Breaking LTE on Layer Two

David Rupprecht
Ruhr-University Bochum
david.rupprecht@rub.de

Katharina Kohls
Ruhr-University Bochum
katharina.kohls@rub.de

Thorsten Holz
Ruhr-University Bochum
thorsten.holz@rub.de

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

***Abstract*—Long Term Evolution (LTE) is the latest mobile communication standard and has a pivotal role in our information society: LTE combines performance goals with modern security mechanisms and serves casual use cases as well as critical infrastructure and public safety communications. Both scenarios are demanding towards a resilient and secure specification and implementation of LTE, as outages and open attack vectors potentially lead to severe risks. Previous work on LTE protocol security identified crucial attack vectors for both the physical (layer one) and network (layer three) layers. Data link layer (layer two) protocols, however, remain a blind spot in existing LTE security research.**

In this paper, we present a comprehensive layer two security analysis and identify three attack vectors. These attacks impair the confidentiality and/or privacy of LTE communication. More specifically, we first present a passive identity mapping attack that matches volatile radio identities to longer lasting network identities, enabling us to identify users within a cell and serving as a stepping stone for follow-up attacks. Second, we demonstrate how a passive attacker can abuse the resource allocation as a side channel to perform website fingerprinting that enables the attacker to learn the websites a user accessed. Finally, we present the ALTER attack that exploits the fact that LTE user data is encrypted in counter mode (AES-CTR) but not integrity protected, which allows us to modify the message payload. As a proof-of-concept demonstration, we show how an active attacker can redirect DNS requests and then perform a DNS spoofing attack. As a result, the user is redirected to a malicious website. Our experimental analysis demonstrates the real-world applicability of all three attacks and emphasizes the threat of open attack vectors on LTE layer two protocols. [1]

I. Introduction

The latest mobile communication standard LTE represents the daily communication infrastructure for billions of people in the world and has a pivotal role in our information society. LTE is designed to combine performance goals such as high transmission rates and low latency with a series of security features like formally proven mutual authentication, well-established encryption algorithms such as AES, and separated security domains. Besides casual use cases, LTE also has an emerging relevance for critical infrastructures and public safety communications [2]. Both scenarios are demanding towards a resilient and secure specification and implementation of LTE, as outages and open attack vectors potentially lead to severe risks. While the LTE specification considers a diverse set of security features, it can hardly predict all potential attacks, and it is even harder to cover sets of restrictions in real-world implementations.

Consequently, recent academic and non-academic work identified various potential vulnerabilities on different layers of the LTE protocol stack. On the network layer (layer three), passive or active attackers can either localize a user or deny the service and thus downgrade the phone to the insecure GSM network [3]–[5]. On the physical layer (layer one), LTE can be the target of jamming attacks that aim to deny the service [6]–[9]. As a matter of fact, the previous research efforts focused only on layer one or layer three protocols and—to the best of our knowledge—no security analysis of data link layer (layer two) protocols exists to date. This leads to a situation of uncertainty about potential security and privacy threats that arise from the specification or implementation flaws of the data link layer and its three protocols: Medium Access Control (MAC), Radio Link Control (RLC), and Packet Data Convergence Protocol (PDCP).

In this paper, we perform a security analysis of LTE on layer two and analyze these protocols for potential vulnerabilities. As a result, we introduce two passive attacks and one active attack that impair the confidentiality and privacy of LTE communication. Table I shows an overview of the attacks and their properties. We first focus on a passive adversary who can remain stealthy during an attack, i. e., being successful does not depend on any active interference with the network entities or protocols. Our first passive attack, the *identity mapping* attack, allows an adversary to map the user's temporary network identity (TMSI) to the temporary *radio* identity (RNTI). More specifically, we demonstrate how an attacker can precisely localize and identify a user within the cell, distinguish multiple transmission streams, and use this information as a stepping stone for subsequent attacks. One example for this is our second attack vector, the *website fingerprinting* attack. Website fingerprinting is known from other contexts like Tor [10], where traffic analysis reveals the browsing behavior of users despite Tor's onion encryption. In the context of LTE, we demonstrate a comparable information leak in the resource allocation: even though transmissions are encrypted, we can access plaintext information up to the PDCP and learn the transmission characteristics for individual users. This information is sufficient to distinguish accessed websites and de-anonymize a connection that is perceived to be secure due to encryption. Both attacks already harm user privacy separately, but they can be combined to an even stronger version of website fingerprinting, while solely depending on passive (downlink) sniffing.

We further introduce an active attack called ALTER that exploits the missing integrity protection of LTE user data to perform a chosen-ciphertext attack. Our attack is based on the

TABLE I
OVERVIEW OF LAYER TWO ATTACKS

	Model	Attack Vector	Attack Aim	Attack Flaw	Hardware	Implementation
Identity Mapping	Passive	RNTI and TMSI Mapping	Privacy (Identity, Location)	Specification	USRP	Software Stack
Website Fingerprinting	Passive	Layer Two Scheduling Metadata	Confidentiality	Specification	USRP	Software Stack
ALTER	Active	Lack of Integrity Protection	Confidentiality, Redirection	Specification	2x USRPs	Software Stack

insight that user data is encrypted in counter mode (AES-CTR) but not integrity protected, hence the cipher is *malleable*. We show how an adversary can actively manipulate the encrypted payload and control specific parts of the message. More specifically, we demonstrate how an attacker uses a malicious LTE relay to manipulate the IP addresses within an encrypted packet, thereby redirecting a packet to a malicious DNS server in the uplink direction, while maintaining a stable and transparent connection at all times. Even though ALTER solely focuses on layer two, the attack still has cross-layer consequences and impacts overlying protocols like IP and DNS. ALTER affects all LTE devices and has implications up to the application layer. At the same time, the attack is hard to detect by existing countermeasures like rogue base station detection [11], [12] and makes a change in the LTE specification the only viable prevention from user data manipulation.

We have verified all attack vectors within a real-world, commercial network using a Software Defined Radio (SDR) and an open-source LTE stack implementation. Our experiments show that our attacks are feasible in practice and pose a realistic threat to users. In particular, we show that the identity mapping attack can be performed in a commercial network on an estimate of $94.73\,\%$ of connections. Our website fingerprinting attack achieves an average detection rate of approximately $90\,\%$ for the Alexa top 50 in a closed-world scenario, tested with three different devices. Combining both attacks creates a powerful non-invasive attacker that is barely detectable. Finally, we have built a proof-of-concept malicious relay and performed the ALTER attack against a Commercial Off-The-Shelf (COTS) mobile phone in a commercial network. We were able to successfully redirect a mobile phone to visit a malicious website while maintaining a stable LTE connection.

In summary, we provide the following three contributions:

- We perform an extensive LTE layer two analysis. In particular, we examine the control plane for possible information leaks that allow an attacker to gain access to sensitive information. Furthermore, we investigate the effects of missing integrity protection on the user plane.
- Based upon the performed analysis, we present three attacks: Two passive attacks allow identity mapping and website fingerprinting purely based on metadata. The active attack allows to redirect DNS traffic and, thus, perform a DNS spoofing attack.
- We demonstrate the feasibility of all three attacks with realistic setups. For each attack, we discuss the real-world applicability, especially with a focus on attacker capabilities and the impact for the user. Furthermore, we discuss possible countermeasures to mitigate the threats. By sharing our results, we hope to influence the upcoming 5G specification to include countermeasures.

Responsible Disclosure. The lack of integrity protection was an active decision of the LTE specification body, mainly related to the additional overhead induced on the radio layer [13]. We demonstrate that this missing integrity protection can be exploited in practice. We are in contact with the GSM Association (GSMA) and 3rd Generation Partnership Project (3GPP) security groups, following the guidelines of responsible disclosure. We hope to influence the upcoming 5G specifications to add mitigations for the demonstrated attacks and will actively work with GSMA and 3GPP to resolve these attack vectors.

II. TECHNICAL BACKGROUND

The different components of the LTE network infrastructure are defined by the roles they fulfill, e. g., they connect a user to the network, manage the resource allocation, or build the backbone of the network. The capabilities of all these components are defined following the rule set of the LTE protocol stack and its respective layers. Within this paper, we focus on the data link layer of the air interface between the user and the network. In the following, we provide an overview of the network and the LTE stack, along with an introduction of relevant authentication and encryption algorithms. Furthermore, we introduce the two adversary models that we consider in our attacks.

A. LTE Network Overview

The LTE network infrastructure consists of end devices for users (User Equipment (UE)), base stations as intermediate connectors called Evolved NodeB (eNodeB), and the core network for mobility management with the aim to provide permanent Internet access. We conduct our attacks between the victim user and a benign base station.

1) UE: The user equipment is the end device providing services to the user. It has a permanent identity, the International Mobile Subscriber Identity (IMSI), and different temporary identities within the network. One of these temporary identifiers is the Radio Network Temporary Identity (RNTI), which helps to distinguish multiple connections on the radio layer. Besides the connection establishment, the UE also applies encryption/decryption and integrity protection for transmissions through the network.

2) eNodeB: The eNodeBs are the base stations of the LTE network and responsible for radio resource management and user data encryption. Furthermore, an eNodeB sends paging messages on a broadcast channel. For our attacks, we exploit

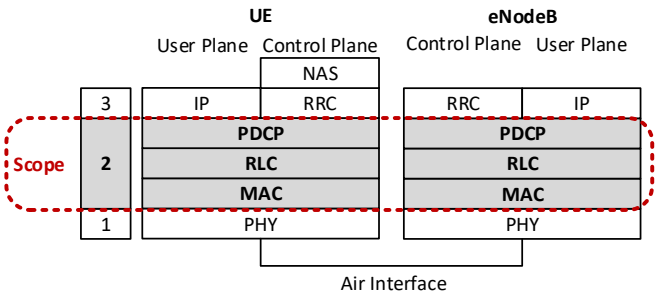

Fig. 1. Overview of the LTE Protocol stack and the scope of our analysis.

the fact that UEs select the eNodeB with the highest signal strength allowing us to establish an active malicious relay.

3) Evolved Packet Core (EPC): The EPC is the core of the network and is responsible for authentication, mobility management, and forwarding of user data. It triggers the procedure for sending out paging requests when user data is incoming.

B. LTE Protocol Stack

The LTE protocol stack between the UE and the eNodeB is depicted in Figure 1. We briefly explain each layer and its tasks from bottom to top. Later on, we describe the individual security mechanisms within the protocol stack separately, since some work in a cross-layer fashion.

1) Physical Layer: As the lowest layer in the protocol stack, the physical layer is responsible for transmitting information over the air interface. The physical layer searches for cell candidates and synchronizes with a selected cell. Further, it controls the transmission power for the physical channel and adapts encoding and modulation schemes. The values for these parameters are adjusted by a channel quality indicator that is regularly sent by the UE's MAC protocol.

2) Data Link Layer: The data link layer extends the physical layer bit pipe by additional services towards the upper layers and provides mechanisms for reliability, security, and integrity. It is organized in three sublayers: (i) MAC protocol scheduling the medium access, (ii) RLC protocol managing the segmentation or concatenation of data units, and (iii) PDCP protocol performing ciphering tasks and optional IP header compression.

Medium Access Control (MAC). The MAC protocol manages the access to the radio resources of LTE. To do so, each UE with an active radio connection must be distinguishable by a unique identity, the RNTI. To obtain such an RNTI, the UE performs the Random Access Preamble (RAP) with the eNodeB of its current cell and exchanges an unencrypted Random Access Response (RAR). In this process, the MAC layer of the eNodeB determines the available radio resources for the UE, matches these assigned resources to the RNTI, and finally signals this information to the UE to be used for the following transmissions. We use the unique information of the RNTI to perform our *identity mapping* attack.

When data needs to be sent in uplink direction, the UE layer issues a scheduling request at a configured location. The eNodeB utilizes the Downlink Control Information (DCI) for notifying the UE when and where the resources are available in uplink and downlink direction. As we will see later, the DCI information leaks sensitive information that enables us to perform a *website fingerprinting* attack.

Radio Link Control (RLC). The RLC protocol offers three transmission modes: (i) Acknowledged Mode (AM), (ii) Unacknowledged Mode (UM), and (iii) Transparent Mode (TM). Depending on the mode, the RLC protocol applies error correction, segmentation, and assembles data into the correct order of upper-layer packets. Furthermore, it manages retransmissions including the detection of retransmitted packets.

Packet Data Convergence Protocol (PDCP). The PDCP protocol provides encryption and integrity protection for control plane messages to the overlying Radio Resource Control (RRC) layer and transfers encrypted user plane data to upper-level protocols like IP. Within the data link layer, the PDCP layer is the first to apply encryption algorithms, hence, we can directly read the payload and header information of all packets *below* this sublayer. This allows us to passively analyze the meta information of layer two transmissions, e. g., the PDCP length of a packet, and perform the *website fingerprinting* attack. Further, we exploit the lack of user data integrity protection for our ALTER attack.

3) Network Layer: There are three sublayers on the network layer: Non-Access Stratum (NAS), Radio Resource Control (RRC), and IP. The NAS layer performs mobility with the core network using encrypted and integrity protected messages. On the RRC sublayer, all radio connections between the UE and the eNodeB are managed, including the configuration of all lower-level protocols down to the physical layer. Finally, the IP protocol handles transmissions to overlying transport protocols like TCP and UDP and, therefore, maintains connections to the Internet.

C. Mobility Management

The mobility of devices in the LTE network holds additional challenges for the specification and implementation of all respective protocols. In the context of this work, the paging procedure is of particular interest.

Paging. The paging procedure is used to notify the UE of incoming data transmissions or a call. Sending paging messages is initiated by the eNodeB, i. e., it broadcasts the Temporary Mobile Subscriber Identity (TMSI) of a certain UE on the paging channel. All UEs within the cell that do not have an active radio connection listen to the paging channel and react to a message in case their TMSI is sent. The paging procedure affects the *identity mapping* attack, as it helps the adversary to learn the unique identifier of a user within the network.

D. Authentication and Encryption

LTE uses a challenge-response protocol for Authentication and Key Agreement (AKA) in which the core network (EPC)

sends an authentication request to the UE. This request contains an authentication token for verification of the network's permanent key on the SIM card of the UE. In case of a successful verification, both the network and the UE can derive a session key from the long-term secret and the random nonce. Using this temporary key material, the NAS and RRC sublayers of the data link layer can establish encryption and integrity protection mechanisms, respectively. The selection of certain security algorithms depends on the network and is defined in the security mode command, sent out by the EPC/eNodeB.

LTE specifies different security mechanisms based on well-established encryption algorithms such as AES. Integrity protection is accomplished by a cipher block chaining message authentication code (CBC-MAC) that is appended to signaling messages. User data is encrypted in counter mode (AES-CTR), where the encryption algorithm is used as a keystream generator, and the ciphertext is computed by XORing the keystream with the plaintext[1]. In fact, this helps us later to perform our ALTER attack given that the cipher is *malleable*.

E. Attacker Model

We use two different attacker models for our layer two security analysis. The *passive attacker* acts as an eavesdropper and can passively sniff radio layer information within the victim's cell and remains unnoticed. In contrast, the *active attacker* extends these capabilities for intercepting messages as a Man-in-the-Middle (MitM) attacker. More specifically, such an attacker can alter message contents and forward the altered packets to the next node. Both attackers depend only on low-budget SDR hardware (in practice, our setup costs about 2600 $ for the active relay) and uses open-source LTE stack implementations [14], [15] that we extended for our attacks. These constraints and requirements render both passive and active attacks a realistic threat in practice. In summary, we assume the following attacker model:

Passive Attacker. The passive attacker eavesdrops transmissions in up- and downlink direction within the same cell the user is located in. Therefore, the attacker can receive and decode signals sent out by the eNodeB and the UE. To do so, it is *not* mandatory to have any knowledge about the established key material.

Active Attacker. In addition to the scope of the passive attacker, the active attack includes capabilities for sending radio signals on certain frequencies. Using these capabilities, the attacker can establish a malicious relay in the network by impersonating a UE towards the network and an eNodeB towards the user. Again, no knowledge of the key material is required for our attacks.

III. Passive Layer Two Attacks

Our passive attacks comprise identity mapping, in which the attacker learns the identity of a user by eavesdropping on the connection establishment procedure. Furthermore, identity mapping serves as a stepping stone for the second attack: website fingerprinting by transmission metadata. Website fingerprinting reveals the browsing behavior of a user by exploiting resource allocation scheduling of the network.

A. Identity Mapping Attack

The identity mapping attack exploits temporary identifiers on layer *two* during the radio connection establishment. It does not depend on any active interference like comparable paging attacks [3], [16], [17]. Compared to the previous mention of this attack vector [4], we describe the attack details and present a practical evaluation in a commercial network using a simple downlink sniffer. In the following, we introduce the attacker assumptions, the connection establishment process, give an overview of the attack procedure, and present experimental results.

Attack Assumption. For the identity mapping attack, we assume that the attacker knows neither the RNTI nor the TMSI of a victim. The attacker learns the mapping between both identities during the radio layer connection establishment, which is triggered every time a user sends or receives data through the network. We exploit the fact that radio packets contain both their own radio layer identity (RNTI) and the TMSI of the overlying Non-Access Stratum (NAS). The mapping can then be further exploited, e. g., the attacker performs a paging attack to map the TMSI to the public phone number or she can perform a website fingerprint attack.

Connection Establishment Process. In the connecting process, the UE sends a Random Access Preamble (RAP) to the eNodeB (cf. Figure 2 ①) and receives the response (RAR) including the Cell Radio Network Temporary Identity (C-RNTI) ②. The C-RNTI serves as a unique identifier of the user within one radio session until the connection is released. In response to receiving the C-RNTI, the UE sends an RRC connection request to the eNodeB ③, which includes the UE's identity. This can either be the TMSI or a random value in case the UE does not possess a valid TMSI at this moment. The eNodeB completes the connection establishment by replying with the RRC connection setup message ④. In our attack, we either exploit the RRC connection request ③ in *uplink* direction, or the RRC connection setup ④ in *downlink* direction.

The Attack. Matching the C-RNTI and the TMSI becomes possible, as packets on the MAC layer use the C-RNTI to be addressed correctly, i. e., delivered to the correct UE. The UE receives the C-RNTI within the Random Access Response (RAR) ② which from now on identifies the UE on the MAC layer. At this point, we benefit from the fact that there are only ten possible Random Access RNTIs (RA-RNTIs), hence, we can monitor all possible RAR and derive the C-RNTI. The information of the RAR in message ② is sufficient for conducting the identity mapping in the following steps ③, ④ of the connection establishment. In particular, we match the C-RNTI *and* the TMSI by (a) using an uplink sniffer or

[1]LTE specifies this as EEAn, where n specifies the underlying encryption algorithm; EEA2 is relevant in our context and the underlying algorithm is AES.

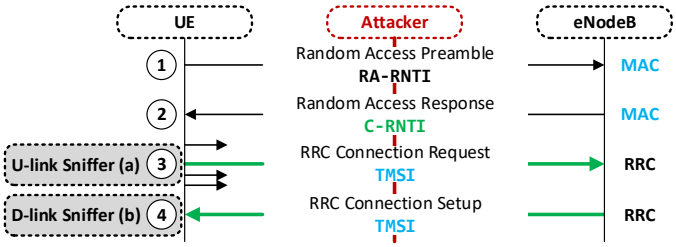

Fig. 2. Radio Connection Establishment Process. We learn the C-RNTI by monitoring all RARs (2) on the downlink shared channel. We now either exploit the RRC connection request (3) or contention-based resolution (4).

by (b) exploiting the contention-based resolution of the RRC connection setup.

(a) In response to the RAR ②, the UE sends the RRC connection request ③ including the TMSI. We use the C-RNTI for identifying the uplink resource allocation for the target UE, e. g., we can distinguish multiple transmissions in the uplink direction (cf. Figure 2, green) and filter out the specific RRC connection request that matches the monitored C-RNTI. In other words, we know when the UE uses the uplink for transmitting the RRC connection request including the TMSI. We can now match the C-RNTI ② and TMSI ③ for a successful attack.

(b) After the RRC connection request, the eNodeB proactively applies contention-based resolution for resolving possible collisions during the random access procedure (cf. Figure 2 ①). Such collisions can occur when more than one UE choose the same RAP within the same time slot. The only case of contention-free RAPs occurs during a handover procedure. In all other cases, the RRC connection setup ④ includes a copy of the RRC connection request ③ with its UE identity. More precisely, the specification states that the UE contention resolution identity field of the RRC connection setup must contain the previous uplink data unit (see [18] in Section 6.1.3.4). In our case the precious uplink data unit is the RRC connection request. As the RRC connection request contains the UE identity, e. g., the TMSI or random value, we can now match the C-RNTI ② and TMSI ④.

1) Experiments: We demonstrate the real-world feasibility of the identity mapping attack by conducting it in a commercial network. In the following, we introduce the technical setup and attack procedure.

Experimental Setup. In our setup, we use two SDRs [14], one representing the target UE (cf. Figure 2), and the other representing the attacker's downlink sniffer (b).

The target UE implements a modified version of srsUE [15], e. g., we extend the software stack such that we can connect to a commercial network. This requires commercial SIM support only, which we realize by using the `PCSC` library [19]. Using these extensions, we can establish an IP connection through the commercial network to the Internet. The second SDR acts as the attacker's passive downlink sniffer. We use it to listen to the broadcast channels of the eNodeB. Again, the sniffer implements the srsLTE software stack. For verifying the success of both attack variants, we record traces at the UE uplink (a) and the downlink sniffer (b).

Procedure. In our experiments, we first assure that all required preconditions are met and subsequently perform the identity mapping attack.

- **Precondition: TMSI.** The UE performs a radio connection establishment with the eNodeB followed by a successful AKA with the core network. The core network replies with the UE's valid TMSI for all further communication. This assures that the UE uses a valid TMSI for the following steps.
- **Precondition: Radio Idle.** The UE remains idle withing the range of the RRC inactivity timer (as default 10 s). Then, the eNodeB signals the UE to transit into the RRC idle state. This assures the performance of the *radio connection establishment* process as soon as the UE intends to send data through the network.

Both preconditions create a setup that is comparable to the characteristics of a real-world scenario, i. e., we assume the possession of a valid TMSI for the user and conduct the attack during the connection establishment.

1) **Attack Step 1.** We setup a new TCP connection to an arbitrary server in the Internet and trigger the radio connection establishment process (cf. Figure 2).
2) **Attack Step 2.** We use the downlink sniffer to eavesdrop the random access responses of the eNodeB for learning all C-RNTI candidates. Up to this point the attack steps are generic, i. e., we can use the C-RNTI of message ② for the up- or downlink sniffer. We continue with attack mode (b).
3) **Attack Step 3.** The eNodeB sends the TMSI in the RRC connection setup ④ within the contention-based resolution. We eavesdrop this information using the downlink sniffer.
4) **Attack Step 4.** We match the set of C-RNTIs of attack step 2 with the TMSI of the contention-based resolution. We can now identify and localize the user within the cell.

The above attack procedure depends on the presence of a valid TMSI within the contention-based resolution. We verify this as an attack procedure with high success probability in our experiments and discuss the use of either an up- or downlink sniffer in the discussion.

2) Results: We successfully repeat the identity mapping attack three times using a downlink sniffer. Furthermore, we provide a theoretical analysis of uplink traces as proof for the feasibility of the uplink sniffer. Figure 3 depicts the Wireshark trace of the RRC connection setup contention-based resolution (attack step 3), recorded by the downlink sniffer. In particular, we see the RRC connection setup message ④ addressed to C-RNTI of the target UE ① that we learned from the RAR of the eNodeB. In the contention-based resolution ②, we find the TMSI assigned to the target UE ③ as part of the RRC connection request. By combining both identifiers, we successfully match layer two and three identities.

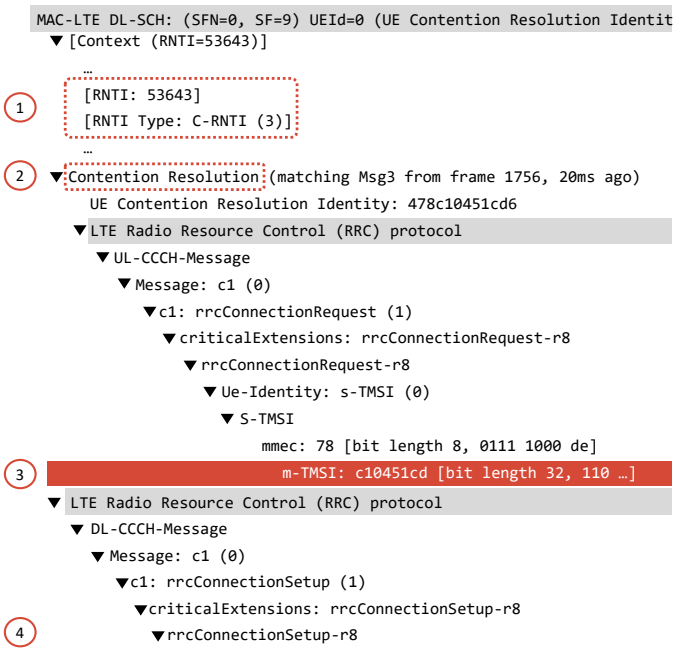

Fig. 3. Identity Mapping Attack: We can decode the TMSI of the RRC connection request as part of the contention resolution identity in the downlink RRC connection setup message. The contention resolution identity (2) is part of the MAC header and located before the RRC connection setup (4). We successfully map the TMSI to the C-RNTI with a downlink sniffer.

As the downlink sniffer depends on the presence of the TMSI within the RRC connection setup, we record a total of 96,911 connection establishment procedures within five days. We conduct these measurements within the cell of a commercial network. Our results show that in 96.85 % of all radio connection establishments we find a contention-based resolution, of which 91.75 % contain the required TMSI. As this covers the majority of connections, the downlink sniffer can be considered a reliable attack variant.

3) Discussion: We next discuss the real-world applicability of identity mapping and compare the deployment of an up- or downlink sniffer.

Real-World Applicability. The identity mapping attack by itself is not detectable, as it is completely passive. Deploying the passive downlink sniffer only depends on standard hardware and an open software stack. Nevertheless, one constraint is the existence of a valid TMSI.

While the proposed identity mapping combines *arbitrary* pairs of C-RNTIs and TMSIs, we can extend the attack by common active paging techniques [3], [4], [17]. This allows us to identify and localize *specific* users for a pre-known TMSI within the cell. We achieve this targeted detection of users at the expense of being detectable through active interference.

Uplink vs. Downlink. The eNodeB synchronizes uplink transmissions depending on the distance between the UE and itself. In particular, it estimates the required transmission delay and signals the time offset for sending data in advance. Deploying an uplink sniffer *between* the UE and the eNodeB requires the attacker to synchronize with this advance offset. Consequently, the attacker must guess the exact location of the UE, which challenges using an uplink sniffer.

In contrast, there is no advance synchronization between the eNodeB and the UE in the downlink direction, i. e., the downlink sniffer can be deployed without any knowledge about the UE's location. In conclusion, it is preferable to use the downlink sniffer on an average of 94.73 % of contention-based resolution access procedures rather than depending on the advance synchronization in the uplink direction.

B. Website Fingerprinting

Tor is a prominent example for website fingerprinting attacks, where an adversary learns the destination of a connection despite the layered encryption of Tor [10], [20]. This becomes possible due to information leaks in the metadata of a connection, e. g., characteristic timing patterns of transmissions that allow distinguishing different websites. In the following, we demonstrate how the challenge of website fingerprinting can be mapped to LTE layer two attacks.

The MAC layer is responsible for scheduling the data transmission of a connection. In particular, the DCI information defines the data allocation for the uplink and downlink for each user individually. As a passive adversary, we can eavesdrop on this information and learn the user data consumption, i. e., the volume of traffic that was sent and received over a connection. This becomes possible by decoding the DCI information that provides unencrypted information up to the PDCP layer. From this information, we learn metadata features, like the length of a PDCP packet, which helps to distinguish requests to different websites in their time series representation.

For conducting a closed-world website fingerprinting attack, we record a corpus of labeled traces for a representative set of websites. Starting from this information set, we analyze traces of new connections and compare their characteristics with the metadata features of the already recorded corpus. An attack can be considered successful if we manage to identify requested websites just from metadata information at an acceptable success rate.

1) Experiments: We conduct the website fingerprinting attack within our own LTE network for recording a sample corpus of layer two traces in up- and downlink connection, according to the following experimental setup and attack procedure.

Experimental Setup. We build a lab LTE network setup by deploying a modified version of the srsLTE eNodeB along with an OpenAirInterface Evolved Packet Core (EPC) [15], [21]. Both components behave specification conform and we can connect COTS mobile phones with a programmable SIM card to our LTE network. In particular, we test three Android phones and access the Alexa top 50 websites overall 100 times with each phone automatically by using Appium [22]. For each new visit, we reset all caches at the phone. Each page visit results in a `pcap` trace, recorded at the eNodeB. We can distinguish user and control plane traffic based on the logical channel ID in the MAC header and thus obtain traces free from control traffic. The raw user plane traces then document

the $(f_1, rnti)$, $(f_2, pdcp_d)$ PDCP direction (up- or downlink), $(f_3, pdcp_s)$ PDCP sequence number, $(f_4, pdcp_l)$ PDCP length, and the timestamp of each packet.

Procedure. Our classification procedure consists of two consecutive analysis steps. First, we compare all captured traces using fast dynamic time warping (*FastDTW*) as a distance metric for the comparison of recorded traces [23]. This time series analysis stretches two input vectors X, Y in a way that the Euclidean distances between corresponding points are minimal. In other words, DTW helps to compute the similarity of measured traffic without depending on synchronization, e. g., we use this for distinguishing websites by individual traffic patterns. Second, we use the distances as an input to the *k-nearest neighbor algorithm (k-NN)* as decision function. In particular, for an unknown trace, we search the closest (1-NN) other trace within the set of labeled traces [24] and use this to classify the new sample. We repeat the analysis using a *10-fold cross-validation* for the verification of our results.

The *standard*, i. e., non-optimized time warping problem, constructs a warp path W given two time series X, Y of lengths $|X|, |Y|$:

$$\begin{aligned} &W = w_1, w_2, \ldots, w_K \\ &max(|X|, |Y|) \leq K < |X| + |Y|, \end{aligned} \tag{1}$$

where K is the length of the warp path, and the k^{th} element of the warp path is $w_k = (i, j)$ with i as index of a time series element in X and j an index of Y, respectively. We get an optimal warp path W_{opt} if the distance is *minimal*:

$$dist(W_{opt}) = \sum_{k=1}^{K} dist(w_k(i), w_k(j)), \tag{2}$$

where $dist(w_k(i), w_k(j))$ is the distance between two data point indexes of $i \in X, j \in Y$ in the k^{th} element of the warp path. The standard implementation of the Dynamic Time Warping, as introduced in Equations 1 and 2, has a complexity of $\mathcal{O}(N^2)$, whereas we refer to the approximate *FastDTW* implementation with complexity $\mathcal{O}(N)$ [23].

Applying *FastDTW* as distance metric, we generate a distance matrix $\mathbf{M}_{dist} = K \times L$ with mutual distances between traces of a training set $g_k \in G$ and a test set $t_l \in T$:

$$\mathbf{M}_{dist} = \begin{pmatrix} d(g_1, t_1) & d(g_1, t_2) & \cdots & d(g_1, t_L) \\ d(g_2, t_1) & d(g_2, t_2) & \cdots & d(g_2, t_L) \\ \vdots & \vdots & \ddots & \vdots \\ d(g_K, t_1) & d(g_K, t_2) & \cdots & d(g_K, t_L) \end{pmatrix}, \tag{3}$$

where $d(g_k, t_l)$ is the distance between the respective training and test trace. The matrix includes all candidate websites of the recorded corpus, e. g., depending on the training and test set size, we draw a defined number of traces from each website. From the distances, we define the 1-NN nearest neighbor, i. e., the lowest distance trace within the training data for the current test trace. More precisely, we determine the minimum of each column in the distance matrix $\mathbf{M}_{dist}$. As a metric for the success of the attack, we derive (1) the average success and standard deviation for a 10-fold cross-validation, as well as (2) the false positive matches for each site in particular.

2) Results: Our results are shown in Table II represent the average true positive (TP) rates, i. e., the relative number of correct website guesses, and the standard deviation (SD) over all ten repetitions of the cross-validation. We achieve an average success rate of $89.63\,\% \pm 10.63$ in uplink and $89.13\,\% \pm 11.2$ in downlink transmissions for individual devices, i. e., when comparing traces for each phone individually.

TABLE II
WEBSITE FINGERPRINTING SUCCESS RATES

Android		Downlink		Uplink	
Device	**OS**	**TP**	**SD**	**TP**	**SD**
LG Nexus 5	`v5.1`	0.949	±0.067	0.936	±0.071
Huawei p9 Lite	`v7.0`	0.932	±0.108	0.922	±0.117
Motorola Moto G4	`v6.0.1`	0.808	±0.144	0.816	±0.148

While we apply comparably simple analysis methods, the success rates of the website fingerprinting attack indicate a promising starting point for future work.

3) Discussion: We present the website fingerprinting attack as a first proof-of-concept for demonstrating the threat of traffic analysis on PDCP sublayer metadata. While our results indicate high success rates for the up- and downlink traffic of different devices, we emphasize that these first insights are limited in several ways. In the following, we discuss the real-world application and how future work can improve our current findings.

Our measurements are biased towards time, location, and network setup, e. g., we recorded all traces from a single position and in closed blocks with our experimental LTE network that is completely under our control.

The choice of conducting the website fingerprinting within our experimental network has two main reasons. First, the configuration of mobile networks is volatile, e.g., features like the physical cell ID or retransmission timers might change over time. Such fluctuations can influence the experimental results and disrupt their reproducibility. A real-world attacker must face short-term and long-term changes of the network configuration and in website contents, i. e., a representative trace corpus requires continuous updates. Second, we are unable to monitor the uplink transmissions on the PDCP layer of a connection in a commercial network (see Section III-A3). Consequently, it would remain unclear whether such uplink metadata is a suitable candidate for website fingerprinting attacks. In contrast to the commercial setup, our experimental LTE network enables us to monitor transmissions also in uplink direction for a coherent evaluation of traffic features.

We use a closed-world setup and identify websites in a set of k candidates, which is very small in comparison to the actual number of existing websites. Open-world setups [20], [25] increase the realism and allow arbitrary page visits for a monitored set of k websites.

We limit the scope of this paper to a first demonstration of website fingerprinting on LTE traffic. While website fingerprinting in general is a well-established research area, the application to LTE traffic is novel. We limit our evaluation to the presented general proof-of-concept and leave the demonstration of the attack in a commercial network, along with the use of sophisticated experiments, to future work.

IV. aLTEr: LTE User Data Manipulation Attack

The lack of integrity protection for LTE user data opens an attack vector for active manipulation of the ciphertext. We exploit this vulnerability in the aLTEr attack, in which we deploy a malicious MitM relay between the UE and the eNodeB to manipulate the (encrypted) payload of user data transmissions. We instantiate aLTEr to perform a DNS redirection attack and describe the individual attack steps in the following.

A. High-level Overview of DNS Redirection Attack

Our goal is to manipulate the destination IP address of a DNS request and detour requests to a malicious rather than the original DNS server. Accordingly, this puts us in the position of redirecting the DNS requests to a server under adversarial control rather than the intended destination. The attack procedure is as follows (cf. Figure 4).

As a precondition for the attack, we deploy a malicious relay within the vicinity of the user and assure a stable radio connection towards both the UE and a commercial eNodeB. As soon as the user's mobile is switched on, the UE and the commercial network perform the Authentication and Key Agreement (AKA) (cf. Figure 4 ⓪) to establish the security parameters for an upcoming connection. Sending a DNS request to the server is triggered under many circumstances, e.g., when the user intends to visit a website or an app contacts a server. To perform a DNS request, the UE first encapsulates the request in a UDP and IP packet and then encrypts the packet using AES in counter mode (AES-CTR). Next, the UE forwards the packet to the intended DNS server, using its original IP destination address ①. Our MitM relay intercepts this transmission, distinguishes DNS packets from other payloads, and applies a manipulation mask to change the original destination IP to the address of our malicious DNS server ②. After the manipulation, our relay forwards the manipulated request (all other packets are relayed unaltered) to the commercial network ③, where it is decrypted and forwarded to the *malicious* instead of the original DNS server ④. In the downlink path, we add another manipulation mask and assure that the source IP address matches the target of the outgoing packet ⑤, such that the manipulation remains undetected.

B. Challenges

While the attack procedure is straightforward, we must consider a set of technical challenges to maintain a stable connection and remain undetected during the attack procedure. In particular, we must assure a connection between the UE, malicious relay, and the commercial eNodeB (IV-B1), reliably distinguish DNS packets from other transmissions (IV-B2), and alter the destination IP without violating the existing checksums of packets (IV-B3).

1) Stable Malicious Relay: Our malicious relay is of fundamental importance for the aLTEr attack. It impersonates a valid eNodeB towards the user and acts as a UE towards the network, i. e., it relays all transmissions between both entities. Deploying a MitM relay means to compete with all other radio connections offered by benign eNodeBs in the vicinity of the user. Therefore, we must motivate the UE to *connect to our relay* rather than the commercial network and provide a *stable and legitimate connection* during the entire attack.

Connecting to the Relay. One option to lure a user into connecting to the malicious relay is overshadowing the authentic frequencies of the commercial network at a *higher* transmission power. This approach holds the risk of letting the malicious relay connect to itself: As we remember, our relay impersonates a UE towards the network and an eNodeB towards the user. We avoid a connection between the UE and eNodeB component of our relay by using the physical cell identity of the commercial network, i. e., we use the physical cell identity of the commercial eNodeB to establish a fixed connection between our UE component and the commercial network.

Stable Radio Connection. For conducting a stealthy attack, our malicious relay must comply with all original protocol capabilities while passing on transmissions between the UE and the eNodeB. In particular, our relay needs to be aware of configuration parameters for the data bearer, the RLC, and the underlying physical layer, as otherwise, the connection would terminate. While the data bearer and RLC configuration remain stable for the network, we guess the parameters of the physical layer that are set for each new radio connection individually.

The idea behind individual guessing is as follows: After the Authentication and Key Agreement (AKA) took place between the UE and the commercial network, the security mode command defines the encryption and integrity protection algorithms for the new radio connection. The eNodeB component of our malicious relay then opens up all possible slots for uplink transmissions, waiting for the UE to use one of the potential slots. Based on the chosen slot, the malicious relay guesses the respective configuration parameter. We can apply the individual guessing for both physical parameters, e. g., the scheduling request index and the channel quality indicator. Both parameters use different uplink slots, and we monitor transmissions, respectively. If the value remains stable, we assume its correctness. We then notify the UE component of our relay about the parameters and set them for the uplink transmission to the commercial network.

2) Identifying DNS Requests and Responses: Since we only redirect DNS *requests* to our malicious DNS server, the destination IP addresses of all other packets must remain intact to maintain the Internet connection of the UE. Therefore, we need a reliable way to distinguish DNS request packets from

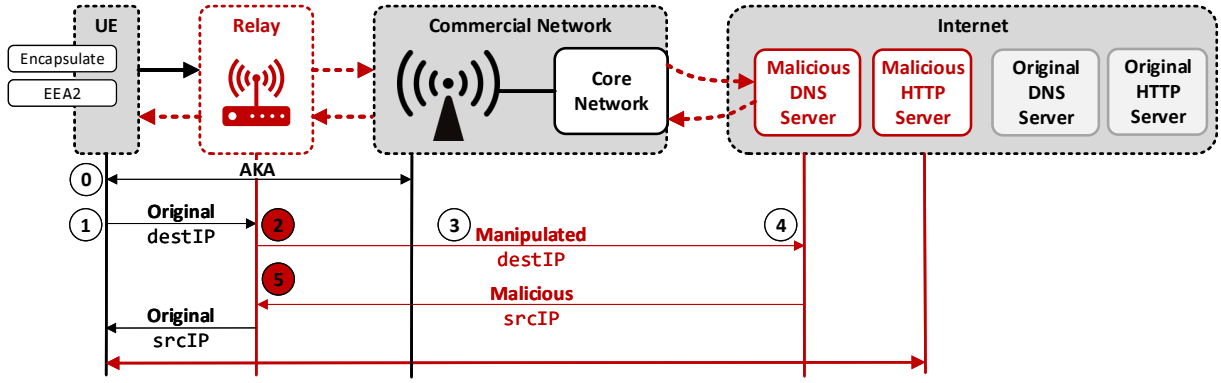

Fig. 4. ALTER: Overview of the DNS redirection attack. We deploy a malicious relay as a MitM between the UE and the commercial network and alter the destination IP address of a DNS request to redirect messages to our malicious DNS server. Eventually, the UE connects to the malicious HTTP server.

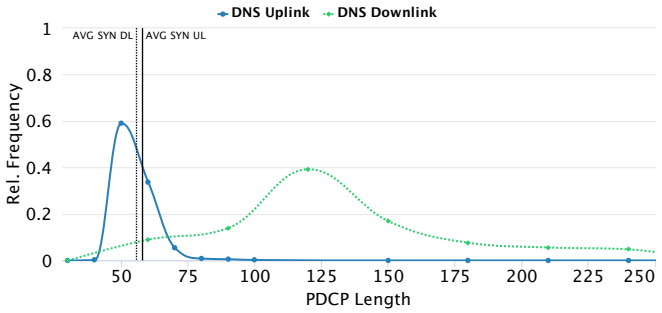

Fig. 5. PDCP lengths of DNS requests in comparison to average TCP SYN packets. We distinguish the relative frequencies of DNS requests in the uplink UL (solid lines) and downlink DL (dotted lines) and the average of SYN packets (vertical lines), respectively. The statistic is based on the corpus of traces for our website fingerprinting.

other transmissions through our relay. This is challenged by the fact that we receive encrypted data, i. e., we cannot identify DNS requests by their disclosed payload. We overcome this by identifying packets through their length: DNS packets are usually smaller than other TCP packets.

Using this simple classification method holds the risk of confusing DNS requests with TCP SYN packets of comparable length. We use our large corpus of website fingerprinting traces to analyze the frequency of DNS, TCP SYN, and all other types of packets in the up- and downlink stream (cf. Figure 5). In the downlink direction, the distribution of DNS packet lengths and the average TCP SYN length differ significantly and allow for a reliable distinction. This becomes more challenging in the uplink direction, therefore, we suggest an interactive approach for increasing the reliability of the decision.

Using the packet length as a filter, we separate approximately 96.21 % of other TCP packets from a set of 3.79 % of either TCP SYN or DNS packets. The relay then alters the destination IP address of the candidate packets and tests the response of the DNS server, i. e., if we receive a valid response the packet was a DNS request and we forward the altered packet. In all other cases, we forward an unaltered packet. We suggest this method for increasing the attack robustness but use fixed values for the demonstration of ALTER.

3) Packet Modification: Once we have identified a DNS packet, we alter the destination IP address for the redirection. We do this by applying a manipulation mask to the original IP and flip bits in a way that results in the malicious server address. In this manipulation, we must maintain the validity of the IP and UDP header. Packet headers are protected against transmission errors through a 16 bit checksum of the header elements [26]. We need to consider this protection mechanism when manipulating the destination IP address, as an invalid checksum results in discarding the IP packet at the first router on the transmission path [27].

Calculating the Manipulation Mask. LTE user data is encrypted in AES in counter mode, i. e., the sender computes the ciphertext c by XORing $\oplus$ the output of the encryption algorithm with the plaintext m [28]. However, the encryption algorithm is malleable, and an adversary can modify a ciphertext into another ciphertext which later decrypts to a related plaintext.

In particular, an active attacker can add a manipulation $mask$ via $\oplus$ to the ciphertext c, with the goal of flipping certain bits in the message (see Figure 6). On the receiver side, the message is decrypted to obtain the plaintext m' by again XORing the manipulated ciphertext c' with the same output of the encryption algorithm. As a result, we can find the same bit flips as in the manipulation $mask$ when inspecting the manipulated plaintext m'. For performing the manipulation in a precise manner, the original plaintext m must be known to the attacker to compute the manipulation mask:

$$mask = m \oplus m'. \tag{4}$$

The mask is flexible in a sense that it does not necessarily cover the whole ciphertext c, but can be restricted to the destination IP field. We know the exact offset to the IP address, as the IP header is embedded at the beginning of the PDCP frame. Therefore, we can apply the manipulation mask without causing collateral damage in other parts of the payload and

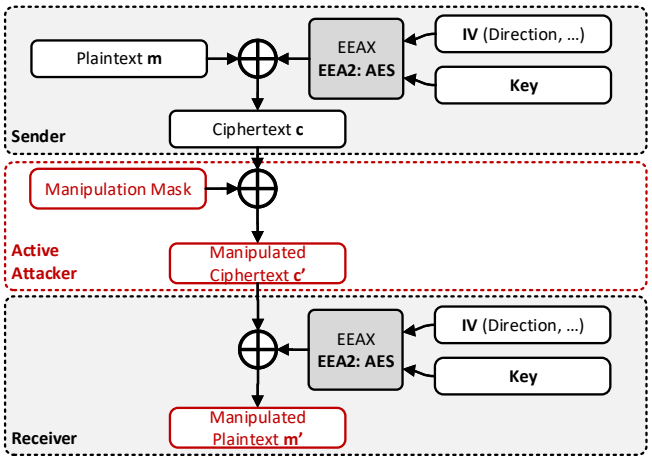

Fig. 6. Overview of ALTER attack: We manipulate the destination IP address of a DNS request using a specific manipulation mask. While maintaining the header checksums of the packet, the manipulated plaintext m' leads to a redirection of the packet.

keep changes in the original message to a minimum. For setting a specific new destination IP address, we benefit from the fact that IP addresses of DNS servers in mobile networks are set by the *core network*, i. e., we can easily obtain the static address of the provider's default DNS server.

Compensation for Changes. Applying the manipulation mask results in bit flips within the ciphertext of the packet. Even though we know where to find the IP address field and can determine an exact mask for the desired address update, this still results in changes of the original payload. Consequently, we compromise the validity of any checksum in the packet and cause a drop of the packet during the transmission. If we restrict our bit manipulation to the target IP, only addresses of the same 16-bit sum as the original DNS resolver are valid candidates.

We can circumvent this when all modifications made to the header *sum up to zero*, i. e., when changing additional bits besides the target IP address, we can restore the original checksum and assure its validity. Having these options for compensation, we gain more degrees of freedom in setting the malicious destination IP address. In the following, we introduce the necessary steps for compensating the IP and UDP header checksum through additional bit manipulations.

IP Header Checksum. We benefit from the fact that, besides the destination IP address field, all other *non-routing* fields in the IP header are open to modifications as long as we can predict or know their contents. A good candidate for compensation in the uplink is the Time To Live (TTL) field, as we can determine the value and a modification has only minor influence on the routing. We can obtain the default value for the UE's TTL by empirical analyses or by analyzing the operating system of the mobile phone. We know that the TTL is not decremented, when we are manipulating the packet before the first router, hence, we know the exact value. Adjusting the TTL field is already sufficient to achieve a valid checksum. The target IP address must fulfill the following requirements:

The 16-bit sum of the original IP address, represented by its octet, e. g., `ip_a.ip_b.ip_c.ip_d`, must equal the sum of the target IP plus the TTL field (cf. Figure 7). In this case, the checksum remains valid even though the IP address and the TTL are manipulated. The TTL field can be incremented or decremented. We need to ensure that packets with a decremented TTL can still reach the malicious server within the remaining time until the hop limit is reached.

In the downlink direction, the exact value of the TTL field is unknown, since it depends on the number of hops that were traversed previously to reaching our malicious relay. This prevents us from altering the TTL field in a deterministic way. Rather than manipulating the TTL field, we exploit the identification field of the IP packet. This field is used for the fragmentation of IP packets and is a 16 bit value. Since we are in control of the malicious DNS server, we set the identification field of the IP packet to a predetermined value. Manipulating the source IP address in downlink direction at the relay, we can now use the identification field to compensate any differences to the original IP address. Consequently, the IP header checksum remains valid on the downlink path and the UE accepts the packet.

We emphasize that the above limitation only applies for IPv4 transmissions, as IPv6 transmissions do not use any header checksums. Consequently, we do not face any limitations in the choice of the target host for IPv6 and the attack can be performed without restrictions.

UDP Header Checksum. Similar to the IP header checksum, altering the IP address also affects the UDP checksum that is a 16 bit sum over the IP pseudo header and UDP payload [29]. While running the malicious servers helps to *ignore* checksums in the uplink direction, we must assure a successful checksum validation in downlink direction for the UE to accept the DNS response. For the downlink direction, we benefit from the fact that UDP checksums set zero should be ignored by the UDP stack [29]. Simply setting the UDP checksum of the DNS response to zero circumvents the checksum validation and the DNS response remains valid, even in cases where the IP source address is modified.

C. Experiments

We demonstrate the feasibility of ALTER in a realistic setup using a commercial network, phone, and SIM card. In the following, we describe the experimental setup including details

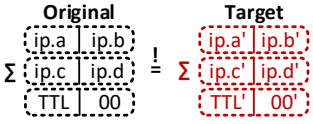

Fig. 7. Manipulations to the IP address must sum up to zero for maintaining valid checksums. We can modify additional non-routing fields to gain more degrees of freedom for the address manipulation.

of the malicious relay. In our demo exploit, we redirect a benign DNS request for the domain `example.org` to a DNS server under our control, which then replies with a malicious IP address. The technical setup and experimental results are as follows.

1) Setup: We use the following components for our experimental setup, as depicted in Figure 9.

- **UE.** We use a COTS mobile phone (LG Nexus 5) isolated from outside radio connections using a shielding box with a commercial SIM card, capable of connecting to a legitimate real-world LTE network. To operate the UE in a deterministic way within the shielding box, we use the Android Debug Bridge (ADB). Furthermore, we use SIMtrace [30] to extract the session key so that we can later analyze the traces recorded by the malicious relay.
- **Malicious Relay.** Our malicious relay consists of two SDRs with a modified version of the srsLTE `v17.09` stack implementation [15]. The first SDR emulates an eNodeB towards the UE, while the second SDR emulates the UE towards the commercial network.
- **Commercial eNodeB and Network.** We connect to a commercial network and use a SIM card according to the analyzed commercial network.
- **Malicious DNS Server.** To operate our rogue DNS server, we use a virtual Ubuntu `v16.04` server entity in the Amazon AWS cloud running a DNS server. We use a modified configuration of the DNS server for redirecting requests to the malicious HTTP server. The DNS server can be reached via an IP address matching the requirements described in Figure 7.
- **Malicious HTTP Server.** The rogue HTTP server uses the same Amazon AWS instance as the malicious DNS server and hosts an Apache web server in standard configuration.

While the DNS and HTTP server function as proof-of-concept destinations for the redirection of DNS packets and do not depend on any characteristic configurations, the specification of the malicious relay is crucial for the ALTER attack. Our implementation of this relay is as follows.

Malicious Relay. Figure 8 depicts the architecture of the malicious relay based upon the open source srsLTE stack [15]. Towards the victim UE, the relay emulates a genuine eNodeB (left side) by broadcasting the identifiers for the corresponding network. This is represented by the eNodeB component of the relay. Towards the network, the relay acts as a UE (right side). Both components (eNodeB and UE) forward control plane and user plane messages in up- and downlink direction.

We leave the physical layer of the UE and the eNodeB component unaltered according to the original implementations of srsUE and srsENB, respectively. On the MAC layer of the eNodeB, we add a component for guessing the encrypted configuration parameters (scheduling requests, channel quality indicator), as introduced in Section IV-B1. The RLC layer simply passes messages to the PDCP layer, which then distinguishes user and control plane messages. We add a message guessing module on the RRC layer for the eNodeB and the UE

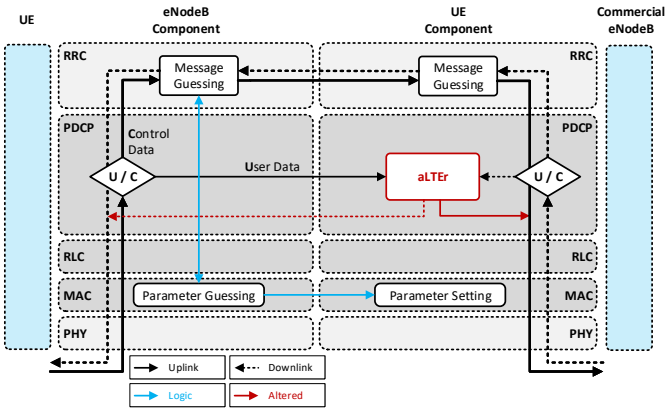

Fig. 8. Implementation of the malicious relay that we use for the ALTER attack. Message guessing and parameter guessing/setting are crucial for maintaining a correct protocol behavior and a stable connection. User data manipulation is applied on DNS requests.

component, where the first triggers the parameter guessing on MAC layer. The user data is simply forwarded on the PDCP layer and passed on the downlink and uplink into the ALTER component of the UE component. The ALTER component first distinguishes DNS traffic from other traffic and, second, modifies the message by applying the manipulation mask if needed. The ALTER component returns the modified and the unmodified packets to the normal packet path. Especially, packets in the uplink direction are sent to the network and packets in the downlink direction are sent to the UE. For our example, we use hard-coded values of the PDCP length for identifying the DNS requests of the domain `example.org` to our malicious DNS server.

D. Results

For the preparation of the experiments, we set the victim phone into flight mode, delete all caches (DNS and HTTP) via an ADB command, and place it in the shielding box. After starting the malicious relay, we disable the flight mode and wait for the successful radio layer connection to our relay. From then on, our malicious relay forwards all messages on

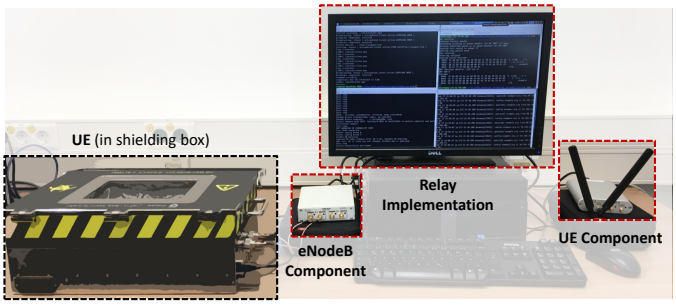

Fig. 9. Experimental lab setup. We use a shielding box for enforcing the UE's connection to the malicious relay; the eNodeB and UE components are deployed in two SDRs; the relay implementation runs on an Ubuntu 17.10 with Intel Core i7-7700.

the RRC layer and PDCP layer to and from the commercial network.

Over an ADB command, we instruct the phone to visit the website `example.org`. The following transmissions trigger the message classifier in our malicious relay and we identify a DNS request according to its PDCP length. In the next step, we apply the manipulation mask for replacing the original DNS server address with the malicious destination and emit the altered packet. Consequently, the DNS request is redirected to our rogue DNS server, which accepts the request despite its invalid UDP checksum. The malicious DNS server performs the DNS spoofing attack and responds with the wrong IP address for `example.org`. On the downlink, we identify the DNS response and apply the manipulation mask to change the source IP address, thus it matches the original IP of the DNS server. Finally, the phone receives the reply packet and connects to the spoofed IP address to perform a `HTTP GET` request, resulting in loading the wrong website content. Further details and results of the attack are provided at the website http://www.alter-attack.net.

E. Discussion

ALTER exploits the specification flaw of missing integrity protection of user data and has consequences for *all* LTE users. In the following, we discuss the real-world applicability of ALTER and possible countermeasures.

Real-World Application. We have demonstrated the feasibility of ALTER using a controlled experimental lab setup. We use a shielding box to prevent our relay from interfering with the commercial network in the licensed spectrum, following ethics guidelines. Further, the shielding box stabilizes the UE's radio connection and prevents non-deterministic behavior of the relay. In other words, the shielding box setup assures that the UE does not connect to any other available cell and the malicious relay does not interfere with itself. While we use this to simplify the experimental procedure, the setup is comparable to IMSI catcher attacks when considering the victim's perspective. Such attacks were conducted successfully in real-world environments, i. e., without shielding equipment [3], [5].

Furthermore, the DNS redirection attack is limited to plain IP traffic. All security measures taken by upper layer protocols cannot be circumvented, e. g., the proper use of DNSSEC or TLS assures the authenticity of the requested server. While DNS spoofing attacks are well-known in different contexts, e. g., DNS spoofing on the Internet depends on the adversarial control of one router, we emphasize the impact of an LTE instantiation. We argue it is even easier to conduct the attack because the accessible radio link is fundamentally more vulnerable to interception than other media [31].

Detection Methods. We discuss two perspectives for potential countermeasures: ALTER can either be detected on the UE side or within the commercial network.

As ALTER deploys a malicious relay on layer two, the general attack setup is comparable to classical rogue base station attacks. However, such attacks are detectable through incorrect protocol behavior, e. g., rogue base stations enforce the downgrade to insecure mobile generations [4]. In contrast, the malicious relay of ALTER forwards *all* messages between the UE and the benign eNodeB. Hence, the proper functioning of all protocols (including the correct integrity protection of control messages) is assured at all times. Consequently, the transmission behavior is as expected and the attack cannot be detected due to protocol anomalies. We argue that the relay integrates—to the best of our knowledge—in a non-detectable way into the existing network infrastructure.

While the malicious relay acts according to the specification on the radio layer, our alteration of destination IP addresses might induce anomalies in overlying levels of the network stack. In particular, our injected addresses differ from common DNS servers. One possible way of detection would be the use of Deep Packet Inspection (DPI), even though this also holds the risk for false positive detections since a user might have set a custom DNS server.

Potential Countermeasure. Even though the LTE Authentication and Key Agreement (AKA) is formally proven secure [32], this attack is still possible due to the lack of integrity protection of user plane data. We argue that the only way to mitigate this attack sustainably is to use authenticated encryption for the user plane. While different suitable schemes exist like AES-GCM (AES-Galois/Counter Mode), we focus on the MAC-then-Encrypt scheme that is already used for the integrity protection of the control plane. We assume that this scheme has the highest potential for being adopted in the specification.

In prior decisions, this was neglected in the specification process due to the additional overhead on the radio layer [13]. The considered worst case scenario assumes small packet lengths of 45 byte on average, the corresponding 4 byte Message Authentication Code (MAC) would, therefore, lead to an overhead of $8.9\,\%$.

Our empirical measurements conducted in the context of the website fingerprinting attack reveal an overhead of $0.63\,\%$ for an average packet length of 634.15 byte for over 18 billion packets. In practice, packet lengths hence seem to differ significantly from the above assumption for the use case of web browsing. The overhead for integrity protection seems to be acceptable when considering the security and privacy impact of ALTER. In the light of the next mobile generation, we hope that we can influence the specification process to add mandatory user plane integrity protection to 5G.

Disclosure Process and Integrity Protection in 5G. As stated before, we have contacted the GSMA following the guidelines of responsible disclosure. The GSMA informed the network providers and issued a liaison statement to inform the 3GPP specification body about the problem [33]. The 3GPP security group evaluated possible actions for LTE and the upcoming 5G specification and composed a statement regarding the attack [34], [35].

The security group "feels that 5G standalone security architecture is in reasonable shape in respect of this attack,

but early implementations may have limited support for UP integrity." [35]. More precisely, the 5G specifies user plane integrity protection as optional [36]. However, for a successful protection against ALTER, the network needs to be configured correctly *and* the UE must support it. We argue that only mandatory integrity protection in 5G is a sustainable countermeasure.

V. Related Work

In the following, we discuss related work in the context of identity mapping attacks, website fingerprinting, and user data manipulation attacks.

A. Identity Mapping

Prior attacks in the context of identity mapping either link the user's *TMSI* to public identities like phone numbers or decode the more volatile *RNTI* of a session. Learning such individual identities enables an adversary to track and localize users within a cell, harming especially their privacy.

TMSI Linking. Paging attacks exploit the *broadcast* wake-up procedure of mobile networks towards idle user devices. Such broadcasts include the individual TMSI of a user, they can be eavesdropped easily, and actively triggering the procedure helps the attacker to learn sensitive information. Kune et al. [16] presented a paging attack in the context of GSM, where the attacker learns the user's TMSI from repeatedly calling the known phone number. The calls trigger the transmission of the TMSI and the attacker can recognize the repeated occurrence of one TMSI. Shaik et al. [3] port the paging attack to LTE and exploit Facebook and Whatsapp typing notifications rather than the phone number as a trigger for the paging procedure. One potential countermeasure against paging attacks is a frequent TMSI reallocation. While this reallocation should protect from the identification and location of users, Hong et al. [17] showed that lack of randomness in the reallocation scheme renders this countermeasure insufficient.

The work above focuses solely on the TMSI, which is an upper layer identifier. In contrast, we map the radio layer identity (RNTI) to the TMSI and, therefore, let the identity mapping attack serve as a stepping stone for follow-up attacks.

RNTI Decoding. While TMSIs can be exploited for the identification and localization of users, RNTI decoding by now was only proposed in the context of performance and interference optimizations. Kumar et al. [37] showed that they could passively decode the RNTI, map it to radio resource allocations, and locate a phone by using radar techniques for optimizing the LTE radio layer. Commercial LTE downlink sniffers [38], [39] are capable of decoding a list of all active RNTIs and monitoring the downlink traffic. Bui et al. [40], [41] presented an open source downlink sniffer also based on the srsLTE stack. While those approaches are technically comparable to ours, our contribution focuses on showing the vulnerability of the LTE downlink traffic.

Most similar to the presented attack is the work by Jover [4] in which the author describes the possibility of mapping a phone number or TMSI to an RNTI. In compaarison to this work, we identify the following differences. First, we have demonstrated the attack on a commercial network. Further, we found out that a simple *downlink* sniffer is in $91.75\,\%$ the cases sufficient to map the more volatile RNTI with a TMSI. Also, we cannot only identify and localize users within a cell but use the scheduling information of the mapped RNTI as a starting point for the website fingerprinting attack.

B. Website Fingerprinting

Website fingerprinting attacks are especially known from anonymity networks such as Tor, where the attacker learns the destination of connections through Tor from analyzing encrypted user traffic. Recent attacks utilize Naive Bayes classifiers [42] or Support Vector Machines [20], [43] and achieve high classification success rates, especially for closed-world scenarios. While website fingerprinting on Tor traffic is a well-established research field, we are the first to present a comparable attack on radio layer LTE traffic. Consequently, we provide the first proof of concept in a closed-world scenario and leave more sophisticated setups [10] to future work.

Furthermore, traffic analysis attacks were analyzed in the context of wireless sensor networks, where traffic patterns might leak the geographical locations of nodes in the networks. Attackers can exploit this information for launching attacks against base stations of the networks [44]. Countermeasures against traffic analysis attacks comprise network coding and homomorphic encryption [45], random path selection [46], or classical countermeasures like mixing and dummy packet injection [47].

In our website fingerprinting attack, we exploit the PDCP lengths using dynamic time warping. Classical countermeasures like dummy packet injection and mixing would induce an enormous performance overhead, as they add a high rate of additional traffic or add artificial delays to a transmission. Encryption is applied in LTE, but it does not obfuscate the PDCP lengths meta information that we exploit in our attack.

C. ALTER: User Data Manipulation

The challenges for conducting the user data manipulation attack are related to three individual research areas. First, we depend on a malicious relay, e. g., acting as a rogue base station towards the user. Second, our relay acts as an unauthenticated user towards the commercial network. Finally, we break the confidentiality aim of LTE as we are able to eavesdrop DNS requests and following connections.

Attacking the User. Rogue Base Stations simulate a benign network and try to lure a victim into its cell, e. g., for deploying an IMSI catcher. Such IMSI catchers help to learn the long-term identifier of a user, perform a Man-in-the-Middle attack, and localize the user's phone within the cell. In the context of LTE, Mjølsnes et al. [5] demonstrated how to build a rogue base station using existing open-source software stacks and performed an IMSI catching attack. Nevertheless, LTE offers mutual authentication and prevents the UE from continuing the connection to a malicious node after the authentication procedure was performed. Hussain et al. [48] describe the

possibility of an authentication relay attack, in which the AKA procedure is relayed between a commercial phone and network. Similar to the authentication relay attack, the presented malicious LTE Man-in-the-Middle relays the LTE AKA messages in a first step to establish mutual authentication between the commercial phone and network. Another way of deploying a MitM was presented by Rupprecht et al. [49], where an implementation flaw of the baseband let the UE connect to a malicious network despite mutual authentication.

The activity of rogue base stations can be detected through dedicated static or mobile sensor networks [11], [12], [50]. Rogue base station detection apps, like Snoopsnitch [51], are unable to identify certain attacks, as the baseband hides crucial information for the detection [52].

The malicious relay in our aLTEr attack differs from conventional rogue base stations in one fundamental characteristic: As we relay *all* messages except for DNS requests, the relay does not interfere with any protocol and a stable connection is maintained during the attack.

Attacking the Network. In contrast to the use of rogue base stations, attacks can also target the LTE network itself. One example for this is the circumvention of a provider's billing mechanism, where the attacker sends malicious data to the network, e. g., by performing IP spoofing [53]. Other attacks emphasize the unreliability of the VoLTE billing mechanism and vulnerabilities in its routing mechanisms [54], [55]. Both classes of attacks depend on a successful authentication towards the LTE network and only interfere with the IP layer and above; however, these limitations do not apply to the set of layer two attacks presented in this work. Other active attacks exploit the pre-authentication traffic towards the network and deny the service for a victim. In particular, Raza et al. describe an attack allowing an attacker to detach a victim from the network as soon as he knows the user identity [56]. We do not depend on a similar exploit of pre-authentication traffic, as we successfully relay all layer two messages of the original transmission.

Eavesdropping. Mobile networks, and GSM in particular, are subject to passive attacks on weak encryption algorithms. Ciphertext-only attacks [57]–[59] enable an attacker to break standard algorithms like A5/1 and A5/2 within a few minutes, just using ordinary hardware and rainbow tables [60]. As a consequence, the attacker can eavesdrop the communication.

In the context of our user data manipulation attack we do not exploit any weaknesses in the cryptographic algorithms of LTE but benefit from the malleability of the cipher, e. g., we perform a chosen-ciphertext attack. This approach has a neglectable overhead and allows to break the data link layer security despite the presence of state-of-the-art encryption.

VI. Conclusion

While lots of research effort in LTE security focuses on the physical and network layers, the data link layer has remained unexplored until now. We present a comprehensive layer two security analysis and reveal open attack vectors. More specifically, we presented three individual attacks on the data link layer of LTE. The identity mapping attack passively matches two temporary identifiers, reveals the location and the radio layer identity of users within the mobile cell, and thereby serves as a starting point for further attacks. One example for this is the website fingerprinting attack, in which we exploit the scheduling information for resource allocation in LTE. On the basis of unencrypted metadata information, we demonstrate how an adversary can derive the accessed websites with severe privacy implications for the user. Finally, we present the user data manipulation attack aLTEr. We perform a chosen-ciphertext attack by deploying a malicious relay and exploiting the missing integrity protection of LTE user data. As a result, we can redirect DNS requests and spoof the DNS responses. We demonstrate the real-world feasibility of all three attacks in realistic setups.

Based on our findings, we urgently demand the implementation of effective countermeasures in the upcoming 5G specification to assure the security and privacy of future mobile communication.

Acknowledgment

This work was supported by the Franco-German BERCOM Project (FKZ: 13N13741) co-funded by the German Federal Ministry of Education and Research (BMBF). In addition, this work was supported in part by Intel (ICRI-CARS). We would like to thank G Data Software AG for supporting our experiments with the shielding box and Software Radio Systems for giving us insights into their LTE software stack. Further, we thank our shepherd Michael Bailey for the guidance towards the camera-ready version.

Acronyms

3GPP	3rd Generation Partnership Project
ADB	Android Debug Bridge
AKA	Authentication and Key Agreement
C-RNTI	Cell Radio Network Temporary Identity
COTS	Commercial Off-The-Shelf
DCI	Downlink Control Information
EEA	EPS Encryption Algorithm
eNodeB	Evolved NodeB
EPC	Evolved Packet Core
GUTI	Globally Unique Temporary Identity
GSM	Global System for Mobile Communications
GSMA	GSM Association
IMSI	International Mobile Subscriber Identity
LTE	Long Term Evolution
MAC	Medium Access Control
MitM	Man-in-the-Middle
NAS	Non-Access Stratum
PDCP	Packet Data Convergence Protocol
RA-RNTI	Random Access RNTI
RAND	Random Number
RAP	Random Access Preamble
RAR	Random Access Response
RLC	Radio Link Control
RNTI	Radio Network Temporary Identity
RRC	Radio Resource Control
TTL	Time To Live
TMSI	Temporary Mobile Subscriber Identity
SDR	Software Defined Radio
UE	User Equipment

REFERENCES

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

[2] FirstNet, "FirstNet: First Responder Network Authority," http://www.firstnet.gov/, [Online; accessed 1-June-2018].

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

[4] R. P. Jover, "LTE Security, Protocol Exploits and Location Tracking Experimentation with Low-Cost Software Radio," *CoRR*, vol. abs/1607.05171, 2016. [Online]. Available: http://arxiv.org/abs/1607.05171

[5] S. F. Mjølsnes and R. F. Olimid, "Easy 4G/LTE IMSI Catchers for Non-Programmers," in *Mathematical Methods, Models, and Architectures for Computer Network Security (MMM-ACNS)*. Springer, 2017, pp. 235–246.

[6] M. Lichtman, J. H. Reed, T. C. Clancy, and M. Norton, "Vulnerability of LTE to Hostile Interference," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2013, pp. 285–288.

[7] M. Lichtman, R. P. Jover, M. Labib, R. Rao, V. Marojevic, and J. H. Reed, "LTE/LTE-A Jamming, Spoofing, and Sniffing: Threat Assessment and Mitigation," *IEEE Communications Magazine*, vol. 54, no. 4, pp. 54–61, 2016.

[8] F. M. Aziz, J. S. Shamma, and G. L. Stüber, "Resilience of LTE Networks Against Smart Jamming Attacks: Wideband Model," in *International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2015, pp. 1344–1348.

[9] R. P. Jover, "Security Attacks Against the Availability of LTE Mobility Networks: Overview and Research Directions," in *Symposium on Wireless Personal Multimedia Communications (WPMC)*. IEEE, 2013.

[10] M. Juarez, S. Afroz, G. Acar, C. Diaz, and R. Greenstadt, "A Critical Evaluation of Website Fingerprinting Attacks," in *ACM Conference on Computer and Communications Security (CCS)*. ACM, 2014.

[11] A. Dabrowski, N. Pianta, T. Klepp, M. Mulazzani, and E. Weippl, "IMSI-Catch Me If You Can: IMSI-Catcher-Catchers," in *ACM Annual Computer Security Applications Conference (ACSAC)*. ACM, 2014, pp. 246–255.

[12] P. Ney, I. Smith, G. Cadamuro, and T. Kohno, "SeaGlass: Enabling City-wide IMSI-Catcher Detection," *Privacy Enhancing Technologies (PETS)*, vol. 2017, no. 3, pp. 39–56, 2017.

[13] 3GPP, "Rationale and track of security decisions in Long Term Evolution (LTE) RAN / 3GPP System Architecture Evolution (SAE)," 3rd Generation Partnership Project (3GPP), TR 33.821, 06 2009. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/33821.htm

[14] "Ettus Research USRP B210," https://www.ettus.com/product/details/UB210-KIT, [Online; accessed 1-June-2018].

[15] "Open Source SDR LTE Software Suite," https://github.com/srsLTE/srsLTE, [Online; accessed 1-June-2018].

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

[17] B. Hong, S. Bae, and Y. Kim, "GUTI Reallocation Demystified: Cellular Location Tracking with Changing Temporary Identifier," in *Symposium on Network and Distributed System Security (NDSS)*. ISOC, 2018.

[18] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification," 3rd Generation Partnership Project (3GPP), TS 36.321, 06 2010. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/36321.htm

[19] "PCSC Lite Project - Middleware to Access a Smart Card using SCard API (PC/SC)." https://pcsclite.apdu.fr/, [Online; accessed 1-June-2018].

[20] T. Wang and I. Goldberg, "Improved Website Fingerprinting on Tor," in *Workshop on Privacy in the Electronic Society (WPES)*. ACM, 2013.

[21] "OpenAirInterface (OAI) - 5G Software Alliance for Democratising Wireless Innovation," http://www.openairinterface.org/, [Online; accessed 1-June-2018].

[22] "Appium: Mobile App Automation Made Awesome," http://appium.io/, [Online; accessed 1-June-2018].

[23] S. Salvador and P. Chan, "Toward Accurate Dynamic Time Warping in Linear Time and Space," *Intelligent Data Analysis*, vol. 11, no. 5, pp. 561–580, 2007.

[24] T. Mitsa, *Temporal Data Mining*. Chapman & Hall/CRC, 2010.

[25] X. Cai, X. C. Zhang, B. Joshi, and R. Johnson, "Touching from a Distance: Website Fingerprinting Attacks and Defenses," in *ACM Conference on Computer and Communications Security (CCS)*. ACM, 2012, pp. 605–616.

[26] J. Postel, "Internet Protocol," Internet Requests for Comments, RFC Editor, STD 5, September 1981, http://www.rfc-editor.org/rfc/rfc791.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc791.txt

[27] F. Baker, "Requirements for IP Version 4 Routers," Internet Requests for Comments, RFC Editor, RFC 1812, June 1995.

[28] 3GPP, "3GPP System Architecture Evolution (SAE); Security architecture," 3rd Generation Partnership Project (3GPP), TS 33.401, 06 2011. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/33401.htm

[29] J. Postel, "User Datagram Protocol," Internet Requests for Comments, RFC Editor, STD 6, August 1980, http://www.rfc-editor.org/rfc/rfc768.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc768.txt

[30] "Osmocom SIMtrace," https://osmocom.org/projects/simtrace/wiki/SIMtrace, [Online; accessed 1-June-2018].

[31] D. Rupprecht, A. Dabrowski, T. Holz, E. R. Weippl, and C. Pöpper, "On Security Research towards Future Mobile Network Generations," *CoRR*, vol. abs/1710.08932, 2017. [Online]. Available: http://arxiv.org/abs/1710.08932

[32] S. Alt, P.-A. Fouque, G. Macario-rat, C. Onete, and B. Richard, "A Cryptographic Analysis of UMTS/LTE AKA," in *Conference on Applied Cryptography and Network Security (ACNS)*. Springer, 2016, pp. 18–35.

[33] GSMA CVD Governance Team/Samantha Saad, "Liaison Statement: LTE and the upcoming 5G standard (S3-181429)," http://www.3gpp.org/ftp/TSG_SA/WG3_Security/TSGS3_91_Belgrade/Docs/S3-181429.zip, [Online; accessed 1-June-2018].

[34] 3GPP Security Group SA3, "Meeting Report 20 April 2018," http://www.3gpp.org/ftp/Meetings_3GPP_SYNC/SA3/Report/MeetingReport_20April.rtf, [Online; accessed 1-June-2018].

[35] Alf Zugenmaier (3GPP Security Group SA3), "Reply to LS on LTE and the upcoming 5G standard (S3-181443)," http://www.3gpp.org/ftp/TSG_SA/WG3_Security/TSGS3_91_Belgrade/Docs/S3-181443.zip, [Online; accessed 1-June-2018].

[36] 3GPP, "NR; Packet Data Convergence Protocol (PDCP) specification," 3rd Generation Partnership Project (3GPP), TS TS38.323, 2018. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/38323.htm

[37] S. Kumar, E. Hamed, D. Katabi, and L. Erran Li, "LTE Radio Analytics Made Easy and Accessible," in *ACM SIGCOMM Computer Communication Review (SIGCOMM)*. ACM, 2014, pp. 211–222.

[38] "Software Radio Systems - Airscope," http://www.softwareradiosystems.com/products/, 2018, [Online; accessed 1-June-2018].

[39] "Sanjole - WaveJudge4900A," http://www.sanjole.com/brochures-2/WaveJudge4900A-LTEHandout-Feb11-2012.pdf, 2018, [Online; accessed 1-June-2018].

[40] N. Bui and J. Widmer, "OWL: A Reliable Online Watcher for LTE Control Channel Measurements," in *Workshop on All Things Cellular: Operations, Applications and Challenges (ATC)*. ACM, 2016, pp. 25–30.

[41] N. Bui, "IMDEA's Online Watcher for LTE (OWL) Control Channel," https://git.networks.imdea.org/nicola_bui/imdeaowl, 2017, [Online; accessed 1-June-2018].

[42] D. Herrmann, R. Wendolsky, and H. Federrath, "Website Fingerprinting: Attacking Popular Privacy Enhancing Technologies with the Multinomial NaïVe-bayes Classifier," in *Workshop on Cloud Computing Security (CCSW)*. ACM, 2009, pp. 31–42.

[43] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel, "Website Fingerprinting in Onion Routing Based Anonymization Networks," in *Workshop on Privacy in the Electronic Society (WPES)*. ACM, 2011, pp. 103–114.

[44] A. Perrig, J. Stankovic, and D. Wagner, "Security in Wireless Sensor Networks," *Communications of the ACM*, vol. 47, no. 6, pp. 53–57, Jun. 2004.

[45] Y. Fan, Y. Jiang, H. Zhu, J. Chen, and X. S. Shen, "Network Coding Based Privacy Preservation Against Traffic Analysis in Multi-Hop Wireless Networks," *IEEE Transactions on Wireless Communications*, vol. 10, no. 3, pp. 834–843, March 2011.

[46] J. Deng, R. Han, and S. Mishra, "Countermeasures Against Traffic Analysis Attacks in Wireless Sensor Networks," in *Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM)*, 2005, pp. 113–126.

[47] X. Luo, X. Ji, and M. S. Park, "Information Science and Applications (ICISA)," in *Information Science and Applications*. IEEE, 2010.

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

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

[50] GSMK mbH, "New Security Systems to Protect Mobile Network Operators against Eavesdropping and Fraud," http://www.cryptophone.de/en/company/news/gsmk-debuts-new-security-systems-to-protect-mobile-network-operators-against-eavesdropping-and-fraud/, 2017, [Online; accessed 1-June-2018].

[51] Security Research Labs, "SnoopSnitch - Mobile Network Security Tests," https://opensource.srlabs.de/projects/snoopsnitch, 2014, [Online; accessed 1-June-2018].

[52] S. Park, A. Shaik, R. Borgaonkar, A. Martin, and J.-P. Seifert, "White-Stingray: Evaluating IMSI Catchers Detection Applications," in *USENIX Workshop on Offensive Technologies (WOOT)*. USENIX Association, 2017.

[53] C. Peng, C.-Y. Li, H. Wang, G.-H. Tu, and S. Lu, "Real Threats to Your Data Bills: Security Loopholes and Defenses in Mobile Data Charging," in *ACM Conference on Computer and Communications Security (CCS)*, 2014, pp. 727–738.

[54] C.-Y. Li, G.-H. Tu, S. Lu, X. Wang, C. Peng, Z. Yuan, Y. Li, S. Lu, and X. Wang, "Insecurity of Voice Solution VoLTE in LTE Mobile Networks," in *ACM Conference on Computer and Communications Security (CCS)*. ACM, 2015, pp. 316–327.

[55] H. Kim, D. Kim, M. Kwon, H. Han, Y. Jang, D. Han, T. Kim, and Y. Kim, "Breaking and Fixing VoLTE : Exploiting Hidden Data Channels and Misimplementations," in *ACM Conference on Computer and Communications Security (CCS)*. ACM, 2015, pp. 328–339.

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

[57] E. Barkan, E. Biham, and N. Keller, "Instant Ciphertext-only Cryptanalysis of GSM Encrypted Communication," *Journal of Cryptology*, vol. 21, no. 3, pp. 392–429, Aug. 2008.

[58] J. D. Golić, "Cryptanalysis of Alleged A5 Stream Cipher," in *Theory and Application of Cryptographic Techniques (EUROCRYPT)*. Springer, 1997, pp. 239–255.

[59] A. Biryukov, A. Shamir, and D. Wagner, "Real Time Cryptanalysis of A5/1 on a PC," in *Workshop on Fast Software Encryption (FSE)*. Springer, 2000.

[60] Security Research Labs, "Kraken: A5/1 Decryption Rainbow Tables," https://opensource.srlabs.de/projects/a51-decrypt, 2010, [Online; accessed 1-June-2018].