernoulli distribution, the distribution over missing HIs can be approximated by a truncated distribution. Over the distribution, we calculated the probability that at least one HI has been missed by HID ($p\{|\text{HID missed HIs}| \geq 1\}$), which turns out to be below 0.5%. This indicates that very likely our HID does not incur any false negative.

For the LTCG, we manually checked all the HIs and found that 68 of them were successfully converted to correct test cases. For the remaining HIs whose test cases were not generated, our findings show: (1) the message involved in the event does not contain UE identity information so cannot be used to attack the victim UE through the MME; (2) the risky operation is to abort a detach procedure, which cannot be exploited to disrupt the normal operations of the victim UE and the target MME; (3) the recipient of the HI's event is not the expected one specified in the test case template (e.g., the message is sent to the malicious UE not the target MME); (4) the HIs are false positives introduced by the TE model and do not contain adequate state/event information for generating attack messages.

For the PI, we checked all the tests it produced, which were all correctly constructed. Also all the vulnerabilities it reported were also confirmed manually. Further, we looked into the 26 test cases (13 under T_1 and 13 under T_2) that did not result in PoCs. Under T_1, 10 test cases could not be executed due to the simulator not implementing related procedures and messages. The remaining case failed to disable the victim UE's connection with the MME due to the ambiguity of the specification, which does not indicate the identity of the UE for aborting its procedure: as a result, the simulator may abort the victim UE's procedure for some messages (e.g., attach request) but stop the attack UE for other messages. For the 13 failed test cases under T_2, we found that some commercial UEs (smartphones like Xiaomi MIX 2S) do not follow the specifications when responding to some messages and continue to send request messages even after receiving reject messages, which causes the attacks to fail.

Performance. Our implementation of Atomic took less than 5 hours to go through the whole LTE 24.301 specification, which contains 549 pages with 13,598 sentences and 283,850 words, and generate and evaluate 68 test cases to find 42 vulnerabilities. Specifically, the HID spent 83.147s to extend two seeds (each for a threat model) to 13 RODs, 4.415s to find 5,652 conditional sentences, and 0.380s on average to run the TE model on one premise-hypothesis pair (totally 73,476 (5,652×13) pairs). Notably, we parallelized the TE analysis using two GPUs, and therefore were able to have this step done within 3.88 hours. The LTCG took 0.201s to process an HI and generate its test case on average. The PI used 64.953s on average (ranging from 40.212s to 95.500s) to configure and execute a test case under T_1, and 40.645s (from 26.090s to 87.011s) under T_2. This result offers strong evidence that our approach is capable of processing a large amount of documentation for vulnerability discovery in cellular networks.

B. Findings

Here we report the findings made in our study. As mentioned earlier, Atomic discovers vulnerabilities through its test environment, which has been built on top of three popular LTE simulators – Open5GS [30], OpenAirInterface [31] and srsLTE [32], the platforms widely used to study cellular network security [18]. We further performed end-to-end attacks to confirm that the threats discovered are realistic, through automatically inferring network states and denying the access of real-world mobile devices (Samsung Galaxy S10, Xiaomi MIX 2S, Google Pixel 2 and Google Nexus 6P) to the core network, even though we could not execute our attacks on commercial carrier networks, an act in violation of telecommunication laws and regulations such as "Title 47 of the United States Code". We have reported all our findings to 3GPP, device manufactures and other authorized parties.

Landscape. Altogether, our approach detected 42 vulnerabilities, including 10 new ones that have never been reported. Specifically, under T_1, Atomic found 10 vulnerabilities confirmed on simulators, all of which belong to a new category of core network flaws. Exploiting these vulnerabilities effectively block a victim UE from receiving the network service unless it enters the area served by a different MME, which can be difficult for the victim given the vast area managed by an MME, including many eNBs [13]. Under T_2, Atomic confirmed 32 vulnerabilities on the four commercial phones. These vulnerabilities allows the adversary running a fake base station to force a victim UE to stop connecting to the LTE network unless it moves into a new place covered by another eNB with a different PLMN identity or TAI, or has been rebooted. Table II and Table III in Appendix show each vulnerability discovered by Atomic under T_1 and T_2. In the rest of the section, we elaborate on these discoveries, their fundamental causes and real-world impacts.

UE DoS through core network. The ten core network vulnerabilities reported under T_1 are summarized in Table II in Appendix. These problems enable an adversary to selectively block the attempt from the victim's UE to attach to the core network, once he knows the UE's IMSI/GUTI. Looking into these vulnerabilities, we found that three of them (No.1, 6 and 7 in Table II) stem from a design issue in the NAS protocol: although the protocol requires mutual authentication between the MME and the UE, before this happens, the messages from the UE to initiate a procedure are not protected and their recipient, the MME, just relies on the identity information in the messages (IMSIs or GUTIs) to process them. Since it is well known that the IMSI and GUTI of a device can be exposed (see [7] and our end-to-end experiments), the adversary who acquires a victim UE's identity can impersonate it to issue new requests, causing the victim's ongoing procedure to stop. For instance, an ATTACH REQUEST sent by the adversary using the victim's IMSI will result in the MME's termination of the victim's attach procedure for processing

the new request. This risk has never been mentioned in the documentation, even though the related operation ("abort the procedure") is described as a response to different requests from the (presumably) same UE. Actually, all the simulators utilized in our research are found to identify the EMM context (the state of an ongoing procedure) using IMSI or GUTI, and are therefore vulnerable to the aforementioned DoS attack through UE impersonation.

After the victim's UE finishes the authentication procedure and the security mode control procedure, which exchanges keys for integrity protection and encryption, it is supposed to be fully protected, no longer vulnerable to an impersonation attack. Still, Atomic reported seven additional weaknesses at this stage in the popular simulators used in our tests, which turned out to be implementation flaws: OAI has no integrity protection in place for GUTI ATTACH REQUEST and other two simulators still clear the GUTI context for the victim UE even after the integrity check on the attack GUTI ATTACH REQUEST fails, and as a result, accept unauthorized messages from the adversary, which leads to denial of the victim UE's access to the core network.

End-to-end attack under T_1. It is important to note here that Atomic does not guarantee that all vulnerabilities it finds can always be exploited end-to-end. Some of them could just serve as a link on a complete kill chain for an end-to-end attack. On the other hand, the specific T_1 vulnerabilities discovered in our study turn out to be all exploitable, since we found that an end-to-end attack on them just needs to roughly estimate time windows at the procedure level: once the attack UE identifies the ongoing procedures with such vulnerabilities, it can continuously probe the MME with attack messages, which will successfully exploit all of them. These vulnerabilities are present in the attach, service request and paging procedures. Among them, the paging procedure can be easily discovered from the paging messages broadcast by the MME. For other two procedures, we found that they will be started by the victim UE based upon a unique identity parameter carried by the paging message issued by the MME: IMSI for the attach procedure and GUTI for the service request procedure. Hence, using a downlink message sniffer, we can identify the victim's target procedure from the paging message and its parameters.

In our research, we ran a sniffer to collect broadcast messages from a leading carrier under related laws and regulations and discovered 77 paging messages with different IMSIs and 86,133 paging with different GUTIs in one hour. In our testing environment, we further confirmed the feasibility of end-to-end attacks through automatically detecting such paging messages using a sniffer and then issuing attack messages (see the video demos online [29]). Note that the cost of the attack is low, requiring only the devices (300 dollars each, configured with open-source srsLTE modules) to eavesdrop on communication and issue messages. Moreover, the adversary only needs to generate messages at a low rate, 10 per second, for continuously blocking the victim UE's service, base upon our measurement on a commercial network that an attach procedure, service request procedure and paging procedure take around 458.9 to 987.3, 0.5 to 199.5 and 0.3 to 345.7 milliseconds respectively. This makes us believe that it is completely feasible to launch a large-scale DoS that prevents idle devices and those disconnected due to temporary network errors from re-connecting to the network. We have reported the vulnerabilities discovered and the end-to-end attacks (in our test enviroment) to authorized parties (CNVD [34] and CNNVD [35]), which have confirmed that the vulnerabilities are indeed present in major commercial carriers' networks (e.g., China Unicom [36]).

UE DoS through fake base station. Actually, messages can also be accepted on the UE side, as observed in the prior research [8]–[10]. As a result, a fake eNB (also 300 dollars, configured with srsLTE) that utilizes its elevated signal power to attract UEs can strategically send messages to the UEs, instructing them to disconnect from the network until the devices moves into a new place covered by another eNB with a different PLMN identity or TAI, or have been rebooted. This type of vulnerabilities have been discovered before [8]–[10], all through ad-hoc manual analyses. We found through experiments that Atomic actually generated test cases covering the previously reported vulnerabilities in this category, but 13 of them were not confirmed in our test environment, since the commercial phones we used act differently than expected from the specifications when responding to some events: for example, Xiaomi MIX 2S continues to send ATTACH REQUEST messages even after receiving ATTACH REJECT with EMM cause #3, which causes the attacks to fail; Google Pixel 2 continues to send TAU REQUEST messages after it receives TAU REJECT with EMM cause #7 (Table III), which violates the specification's requirement that considers the USIM as invalid until the UE is switched off. Table III presents all 32 confirmed vulnerabilities on four real-world phones. Looking into them, we found that 12 (No.1 to 12) are design flaws: a UE may receive messages (such as ATTACH REJECT) before it authenticates the network, so it could be manipulated during this period to connect to a fake network. For the rest 20 vulnerabilities, the phones we tested fail to check the integrity of the messages received after security context establishment, as required by the specifications, which results in implementation errors and opens the avenue for an impersonation attack. We have reported all these problems to phone manufacturers and received acknowledgements from all of them and an award of USD 2,000 from Google due to the gravity of the vulnerabilities.

V. DISCUSSION

Generality. Atomic is meant to find the vulnerabilities whose security implications are described in the specifications, with their hazardous consequences being triggered by an event under a certain system state. In line with prior research [11], we consider the Dolev-Yao model in our research, to identify the weaknesses that enable the attacks from the malicious UE (T_1), the malicious eNB (T_2) and the attacks on their communication. So in addition to what we reported under T_1

and T_2, our framework can also handle other weaknesses such as those subject to man-in-the-middle (MITM) downgrade attacks [10]. For this purpose, we need to utilize a test case template to describe the MITM threat model, choose a related ROD seed (such as "disable capability"), and further customize the test environment and the result analyzer for the new model. In the meantime, the whole workflow, including HID, test case generator and executor, remains unchanged. In our study, we performed a preliminary experiment on the threat model and the result shows that indeed such adjustment allows us to capture 3 known problems [10], [11] in this category (see Table IV at Appendix for details).

Limitations. Although we believe that the Atomic workflow is general, our current design and implementation are still limited: we consider that an HI is included in a single sentence or within a well-formatted multi-sentence structure, and the risky operation of the HI is described as a verb phrase; also our approach is not ready to handle implicit state and event description. These limitations render our system less effective in discovering other weaknesses than those related to DoS, which tend to be more explicitly stated in 3GPP documentation than other risks, given the specifications' utility-oriented nature. For example, the MITM vulnerability reported in the prior work [37] allows the adversary to change the parameter on an attach request message to EEA0 so that all the follow-up messages are not encrypted, leaking out such private information as locations to the eavesdropper; here the risky operation ("be ciphered with the null ciphering algorithm") is implicit and not described as a verb phrase; also its state and event have been scattered far away from its sentence. Although an effective cross-sentence analysis can certainly improve our current implementation and help detect such weaknesses from the implicit description like the example above, and even directly confirm design errors from the specifications (such as finding security-critical operations not covered by the security requirements located in other part of the documentation or even in a different document), it is known to be hard in the NLP community and new techniques need to develop to make a step closer to this end.

Future research. We believe that intelligent, data-driven and systematic security analysis is the key to the enhancement of cellular network's security assurance. Our study on Atomic has made the first step toward this end, but its application just scratched the surface of a large problem space such a technique can make inroads into. More specifically, we just ran our framework on the LTE NAS protocol, one of the hundreds of 3GPP protocols [38], while the design of the framework is meant to be general, supporting vulnerability discovery in other protocols on different layers of LTE and 5G, under various threat models (not only DoS but also other threats like information leak). Further, in addition to hazard indicators, other clues about security risks should also be discovered from documentation and utilized for security analysis, particularly those directly confirming the presence of design flaws (such as descriptions in conflict with the cellular network's security requirements) or strongly suggesting the existence of implementation errors (e.g., the absence of the description about a critical security check). Automatic discovery of these security weaknesses will contribute to the great elevation of cellular security and will be studied in our future research.

VI. RELATED WORK

DoS on cellular network. Denial of service (DoS) is among most serious security risks introduced by LTE vulnerabilities, which has been extensively studied. Some prior research focuses on the DoS attacks on the two essential LTE services, SMS and VoLTE. For example, Enck et al. [39]–[41] built DoS attacks by overloading the control channel of SMS. Kim et.al [42], [43] studied VoLTE security and identified several vulnerabilities that enable DoS attacks on the VoLTE. Besides the studies focusing on services, some previous work analyzes other LTE layers. For example, on the first layer, prior studies [1]–[6], [44] look into the physical channels and signals and demonstrate that both UEs and the eNBs are vulnerable to jamming attacks. On the third layer protocols, prior research investigates the DoS attacks launched by a fake eNB issuing sensitive control plane messages [7]–[10], [45], a malicious UE remotely disrupting the service of a victim UE [11]–[13], and a malicious man-in-the-middle modifying messages in a public channel [10]. The new technique we developed has been utilized to find DoS vulnerabilities in the NAS protocol on the layer three, which leads to the discovery of a new category of vulnerabilities that enable a malicious UE to block a victim UE's access to the cellular core network, and new vulnerabilities that can be exploited by a fake eNB to disable the UE it connects to.

Other reported vulnerabilities. Prior research also investigates various privacy risks in cellular networks, such as disclosing a user's IMSI [17], [20], [46]–[48], exposing one's locations, allowing user tracking [8], [20], [49], [50], eavesdropping on and tempering with user communication through man-in-middle-attacks [11], [18], [45]. Moreover, several studies [11], [14], [15], [51] demonstrate the attacks that violate authentication properties and further bypass the AKA procedure even though the mutual authentication procedure becomes mandatory from 4G. In addition, prior research also shows that the data charging system of the cellular network can be exploited. For example, Pent et al. [52]–[55] present accounting attacks that use the network service for free; Tu et.al [54], [56], [57] describe overbilling attacks on victim users. Another threat to LTE is to drain the battery of mobile phones, for example, through low-rate of retrieval of malicious MMS [58]. Although the Atomic framework has only been applied to discover DoS vulnerabilities in our study, our approach is meant to be general. Further effort will be made to extend it to cover some of the threats mentioned above.

Systematic method for LTE flaw discovery. With most LTE/5G vulnerabilities discovered manually, recent effort is moving toward seeking systematic solutions, leveraging two classic system analysis techniques: fuzzing and formal verification. Prominent examples include novel fuzzing techniques

[13], [45], [59] that discover the implementation errors in the core network and on the UE side, model checking effort [11], [14]–[16], [60] that evaluate the LTE control plane protocol and the 5G's authentication procedure. Formal verification helps find design flaws but requires significant human effort in building detailed models and identifying security properties, and is often error-prone. By comparison, our approach only entails a relatively small amount of human involvement in preparing knowledge base and utilizes NLP to automatically discover potential hazards from the LTE documentation and validate them in a test environment. This makes an important step towards intelligent and fully automated end-to-end vulnerability discovery in cellular networks.

VII. CONCLUSION

The security of cellular networks has been scrutinized in recent years, leading to the discovery of numerous vulnerabilities. However, most of them have been found through ad-hoc manual analysis. In our research, we developed a novel framework that makes a first step towards automated discovery of vulnerabilities in cellular networks from the documentation. Leveraging the observation that specifications often contain hazard indicators implying that a risky operation will take place if a specific event is triggered at a certain state, our approach utilizes NLP techniques to automatically discover HIs from a large amount of LTE documentation, recover state-event information from the HIs to generate test case, and further constructs tests and executes them in an LTE simulation environment to detect vulnerabilities both in LTE design and implementations. Running Atomic on the 549-page LTE NAS specification, we discovered 42 vulnerabilities, including 10 never reported before in both the core network and commercial UEs. All of them have been confirmed through end-to-end attacks and reported to 3GPP, device manufacturers, and other authorized parties. With its efficacy demonstrated on LTE NAS protocols under DoS threat models, we believe that the Atomic framework has great potentials to be applied to other LTE protocols and 5G for detecting other types of vulnerabilities.

VIII. ACKNOWLEDGMENT

We would like to thank the anonymous reviewers for their insightful comments, particularly our shepherd Thorsten Holz for the guidance for preparing the final version.

REFERENCES

[1] R. P. Jover, J. Lackey, and A. Raghavan, "Enhancing the security of lte networks against jamming attacks," *EURASIP Journal on Information Security*, vol. 2014, no. 1, p. 7, 2014.

[2] M. Lichtman, R. P. Jover, M. Labib, R. Rao, V. Marojevic, and J. H. Reed, "LTE/LTE-A jamming, spoofing, and sniffing: threat assessment and mitigation," *IEEE Communications Magazine*, vol. 54, no. 4, pp. 54–61, 2016.

[3] F. M. Aziz, J. S. Shamma, and G. L. Stüber, "Resilience of lte networks against smart jamming attacks: Wideband model," in *2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2015, pp. 1344–1348.

[4] R. P. Jover, "Security attacks against the availability of lte mobility networks: Overview and research directions," in *2013 16th international symposium on wireless personal multimedia communications (WPMC)*. IEEE, 2013, pp. 1–9.

[5] M. Lichtman, J. H. Reed, T. C. Clancy, and M. Norton, "Vulnerability of lte to hostile interference," in *2013 IEEE Global Conference on Signal and Information Processing*. IEEE, 2013, pp. 285–288.

[6] J. Xiao, X. Wang, Q. Guo, H. Long, and S. Jin, "Analysis and evaluation of jammer interference in lte," in *Proceedings of the Second International Conference on Innovative Computing and Cloud Computing*, 2013, pp. 46–50.

[7] 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, pp. 235–246.

[8] R. P. Jover, "Lte security, protocol exploits and location tracking experimentation with low-cost software radio," *arXiv preprint arXiv:1607.05171*, 2016.

[9] Y. Chuan and S. Chen, "On effects of mobility management signalling based dos attacks against lte terminals," in *2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC)*. IEEE, 2019, pp. 1–8.

[10] A. Shaik, R. Borgaonkar, N. Asokan, V. Niemi, and J.-P. Seifert, "Practical attacks against privacy and availability in 4G/LTE mobile communication systems," in *Network and Distributed System Security Symposium*. Internet Society, 2016.

[11] S. Hussain, O. Chowdhury, S. Mehnaz, and E. Bertino, "LTEInspector: A systematic approach for adversarial testing of 4G LTE," in *Network and Distributed Systems Security (NDSS) Symposium 2018*, 2018.

[12] 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.

[13] H. Kim, J. Lee, E. Lee, and Y. Kim, "Touching the untouchables: Dynamic security analysis of the lte control plane," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 1153–1168.

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

[15] C. Cremers and M. Dehnel-Wild, "Component-based formal analysis of 5G-AKA: Channel assumptions and session confusion," 2019.

[16] S. R. Hussain, M. Echeverria, I. Karim, O. Chowdhury, and E. Bertino, "5GReasoner: A Property-Directed Security and Privacy Analysis Framework for 5G Cellular Network Protocol," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 669–684.

[17] B. Hong, S. Bae, and Y. Kim, "Guti reallocation demystified: Cellular location tracking with changing temporary identifier." in *NDSS*, 2018.

[18] D. Rupprecht, K. Kohls, T. Holz, and C. Pöpper, "Breaking lte on layer two," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 1121–1136.

[19] D. Rupprecht, K. Kohls, T. Holz, and C. Popper, "IMP4GT: Impersonation attacks in 4G networks," in *Symposium on Network and Distributed System Security (NDSS). ISOC*, 2020.

[20] S. R. Hussain, M. Echeverria, O. Chowdhury, N. Li, and E. Bertino, "Privacy attacks to the 4g and 5g cellular paging protocols using side channel information." in *NDSS*, 2019.

[21] E. M. Clarke Jr, O. Grumberg, D. Kroening, D. Peled, and H. Veith, *Model checking*. MIT press, 2018.

[22] B. Blanchet, "Automatic verification of correspondences for security protocols," *Journal of Computer Security*, vol. 17, no. 4, pp. 363–434, 2009.

[23] 3GPP TS 24.301, "Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3," https://www.3gpp.org/ftp/Specs/archive/24_series/24.301/24301-g60.zip.

[24] A. Sasan, A. Mobasher, and T. Tofi, *Fourth-Generation Wireless Networks: Applications and Innovations: Applications and Innovations*. Information Science Reference, 2009.

[25] I. Dagan, O. Glickman, and B. Magnini, "The pascal recognising textual entailment challenge," in *Machine Learning Challenges Workshop*. Springer, 2005, pp. 177–190.

[26] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[27] AI2 Allen Institute for AI, "AllenNLP," https://allennlp.org.

[28] D. Chen and C. D. Manning, "A fast and accurate dependency parser using neural networks," in *Proceedings of the 2014 conference on*

empirical methods in natural language processing (EMNLP), 2014, pp. 740–750.
[29] "Atomic," https://sites.google.com/view/atomic-bookworm.
[30] "Open5GS," https://open5gs.org.
[31] "OpenAirInterface," https://openairinterface.org/.
[32] "srsLTE," https://github.com/srsLTE/srsLTE.
[33] "SCAT," https://github.com/fgsect/scat.
[34] "China National Vulnerability Database," https://www.cnvd.org.cn.
[35] "China National Vulnerability Database of Information Security," http://www.cnnvd.org.cn.
[36] "China Unicom," http://www.chinaunicom.com.
[37] M. Chlosta, D. Rupprecht, T. Holz, and C. Pöpper, "Lte security disabled: misconfiguration in commercial networks," in *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, 2019, pp. 261–266.
[38] "3GPP Specification Release version matrix," https://www.3gpp.org/DynaReport/SpecReleaseMatrix.htm.
[39] 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*, 2005, pp. 393–404.
[40] P. Traynor, W. Enck, P. McDaniel, and T. La Porta, "Mitigating attacks on open functionality in sms-capable cellular networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 1, pp. 40–53, 2008.
[41] P. Traynor, P. McDaniel, T. La Porta *et al.*, "On attack causality in internet-connected cellular networks," in *Proceedings of 16th USENIX Security Symposium*, vol. 21. USENIX Association, 2007, pp. 1–21.
[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*, 2015, pp. 328–339.
[43] G.-H. Tu, C.-Y. Li, C. Peng, and S. Lu, "How voice call technology poses security threats in 4g lte networks," in *2015 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2015, pp. 442–450.
[44] H. Yang, S. Bae, M. Son, H. Kim, S. M. Kim, and Y. Kim, "Hiding in Plain Signal: Physical Signal Overshadowing Attack on LTE," in *28th USENIX Security Symposium (USENIX Security 19)*, 2019.
[45] 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 16)*, 2016.
[46] A. Dabrowski, N. Pianta, T. Klepp, M. Mulazzani, and E. Weippl, "Imsi-catch me if you can: Imsi-catcher-catchers," in *Proceedings of the 30th annual computer security applications Conference*, 2014, pp. 246–255.
[47] M. S. A. Khan and C. J. Mitchell, "Trashing imsi catchers in mobile networks," in *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2017, pp. 207–218.
[48] S. Park, A. Shaik, R. Borgaonkar, A. Martin, and J.-P. Seifert, "White-stingray: Evaluating IMSI catchers detection applications," in *11th USENIX Workshop on Offensive Technologies (WOOT 17)*, 2017.
[49] A. Shaik, R. Borgaonkar, S. Park, and J.-P. Seifert, "New vulnerabilities in 4g and 5g cellular access network protocols: exposing device capabilities," in *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, 2019, pp. 221–231.
[50] A. J. Alzahrani and A. A. Ghorbani, "Sms mobile botnet detection using a multi-agent system: research in progress," in *Proceedings of the 1st International Workshop on Agents and CyberSecurity*, 2014, pp. 1–8.
[51] B. Blanchet, "A computationally sound mechanized prover for security protocols," *IEEE Transactions on Dependable and Secure Computing*, vol. 5, no. 4, pp. 193–207, 2008.
[52] Y. Go, J. Won, D. F. Kune, E. Jeong, Y. Kim, and K. Park, "Gaining control of cellular traffic accounting by spurious tcp retransmission," in *Network and Distributed System Security (NDSS) Symposium 2014*. Internet Society, 2014, pp. 1–15.
[53] Y. Go, D. F. Kune, S. Woo, K. Park, and Y. Kim, "Towards accurate accounting of cellular data for tcp retransmission," in *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications*, 2013, pp. 1–6.
[54] C. Peng, C.-y. Li, G.-H. Tu, S. Lu, and L. Zhang, "Mobile data charging: new attacks and countermeasures," in *Proceedings of the 2012 ACM conference on Computer and communications security*, 2012, pp. 195–204.
[55] C. Peng, C.-Y. Li, H. Wang, G.-H. Tu, and S. Lu, "Real threats to your data bills: Security loopholes and defenses in mobile data charging," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014, pp. 727–738.
[56] C. Peng, G.-h. Tu, C.-y. Li, and S. Lu, "Can we pay for what we get in 3g data access?" in *Proceedings of the 18th annual international conference on Mobile computing and networking*, 2012, pp. 113–124.
[57] G.-H. Tu, C. Peng, C.-Y. Li, X. Ma, H. Wang, T. Wang, and S. Lu, "Accounting for roaming users on mobile data access: Issues and root causes," in *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, 2013, pp. 305–318.
[58] R. Racic, D. Ma, and H. Chen, "Exploiting mms vulnerabilities to stealthily exhaust mobile phone's battery," in *2006 Securecomm and Workshops*. IEEE, 2006, pp. 1–10.
[59] W. Johansson, M. Svensson, U. E. Larson, M. Almgren, and V. Gulisano, "T-fuzz: Model-based fuzzing for robustness testing of telecommunication protocols," in *2014 IEEE Seventh International Conference on Software Testing, Verification and Validation*. IEEE, 2014, pp. 323–332.
[60] G.-H. Tu, Y. Li, C. Peng, C.-Y. Li, H. Wang, and S. Lu, "Control-plane protocol interactions in cellular networks," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 223–234, 2014.

IX. APPENDIX

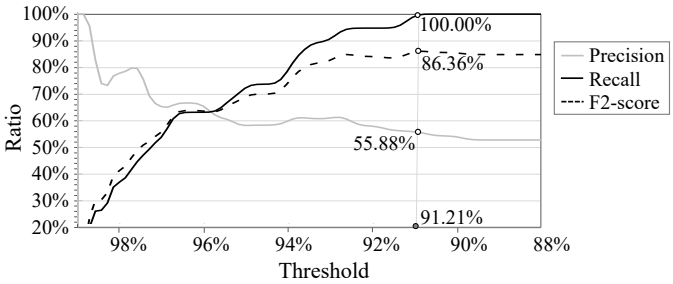
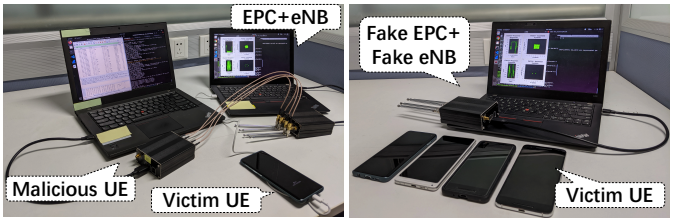


Fig. 10: The precision, recall, and f2-score of TE.



(a) T_1 test environment (b) T_2 test environment

Fig. 11: Test environment.

TABLE I: The summary of extended RODs.

a/b: a is the number of test cases or vulnerabilities only identified by the ROD, b is the number of test cases or vulnerabilities identified by the ROD but not only such one.

			Test Case (*a/b*)	**Vulnerability** (*a/b*)
T_1	**ROD seed**	abort procedure	2/18	2/8
	extended ROD	progress procedure	2/20	1/8
		initiate procedure	0/3	0/2
		release resource	0/2	0/0
		deactivate context	0/0	0/0
T_2	**ROD seed**	consider usim	12/12	11/11
	extended ROD	delete TAI	3/30	3/20
		delete list	1/28	1/18
		delete GUTI	0/27	0/17
		delete eKSI	0/20	0/16
		delete KSIAME	0/1	0/1
		delete TMSI	0/0	0/0
		delete LAI	0/0	0/0

TABLE II: Summary of vulnerabilities under T_1.

/rs: short for "ready to send", /wf: short for "waiting for", D: design weakness, I: implementation error,
A: Attach, P: Paging, T: Identification, S_1: Security mode control, S_2: Service request,
■: confirmed PoC (new vulnerability), ×: test case that cannot be confirmed in our testing environment

No.	FSM	State	Event (UEm to MME)	Open5GS	OAI	srsLTE	Vulnerability Type
1	A	MME /rs ATTACH ACCEPT/REJECT	ATTACH REQUEST	■	■	■	D
2	A	MME /wf ATTACH COMPLETE	ATTACH REQUEST	■	■	■	I
3	A	MME /wf ATTACH COMPLETE	DETACH REQUEST	×	■	■	I
4	T	MME /wf IDENTITY RESPONSE	ATTACH REQUEST	■	■	■	I
5	T	MME /wf IDENTITY RESPONSE	DETACH REQUEST	×	■	■	I
6	S_1	MME /wf SECURITY MODE COMPLETE	ATTACH REQUEST	■	■	■	D
7	S_1	MME /wf SECURITY MODE COMPLETE	DETACH REQUEST	×	■	■	D
8	S_2	MME /rs SERVICE REJECT	ATTACH REQUEST	■	■	■	I
9	P	MME /wf SERVICE REQUEST	ATTACH REQUEST	■	■	■	I
10	P	MME /wf EXTENDED SERVICE REQUEST	ATTACH REQUEST	■	■	■	I

TABLE III: Summary of vulnerabilities under T_2.

□: confirmed PoC, ×: test case that cannot be confirmed in our testing environment, D: design weakness, I: implementation error, /w: short for "with"

No.	State	Event (MMEm to UE)	Samsumg Galaxy S10	Xiaomi MIX 2S	Google Pixel 2	Google Nexus 6P	Vulnerability Type
1	UE /wf ATTACH ACCEPT/REJECT	ATTACH REJECT /w EMM cause #3	□	×	□	×	D
2	UE /wf ATTACH ACCEPT/REJECT	ATTACH REJECT /w EMM cause #6	□	×	□	□	D
3	UE /wf ATTACH ACCEPT/REJECT	ATTACH REJECT /w EMM cause #7	□	□	□	×	D
4	UE /wf ATTACH ACCEPT/REJECT	ATTACH REJECT /w EMM cause #8	□	×	□	×	D
5	UE /wf ATTACH ACCEPT/REJECT	ATTACH REJECT /w EMM cause #11	□	□	□	×	D
6	UE /wf ATTACH ACCEPT/REJECT	ATTACH REJECT /w EMM cause #12	□	□	□	×	D
7	UE /wf ATTACH ACCEPT/REJECT	ATTACH REJECT /w EMM cause #13	□	□	□	×	D
8	UE /wf ATTACH ACCEPT/REJECT	ATTACH REJECT /w EMM cause #15	□	□	□	×	D
9	UE /wf ATTACH ACCEPT/REJECT	ATTACH REJECT /w EMM cause #42	□	□	□	×	D
10	UE /wf ATTACH ACCEPT/REJECT	DETACH REQUEST /w EMM cause #3	×	×	×	□	D
11	UE /wf ATTACH ACCEPT/REJECT	DETACH REQUEST /w EMM cause #6	×	×	×	□	D
12	UE /wf ATTACH ACCEPT/REJECT	DETACH REQUEST /w EMM cause #8	×	×	×	□	D
13	UE /rs AUTHENTICATION RESPONSE	AUTHENTICATION REJECT	□	×	□	□	I
14	UE /wf TAU REJECT	TAU REJECT /w EMM cause #3	□	×	□	□	I
15	UE /wf TAU REJECT	TAU REJECT /w EMM cause #6	□	×	□	□	I
16	UE /wf TAU REJECT	TAU REJECT /w EMM cause #7	□	□	×	□	I
17	UE /wf TAU REJECT	TAU REJECT /w EMM cause #8	×	×	×	□	I
18	UE /wf TAU REJECT	TAU REJECT /w EMM cause #11	□	□	□	□	I
19	UE /wf TAU REJECT	TAU REJECT /w EMM cause #12	□	□	□	□	I
20	UE /wf TAU REJECT	TAU REJECT /w EMM cause #13	□	□	□	□	I
21	UE /wf TAU REJECT	TAU REJECT /w EMM cause #14	□	□	□	□	I
22	UE /wf TAU REJECT	TAU REJECT /w EMM cause #15	□	□	□	□	I
23	UE /wf TAU REJECT	TAU REJECT /w EMM cause #42	□	□	□	□	I
24	UE /wf SERVICE REJECT	SERVICE REJECT /w EMM cause #3	□	×	□	□	I
25	UE /wf SERVICE REJECT	SERVICE REJECT /w EMM cause #6	□	×	□	□	I
26	UE /wf SERVICE REJECT	SERVICE REJECT /w EMM cause #7	□	□	□	□	I
27	UE /wf SERVICE REJECT	SERVICE REJECT /w EMM cause #8	□	×	□	×	I
28	UE /wf SERVICE REJECT	SERVICE REJECT /w EMM cause #11	□	×	□	□	I
29	UE /wf SERVICE REJECT	SERVICE REJECT /w EMM cause #12	□	□	□	□	I
30	UE /wf SERVICE REJECT	SERVICE REJECT /w EMM cause #13	□	□	□	□	I
31	UE /wf SERVICE REJECT	SERVICE REJECT /w EMM cause #15	□	□	□	□	I
32	UE /wf SERVICE REJECT	SERVICE REJECT /w EMM cause #42	□	□	□	□	I

TABLE IV: Summary of vulnerabilities under $MITM$

□: confirmed PoC, ×: test case that cannot be confirmed in our testing environment, /w: short for "with"

No.	HI	ROD	State	Event	Samsumg Galaxy S10	Xiaomi MIX 2S	Google Pixel 2	Google Nexus 6P
1	If the ATTACH ACCEPT message includes the Additional update result IE with value "SMS only" or "CS Fallback not preferred", a UE operating in CS/PS mode 1 with "IMS voice not available" shall attempt to select GERAN or UTRAN radio access technology and **disable the E-UTRA capability** (see subclause 4.5).	disable capability	UE /wf ATTACH ACCEPT/REJECT	ATTACH ACCEPT /w additional update result "SMS only" / "CS Fallback not preferred"	×	□	□	□
2	If the TRACKING AREA UPDATE ACCEPT message includes the Additional update result IE with value "SMS only" or "CS Fallback not preferred", a UE operating in CS/PS mode 1 with "IMS voice not available" shall attempt to select GERAN or UTRAN radio access technology and **disable the E-UTRA capability** (see subclause 4.5).	disable capability	UE /wf TAU ACCEPT	TAU ACCEPT /w additional update result "SMS only" / "CS Fallback not preferred"	×	□	□	□
3	if the UE is in WB-S1 mode and the Extended EMM cause IE with value "E-UTRAN not allowed" is included in the TRACKING AREA UPDATE REJECT message, the UE ...; then **the UE shall disable the E-UTRA capability** as specified in subclause 4.5 and search for a suitable cell in another location area or 5GS tracking area;	disable capability	UE /wf TAU REJECT	TAU REJECT /w extended EMM cause "E-UTRAN not allowed"	□	□	□	□

TABLE V: Summary of cases from a ROD to HI, test case and PoC under T_1

■: confirmed PoC, ×: test case that cannot be confirmed in our testing environment, /wf: short for "waiting for", /rs: short for "ready to send"

HI No.	HI	ROD	State	Event	Open5GS	OAI	srsLTE
1	e) More than one ATTACH REQUEST received and no ATTACH ACCEPT or ATTACH REJECT message has been sent - If one or more of the information elements in the ATTACH REQUEST message differs from the ones received within the previous ATTACH REQUEST message, **the previously initiated attach procedure shall be aborted and the new attach procedure shall be executed**;	progress procedure abort procedure	MME /rs ATTACH ACCEPT/REJECT	ATTACH REQUEST	■	■	■
2	d) ATTACH REQUEST received after the ATTACH ACCEPT message has been sent and before the ATTACH COMPLETE message is received - If one or more of the information elements in the ATTACH REQUEST message differ from the ones received within the previous ATTACH REQUEST message, **the previously initiated attach procedure shall be aborted** if the ATTACH COMPLETE message has not been received and **the new attach procedure shall be progressed**; or	progress procedure initiate procedure abort procedure	MME /wf ATTACH COMPLETE	ATTACH REQUEST	■	■	■
3	h) DETACH REQUEST message received before ATTACH COMPLETE message. **The network shall abort the attach procedure and shall progress the detach procedure** as described in subclause 5.5.2.2.	progress procedure abort procedure	MME /wf ATTACH COMPLETE	DETACH REQUEST	×	■	■
4	If the network receives an ATTACH REQUEST message before the ongoing identification procedure has been completed and no attach procedure is pending on the network (i.e. no ATTACH ACCEPT/REJECT message has still to be sent as an answer to an ATTACH REQUEST message), **the network shall proceed with the attach procedure**.	progress procedure	MME /wf IDENTITY RESPONSE	ATTACH REQUEST	■	■	■
5	If the network receives a DETACH REQUEST message before the ongoing identification procedure has been completed, **the network shall abort the identification procedure and shall progress the detach procedure**.	progress procedure abort procedure	MME /wf IDENTITY RESPONSE	DETACH REQUEST	×	■	■
6	c) Collision between security mode control procedure and attach, service request, tracking area updating procedure or detach procedure not indicating switch off **The network shall abort the security mode control procedure and proceed with the UE initiated procedure**.	progress procedure abort procedure	MME /wf SECURITY MODE COMPLETE	ATTACH REQUEST	■	■	■
6	c) Collision between security mode control procedure and attach, service request, tracking area updating procedure or detach procedure not indicating switch off **The network shall abort the security mode control procedure and proceed with the UE initiated procedure**.	progress procedure abort procedure	MME /wf SECURITY MODE COMPLETE	DETACH REQUEST	×	■	■
7	If an ATTACH REQUEST message is received and the service request procedure has not been completed or a SERVICE REJECT message has not been sent, **the network may initiate the EMM common procedures**, e.g. the EMM authentication procedure.	progress procedure initiate procedure	MME /rs SERVICE ACCEPT/REJECT	ATTACH REQUEST	■	■	■
8	a) ATTACH REQUEST message received when paging procedure is ongoing. If an integrity-protected ATTACH REQUEST message is received from the UE and successfully integrity checked by the network, **the network shall abort the paging procedure**.	abort procedure	MME /wf SERVICE REQUEST	ATTACH REQUEST	■	■	■
8	b) ATTACH REQUEST message received when paging procedure is ongoing. If an integrity-protected ATTACH REQUEST message is received from the UE and successfully integrity checked by the network, **the network shall abort the paging procedure**.	abort procedure	MME /wf EXTENDED SERVICE REQUEST	ATTACH REQUEST	■	■	■

TABLE VI: Examples of cases from a ROD to HI, test case and PoC under T_2

Note: All the cases can be found online [29]

■: confirmed PoC, ×: test case that cannot be confirmed in our testing environment, /w: short for "with", /wf: short for "waiting for", /rs: short for "ready to send"

No.	HI	ROD	State	Event	Samsumg Galaxy S10	Xiaomi MIX 2S	Google Pixel 2	Google Nexus 6P
1	The UE shall take the following actions depending on the EMM cause value received in the ATTACH REJECT message. #3 (Illegal UE); **The UE shall consider the USIM as invalid** for EPS services and non-EPS services until switching off or the UICC containing the USIM is removed.	consider USIM	UE /wf ATTACH ACCEPT/REJECT	ATTACH REJECT /w EMM cause #3	■	×	■	×
2	The UE shall take the following actions depending on the EMM cause value received in the ATTACH REJECT message. #11 (PLMN not allowed); The UE shall set the EPS update status to EU3 ROAMING NOT ALLOWED (and shall store it according to subclause 5.1.3.3) and **shall delete any GUTI, last visited registered TAI, TAI list and eKSI**, and reset the attach attempt counter.	delete GUTI delete TAI delete list delete eKSI	UE /wf ATTACH ACCEPT/REJECT	ATTACH REJECT /w EMM cause #11	■	■	■	×
3	If the detach type indicates "re-attach not required", the UE shall take the following actions depending on the received EMM cause value: #6 (Illegal ME); **The UE shall consider the USIM as invalid** for EPS services until switching off or the UICC containing the USIM is removed.	consider USIM	UE /wf ATTACH ACCEPT/REJECT	DETACH REQUEST /w EMM cause #6	×	×	×	■
4	If the detach type indicates "re-attach not required", the UE shall take the following actions depending on the received EMM cause value: #8 (EPS services and non-EPS services not allowed); **The UE shall consider the USIM as invalid** for EPS services until switching off or the UICC containing the USIM is removed.	consider USIM	UE /wf ATTACH ACCEPT/REJECT	DETACH REQUEST /w EMM cause #8	×	×	×	■
5	Upon receipt of an AUTHENTICATION REJECT message, a) if the message has been successfully integrity checked by the NAS, the UE shall set the update status to EU3 ROAMING NOT ALLOWED, **delete the stored GUTI, TAI list, last visited registered TAI and KSIASME**.	delete GUTI delete TAI delete list delete KSIASME	UE /rs AUTHENTICATION RESPONSE	AUTHENTICATION REJECT	■	×	■	■
6	The UE shall take the following actions depending on the EMM cause value received in the TRACKING AREA UPDATE REJECT message. #13 (Roaming not allowed in this tracking area); The UE shall set the EPS update status to EU3 ROAMING NOT ALLOWED (and shall store it according to subclause 5.1.3.3) and shall **delete the list of equivalent PLMNs**.	delete list	UE /wf TRACKING AREA UPDATE REJECT	TRACKING AREA UPDATE REJECT /w EMM cause #13	■	■	■	■
7	The UE shall take the following actions depending on the EMM cause value received in the TRACKING AREA UPDATE REJECT message. #15 (No suitable cells in tracking area); The UE shall store the current TAI in the list of "forbidden tracking areas for roaming" and shall **remove the current TAI from the stored TAI list** if present and:	delete TAI	UE /wf TRACKING AREA UPDATE REJECT	TRACKING AREA UPDATE REJECT /w EMM cause #15	■	■	■	■
8	The UE shall take the following actions depending on the received EMM cause value in the SERVICE REJECT message. #42 (Severe network failure); The UE shall set the EPS update status to EU2 NOT UPDATED, and shall **delete any GUTI, last visited registered TAI, eKSI, and list of equivalent PLMNs**.	delete GUTI delete TAI delete list delete eKSI	UE /wf SERVICE REJECT	SERVICE REJECT /w EMM cause #42	■	■	■	■