LTEInspector: A Systematic Approach for Adversarial Testing of 4G LTE

Syed Rafiul Hussain
Purdue University
hussain1@purdue.edu

Omar Chowdhury
The University of Iowa
omar-chowdhury@uiowa.edu

Shagufta Mehnaz
Purdue University
smehnaz@purdue.edu

Elisa Bertino
Purdue University
bertino@purdue.edu

***Abstract*—In this paper, we investigate the security and privacy of the three critical procedures of the 4G LTE protocol (i.e., attach, detach, and paging), and in the process, uncover potential design flaws of the protocol and unsafe practices employed by the stakeholders. For exposing vulnerabilities, we propose a model-based testing approach LTEInspector which lazily combines a symbolic model checker and a cryptographic protocol verifier in the symbolic attacker model. Using LTEInspector, we have uncovered 10 new attacks along with 9 prior attacks, categorized into three abstract classes (i.e., security, user privacy, and disruption of service), in the three procedures of 4G LTE. Notable among our findings is the *authentication relay attack* that enables an adversary to spoof the location of a legitimate user to the core network without possessing appropriate credentials. To ensure that the exposed attacks pose real threats and are indeed realizable in practice, we have validated 8 of the 10 new attacks and their accompanying adversarial assumptions through experimentation in a real testbed.**

I. Introduction

The adoption of Fourth generation Long Term Evaluation (4G LTE)—the de facto standard for cellular telecommunication—has seen a stable growth in recent years, replacing prior generations due to its promise of improved assurances (e.g., higher bandwidth, reliable connectivity, enhanced security). Cellular networks which are part of a nation's critical infrastructure not only influence our society as a whole (e.g., business, public-safety message dissemination) but also can impact us at a more personal level by enabling applications that often improve our quality of life (e.g., navigation). Because of their ubiquitous presence and use for critical applications, cellular networks are, however, attractive attack targets for malicious parties. Resourceful adversaries (e.g., nation-states, foreign intelligence agencies, terrorists) can wreak havoc by exploiting vulnerabilities of the cellular network ecosystem (e.g., surveillance [1], cyberwarfare [2]). It is hence pivotal to investigate the existing design and deployments of cellular networks for detecting potential vulnerabilities.

There is already a substantial work that has analyzed the security and privacy of telecommunication systems, and also identified design weaknesses of the 3GPP (Third Generation Partnership Project) standard [3], [4] and unsafe practices by the responsible stakeholders [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17]. Such work, however, has at least one of the following limitations: (A) Analyses do not use systematic methodologies for attack discovery [5], [6], [7], [8], [9], [10], [11], [12], [13], [15]; (B) Analyses focus on prior generations of the protocols only, and hence some of the findings do not directly apply to 4G LTE [5], [6], [7], [8], [9], [10], [11], [12], [16]; (C) Analyses do not explicitly reason about adversarial actions [17].

Problem and scope. The 4G LTE protocol can be viewed as an amalgamation of multiple critical procedures, such as attach, paging, detach, handover, and calls — to name a few. Each of these procedures is complex and requires an in-depth security and privacy analysis of its own. Among the different procedures, attach, paging, and detach are critical for the correct and reliable functionality of the other procedures. For instance, without a correct and secure attach procedure (i.e., the initial secure connection setup), the security bootstrapping process is likely to be vulnerable; this may have serious consequences, such as man-in-the-middle attacks, spurious mobile billing, or even life threatening risks. This paper thus addresses the following central research question: *is it possible to develop a systematic approach for scrutinizing the attach, detach, and paging procedures to uncover vulnerabilities that can be shown to be realizable in practice by an adversary*?

Challenges. Any testing approach analyzing the security and privacy of the 4G LTE protocol needs to address the following challenges: (a) *Specification*: The cellular protocol lacks a formal specification, and the standard [3], [4] often suffers from ambiguity and under-specification. (b) *Protocol Complexity*: The cellular protocol—comprising of multiple (cryptographic) sub-protocols—is *stateful* in nature [17]. Also, analyses will likely experience scalability challenges due to the presence of multiple types of protocol participants, and messages containing data with a large domain. (c) *Closed system*: A majority of the deployed cellular systems are proprietary and closed systems which require any testing approach to be black-box and system-agnostic. (d) *Legal barrier*: Regulatory requirements [18] prohibit transmission in the licensed spectrum making dynamic network testing and attack validation challenging.

Approach. For analyzing the critical procedures of 4G LTE, in this paper, we take a first step by proposing LTEInspector which employs a property-driven adversarial model-based testing philosophy. LTEInspector considers the standard symbolic adversary model (alternatively known as the Dolev-Yao model

Network and Distributed Systems Security (NDSS) Symposium 2018
18-21 February 2018, San Diego, CA, USA
ISBN 1-1891562-49-5
http://dx.doi.org/10.14722/ndss.2018.23313
www.ndss-symposium.org

[19]) for its analysis. LTEInspector takes the relevant 4G LTE abstract model $\mathcal{M}$ and a desired property φ, and tries to find a violation of φ in $\mathcal{M}$. The set of properties that LTEInspector aims to check include *authenticity* (e.g., disallowing impersonation), *availability* (e.g., preventing service denial), *integrity* (e.g., restricting unauthorized billing), and *secrecy* of user's sensitive information (e.g., preventing activity profiling).

As a prerequisite of LTEInspector's analysis, we first construct the 4G LTE ecosystem model by consulting the standard [3], [4]. Our model $\mathcal{M}$ (publicly available in https://github.com/relentless-warrior/LTEInspector) captures the *abstract* functionality (ignoring low-level implementation details) of the 4G LTE ecosystem—only relevant to the analysis of the three procedures—as synchronous communicating finite state machines (FSM). Each FSM captures the stateful functionality of a protocol participant's (i.e, user's cellular device and the core network) at the Non-Access Stratum (NAS) protocol layer [3], [4]. The two FSMs communicate with each other through public (adversary-controlled) communication channels by sending each other NAS layer messages.

Our analysis is an instance of the parameterized system verification problem (i.e., parameterized by the number of protocol participants) which is generally undecidable [20]; achieving both soundness and completeness is thus impossible. Consequently, we follow the conventional method of aiming for soundness instead of completeness, that is, if our approach reports a violation, it is indeed a violation; we cannot, however, detect all violations. Also, checking compliance of the protocol model against desired security and privacy properties often requires simultaneously reasoning about: (❶) temporal ordering of different events/actions (i.e., trace properties such as response properties [21]), (❷) cryptographically-protected messages and constructs (e.g., encryption, hashing), and (❸) other rich constraints (e.g., linear integer arithmetic constraints). General purpose model checkers [22], [23] have shown promise in successfully reasoning about properties concerning ❶ and ❸. Cryptographic protocol verifiers [24], [25], [26], [27], [28], [29], although proficient in verifying cryptography related properties, for tractability reasons only provide primitive support at best for properties concerning ❶ and ❸. This naturally leads us to the question: *is it possible to get the best of both these techniques*?

To this end, LTEInspector lazily (or, on an on-demand basis) combines the reasoning powers of a symbolic model checker and a cryptographic protocol verifier. To the best of our knowledge, in the context of 4G LTE, the use of symbolic model checking and a cryptographic protocol verifier to reason about rich temporal trace properties is novel. In this approach, we first abstract away all cryptography-related constructs from the model $\mathcal{M}$ and the desired property φ and only reason about aspects ❶ and ❸ of the φ (denoted by φ_{abs}). For any violation of φ_{abs} in $\mathcal{M}$, the symbolic model checker would yield a counterexample π demonstrating the violation. Now, π may include adversary actions which may not be realizable due to cryptographic assumption violations (e.g., constructing a valid ciphertext of a message without possessing the encryption key). To rule out such cases, for each adversary action in π, we query a cryptographic protocol verifier to check the action's feasibility with accordance to the cryptographic assumptions. In case all adversary actions in π turn out to be feasible, we can report π to be a feasible vulnerability. If, however, there exists one adversary action in π which is not feasible, we refine the property φ_{abs} to rule out traces in which the adversary takes that action. The analysis is then run again with the refined property. For further confidence, we validate π by concretely executing it in a testbed.

Finally, we show the application of a technique we call *attack chaining* in which seemingly low-impact attacks are stitched together to yield a damaging high-impact attack. We show its successful application by chaining together attacks, exposed using LTEInspector, to allow an adversary to carry out an authentication relay attack in the 4G LTE network.

Findings. Notable among our findings is the *authentication relay attack* which enables an adversary to connect to the core networks—without possessing any legitimate credentials—while impersonating a victim cellular device. Through this attack the adversary can poison the location of the victim device in the core networks, thus allowing setting up a false alibi or planting fake evidence during a criminal investigation.

Other notable attacks reported in this paper enable an adversary to obtain user's coarse-grained location information and also mount denial of service (DoS) attacks. In particular, using LTEInspector, we obtained the intuition of an attack which enables an adversary to possibly hijack a cellular device's paging channel with which it can not only stop notifications (e.g., call, SMS) to reach the device but also can inject fabricated messages resulting in multiple implications including energy depletion and activity profiling.

Contributions. In summary, the paper makes the following technical contributions:

(1) We propose LTEInspector—a systematic model-based adversarial testing approach—that leverages the combined power of a symbolic model checker and a protocol verifier for analyzing three critical procedures (i.e., attach, detach, and paging) of the 4G LTE network. The general principle employed by LTEInspector is to be tool-agnostic, that is, it can be instantiated through any generic symbolic model checker and cryptographic protocol verifier.

(2) We show the effectiveness of our approach in finding new vulnerabilities as well as 9 prior attacks. Our approach has contributed to exposing 10 new attacks.

(3) We show that the majority of our new attacks (i.e., 8 out of 10) are realizable in practice through experimentation in a low cost (i.e., $\$3,900$), real test-bed while adhering to ethical, legal, and moral practices.

II. LTE Preliminaries

In this section, we provide a brief primer of 4G LTE. For the ease of exposition, we simplify the network architecture substantially (see Figure 1) and only focus on the aspects relevant to the attach, detach, and paging procedures.

A. LTE Network Architecture

The LTE network is broadly comprised of the following three components: the cellular device (also known as user equipment or UE), the radio access network (or, (E-UTRAN), and the core network or Evolved Packet Core (EPC).

UE: UE is the cellular device equipped with a universal subscriber identity module known as SIM card. The SIM card

securely stores the unique international mobile subscriber identity (IMSI) number and its associated cryptographic keys used for the UE identification and authentication during the UE's connection initiation with the EPC. The UE also has its own device-specific unique identity, called the international mobile equipment identity (IMEI) also used for identification. The IMSI and IMEI are sensitive in the sense that exposing them can make the UE prone to illegitimate tracking/impersonation.

E-UTRAN: A geographical area, in the context of LTE, is partitioned into hexagonal cells (see Figure 1) where each cell is serviced by a single base station (i.e., eNodeB); providing its nearby cellular devices the connectivity to the Internet through the carrier's core network. The eNodeB can be roughly viewed as an intermediary facilitating the connection between the UE and the EPC. In essence, the E-UTRAN is the network between a UE and the eNodeB, and between pairs of eNodeBs.

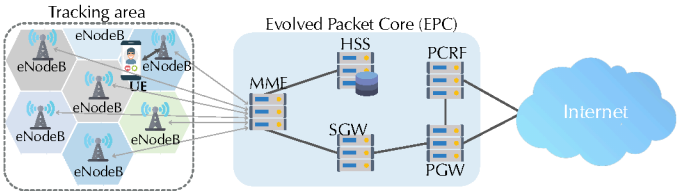

Figure 1: The LTE Network Architecture

EPC: We now describe those EPC components that are relevant to our discussion, i.e., the MME (Mobility Management Entity) and the HSS (Home Subscriber Server).

(1) Mobility Management Entity (MME): The MME manages attach (including authentication and key agreement), paging, and detach procedures of the UEs in a particular tracking area (formed by a set of hexagonal cells). It is also responsible for keeping track of locations of the UEs residing in its designated tracking area.

(2) Home Subscriber Server (HSS): The HSS stores UEs' identities (e.g., IMSI and IMEI) and subscription data (e.g., QoS profile) along with the cryptographic master keys from which it generates the authentication challenges and the symmetric session keys for each subscriber.

B. Attach Procedure

When a UE wants to connect to the EPC (e.g., at the time of device reboot), the UE starts off by scanning for the system_info_block messages broadcasted by the surrounding eNodeBs. *The UE then establishes a connection (see Figure 2) with the eNodeB whose signal power it perceives to be the highest.* As we illustrate later, this step (i.e., connecting to the highest powered eNodeB) can be exploited to set up a malicious eNodeB, prevalently in the context of IMSI catchers [14], [16]. Once the UE has established a connection with the eNodeB, the attach procedure can proceed according to the following four stages.

Identification: The UE starts the attach procedure by sending the attach_request message to the MME through the eNodeB (see Figure 2). The UE includes its identity, i.e., the IMSI/IMEI and its security capabilities (e.g., supported cipher suites) in this attach_request message in plaintext.

Authentication: For verifying the authenticity of the UE, the MME, upon reception of an authentication challenge generated by the HSS, sends an authentication_request message including this challenge to the UE. The UE using its master key solves the challenge and sends an authentication_response message to the UE. If the authentication is successful, the UE and the MME enter next stage, that is, security algorithm negotiation.

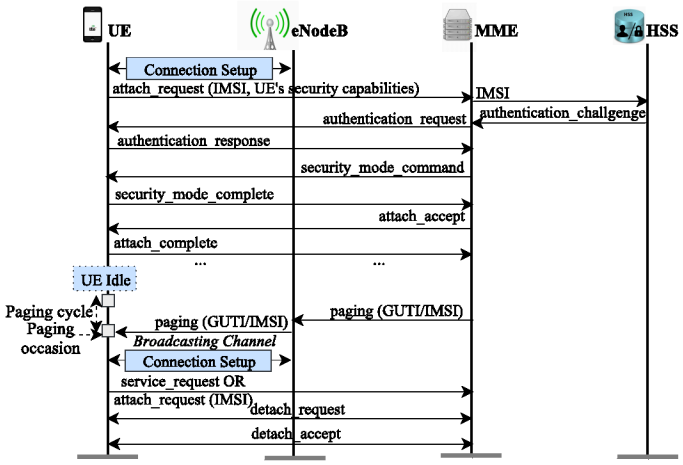

Figure 2: Attach, paging, and detach procedures of 4G LTE.

Security algorithm negotiation: The MME chooses one of the algorithm pairs (i.e., encryption and integrity) that the UE supports, obtained from the security capabilities sent with the attach_request. The MME then sends the integrity protected security_mode_command message to the UE in which the MME replays with the UE's security capabilities so that the UE verifies whether the security capabilities in the security_mode_command message are same as the ones sent by the UE in the attach_request message. After a successful verification of the message authentication code (MAC) included in the security_mode_command, the MME then sends an encrypted and integrity protected security_mode_complete message. The UE and the MME then create a shared security context, i.e., the shared keys for protecting confidentiality and integrity of the future message exchanges.

Secure temporary identifier exchange: The MME then sends an encrypted and integrity-protected attach_accept message which includes a temporary identity called GUTI (Globally Unique Temporary Identity)[1] for the UE. To limit the exposures of sensitive IMSI/IMEI, GUTI is used in all subsequent communications between the UE and the eNodeB/MME. The UE concludes the *attach* procedure by sending an attach_complete message. The UE and the eNodeB then also create a security context by generating a pair of shared keys for their secure communication.

C. Paging procedure

While a UE is actively communicating with the network, it keeps updating the MME about its location changes via tracking_area_update_request messages (the tracking area update procedure is very similar to the attach procedure). However, when an attached UE has no data to send, it goes to

[1]GUTI = MME identifier + TMSI (Temporary Mobile Subscriber Identity). The MME/eNodeB uses few bytes of GUTI as TMSI in *paging* procedure to represent the temporary identifier. Therefore, for the sake smooth discussion, we use the terms GUTI and TMSI interchangeably for the rest of the paper.

the low energy/idle mode and wakes up periodically (called, *paging occasion*) once every discontinuous reception (DRX) cycle (called, *paging cycle*) to check for paging messages.

MME-initiated Paging: When the MME needs to locate an idle UE in a particular tracking area or requires to deliver a network service such as incoming calls or SMS, the MME instructs all eNodeBs in its tracking area to generate paging messages for that particular UE. The eNodeBs present in the paged tracking area then broadcast the paging message to all the UEs. The paging messages contain the identities of the UEs – GUTI(s) or the IMSI(s). The MME usually sends paging messages with the GUTI for which it receives service_request message from the UE (see Figure 2). However, if the MME fails to allocate a new GUTI to the UE due to network failure during the *attach* procedure, the MME sends paging messages with the IMSI. Upon reception of the paging messages with the IMSI, the UE tears down all the existing radio bearers, moves to the detached state, and re-initiates the *attach* procedure.

eNodeB-initiated Paging: The eNodeB can generate paging messages without the MME involvement when it needs to notify a UE about one of the following: (i) A change in the system; (ii) Emergency (e.g., amber alert); (iii) Earthquake/tsunami warning.

D. Detach Procedure

The UE/MME can choose to terminate the established connection by generating a detach_request including the cause of detach. In response to the detach_request, the UE/MME is *expected* to send a detach_accept message.

III. Design Overview of LTEInspector

In this section, we first present our threat model, and then describe the major components of LTEInspector's architecture (see Figure 3). Finally, we explain LTEInspector's adversarial-testing approach with a concrete example.

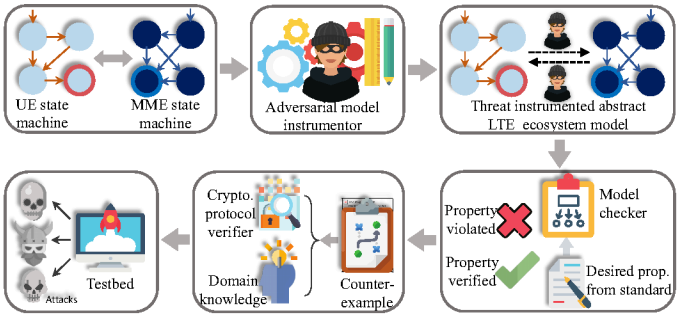

Figure 3: Architecture of LTEInspector

Adversary model. For our analysis, we consider a Dolev-Yao-style network adversary Adv^{+c} [19] with the following capabilities: **(A-1)** It can *drop* or *modify* any messages in the public communication channel. **(A-2)** It can impersonate a legitimate protocol participant and can *inject* messages in the public communication channel on the victim's behalf. **(A-3)** It adheres to all cryptographic assumptions. For instance, Adv^{+c} can decrypt an encrypted message *only if* it possesses the decryption key. Cryptographic constructs are considered to be perfectly secure in this model.

Our choice of Adv^{+c} is motivated by the following three aspects: (1) Adv^{+c} is very powerful and any protocol that is secure against it, is *likely* to be secure in weaker threat models; (2) Automatic tools can analyze protocols in this model (e.g., ProVerif [24], Tamarin [25]); (3) It is often possible to realize (a majority of the) Adv^{+c} capabilities in our context. We do not adopt the computational model [30], [31] as proving properties in this model often requires manual intervention.

A. High-Level Approach

LTEInspector in an on-demand basis combines the reasoning power of a general purpose model checker (**MC**) and a cryptographic protocol verifier (**CPV**). We first construct a protocol model in the propositional logic level and use a **MC** [22], [23] to check for violations of the abstracted input property. Along with the propositional level abstraction, we also abstract away cryptographic constructs from the model and the input property. During checking compliance of the model with respect to the property with a **MC**, as part of abstracting the cryptographic constructs, we carry out our analysis with respect to an adversary Adv^{-c} which is the cryptography-agnostic version of the Adv^{+c}. Any counterexample generated by the **MC** hence may not be feasible due to such an abstraction. To rule out such infeasible counterexamples, we check the feasibility of the counterexample with a cryptographic protocol verifier (**CPV**) [24], [25] which although operates on the first-order logic level can only verify certain types of queries. In the case **CPV** cannot find an attack, we refine the input property to rule out such spurious counterexamples.

One natural question readers may have is that why do not we use a **CPV** [24], [25] to begin with. This is because the level of abstraction and the scope with which we model the protocol enables us to efficiently reason about rich *temporal trace properties* (e.g., safety and liveness [21]). **CPV**s, even though can support unbounded parallel sessions, cryptographically sophisticated adversaries, and rich constructs, for the sake of tractability often limit their analyses to specific syntactic forms of safety properties (e.g., *correspondence* [32], *secrecy* [33]) which may not be sufficient for capturing all the desired properties we have observed. In the same vein, for tractability reasons, **CPV**s do not allow us to model the rich constructs (e.g., constraints on linear integer arithmetic) on which a **MC** can reason very efficiently, for instance, authentication desynchronization vulnerability described in our running example. Finally, even infeasible counterexamples may give us insights about other possible attacks.

B. LTEInspector Components

We now describe the major components of LTEInspector.

Abstract LTE model. We model the LTE protocol from the point of view of two participants: a UE and an MME. Although there are other protocol participants (e.g., eNodeBs, HSS), for ease of modeling, we combine their functionality inside the MME as their identity distinction does not impact the analysis of the attach, detach, and paging procedures. Also, we only consider the NAS layer messages between the two entities[2].

We abstractly model only the portion of the 4G LTE protocol that is relevant to the analysis of attach, detach, and paging

[2]Although paging messages are Radio Resource Control (RRC) protocol layer messages [3], [4], as we model the core network and the base-station as a single entity, without loss of generality, we simplify the modeling by considering paging messages as NAS layer messages in our abstract model.

procedures—without fine-grained implementation details—as two synchronous communicating finite state machines (FSM) (denoted by $\mathcal{M}_{\text{vanilla}}$). The two FSMs in $\mathcal{M}_{\text{vanilla}}$ (one for the UE and another for the MME) communicate with each other by sending messages through public communication channel. We model the communication channel between the two FSMs $\mathcal{M}_{\text{UE}}$ and $\mathcal{M}_{\text{MME}}$ with two uni-directional channels; one from $\mathcal{M}_{\text{UE}}$ to $\mathcal{M}_{\text{MME}}$ and another from $\mathcal{M}_{\text{MME}}$ to $\mathcal{M}_{\text{UE}}$. The choice of two unidirectional channels instead of a single bidirectional channel is not only for modeling convenience but also for effortlessly modeling weaker adversaries than $\mathsf{Adv}^{+\mathsf{c}}$ during adversary model instrumentation (e.g., only one direction of the public channel to be adversary controlled).

To keep the analysis tractable, in $\mathcal{M}_{\text{vanilla}}$, we do not model message data with arbitrarily large domains. For instance, the attach_request message can possibly contain IMSI as a data; in our model, we do not capture the IMSI and just model attach_request as a possible message type. We, however, model message data-dependent conditions as environment-controlled (or, in short, *environmental*) Boolean variables. For instance, for each message that can have integrity protection, we capture its integrity verification with a *unique* Boolean variable mac_failure whose value is nondeterministically set by the environment during model checking. $\mathcal{M}_{\text{vanilla}}$ can capture an unbounded number of sequential sessions. It, however, can neither capture an unbounded number of parallel sessions nor an unbounded number of protocol participants; the latter is shown to be undecidable [34].

Adversarial model instrumentor. The adversarial model instrumentor takes $\mathcal{M}_{\text{vanilla}}$ and instruments it to incorporate the presence of an adversary to obtain a new model $\mathcal{M}_{\text{adv}}$. We model a cryptography-agnostic adversary $\mathsf{Adv}^{-\mathsf{c}}$ which possesses the same capabilities of $\mathsf{Adv}^{+\mathsf{c}}$ except for its cryptographic proficiency (i.e., **A-3**). We model the $\mathsf{Adv}^{-\mathsf{c}}$ capabilities for each unidirectional public channel **ch** in the following manner. Capability **A-1** is modeled as **ch**'s property, that is, **ch** *nondeterministically* drops any message **msg** passing through it (represented as a no_operation) or replaces it with another plausible message including the current message **msg**.

Modeling capability **A-2** for **ch** requires considering an adversarial FSM which nondeterministically injects one of the possible messages including no_operation. We call such adversary FSMs the *injection adversaries*. In the case both the legitimate protocol participant and the adversary simultaneously push messages $\mathbf{msg}_{\mathsf{v}}$ and $\mathbf{msg}_{\mathsf{adv}}$, respectively, into **ch**, the message received on the other side is decided by the value of an environmental Boolean variable adv_turn; the value of adv_turn is nondeterministically chosen by the environment. Precisely, the other side of **ch** will receive $\mathbf{msg}_{\mathsf{adv}}$ only if the value of adv_turn is set to be true by the environment. *The nondeterministic behavior of the channels and the injection adversaries are crucial for reasoning about all possible adversary strategies*. Our instrumentation makes it effortless to customize the capabilities of the adversary, for instance, independently making each **ch** to have adversarial interference.

General-purpose Model Checker (MC). **MC** takes as input $\mathcal{M}_{\text{adv}}$ and a desired abstract property φ, and checks to see whether all possible executions of $\mathcal{M}_{\text{adv}}$ (considering all possible values of the environmental variables) satisfy φ. In the case **MC** finds an execution π of $\mathcal{M}_{\text{adv}}$ which violates φ, **MC** outputs π as an evidence of the violation (also, known as the *counterexample*). π includes the adversary actions which were used to violate φ, and alternatively, can be viewed as the attack strategy. The $\mathsf{Adv}^{-\mathsf{c}}$ used when model checking $\mathcal{M}_{\text{adv}}$ does not have the necessary cryptographic proficiency of $\mathsf{Adv}^{+\mathsf{c}}$ (i.e., **A-3**), and hence π can violate cryptographic assumptions, making it unrealizable in practice. We rule out such spurious πs through the following process.

Validating counterexamples with CPV. For a given counterexample π, we check each sub-step of π that requires manipulating some crytographically-protected message type. We model each small sub-step in a **CPV**, denoted as $\mathsf{M}_{\mathsf{crypto}}$. We then pose a query to **CPV** that will be violated in $\mathsf{M}_{\mathsf{crypto}}$ only if the $\mathsf{Adv}^{+\mathsf{c}}$ has the specific capability that π requires. For few message types, such as, paging, we know from the 3GPP standard that there are no confidentiality and integrity protections; for those message types we do not invoke the **CPV**.

Testbed experimentation. Once both **MC** and **CPV** adjudicate a given π to be feasible, we try to realize this attack in a testbed. This is essential because π may not be realizable in practice due to possible technical safeguards. Any π validated in the testbed experiment can thus be considered a vulnerability. We built a testbed using low-cost software defined radios and open-source LTE software stack having a price tag of around $\$3,900$ which we would argue is within the reach of a motivated adversary.

C. Example Demonstrating LTEInspector's Effectiveness

We now show the effectiveness of LTEInspector's vulnerability detection through a concrete example. For ease of exposition, we rely on a simplified and partial model of the LTE ecosystem shown in Figure 4 for this example.

In the example, the UE FSM (the top FSM) has 3 states and 9 transitions whereas the MME FSM (the bottom FSM) has 3 states and 6 transitions. Transition labels are of the form "condition/actions" in which condition is a propositional logic formula specifying the condition under which the transition will be triggered whereas the actions component refers to a sequence of actions that will be performed (in their appearance order) by the FSM after the transition is taken. Although the actions component can be empty (denoted with –), the condition component cannot be empty. We represent the states with barebone arrows (i.e., arrows with no condition and action) as the initial states of the FSMs. We use ❶, ❷,... to denote the UE transitions whereas we use ①, ②,..., to denote the MME transitions. The FSMs have the following environmental variables: mobile_restart (signifying UE rebooting); mac_failure (improper MAC for auth_request message); xres_matches_sres (correct authentication_response message for a given authentication_request message). Both the FSMs start with their respective sequence numbers to be 0.

The response property we want to verify is the following: "*It is always the case that whenever the UE FSM is in the* wait for auth_request *state, it will eventually move to the state where the UE authenticates the MME*" (denoted by φ_1). The property is desirable as its violation signifies a denial-of-service attack in which the UE cannot proceed to the next stage of the attach procedure after initiating it.

ID	MC property details
φ_1	It is always the case that whenever the UE FSM is in the wait for auth_request state, it will eventually move to the state where the UE authenticates the MME.
φ_2	Refinement over φ_1: Once UE FSM moves to the wait for &auth_request state, the environment will never set the value of mobile_restart to be true.
φ_3	Refinement over φ_2: mac_failure is never set to true by the environment.
φ_4	Refinement over φ_3: UE FSM never receives the detach_req message.
φ_5	Refinement over φ_4: UE FSM never receives the auth_reject message.

Table I: **MC** properties used in the motivating example.

ID	CPV property details
Ψ_1	Every attach_request message received by the MME should be preceded by a unique attach_request message sent by the UE.

Table II: CPV Properties used in the motivating example.

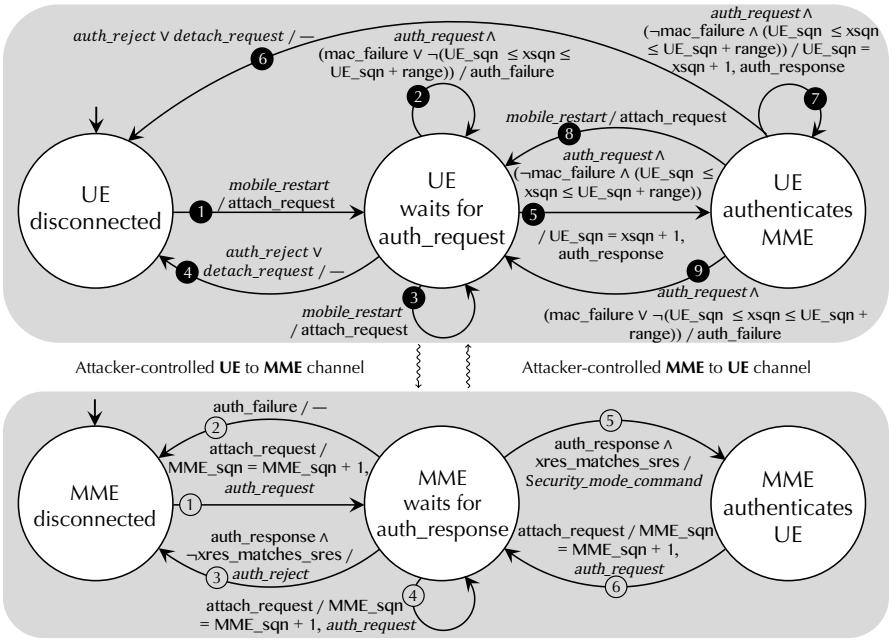

Figure 4: A simplified LTE ecosystem model.

When φ_1 is checked against the given $\mathcal{M}_{\text{adv}}$, it gives the trivial counterexample π in which after the UE FSM moves to the wait for auth_request state, it continuously observes the value of the environmental variable mobile_restart to be true (triggering transition ❸)—signifying the repeated restart of the UE—which even though plausible, is not interesting as the Adv^{+c} has no control over UE reboot. One possible way of removing this π is to refine φ_1 to add the restriction that once UE FSM moves to the wait for auth_request state, the environment will never set the value of mobile_restart to be true. When this refined property (denoted by φ_2—see Table I) is checked, the **MC** yields a π in which all the auth_request messages received by the UE fails the MAC verification (triggering transition ❷) because the **MC** assigns the value of the mac_failure to be continuously true. We then refine φ_2 further to ensure that mac_failure is never set to true by the **MC** and obtain the property φ_3.

Attack 1: Checking $\mathcal{M}_{\text{adv}}$ against φ_3 using **MC** yields another π in which after UE FSM moves to the wait for auth_request state, it receives a network initiated detach request (detach_req)–injected by the adversary—triggering transition ❹ which moves it to the disconnected state and due to avoiding reboot after transitioning to wait for auth_request state (in φ_2), stops the UE FSM to get out of the disconnected state. This is a legitimate attack and we would like to know whether it is possible for the attacker to forge a network initiated detach_request. A close inspection of the standard reveals that once the security context has been established between the UE and the MME, the network initiated detach_request should be integrity protected only. Our experiment with the UE, however, revealed that the UE does not actually check the validity of the MAC for detach_req even in the case the security context has been established. This means such an attack is plausible and we have verified it in our testbed. We then refine φ_3 further to exclude this π and ensure that the UE FSM never receives the network initiated detach request; as a result, we obtain the refined property φ_4.

Attack 2: We then check $\mathcal{M}_{\text{adv}}$ against φ_4, and it yields another similar π in which the UE FSM receives an auth_reject message (triggering transition ❹); this moves the FSM to the disconnected state. Again, one needs to determine whether the adversary can fabricate an auth_reject message. A close inspection of the standard revealed that the auth_reject message is never integrity protected and our experimentation validated it. In a similar way, we refine φ_4 to exclude any auth_reject message and obtain our final property φ_5.

Attack 3: Finally, checking $\mathcal{M}_{\text{adv}}$ against φ_5 with **MC** results in a very interesting π in which the adversary sends the **range** (a UE-specific constant non-negative integer) number of fake attach_request messages to the MME, before the UE reboots and sends the attach request. After receiving each attach_request message, the MME increases its own sequence number (according to transitions ① and ④) and uses the value of the sequence number to provide replay protection to the auth_request message which it sends to the UE. As the UE is still in the disconnected state (with its sequence number 0, i.e., $UE\_sqn = 0$) and there is no transition that is triggered by the auth_request when the UE is in the disconnected state (see Figure 4), these auth_request messages are ignored. Then the UE observes a mobile_restart (triggering transition ❶) and sends an attach request to the MME which the adversary allows to reach the MME. After the MME receives it (transitions ①, ④, and ⑥), the MME as usual responds with an auth_request with the sequence number $\textbf{range} + 1$. The adversary also allows the auth_request from the MME to reach the UE. Upon receiving the message, the UE checks for mac_failure (which cannot be true as we excluded it while refining φ_2 to obtain φ_3) and checks whether the received sequence number from MME (denoted with $xsqn$) satisfies the following: $UE\_sqn \leq xsqn \leq UE\_sqn + \textbf{range}$; this condition will fail as $xsqn = \textbf{range} + 1$ resulting in the UE not being able to attach with the MME. To ensure that validity of the π, one has to verify whether the Adv^{-c} can inject a fake attach_request message; we use the **CPV** to

validate it. We pose an injective-correspondence [32] query, Ψ_1 (shown in Table II) which asserts that every attach_request message received by the MME should be preceded by a unique attach_request message sent by the UE. The **CPV** produced an attack in which the adversary injected an attach_request; validating the desired sub-step of π. We have also observed the feasibility of this attack in our testbed. We call this attack **the authentication synchronization failure attack**; this attack is also applicable against a recent proposal for defeating IMSI catchers [16]. One of the challenges of detecting such an attack through **CPV**s is to precisely capture the sanity check corresponding to the sequence number. It is not immediately clear to us how one would directly capture such a requirement in the **CPV** in a fine-grained fashion. The **MC**, however, can efficiently reason about such requirement precisely. *This shows the effectiveness of combining a* **MC** *with a* **CPV** *for attack discovery.*

This example demonstrates the analysis power of LTEInspector's approach, that is, based on the violation of a single desired property (and, its refinements) LTEInspector was able to find three different attacks which have been shown to be realizable in practice. As we will demonstrate later, Attacks 1 or 2 can be stitched with a relay attack to yield the authentication relay attack.

D. Implementation

We now discuss some additional details of LTEInspector.

MC and CPV: We instantiate LTEInspector's **MC** component with NuSMV [22] and use ProVerif [24] for **CPV**.

Model. We manually construct the abstract LTE model by consulting the 3GPP standard [3], [4]. Our model has a total of 13 states and 107 transitions. Our current model and the respective properties are publicly available at https://github.com/relentless-warrior/LTEInspector.

Properties. Our properties were extracted from the 3GPP standard [3], [4]. We have tested $\mathcal{M}_{\text{adv}}$ against 14 properties in total in which 7 properties were analyzed with NuSMV whereas 7 properties were analyzed with ProVerif. Note that, we do not claim our list of properties to be exhaustive.

IV. LTEInspector Findings

In this section, we highlight the findings of LTEInspector. For readers' convenience, we have provided a summary of the attacks and their implications in Table III.

A. Attacks Against Attach Procedure

We now present our findings on the attach procedure.

A-1 *Authentication Synchronization Failure Attack:* This attack exploits the UE's sequence number sanity check to disrupt its attach procedure. Precisely, the adversary interacts with the HSS through the MME to ensure that the sequence numbers of the UE and the HSS are out-of-sync. As a result, the authentication challenge received through the legitimate auth_request message fails the UE's sanity check and consequently is discarded by the UE.

Adversary assumptions. For successfully carrying out this attack, the adversary is required to set up a malicious UE and also is required to the know the victim UE's IMSI. Such a threat is very practical and has been validated through experimentation in our testbed (see Section V-A1).

Detection. We exposed this attack by first model checking $\mathcal{M}_{\text{adv}}$ with respect to a refinement φ_5 of the following property: "*It is always the case that whenever the UE FSM is in the wait for* auth_request *state, it will eventually move to the state where the UE authenticates the MME*" (see Table I for details). We observed a violation of φ_5 in $\mathcal{M}_{\text{adv}}$ where the $\text{Adv}^{-\text{c}}$ fabricates attach_request messages and sends them to the MME. To validate the $\text{Adv}^{-\text{c}}$'s capability of forging an attach_request message, we leverage ProVerif which showed that forging attach_request messages are possible; validating the feasibility of the attack.

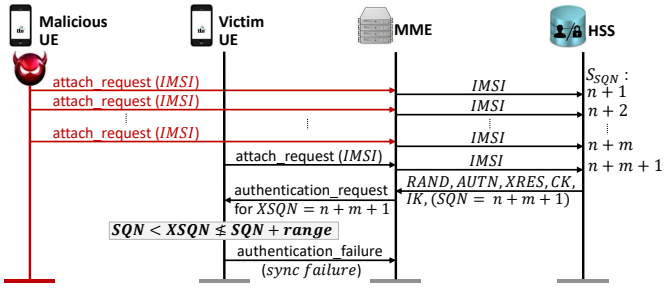

Figure 5: Authentication synchronization failure attack.

Attack description: The steps of this attack are shown in Figure 5. It is very similar to the description of Attack 3 in Section III-C with one caveat. It is just not sufficient to send the same attach_request message m times (where $m >$ **range**). The malicious UE needs to send different security capabilities (by selecting different encryption and integrity protection algorithms) in successive attach_request messages. This is crucial as the HSS only processes an attach_request message only if one or more of the information elements in the current attach_request message differs from the already received one. In which case, in accordance to subclause 5.5.1.2 of 3GPP standard [35], the previously initiated attach procedure is aborted and the new attach_request message is processed (including, the increment of the sequence number).

Re-synchronization: When the sequence number sanity check fails on the UE side, it sends an auth_failure message (cause: sync. failure) to the EPC with $AUTS$ parameter containing the UE's current sequence number resulting in the EPC to re-synchronize its sequence number. Even after re-synchronization, the adversary can continue repeating step 1 to make the UE and the HSS sequence numbers to go out-of-sync again; preventing the UE from connecting to the EPC.

Implication: The major implication of this attack is the service disruption suffered by the victim UE.

A-2 *Traceability Attack:* This attack exploits the responses of security_mode_command messages to track a particular victim UE. Typically during the attach procedure, the MME uses the security_mode_command message to choose one of the UE-supported cipher suites to use for communication. When the UE receives this message, it is expected to respond with a security_mode_complete message when the received message satisfies the MAC validation. In case of MAC failure, the UE responds with a security_mode_reject message. The MME can also send a security_mode_command

message to an already attached UE for changing the current cipher suite. This message is recommended to be integrity- and replay-protected [35].

Adversary assumptions. We assume that the adversary has already obtained an authentic security_mode_command message sent to the victim UE in the previous attach procedure. We also assume that the adversary can set up a malicious eNodeB.

Detection. We model check the $\mathcal{M}_{\text{adv}}$ against the following property: *the UE responds with a* security_mode_complete *message only if the MME sent a* security_mode_command *message which passes the sanity checks.* This is trivially violated by a counterexample in which the adversary injects a security_mode_command message. As the security_mode_command message is cryptographically protected this seems to be a spurious counterexample. Coincidentally, we observed that none of the four major US carriers make the security_mode_command message replay-protected (they do not include the recommended fresh nonce). Then, we used ProVerif's capability of reasoning about observational equivalence to pose the query: *is it possible for the adversary to distinguish two UEs based on their responses to a* security_mode_command *message? ProVerif provided an attack strategy for distinguishing two UEs.*

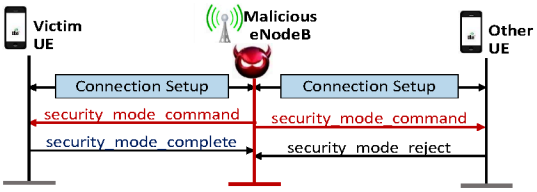

Figure 6: Traceability attack using security_mode_command.

Attack description: The UEs in a particular eNodeB cell get connected with the malicious eNodeB. The malicious eNodeB then replays the captured security_mode_command message to all the UEs as shown in Figure 6. The victim UE verifies the integrity and the UE's security capabilities of the received message, and responds with the security_mode_complete message to the malicious eNodeB whereas the rest of the UEs in that cell respond with security_mode_reject messages due to integrity check failure. The malicious eNodeB thus identifies the presence of a particular UE in an area.

Implications: This attack can enable an adversary to track a particular victim UE.

A-3 *Numb Attack:* In this attack, the adversary injects an out-of-sequence control-plane protocol message to severely disrupt the service of a victim UE.

Adversary assumption. For successfully carrying out this attack, the adversary is required to set up a malicious eNodeB.

Detection. We observed this attack after model checking $\mathcal{M}_{\text{adv}}$ with respect to a refinement φ_3 of the property φ_1 in Table I. The counterexample produced by the model checker shows that whenever the UE receives an auth_reject message injected by the Adv^{-c}, the UE FSM moves to the disconnected state. To realize this attack the Adv^{-c} has to be able to inject auth_reject message which is feasible as the message is not cryptographically protected according to the standard.

Attack description: As soon as the victim UE connects with the malicious eNodeB, the malicious eNodeB sends an auth_reject message to the victim UE irrespective of the context of the victim UE.

Implications: Along with the most straightforward implication of severe service disruption, this finding may be chained with another attack (possibly, some form of impersonation attack) which requires the UE to be inactive or re-initiate the attach procedure. We have observed that upon reception of the auth_reject message in one of the popular cellular device, the UE first detaches itself from the network, completely shuts down all cellular activities, and does not even attempt to downgrade/connect to 3G/2G networks. In this situation, even re-insertion of SIM card does not allow the victim UE to connect to the EPC again. The victim UE remains in such a numb state until the user restarts her UE.

B. Attacks Against Paging Procedure

In this section, we present attacks that we have exposed against the paging procedure.

Adversary assumptions. For the following attacks on the paging procedure, the adversary needs to setup a malicious eNodeB and also needs to know the victim UE's IMSI. For the linkability attack, we assume the adversary knows the previous **pseudo-IMSIs** (or, in short, PMSIs[3]) [16].

Detection. We obtain the intuition for the *paging channel hijacking* attack (**P-1**) after observing our model $\mathcal{M}_{\text{adv}}$'s violation of the following property: *the UE sends a service request only if the MME has sent a* paging *message that is pending.* The model checker produces a counterexample in which the Adv^{-c} injects a paging message. After consulting the standard [35], we observed that paging messages do not have any cryptographic protection. This signifies that an adversary can inject paging messages. *The rest of the attacks in this section are direct consequences of the paging channel hijacking attack.*

P-1 *Paging Channel Hijacking:* For hijacking the paging channel, the malicious eNodeB operates in the same frequency band as the legitimate eNodeBs so that the victim UE does not perceive any network changes. The malicious eNodeB then broadcasts fake **empty** paging messages in the shared paging channel. However, a UE does not listen to the paging channel continuously for incoming paging messages. It usually remains in the *sleep* state and wakes up in its paging cycle for pending paging message. Therefore, it is crucial for the adversary to make its eNodeB's paging cycle same as the victim UE's. Detailed synchronization procedure is presented in Section V.

Although both malicious and legitimate eNodeBs broadcast the paging messages at the same time intervals, the UE only responds to the first received message. To address this challenge, the malicious eNodeB broadcasts paging messages with higher signal power. Thereby, the adversary hijacks the victim UE's paging channel and makes the victim UE unable to receive legitimate paging messages from the MME. This means that the victim does not receive any service (e.g., incoming phone calls/SMS) notifications which would result in

[3]To protect against IMSI catching attacks, Broek et al. [16] proposed an enhancement over the 3GPP standard where the IMSI is replaced with a changing pseudonym, called Pseudo-IMSI or PMSI, that only the SIMs HSS can link to the SIMs identity. Therefore, instead of sending the IMSI, the UE uses different PMSIs in different attach_request messages. To the best of our knowledge, support for PMSIs have not been implemented in 4G LTE.

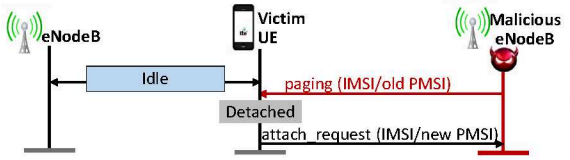

Figure 7: Detach and linkability attacks using paging.

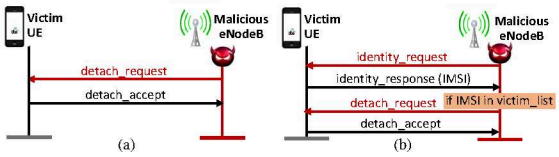

Figure 8: (a) Indiscriminately- (b) targeted- detach/downgrade a UE using the network initiated detach_request message.

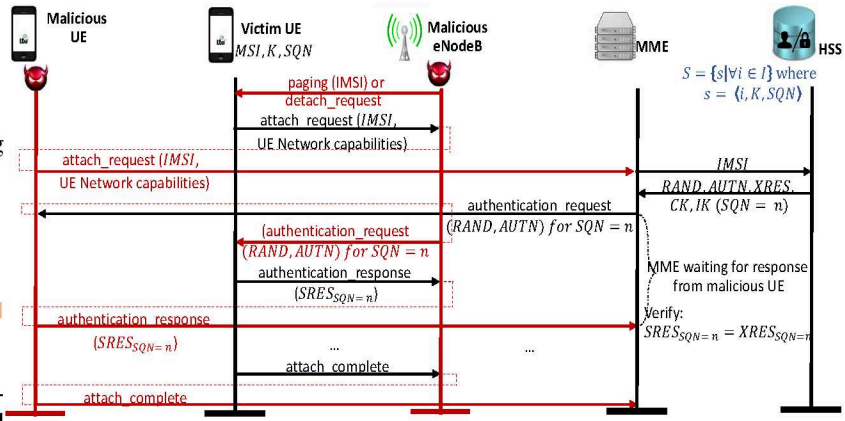

Figure 9: Authentication relay attack

customer dissatisfaction, revenue loss and reputation damage of the network operators. *One of the interesting consequences of this attack is that the victim UE is completely unaware of paging channel hijacking which lets the adversary silently drop incoming services.*

P-2 *Stealthy kicking-off Attack:* The steps of this attack are shown in Figure 7 and described as follows:

After hijacking the victim UE's paging channel, the malicious eNodeB creates a paging message with one paging record consisting of the victim UE's IMSI. The adversary sets other fields of the paging record similar to an original paging message. Upon reception of the paging message with IMSI, the victim UE finds its IMSI in the first paging record. As a result, it first disconnects from the EPC and then sends an attach_request message. *This attack can also be used as a prerequisite of the authentication relay attack.* The major implication of this attack is service disruption.

P-3 *Panic Attack:* In this attack, the adversary wants to inject fake emergency paging messages to a large number of UEs. The adversary thus sends a paging message with empty records but with fake emergency warnings. To ensure that such a fake paging message reaches a large number of UEs, the adversary keeps broadcasting this message for all possible paging occasions of the legitimate eNodeB. *This can create artificial emergency which can be exploited by malicious parties for hiding their agenda.*

P-4 *Energy Depletion Attack:* The idea of this attack is to make the victim UE perform expensive cryptographic operations. One way to achieve this is to force the UE to keep carrying out the expensive attach procedure over and over again, by sending a paging message with IMSI between two successive attach procedures. In case the adversary knows the GUTI of the victim [8], it can send a paging message with GUTI which the UE responds with a cryptographically-involved service_request message.

P-5 *Linkability Attack:* This attack (see Figure 7) focuses on breaking the unlinkability guarantees (i.e., attacker cannot link any two successive pseudo-IMSIs/PMSIs) provided by Broek et al. [16]. From the assumption that the adversary knows the old PMSI which it uses to issue a paging message which the victim responds with an attach_request with the new PMSI enabling us to link the two PMSIs. *Note that, the above attack is not applicable to 4G LTE because the mechanism of Broek et al. [16] is not adopted for 4G LTE.* In case of 4G LTE, however, the adversary may use the same philosophy for tracing a victim UE in a cell area. After broadcasting a paging with the victim UE's IMSI, if the adversary observes an attach_request with the same IMSI, the adversary can confirm the victim UE's presence.

C. Attacks Against Detach Procedure

We now describe an attack on the detach procedure that LTEInspector has exposed.

D-1 *Detach/Downgrade Attack:* In this attack, the adversary injects the network initiated detach_request to disrupt the service of a victim UE irrespective of the UE context.

Adversary assumptions. The adversary needs to setup a malicious eNodeB and also needs to know the IMSI of the victim.

Detection. Attack detection is similar to the numb attack.

Attack description. The steps of this attack are shown in Figure 8(a). Whenever the victim UE connects to the malicious eNodeB, it sends a network initiated detach_request message which force the UE to move to the disconnected state and to send detach_accept message.

Targeted variant of detach/downgrade attack. This attack (see Figure 8(b)) can be adopted to a more targeted setting in which the adversary targets specific UEs. For the targeted variation, before the detach_request is sent, the eNodeB will send an identity_request message which the UE will respond with an identity_response message containing the UE's IMSI. If the IMSI is in the attacker's victim list, it will send the detach_request, otherwise, it will ignore that UE.

Implications: Along with its direct consequence of severe service disruption, this finding can be stitched with another attack (possibly, some form of impersonation attack) which requires the UE to be inactive and re-initiate the attach process.

ID	Attack name	Affected procedure	Adversary assumptions	Assumption validation	Standard/ Stakeholder slip-up	New attack?	Detection process	Notable implications	Validated	Setup cost
A-1	Authentication sync. failure	Attach	Known IMSI, malicious UE	IMSI [13], Section V-A1	3GPP	Yes	LTEInspector	Denial-of-attach or Denial-of-services.	●	$2600 (2 USRPs)
A-2	Traceability	Attach	Valid security_mode command, malicious eNodeB	Section V-A1	Operational networks, mobile devices	Inspired by [10]	LTEInspector	Coarse-grained location information leakage.	●	$2600 (2 USRPs)
A-3	Numb using auth_reject	Attach	Malicious eNodeB	Section V-A1	3GPP	Yes	LTEInspector	Denial of all cellular services.	●	$1300 (1 USRP)
A-4	Authentication relay	Attach	Known IMSI, malicious eNodeB	IMSI [13]; Section V-A1	3GPP, operational networks	Yes	LTEInspector, attack chaining	Reading incoming/outgoing messages of victim, stealthy denial of all/selective services, location history poisoning.	●	$3900 (3 USRPs)
P-1	Paging channel hijacking	Paging	Known IMSI, malicious eNodeB	IMSI [13], Section V-A1	3GPP	Yes	Intuition from LTEInspector, domain Knowledge	Stealthy denial of incoming services.	●	$1300 (1 USRP)
P-2	Stealthy kicking-off	Paging	Known IMSI, malicious eNodeB	IMSI [13], Section V-A1	3GPP	Yes	Consequence of **P-1**, domain knowledge	Detaching a victim from the network surreptitiously.	●	$1300 (1 USRP)
P-3	Panic	Paging	Malicious eNodeB	Section V-A1	3GPP	Yes	Consequence of **P-1**, domain knowledge	Life threatening impact against mass people, e.g., artificial chaos for terrorist activity.	○	$1300 (1 USRP)
P-4	Energy depletion	Paging	Known IMSI, GUTI, malicious eNodeB	IMSI [13]; GUTI [8], Section V-A1	3GPP	Inspired by [36], [37]	Consequence of **P-1**, domain knowledge	Battery depletion.	◓	$1300 (1 USRP)
P-5	Linkability	Paging	Known IMSI or old pseudo-IMSI	Section V-A1	3GPP, enhanced AKA [16]	Yes	ProVerif only	Coarse-grained location information leakage	○	$1300 (1 USRP)
D-1	Detach/ Downgrade	Detach	Malicious eNodeB, known IMSI (for targeted version)	IMSI [13]; Section V-A1	3GPP	Inspired by [13]	LTEInspector, attack chaining (for targeted version)	Denial of services/Downgrade to 2G/3G.	●	$1300 (1 USRP)

Table III: Summary of our findings (●=validated, ◓=partially validated, ○=not validated) of the reported attacks.

D. Attack Chaining

We now demonstrate the application of the attack chaining technique through the authentication relay attack.

A-4 *Authentication Relay Attack:* In this attack, one of our exposed attacks (e.g., paging with IMSI) is stitched with a relay attack with which an adversary impersonates the victim UE to connect to the EPC without possessing proper credentials, and in the process, spoof the victim UE's location in the core networks.

Adversary assumptions. For this attack, the adversary is required to setup a malicious eNodeB ($\mathsf{eNodeB_{adv}}$) and a malicious UE ($\mathsf{UE_{adv}}$), and also needs to know the IMSI of the victim UE. We assume there is a private channel between the $\mathsf{eNodeB_{adv}}$ and the $\mathsf{UE_{adv}}$.

Attack description. In this attack, the adversary impersonates an already attached victim UE ($\mathsf{UE_{vic}}$) to connect to the EPC by collaborating with the $\mathsf{eNodeB_{adv}}$ and the $\mathsf{UE_{adv}}$. Suppose the $\mathsf{UE_{vic}}$ is already attached with the legitimate eNodeB denoted by $\mathsf{eNodeB_{benign}}$. The attack can be broken down into two main goals: (i) Force $\mathsf{UE_{vic}}$ to disconnect from the EPC; (ii) $\mathsf{UE_{adv}}$ pretends to be $\mathsf{UE_{vic}}$ to connect to the EPC.

Disconnecting $\mathsf{UE_{vic}}$ *from the EPC*: For disconnecting the $\mathsf{UE_{vic}}$ from the EPC, we use our paging with IMSI attack. This can also be achieved with our network initiated detach_request or auth_reject attacks.

$\mathsf{UE_{adv}}$ *connecting to the EPC by impersonating as* $\mathsf{UE_{vic}}$: As the $\mathsf{UE_{vic}}$ detached itself from the EPC due to a paging with IMSI message, it will try attach to the eNodeB with the highest signal strength; which is the $\mathsf{eNodeB_{adv}}$. The $\mathsf{UE_{vic}}$ will send an attach_request message m_{req} to the $\mathsf{eNodeB_{adv}}$ which the $\mathsf{eNodeB_{adv}}$ forwards to the $\mathsf{UE_{adv}}$. The $\mathsf{UE_{adv}}$ then sends the same attach request m_{req} to the $\mathsf{eNodeB_{benign}}$. The legitimate MME will send an authentication challenge c to the $\mathsf{UE_{adv}}$ through the $\mathsf{eNodeB_{benign}}$ upon receipt of m_{req}. The $\mathsf{UE_{adv}}$ will forward the c to the $\mathsf{eNodeB_{adv}}$ which the $\mathsf{eNodeB_{adv}}$ will send to the $\mathsf{UE_{vic}}$. After the $\mathsf{UE_{vic}}$ receives c, unaware that the $\mathsf{eNodeB_{adv}}$ sent it, it will solve the challenge c to generate the correct response r. The $\mathsf{UE_{vic}}$ will then send r to the $\mathsf{eNodeB_{adv}}$ which it will forward to the $\mathsf{UE_{adv}}$. The $\mathsf{UE_{adv}}$ then will use r to respond to the MME challenge. Using the same principle, the $\mathsf{UE_{adv}}$ will finish the rest of the steps of the attach procedure.

Discussion. Unlike a typical man-in-the-middle attack, the adversary in this attack can neither decrypt the encrypted traffic between the victim UE and the core networks, nor can inject valid encrypted traffic unless the service provider blatantly disregards the standard's security recommendations and choose a weak-/no- security context during connection establishment.

Implications: The implications of this attack include:

(1) Deception: The adversary deceives the victim into believing that the $\mathsf{UE_{vic}}$ is connected to the core network.

(2) Location History Poisoning: Since the $\mathsf{UE_{adv}}$ does not need to be in the same tracking area as the $\mathsf{UE_{vic}}$, it can authenticate itself to the EPC from a different tracking area and thus provide misleading location information about the $\mathsf{UE_{vic}}$. Thus, the $\mathsf{UE_{adv}}$ can poison the location history of the $\mathsf{UE_{vic}}$ by performing this attack successively from different tracking areas. As a result, a fugitive or criminal hiding in one location can deceive the core network into believing that the criminal has attached to the core network from a different location.

(3) Loss of confidentiality: The security_mode_command message sent by the MME during the attach procedure includes the selected cipher (EEA0-EEA7) and integrity protection (EIA0-EIA7) algorithms of the MME. By observing the security_mode_command messages of all four major network providers in the US, **we have observed that at least one carrier (OP-I) never used encryption (i.e., uses EEA0—no cipher)**. Note that, to keep the four major US network operators anonymous, we use pseudonyms (i.e., OP-I, OP-II, OP-III, OP-IV) to identify them. We have observed this insecure

#	Prior attack	Detected	How/Why?
1	Downgrade using tau_reject [13]	Yes	LTEInspector.
2	Denial of all services [13]	Yes	LTEInspector.
3	Denial of selected services [13]	No	Do not model data for attach_request.
4	Location tracking through mapping user's phone number/social network ID to GUTI [8], [13].	No	Do not model multiple instance of UEs.
5	IMSI catching [16]	Yes	LTEInspector.
6	Fine-grained location exposure [13]	No	Do not model RRC layer messages.
7	DoS exploiting race condition with paging_response [38] in 2G	Yes	LTEInspector.
8	Service hijacking exploiting race condition with paging_response [38] in 2G	Yes	LTEInspector.
9	Linkability using TMSI_reallocation_command [11] in 3G	Yes	LTEInspector.
10	Linkability of IMSI to GUTI using paging_request [10] in 3G	Yes	LTEInspector.
11	Linkability using auth_sync_failure [10] in 3G	Yes	LTEInspector.
12	Man-in-the-Middle in 2G [9]	Yes	LTEInspector.
13	Man-in-the-Middle in 3G [12]	No	Do not model data.

Table IV: Prior attacks (related to attach, detach, and paging procedures) that are detected/not detected by LTEInspector.

practice multiple times in two different geographical locations. The adversary hence can learn the UE_{vic}'s conversation, SMSs, and data through the UE_{adv} and the $eNodeB_{adv}$. We reported this to the affected carrier which has now been addressed.

(4) Complete or Selective DoS: Using this attack, the UE_{adv} and the $eNodeB_{adv}$ can relay the incoming/outgoing traffic of the UE_{vic} and the EPC. Therefore, the UE_{adv} and the $eNodeB_{adv}$ can deny the UE_{vic}'s phone-calls/SMS/data-transfers completely/selectively. Consequently, the operational network is deprived of the charges for the incoming/outgoing calls and SMSs.

(5) Profiling victim's service usage: Since all the incoming/outgoing communications of the UE_{vic} take place through the UE_{adv} and the $eNodeB_{adv}$, the adversary can profile the service usage pattern (i.e., patterns of phone calls, SMSs, data) of the victim.

E. Prior Attacks Detected by LTEInspector

In addition to the new attacks, LTEInspector is capable of detecting 9 [16], [13], [38], [10], [11], [9] out of 13 prior attacks (see Table IV) that are relevant in the context of attach, detach, and paging procedures. The previous attacks [13], [8], [12] that LTEInspector cannot detect exploit one of the following which LTEInspector currently does not support: (1) message data, (2) multiple instances of UEs or MMEs, (3) other layers' (e.g., RRC layer) messages, (4) 2G/3G procedures that are different from 4G LTE, (5) properties about sets of traces, and (6) performance related parameters (e.g., data transmission and reception rate).

V. Validation of Attacks with Testbed

In this section, we describe the verification of the new attacks (along with their adversarial assumptions) detected by LTEInspector. We have tried to exercise restraint—conforming to best practices—in validating the effectiveness of the different vulnerabilities while maintaining the validation process meaningful. To limit the impact of our attacks, we use both a custom-built LTE network and commercial networks with a logical Faraday cage [13].

A. Testbed Setup and Assumption Validation

We now describe our testbed setup for attack validation.

1) Malicious eNodeB Setup: We have used a Universal Software-defined Radio Peripheral device (i.e., USRP B210 [39]) connected to an Intel Core i7 machine running Ubuntu 14.04 as the hardware component and OpenLTE [40], an open source LTE protocol stack implementation, to set up a malicious eNodeB which costs around $1300 for the dedicated hardware (i.e., excluding the core i7 machine). We used OpenLTE's `LTE_Fdd_enodeb` application which simultaneously acts as a bare-minimal eNodeB, a mobility management entity (MME), and a home subscriber server (HSS). We have implemented support for the *detach* procedure in OpenLTE as it originally had support for the *attach* procedure only. We have also instrumented OpenLTE to inject different fabricated messages (e.g., network initiated detach_request message) when necessary. For validating attacks against the paging procedure, we use srsLTE [41] which we enhanced to support eNodeB-initiated paging messages; its original support only included MME-initiated paging messages.

eNodeB configuration. Our malicious eNodeB can impersonate the legitimate eNodeB of a network operator (i.e., OP-I to OP-IV) by broadcasting system_info_block messages with higher signal power. For successful impersonation, these messages must include parameter values that are equal to that of an operator's legitimate eNodeB. The adversary uses a UE with the operator's SIM to learn the parameters in the system_info_block messages sent by the operator's eNodeB. In our setup, we use both our custom-built sniffer and QXDM [42] to sniff the incoming and outgoing LTE messages on a consumer UE to learn the operator's parameters. Table V shows the parameters that we capture from the operator eNodeB's system_info_block messages. We use them to configure the malicious eNodeB with OpenLTE.

Parameters	Description
band	The frequency band number of the network operator.
dl_earfcn	E-UTRA absolute radio frequency channel number.
mcc	Mobile country code specific to a country.
mnc	Mobile network code specific to a network operator.
p0_nominal_pucch	Power control parameter.
p0_nominal_pusch	Power control parameter.
q_rx_lev_min	Used for cell re-selection.
q_hyst	Used for cell re-selection
DRX cycle	Paging cycle

Table V: Configuration parameters captured from Operator's system_info_block messages.

Learning IMSI/IMEI. As soon as the victim UE is forced to connect with the malicious eNodeB, the malicious eNodeB sends an identity_request (IMSI/IMEI) message to the victim UE which responds with the identity_response message including its IMSI/IMEI.

Learning GUTI. We use the well-known set intersection technique to find the GUTI as described in [8].

2) Malicious UE Setup: We use a USRP B210 [39] running srsUE [43] (open source protocol stack implementation for UE) as the malicious UE which costs around $1300.

3) Victim UEs and EPC Networks: We have used 3 different models of LTE-capable mobile phones and the 4 major network operators in the US. For the authentication relay and authentication synchronization failure attacks, the adversary requires a malicious UE to send messages to a commercial EPC. Since this is a violation of the Federal Communications Commission's (FCC) regulations [18], we use our custom-built network (as the EPC) and a USIM (Universal Software Identity Module) instead of commercial EPCs and their SIMs, respectively. Our custom-built network similar to [44], operates on an experimental licensed spectrum.

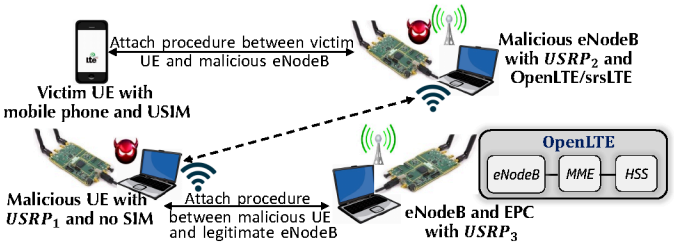

Figure 10: Experiment setup for custom-built network.

4) Sniffer setup: We have built a low-cost, real-time LTE channel decoder (costs around $1300) using USRP [39] as the hardware whereas we used Owl [45] and srsLTE [41] as the software components. Our sniffer identifies the new c-RNTI (a lower-layer UE identifier) of a UE that joins the cell, and then decodes the unencrypted downlink messages from the MME/eNodeB.

Attack setup cost. Since different attacks (e.g., **A-4** and **P-1**) require different attack setups, we list only the dedicated hardware cost for individual attack in Table III. which shows $3900 as the maximum cost required to validate the attacks.

B. Validation using Custom-built Network

We now discuss how we successfully verified the authentication synchronization failure and authentication relay attacks in our custom-built network (See Figure 10).

A-1 *Authentication Synchronization Failure Attack:* In our verification setup, the malicious UE (using the victim UE's IMSI) sent 100 attach_request messages (with different security configurations) to the legitimate MME. After the attack step, we reboot the victim UE which initiated the attach procedure by sending an attach_request message for which it received an authentication_request message from the MME. In response to that message, using our sniffer we observed that the victim UE responded with an auth_failure message (with cause sync. failure); confirming our attack.

A-4 *Authentication Relay Attack:* For this attack, we built a relay channel (See Figure 10) between the malicious UE ($USRP_1$) and the malicious eNodeB ($USRP_2$) using the Wi-Fi interfaces of the machines co-located with those USRPs. We also configured both the malicious eNodeB and the legitimate eNodeB to broadcast different tracking areas in their system_info_block messages. After performing the attack steps, we observed that both the victim and malicious UE received the same attach_accept message and also completed the attach procedure. From the logs of the legitimate EPC and using the tracking area numbers, we confirmed that only the malicious UE was connected with the legitimate MME. On the other hand, although the victim UE was actually connected with the malicious EPC, it was deceived to realize that it was connected with the legitimate EPC. Thus, the legitimate EPC was duped about the actual location of the legitimate UE; confirming the attack.

We have also successfully verified the authentication synchronization failure and relay attacks with OpenEPC [46], which is a licensed prototype implementation of the 3GPP Evolved Packet Core (EPC).

C. Validation using Commercial Mobile Phones

For all the other attacks (except **A-1** and **A-4**), we use commercial network operators' (e.g., OP-I) SIMs for the victim UE and a malicious eNodeB as discussed in Section V-A1).

A-2 *Traceability Attack:* For this attack verification, we use multiple UEs among which one of them is designated as the victim UE; the one being tracked. First, using the sniffer, we capture a legitimate security_mode_command message sent by the MME to the victim UE during a benign interaction. We decoded the captured security_mode_command message and extracted all its field values including the MAC. For tracing, the malicious eNodeB then injected fabricated security_mode_command messages to all the UEs using the extracted field values. We observed that in response to the injected messages only the victim UE responded with the security_mode_complete message whereas the rest of UEs responded with a security_mode_reject message; confirming our attack. Note that, the malicious eNodeB may not always have success with this attack when it does not receive security_mode_complete messages from any of the UEs. This can be attributed to either the victim UE's absence in that cell area or the victim UE's use of a different a cipher suite.

This attack can also be performed for a specific user with only the knowledge of victim's phone number. The adversary first determines the victim UEs GUTI and her paging occasion using [8], and then hijacks her paging channel using **P-1**. Now the adversary initiates phone calls to victim's phone number which trigger the MME to send paging messages to the victim's UE. However, the victim UE's unresponsiveness due to **P-1** causes the MME to send paging with IMSI (subclause 5.5.3.2.7 [4]) which the adversary may intercept. In case the attacker receives multiple such paging with IMSI, he would use the set intersection technique [8] to uniquely identify the victim UE's IMSI. The adversary then performs the traceability attack as follows—(i) forces the victim UE to complete an attach procedure with the legitimate EPC; (ii) captures a valid security_mode_command message; (iii) and finally replays that message to all nearby UEs.

A-3 *Numb Attack and* ***D-1*** *Detach/Downgrade Attack:* For verifying both these attacks, we made the malicious eNodeB inject auth_reject and network initiated detach_request (with different causes) messages in different stages of the protocol, and observed the UE responses to these messages. For auth_reject messages, we observed a complete unresponsiveness of the victim UE until the SIM is re-inserted and the mobile phone is rebooted. Our observation of victim UE's responses in reaction to the detach_request messages (with different causes) are summarized in Table VI.

P-1 *Paging Channel Hijacking Attack:* Successfully carrying out this attack first requires determining the victim UE's

Detach Type	Our observation of victim UE's response
Re-attach required	No cellular signals (shows "No Service"). Requires mobile restart or SIM re-insert to get the 4G LTE back again.
Re-attach not required	Detaches from 4G LTE. Immediately downgrades to 3G/2G and sends attach_request to the 3G/2G network.
IMSI detach	Does not detach from the 4G LTE network.

Table VI: Victim UE's responses to different types of detach.

paging cycle/occasions. To this end, we captured and decoded the system_info_block and attach_request messages—sent in plaintext by the carrier's eNodeB and the victim UE, respectively, and learned the parameters relevant for computing the victim UE's paging occasion (e.g., DRX_cycle, IMSI). The malicious eNodeB then injected fake paging messages (with no paging records) at the paging occasions of the victim UE. We observed that the victim UE only received the fake paging messages instead of the legitimate messages.

After hijacking the victim UE's paging channel, we allowed two senders to place phone calls and send SMSs to victim's phone number triggering the (benign) MME to send multiple paging messages to the victim UE. We observed that the victim did not receive any of the legitimate paging messages, i.e., the service notifications. The victim's unresponsiveness was also noticed on the sender-side.

P-2 *Stealthy Kicking-off Attack:* For this attack, instead of injecting empty paging messages, the malicious eNodeB fabricated the paging messages with a paging record containing the victim UE's IMSI. As soon as the victim UE received this message, we observed that it locally detached itself from the network and sent an attach_request, confirming the attack.

P-3 *Panic Attack:* To inject fake paging messages to arbitrary neighboring UEs, the malicious eNodeB broadcasted paging messages at all possible paging occasions. Each of these paging messages had the ETWS (earthquake and tsunami warning system) bits set to provide the UEs an alert notification. Upon receiving such alert notification, a UE looks for the actual warning message which the eNodeB broadcasts through the system_information_block type 10 or 11 or 12 messages. Since such warning messages may be received by other mobile phones which are not subject to our experiment, we refrained the malicious eNodeB from sending the actual warning messages.

P-4 *Energy Depletion Attack:* We quantitatively measure the UE's energy depletion due to this attack. In particular, we leverage the strong correlation between energy consumption with message transmission rate [47]. We essentially measured the message transmission rate in the benign and attack case, and drew conclusions about energy consumption. To realize this attack, we configured the malicious eNodeB to broadcast paging message with the victim's GUTI at every third paging occasion (i.e., ~ 3 seconds) of the victim UE. Upon reception of this paging message, we observed that the victim UE sent an encrypted and integrity protected service_request message to the malicious eNodeB. We also carried out this attack where the paging message included the victim's IMSI in which case, however, the victim initiated the attach procedure. For the paging with GUTI, we carried out the attack for an hour and observed that the victim sent 1200 service_request messages. In the benign case (measured from 4G LTE traces [48]), however, on average the UE responds to 156 (std. dev. 14.27) paging messages . Roughly, the energy depletion due to this attack ~ 8 times to that of the benign condition. The attacker can make it worse, in case it chooses to inject the paging with GUTI in every paging occasion.

VI. Related Work

In this section, we present existing efforts that focus on the security, privacy, and availability of cellular networks. In this context, although prior work has extensively investigated the security and privacy issues of 2G and 3G protocols [8], [6], [7], [49], [50], there is less work that addresses the same concerns for the 4G LTE networks [13].

The closest to LTEInspector's approach is by Tu et al. [17] where they focus on identifying non-trivial interactions—using an explicit-state model checker [23]—between the different control-plane protocol layers of LTE. Unlike LTEInspector, their work, however, can neither explicitly reason about adversarial actions nor can support cryptographic constructs.

Man-in-the-Middle Attacks: Meyer et al. [12] exploit the null integrity of the security_mode_command message in 3G networks to perform a man-in-the-middle attack. In contrast, our authentication relay attack in 4G LTE allows the adversary to connect to the EPC without nullifying the security capabilities. In this attack the adversary, however, cannot decrypt or inject valid encrypted messages unless the operator uses a weak or no security context. Rupprecht et al. [44] used an implementation bug in a particular LTE dongle (Huawei E3276 USB Dongle) to demonstrate a man-in-the-middle attack for 4G LTE.

IMSI Catching Attacks and 5G AKA Solutions: The IMSI catching attacks [51], [5], [52] still prevail for 4G LTE networks as they did for 2G and 3G networks. Arapinis et al. [10] propose to use public key encryption mechanism for protecting against the IMSI exposure attacks. However, public key encryption of the IMSI will likely incur high overhead during the *attach* procedure. Also, this mechanism will require managing multiple public keys for different MMEs (i.e., the serving network) in the SIM, as well as modification of the LTE protocol implementation in the legacy UEs. Due to these reasons, the 3GPP community has not yet adopted this solution for real deployments. Broek et al. [16] and Khan et al. [53] both proposed changing pseudonym-based (PMSI) defense against IMSI catching attack which have several limitations as discussed in Section IV. The 5G AKA protocol proposed by Norman et al.[54] can be viewed as a hybrid of the approaches by Arapinis et al. [10] and Broek et al. [16]. In such a protocol, the UE encrypts the IMSI using HSS' public key—unlike the public-key of MME as in Arapinis et al. [10]—during the very first occurrence of the attach procedure and then subsequently uses pseudo-IMSIs for communicating with MME. Although this approach does not have to manage multiple MME public keys, it will likely incur additional overhead due to the use of public key encryption and also suffers from the same weakness as the solution proposed in [16]. Dabrowski et al. [14] and Nohl [52] have designed a mobile application that warns the users about the likely presence of IMSI catchers. In contrast, LTEInspector can identify the IMSI catching attack and other attacks that the adversary may launch after knowing the IMSIs.

Linkability Attacks: Arapinis et al. [10] exploit the paging messages with GUTI for linking the IMSI to the GUTI in 3G protocols. In contrast, using paging message with IMSI in 4G LTE we demonstrate how an adversary along with other attacks

(discussed in Section IV-B) can link the IMSIs in 3GPP [35] and the old PMSI with the new PMSI in the enhanced Authentication and Key Agreement (AKA) mechanism [16].

Traceability Attacks: Arapinis et al. [11] presented a traceability attack by exploiting an implementation bug in the 3G network which violates the 3GPP standard recommendation of adopting new temporary identity for the UE with a tracking area change. In another work, Arapinis et al. [10] demonstrate another traceability attack in which the adversary replays the auth_request for the victim UE to all the UEs in an area and detects the presence of the victim UE from the causes (MAC failure or SQN synchronization failure). In contrast, our traceability attack with the *security mode command* procedure exploits the missing nonces in security_mode_command, a different implementation bug of the commercial networks.

Denial-of-Service (DoS) Attacks: Shaik et al. [13] demonstrate 3 DoS attacks against UEs in which downgrading to 3G/2G and denial of all network services are performed with the same tracking_area_update_reject message with just two different causes. In contrast, our DoS attacks use four new techniques as discussed in Section IV. As opposed to the DoS attacks against the UE, Jover et al. [15] discuss a DoS attack against the EPC using a compromised UE/eNodeB that sends huge number of attach_request messages to the EPC and thus takes the EPC down. Jermyn et al. [55] show a similar DoS attack and simulate a set of compromised UEs that exhaust the victim UEs' traffic capacity. Golde et al. [38] exploit a race condition in the paging message responses in 2G networks that enables a malicious UE to send the paging_response message before the victim UE, and thus preventing the victim UE from receiving incoming phone calls/SMS.

VII. Discussion

Properties amenable to our analysis. LTEInspector can reason about temporal trace properties (with cryptographic constructs) of both safety and liveness variations. Our current model cannot handle properties that require reasoning about sets of traces (e.g., noninterference) instead of a single trace. For such properties, we mainly rely on the protocol verifier.

Defenses. We deliberately do not discuss defenses for the observed attacks as retrospectively adding security into an existing protocol without breaking backward compatibility often yields band-aid-like-solutions which do not hold up under extreme scrutiny. It is also not clear, especially, for the authentication relay attack whether a defense exists that does not require major infrastructural or protocol overhaul. A possibility is to employ a distance-bounding protocol; realization of such protocol is, however, rare in practice [56]. Due to the strict performance requirements of 4G LTE, any public-key cryptography-based solution is unlikely to be feasible. On the contrary, symmetric-key cryptography often have scalability issues due to establishing a common shared key for broadcasting sensitive (e.g., paging) messages. Further investigation on defenses is, therefore, required for balancing the performance and scalability as well as guaranteeing the security.

Limitations. Although LTEInspector suggests a systematic approach, it currently requires human intervention, for instance, deciding which sub-steps of the counterexample is required to be modeled in ProVerif and how. Also, our FSM extraction is currently manual and the extracted FSMs are not complete. In the same vein, the list of properties we have checked is not exhaustive. Our current model also does not capture all the data embedded in the messages.

Threat to Validity. Our manually extracted FSMs from the 3GPP standard may not reflect the behavior of real operational networks. Inaccuracies in the FSMs may induce false positives, although, we have not observed any. Due to ethical considerations, we limit our experiments to a custom-built network for some attack validations which may not faithfully capture the operational network behavior.

VIII. Conclusion and Future Work

In this paper, we propose LTEInspector which employs an adversarial model-based testing philosophy for exposing attacks against three critical procedures of 4G LTE. LTEInspector harnesses the strengths of both a symbolic model checkers and a protocol verifier and is demonstrated to be effective in finding 10 novel and 9 prior attacks. We have also validated most of our attacks (i.e., 8 out of 10) in a testbed.

Future work. In future, we would like to explore the following four directions: (i) supporting analysis of other procedures and protocol layers (e.g., RRC and PDCP) messages; (ii) automating some of the manual tasks in LTEInspector; (iii) enriching the model features and analysis of LTEInspector to handle message data; (iv) investigating the defense techniques.

Acknowledgement

We thank our Shepherd, Yongdae Kim, and the anonymous reviewers for their valuable suggestions. This work was supported by Intel/CERIAS RAship, FFTF fellowship by Schlumberger Foundation, NSF grant CNS-1657124, NSF grant CNS-1719369, and Intel as part of the NSF/Intel ICN-WEN program.

References

[1] *Hackers Are Tapping Into Mobile Networks' Backbone, New Research Shows*, https://www.forbes.com/sites/parmyolson/2015/10/14/hackers-mobile-network-backbone-ss7.

[2] *Hackers Take Down the Most Wired Country in Europe*, https://www.wired.com/2007/08/ff-estonia/.

[3] *3GPP Standard*, www.3gpp.org.

[4] *3GPP Standard. Release 12.*, http://www.3gpp.org/specifications/releases/68-release-12.

[5] D. Abodunrin, Y. Miche, and S. Holtmanns, "Some dangers from 2g networks legacy support and a possible mitigation," in *Communications and Network Security (CNS)*, Sept 2015, pp. 585–593.

[6] P. P. C. Lee, T. Bu, and T. Woo, "On the detection of signaling dos attacks on 3g wireless networks," in *26th International Conference on Computer Communications (INFOCOM)*, May 2007, pp. 1289–1297.

[7] N. Golde, K. Redon, and R. Borgaonkar, "Weaponizing Femtocells: The Effect of Rogue Devices on Mobile Telecommunications," in *Proceedings of the 19th Annual Network & Distributed System Security Symposium*, Feb. 2012.

[8] D. F. Kune, J. Koelndorfer, and Y. Kim, "Location leaks on the gsm air interface," in *NDSS*, 2012.

[9] I. Androulidakis, "Intercepting mobile phone calls and short messages using a gsm tester," in *18th Conference on Computer Networks, Ustron, Poland*, A. Kwiecień, P. Gaj, and P. Stera, Eds., 2011, pp. 281–288.

[10] M. Arapinis, L. Mancini, E. Ritter, M. Ryan, N. Golde, K. Redon, and R. Borgaonkar, "New privacy issues in mobile telephony: Fix and verification," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, ser. CCS '12. ACM, 2012, pp. 205–216.

[11] M. Arapinis, L. I. Mancini, E. Ritter, and M. Ryan, "Privacy through pseudonymity in mobile telephony systems," in *NDSS*, 2014.

[12] U. Meyer and S. Wetzel, "A man-in-the-middle attack on umts," in *Proceedings of the 3rd ACM Workshop on Wireless Security*, ser. WiSe '04. ACM, 2004, pp. 90–97.

[13] A. Shaik, J. Seifert, R. Borgaonkar, N. Asokan, and V. Niemi, "Practical attacks against privacy and availability in 4g/lte mobile communication systems," in *23nd Annual Network and Distributed System Security Symposium, NDSS, San Diego, CA, USA, February 21-24*, 2016.

[14] 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*, ser. ACSAC '14. ACM, 2014, pp. 246–255.

[15] R. P. Jover, "Security attacks against the availability of lte mobility networks: Overview and research directions," in *Wireless Personal Multimedia Communications (WPMC), 2013 16th International Symposium on*, June 2013, pp. 1–9.

[16] F. van den Broek, R. Verdult, and J. de Ruiter, "Defeating imsi catchers," in *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '15. ACM, 2015, pp. 340–351.

[17] G.-H. Tu, Y. Li, C. Peng, C.-Y. Li, H. Wang, and S. Lu, "Control-plane protocol interactions in cellular networks," in *Proceedings of the 2014 ACM Conference on SIGCOMM*. ACM, 2014, pp. 223–234.

[18] *Cellular Service*, https://www.fcc.gov/wireless/bureau-divisions/mobility-division/cellular-service.

[19] D. Dolev and A. C. Yao, "On the security of public key protocols," Stanford, CA, USA, Tech. Rep., 1981, http://www.ncstrl.org:8900/ncstrl/servlet/search?formname=detail&id=oai

[20] K. R. Apt and D. C. Kozen, "Limits for automatic verification of finite-state concurrent systems," *Inf. Process. Lett.*, vol. 22, no. 6, pp. 307–309, May 1986.

[21] Z. Manna and A. Pnueli, "A hierarchy of temporal properties (invited paper, 1989)," in *Proceedings of the ninth annual ACM symposium on Principles of distributed computing*. ACM, 1990, pp. 377–410.

[22] A. Cimatti, E. M. Clarke, F. Giunchiglia, and M. Roveri, "Nusmv: A new symbolic model verifier," in *Proceedings of the 11th International Conference on Computer Aided Verification*, ser. CAV '99. London, UK, UK: Springer-Verlag, 1999, pp. 495–499.

[23] G. Holzmann, *Spin Model Checker, the: Primer and Reference Manual*, 1st ed. Addison-Wesley Professional, 2003.

[24] B. Blanchet, "Modeling and verifying security protocols with the applied pi calculus and proverif," *Foundations and Trends in Privacy and Security*, vol. 1, no. 1-2, pp. 1–135, 2016.

[25] S. Meier, B. Schmidt, C. Cremers, and D. Basin, "The tamarin prover for the symbolic analysis of security protocols," in *Proceedings of the 25th International Conference on Computer Aided Verification (CAV)*. Springer-Verlag, 2013, pp. 696–701.

[26] A. Armando and et al., "The avispa tool for the automated validation of internet security protocols and applications," in *Proceedings of the 17th International Conference on Computer Aided Verification*, ser. CAV'05. Springer-Verlag, 2005, pp. 281–285.

[27] J. Bengtson, K. Bhargavan, C. Fournet, A. D. Gordon, and S. Maffeis, "Refinement types for secure implementations," *ACM Trans. Program. Lang. Syst.*, vol. 33, no. 2, pp. 8:1–8:45, Feb. 2011.

[28] G. Lowe, "Breaking and fixing the needham-schroeder public-key protocol using fdr," in *Proceedings of the Second International Workshop on Tools and Algorithms for Construction and Analysis of Systems*, ser. TACAs '96. Springer-Verlag, 1996, pp. 147–166.

[29] C. J. Cremers, "Unbounded verification, falsification, and characterization of security protocols by pattern refinement," in *Proceedings of the 15th ACM Conference on Computer and Communications Security*, ser. CCS '08. ACM, 2008, pp. 119–128.

[30] A. C. Yao, "Theory and application of trapdoor functions," in *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, ser. SFCS '82. IEEE Computer Society, 1982, pp. 80–91.

[31] J.-K. Tsay and S. F. Mjølsnes, "A vulnerability in the umts and lte authentication and key agreement protocols," in *Proceedings of the 6th MMM-ACNS*. Springer-Verlag, 2012, pp. 65–76.

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

[33] M. Abadi and B. Blanchet, "Analyzing Security Protocols with Secrecy Types and Logic Programs," in *29th Annual ACM SIGPLAN - SIGACT Symposium on Principles of Programming Languages (POPL 2002)*. Portland, Oregon: ACM Press, Jan. 2002, pp. 33–44.

[34] E. A. Emerson and K. S. Namjoshi, "Reasoning about rings," in *Proceedings of the 22Nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*. ACM, 1995, pp. 85–94.

[35] *3GPP. Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3 Specification 3GPP TS 24.301 version 12.8.0 Release 12.*, [Online]. Available: http://www.3gpp.org/dynareport/24301.htm.

[36] R. Racic, D. Ma, and H. Chen, "Exploiting mms vulnerabilities to stealthily exhaust mobile phone's battery," in *Securecomm and Workshops*, Aug 2006, pp. 1–10.

[37] J. Serror, H. Zang, and J. C. Bolot, "Impact of paging channel overloads or attacks on a cellular network," in *Proceedings of the 5th ACM Workshop on Wireless Security*, 2006, pp. 75–84.

[38] N. Golde, K. Redon, and J.-P. Seifert, "Let me answer that for you: Exploiting broadcast information in cellular networks," in *Proceedings of the 22Nd USENIX Conference on Security*, 2013, pp. 33–48.

[39] *USRP B210*, https://www.ettus.com/product/details/UB210-KIT.

[40] *OpenLTE*, http://openlte.sourceforge.net/.

[41] *srsLTE*, https://github.com/srsLTE.

[42] *Qxdm Professional Qualcomm Extensible Diagnostic-Monitor*, www.qualcomm.com/media/documents/files/qxdm-professional-qualcomm-extensible-diagnostic-monitor.pdf.

[43] *srsUE*, https://github.com/srsLTE/srsUE.

[44] D. Rupprecht, K. Jansen, and C. Pöpper, "Putting lte security functions to the test: A framework to evaluate implementation correctness," in *Proceedings of the 10th USENIX Conference on Offensive Technologies*, ser. WOOT'16, 2016, pp. 40–51.

[45] N. Bui and J. Widmer, "Owl: A reliable online watcher for lte control channel measurements," in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*, ser. ATC '16. ACM, 2016, pp. 25–30.

[46] *OpenEPC*, www.openepc.com.

[47] P. Y. Kong, "Power consumption and packet delay relationship for heterogeneous wireless networks," *IEEE Communications Letters*, vol. 17, no. 7, pp. 1376–1379, July 2013.

[48] Y. Li, C. Peng, Z. Yuan, J. Li, H. Deng, and T. Wang, "Mobileinsight: Extracting and analyzing cellular network information on smartphones," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '16, 2016, pp. 202–215.

[49] P. Traynor, P. McDaniel, and T. La Porta, "On attack causality in internet-connected cellular networks," in *Proceedings of 16th USENIX Security Symposium*, 2007, pp. 21:1–21:16.

[50] P. Traynor, M. Lin, M. Ongtang, V. Rao, T. Jaeger, P. McDaniel, and T. La Porta, "On cellular botnets: Measuring the impact of malicious devices on a cellular network core," in *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS)*, 2009, pp. 223–234.

[51] R. Borgaonkar and S. Udar, "Understanding imsi privacy," in *BlackHat*, 2014.

[52] K. Nohl, "Mobile self-defense." [Online]. Available: https://events.ccc.de/congress/2014/Fahrplan/system/attachments/2493/original/Mobile_Self_Defense-Karsten_Nohl-31C3-v1.pdf

[53] 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.

[54] *Protecting IMSI and User Privacy in 5G Networks*, www.ericsson.com/res/docs/2016/protecting-imsi-and-user-privacy-in-5g-networks.pdf.

[55] J. Jermyn, G. Salles-Loustau, and S. Zonouz, "An analysis of dos attack strategies against the lte ran," vol. 3, 2014, pp. 159–180.

[56] K. B. Rasmussen and S. Čapkun, "Realization of rf distance bounding," in *Proceedings of the 19th USENIX Conference on Security*, ser. USENIX Security'10, 2010, pp. 25–25.