IMP4GT: IMPersonation Attacks in 4G NeTworks

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
NYU Abu Dhabi
christina.poepper@nyu.edu

***Abstract*—Long Term Evolution (LTE/4G) establishes mutual authentication with a provably secure Authentication and Key Agreement (AKA) protocol on layer three of the network stack. Permanent integrity protection of the control plane safeguards the traffic against manipulations. However, missing integrity protection of the user plane still allows an adversary to manipulate and redirect IP packets, as recently demonstrated.**

In this work, we introduce a novel cross-layer attack that exploits the existing vulnerability on layer two and extends it with an attack mechanism on layer three. More precisely, we take advantage of the default IP stack behavior of operating systems and show that combining it with the layer-two vulnerability allows an active attacker to impersonate a user towards the network and vice versa; we name these attacks IMP4GT (IMPersonation attacks in 4G neTworks). In contrast to a simple redirection attack as demonstrated in prior work, our attack dramatically extends the possible attack scenarios and thus emphasizes the need for user-plane integrity protection in mobile communication standards. The results of our work imply that providers can no longer rely on mutual authentication for billing, access control, and legal prosecution. On the other hand, users are exposed to any incoming IP connection as an adversary can bypass the provider's firewall. To demonstrate the practical impact of our attack, we conduct two IMP4GT attack variants in a live, commercial network, which—for the first time—completely break the mutual authentication aim of LTE on the user plane in a real-world setting.

I. Introduction

Long Term Evolution (LTE) is the latest widely deployed mobile communication standard and is used by hundreds of millions of people worldwide. The protocol offers high-speed Internet access and packet-based telephony services and has become an integral component of our daily communication. We fundamentally rely on the security of LTE for a variety of applications. The security goals of LTE include, amongst others, mutual authentication, traffic confidentiality, and location privacy; any attack vector undermining these security aims has far-reaching implications to the use of LTE as a communication medium.

In the context of mobile communication, mutual authentication is an important security aim since it ensures that both communication parties (i. e., the user equipment and the network) mutually verify their identities. As the wireless medium is accessible for everyone in the vicinity and identifiers can be easily forged, mutual authentication is essential for building trust between communication parties. Telecommunication providers rely on *user authentication* for accounting, authorization, and the association of data sessions to a legal person. The latter case is of particular importance in prosecution, in which a possible offender is accused of committing a crime via a mobile Internet connection. Additionally, users rely on *network authentication* for the confidentiality of their communication. One important example for missing network authentication is the second mobile network generation GSM (Global System for Mobile Communications): by faking the identity of a legitimate network, an attacker can impersonate the network in GSM and eavesdrop on the communication of the victim.

In contrast to earlier network generations, LTE establishes *mutual* authentication on layer three of the network stack using a provably secure Authentication and Key Agreement (AKA) protocol [6], [8]. Based on this protocol, subsequent encryption ensures the confidentiality of user and control data. Permanent integrity protection, however, is only applied to the control data. A recent study has revealed that missing integrity protection of the user plane on layer two allows to manipulate user data in a deterministic way [40]. More specifically, a layer-two attacker in a Man-in-the-Middle (MitM) position between the phone and the network can introduce undetectable bit flips due to malleable encryption and *redirect* traffic to another destination. While this attack demonstrates the potential consequences of traffic manipulation, it is solely limited to redirecting traffic to another destination.

In this work, we introduce a novel *cross-layer attack concept* that complements the known layer-two vulnerability (i. e., missing integrity protection on the user plane [40]) with exploiting the default IP stack behavior of operating systems on layer three. More precisely, we make use of the reflection mechanism of certain IP packets, which allows us to not only redirect user-plane traffic, but also to create an encryption and decryption oracle that enables an adversary to perform a *full impersonation* of the phone or network on the user plane. We call this concept IMP4GT (IMPersonation in 4G neTworks, pronounced [ˈimˌpæk(t)]). IMP4GT completely breaks the mutual authentication property for the user plane on layer three, as an attacker can send and receive arbitrary IP packets despite any encryption.

This attack has far-reaching consequences for providers and users. Providers can no longer assume that an IP connection originates from the user. Billing mechanisms can be triggered by an adversary, causing the exhaustion of data limits, and any access control or the providers' firewall can be bypassed. A possible impersonation also has consequences for legal pros-

Network and Distributed Systems Security (NDSS) Symposium 2020
23-26 February 2020, San Diego, CA, USA
ISBN 1-891562-61-4
https://dx.doi.org/10.14722/ndss.2020.24283
www.ndss-symposium.org

ecution, as an attacker can establish arbitrary IP connections associated with the victim's identity.

IMP4GT can be deployed in two variants: i) In the uplink impersonation variant, the attacker acts as a user towards the network; this variant can be used to establish a TCP/IP connection towards the Internet that is associated with the victim's identity. ii) In the downlink variant, the attacker impersonates the network and can establish a TCP/IP connection towards the phone. In doing so, the attacker circumvents the provider's firewall and can potentially use this connection for malware deployment or data exfiltration. In contrast to the layer-two redirection presented in earlier work, IMP4GT allows the attacker to not only manipulate the content of a connection, but adds substantially more degrees of freedom (e. g., establishing arbitrary network connections) to possible attack scenarios.

We are the first to combine the known layer-two vulnerability with a layer-three attack to extend the adversary's capabilities. This broader view on the problem of missing integrity protection leads to the discovery of new vulnerabilities that allow a full impersonation attack. In a series of empirical experiments, we provide a comprehensive view of the problem statement and explain the characteristics we make use of for IMP4GT. Furthermore, we show the real-world applicability of the uplink and downlink attacks in an actual commercial network. To this end, we demonstrate how an attacker can access a service site that should only be accessible for the victim and how an attacker can bypass the provider's firewall. The feasibility of such an impersonation reveals that the dimension of missing integrity protection is more far-reaching than previously assumed. We describe the analysis in a step-by-step manner for the uplink and downlink variants of IMP4GT. By performing the analysis and demonstrating the attack, we also aim at influencing the current 5G specification to mandate user plane integrity. In summary, we make the following three contributions:

- We introduce IMP4GT, an attack that exploits the missing integrity protection on layer two along with standard IP stack behavior on layer three. This cross-layer approach aggravates a prior redirection attack with the ability to perform a *full impersonation* on the user plane in both uplink and downlink direction.
- We provide a comprehensive series of experiments that enable us to understand the network characteristics we exploit for the IMP4GT attacks. In particular, we analyze the default IP stack behavior for two types of reflections, which allows us to build the encryption and decryption oracle for the impersonation attack.
- Finally, we successfully demonstrate full end-to-end implementations of the uplink and downlink variants of IMP4GT with a mobile phone in a commercial network. Furthermore, we discuss the implications of our attack for the current and upcoming mobile generations for both users and providers.

Responsible Disclosure. Following the guidelines of responsible disclosure, we informed providers and vendors about our findings through the GSMA's coordinated vulnerability disclosure program [18]. By that, we hope to influence the LTE and 5G specifications to add full rate, mandatory integrity protection.

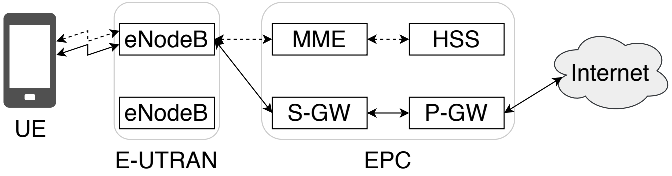

Fig. 1. Overview of an LTE network.

II. Preliminaries

We first provide an overview of the LTE network and relevant protocols, including the LTE protocol stack. Afterward, we introduce the security establishment in LTE and the IP stack's reflection mechanisms.

A. LTE Network

An LTE network comprises the User Equipments (UEs), the Evolved NodeBs (eNodeBs), and the Evolved Packet Core (EPC) network, which in turn consists of different entities (cf. Figure 1). In the following, we briefly introduce these network entities.

UE. The User Equipment (UE) is the user's communication device, such as a smartphone. It contains a SIM card that stores a shared key along with the permanent identity called International Mobile Subscriber Identity (IMSI). Furthermore, the UE consists of a modem that communicates with the network and thus abstracts the communication for an operating system. On top of that, the operating system implements the IP protocol along with transport-layer protocols, e. g., TCP or UDP, for applications.

eNodeB. The base stations in LTE are called eNodeBs. They manage the radio resources and encrypt and decrypt the user data. Usually, the UE selects the eNodeB with the highest signal strength. Fake base stations exploit this behavior to lure a victim into their cell [41]. In this work, we assume a MitM attacker with similar capabilities as a fake base station.

EPC. The LTE core network, called EPC, consists of multiple components. The Mobility Management Entity (MME) is responsible for the mobility management and user authentication. The Home Subscriber Server (HSS) stores the shared key and generates an authentication vector when the authentication is established. The Serving Gateway (S-GW) and Packet Data Network Gateway (P-GW) forward the user data to and from the Packet Data Network (PDN) and are responsible for accounting, authorization, and lawful interception. In most of the cases, the PDN is the Internet. It is also possible to connect private IP networks as PDN, e. g., company networks.

Most mobile network providers implement IPv4 *and* IPv6 [26]. In the case of IPv4, providers use Network Address Translation (NAT) at the P-GW to allocate an internal IP address for the UE [21]. For Internet communication, the P-GW maps the internal IP address to a public IP address. With the help of NAT, incoming packets are filtered in cases where the connection was not established from the internal network. For IPv6, a firewall at the P-GW protects the user from incoming traffic. Later on, we will show how the downlink variant of IMP4GT allows an adversary to circumvent the security mechanisms of the NAT gateway and firewall.

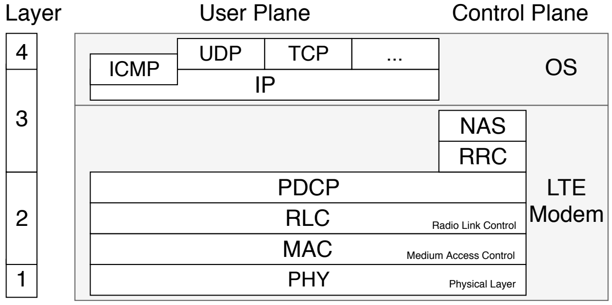

Fig. 2. Overview of the LTE and IP stack. Note that ICMP is a layer-three protocol, but ICMP packets are encapsulated in an IP packet similar to transport-layer packets.

B. LTE Protocol Stack

Figure 2 depicts the LTE protocol stack in cooperation with the IP stack. While the LTE stack is part of the LTE modem, the IP stack is implemented by the operating system. We briefly explain the protocols beginning with the PDCP protocol. In particular, we focus on the behavior of IP stacks, which is the mechanism exploited by the IMP4GT attack.

PDCP. The Packet Data Convergence Protocol (PDCP) transfers both user and control plane data. For the control plane, PDCP provides encryption and integrity protection. For the user plane, the protocol only provides encryption without any integrity protection, which leads to malleable encryption [40]. We exploit the missing integrity protection of the user plane for the IMP4GT attack.

RRC. The Radio Resource Control (RRC) protocol is part of the control plane and manages all radio connections between the UE and the eNodeB. This includes the configuration of all lower-level protocols down to the physical layer (PHY).

NAS. The Non-Access Stratum (NAS) protocol is responsible for mobility management with the core network. As part of the NAS protocol, the AKA establishes mutual authentication and a shared session key. Further security mechanisms on the NAS and PDCP layer build upon the established session key.

IP. The IP protocol allows to communicate with Internet services and is implemented by the operating system. Nowadays, most operating systems support both IPv4 and IPv6. An IP packet contains the transport-layer protocols, whose types are signaled by the IPv4 protocol field or the IPv6 next header field. The most common transport-layer protocols are the TCP and UDP protocols.

C. Security Establishment

The shared symmetric keys that are stored on the SIM card and in the HSS are the anchor for all security mechanisms in LTE. The keys are used during the AKA protocol to establish mutual authentication and to derive session keys for ongoing security mechanisms.

The AKA protocol takes place when the UE connects to the network. In this situation, the MME sends an authentication request to the UE that contains (a) an authentication token and (b) a random nonce. The authentication token verifies the authenticity of the network along with the shared key and a sequence number. The random nonce serves as a challenge for the UE and is used to derive session keys. The UE calculates a response and sends it back to the MME. The random nonce with the shared key is used by the MME and the UE to derive a session key, based on which the NAS and the RRC layer derives temporary key material for the ongoing security mechanisms. The EPC and eNodeB activate the security in the NAS and PDCP protocols with a *security mode command* and thus define the used security algorithms.

In LTE, the security algorithms for encryption and integrity protection are based on three basic ciphers: Snow3G, AES, and ZUC. Snow3G and ZUC are stream ciphers, while AES is a block cipher. The ciphers are used in a mode of operation for performing either integrity protection or encryption. For integrity protection, a Cipher Block Chaining Message Authentication Code (CBC-MAC) is calculated over the message and appended. For encryption, the ciphertext is computed by xor-ing the plaintext with a keystream. If the underlying cipher is already a stream cipher (i. e., Snow3G or ZUC), no further processing is required. In case of a block cipher (AES), the algorithm is turned into a stream cipher with counter mode. Each PDCP frame is encrypted with a separate keystream that is realized by increasing a counter as an input parameter for the cipher.

An active attacker can introduce bit flips to the ciphertext that are inherited to the plaintext—called malleable encryption. The aLTEr attack [40] exploits the malleable encryption of user data in LTE for a DNS spoofing attack, where the targeted manipulation of DNS requests allows to manipulate the destination IP address of DNS requests. For the IMP4GT attacks, we apply the aLTEr DNS spoofing as one of the building blocks of our attack to establish a cryptographic oracle. In addition to the previously introduced traffic redirection, an adversary can impersonate a user towards the network and vice versa.

D. Unreachable and Ping Reflection

The Internet Control Message Protocol (ICMP) maintains an IP connection by exchanging additional information or error messages. How systems support and handle ICMP and other protocol messages is defined by the stack implementation of the operating system. Part of this protocol is the reflection of messages, a mechanism which we exploit in our attack. In the following, we introduce two relevant reflection types and document their limitations for our attack scenario.

1) Reflection Types: One functionality of the ICMP protocol is the designated notification [20], [2] about lacking support of transport protocols in the operating system. This ICMP message is of type "destination unreachable/protocol unreachable" ($type$ = 3 / $code$ = 2) and contains a copy of the original incoming IP packet. We call this mechanism *unreachable reflection*. Another ICMP functionality is the echo (ping) mechanism that tests if a host is reachable. In response to an *echo request*, the ICMP stack sends an *echo reply* that copies also the payload of the request. We call this *ping reflection*.

While both mechanisms reflect the payload, they differ in the length, rate, and foreknowledge of the payload. This difference influences what type of reflection we use in distinct attack parts. The unreachable reflection is often limited in rate and size (cf. Section IV), but it does not rely on a checksum for the correctness of the unsupported transport protocol. On the other side, the ping reflection is not limited in rate and size, but the *echo request* has its own ICMP checksum that is checked by the operating system.

2) Limitations: The rate and size limitation of the unreachable reflection affects the attack performance and its use should be minimized as far as possible. However, it fits the situation in which the payload is unknown to the attacker and the inner checksum cannot be computed. In contrast, the ping reflection does not impair the attack performance, but is only suitable for situations in which the payload is known, i. e., the correct ICMP checksum can be computed. Consequently, we use the unreachable reflection for conditions where the payload is not known (decryption) and use the ping reflection in cases of known plaintext (encryption).

III. IMP4GT ATTACKS

The lack of integrity protection for user data allows to deterministically manipulate and redirect IP packets sent in uplink and downlink direction—this is how far the ALTER attack goes [40]. However, we can go further and exploit the missing integrity protection to establish an encryption and decryption oracle that allows to *inject arbitrary* packets and access the payload of *existing* packets. We achieve this through a cross-layer attack that takes the default IP stack behavior of mobile operating system further into account. In the end, the combination of both attack vectors allows us to perform a *full impersonation* towards the UE and the network.

In the following, we first explain the general concept of IMP4GT and then dive into more detail to document the preparation phase and the different attack phases of an uplink and downlink impersonation.

A. Attack Concept

For extending the ALTER attack to a full impersonation, we depend on the ability to encrypt and decrypt packets in uplink and downlink direction. Therefore, the construction of a cryptographic oracle is a core requirement for the IMP4GT attacks. We now provide an overview of the different phases of the attack procedure, document the steps of the oracle construction, and define the attacker model.

1) Phases: As a preliminary step, we first pass through a *preparation phase* which is followed by the actual *attack phase* (cf. Figure 3). The preparation phase aims to retrieve internal information of the victim's UE and to establish a connection to a plaintext generation server. In the attack phase, two variants of the impersonation attack can be conducted. The uplink impersonation allows an attacker to establish an arbitrary IP connection towards the Internet, e. g., a TCP connection to an HTTP server. With the downlink variant, the attacker can build a TCP connection to the UE.

Note that while the general attack procedure is rather simple, the notation of different traffic directions, encryption, and decryption in the following paragraphs can sometimes get a bit confusing. Hence we first provide an overview of the abstract idea before diving into details. Both attacks variants require to encrypt and decrypt packets for bidirectional communication: for the uplink impersonation, uplink packets need to be encrypted, and downlink packets need to be decrypted, whereas the downlink impersonation requires downlink packet encryption and uplink packet decryption. In both cases, the encryption and decryption is achieved by an oracle.

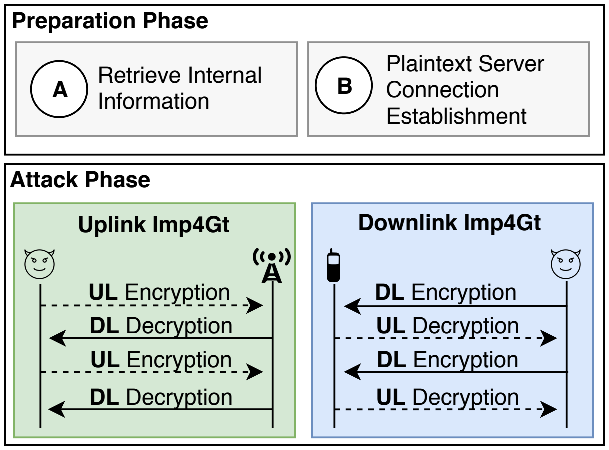

Fig. 3. IMP4GT attack concept. The preparation phase consists of two preliminary steps to Ⓐ derive internal information from the UE and to Ⓑ establish a connection with the plaintext server. After the preparation, either an *uplink* or a *downlink* impersonation can be performed.

2) Oracle Construction: In the following, we describe the abstract idea of the (a) encryption and (b) decryption oracle. For the sake of abstraction, we use the term *system* as a combination of LTE network entities. The realization of the oracles and the entities exploited are specific for the different attack variants described later in more detail.

(A) Encryption Oracle. The goal of an encryption oracle is to learn the keystream of a connection, which later allows to encrypt and inject arbitrary packets. Figure 4 depicts the encryption oracle for the IMP4GT attack. For encrypting a target plaintext, the oracle injects a known plaintext to the system ①. The system encrypts the packet by xor-ing the known-plaintext with a valid keystream for transmission, which is returned to the oracle ②. Now, the oracle can extract the valid keystream by xor-ing the known-plaintext on the encrypted packet. Any arbitrary payload can now be encrypted by xor-ing the target plaintext and the keystream ③.

(B) Decryption Oracle. The goal of a decryption oracle is to decrypt and access the payload of an encrypted packet ①. The high-level concept of the decryption oracle is depicted in Figure 5. To achieve the decryption of a packet, the oracle manipulates the to-be-decrypted ciphertext and sends it to the system ②. The system decrypts the packet *and* subsequently sends it back to oracle ③. This way, we can receive the plaintext of encrypted packets.

Both oracles vary in their implementation, i. e., in the used entities as *system* and mechanisms for the uplink and downlink impersonation. We document the technical details of the system, along with the exploited protocol properties, in the following.

3) Attacker Model: We consider an *active attacker* that has radio capabilities with full protocol knowledge, but does *not*

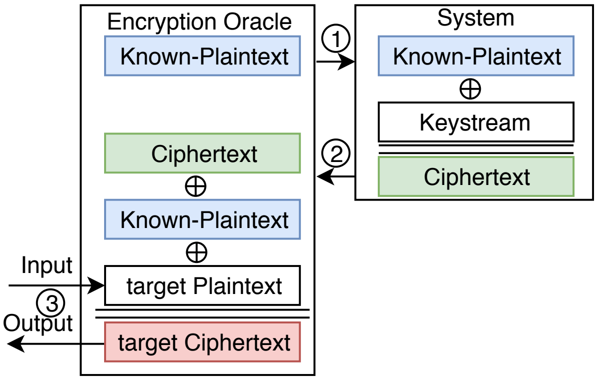

Fig. 4. The basic principle of the encryption oracle for the IMP4GT attack.

possess any key material or access to the core network. In particular, we analyze a layer-two attacker in a MitM position between the eNodeB and UE. In this position, the attacker can intercept, drop, and forward messages with unaltered or altered content. Furthermore, we assume that an attacker can deploy IP-based services on the Internet. In particular, the attacker deploys the following entities:

- **Relay.** The relay is in a MitM position between the UE and the network and forwards layer-two traffic between both entities. Because of this position, the relay can detect the length of a frame. Furthermore, the missing integrity protection allows modifying the content of encrypted layer-two frames.
- **DNS Server**. The DNS server is deployed on the Internet and is only active during the preparation phase. It performs DNS spoofing to redirect the request, and redirects the subsequent TCP connection to the TCP proxy.
- **TCP Proxy.** The TCP proxy is in a MitM position between the UE and the original TCP server and relays a TCP connection during the preparation phase. The TCP proxy allows to intercept and hijack the original TCP connection to inject additional packets into the connection with correct sequence numbers.
- **Decryption Server.** The decryption server receives decrypted packets and shares the information with other entities via the control connection. Those packets are encapsulated within an ICMP frame and need to be decapsulated by the decryption server. In the preparation phase, the decryption server receives internal information about the phone. During the attack phase, the server receives the decrypted TCP packets encapsulated in ICMP packets of the impersonated TCP connection.
- **Plaintext Generation Server.** The plaintext generation server generates a known plaintext and sends it in downlink direction to the UE. The relay uses the known plaintext for extracting the keystream and re-encrypting a crafted packet. We instantiate the plaintext generation server as a UDP server.

The attack requires to react on parameters set by the network for a new radio connection. For example, the victim's internal IP address influences the plaintext prediction that is performed at the relay. However, only the decryption server can access the information. Therefore, all entities share information via a separate control connection.

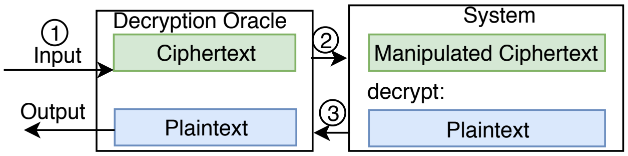

Fig. 5. The basic principle of the decryption oracle for the IMP4GT attack.

We consider the following initial situation: The victim connects to the attacker's relay, which can be achieved by increasing the signal strength of the relay or by jamming the legitimate cell [31]. Furthermore, the victim's UE requests the default DNS server of the network for a subsequent TCP connection. This situation can be triggered either by the user's action or automatically: In the former case, the victim visits a website or uses one of the installed applications; the latter situation occurs when background services periodically check for incoming data. We demonstrate the attack without requiring *any* specific action of the victim. In particular, we intercept the initial DNS request and the subsequent TCP connection that checks the Internet connectivity (e. g., on Android a connection to `connectivity.android.com` and on iOS a connection to `captive.apple.com`).

B. *Preparation Phase*

The preparation phase allows the attacker to Ⓐ learn mandatory internal information about the UE, e. g., the IP address and TCP port behind the NAT, and to Ⓑ connect to a plaintext generation server that later is required for maintaining a plausible connection (cf. Figure 6). Both steps Ⓐ and Ⓑ make use of a preliminary connection establishment to a malicious *TCP proxy* that allows to hijack the TCP connection. The attacker can use this hijacked TCP connection for sending additional packets to the UE, which is one requirement for the *unreachable reflection*, the subsequent information retrieval, and the keystream server connection. Next we describe each of these steps in more detail.

1) Initial DNS Request: The preparation phase begins when the UE requests the default DNS server, e. g., when the victim visits a website or the UE initially checks the Internet connectivity. As the malicious relay forwards all packets between the UE and the LTE network, it can detect DNS requests based on the packet lengths that differ from other types of traffic. Following the successful detection, the attacker performs the aLTEr attack (described in Sec. II-C) to alter the destination IP address of the DNS request accordingly. When the LTE network decrypts the manipulated request, it is redirected to the malicious DNS server.

2) Establishing TCP Proxy: With the malicious DNS server in charge of resolving the DNS request, the attacker performs DNS spoofing (1) and replies with the IP address of the TCP proxy. In the following, the UE establishes a TCP connection to the attacker's TCP proxy (2a), which connects to the original TCP server (2b). This allows the attacker to relay TCP connections and hijack the underlying TCP connection. More precisely, the attacker can inject additional TCP packets at the end of the TCP connection with the correct sequence numbers. In this way, the LTE network's firewall/NAT routes those packets to the UE. By injecting *two* additional TCP

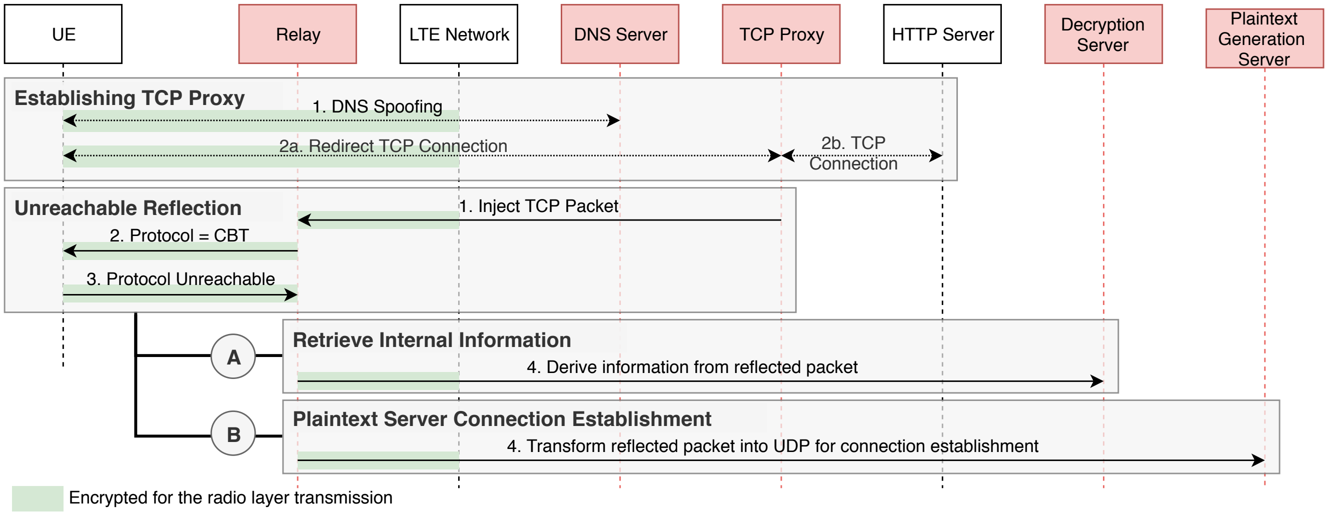
UE
Relay
LTE Network
DNS Server
TCP Proxy
HTTP Server
Decryption Server
Plaintext Generation Server
Establishing TCP Proxy
1. DNS Spoofing
2a. Redirect TCP Connection
2b. TCP Connection
Unreachable Reflection
1. Inject TCP Packet
2. Protocol = CBT
3. Protocol Unreachable
A
B
Retrieve Internal Information
4. Derive information from reflected packet
Plaintext Server Connection Establishment
4. Transform reflected packet into UDP for connection establishment
🟩 Encrypted for the radio layer transmission

Fig. 6. In the preparation phase, the attacker first hijacks a TCP connection with a TCP Proxy using a DNS spoofing attack via the ALTER attack. In the next step, the unreachable reflection helps to Ⓐ retrieve internal information about the UE and to Ⓑ establish a UDP connection to the Plaintext Generation Server.

packets, the attacker can Ⓐ extract internal information and Ⓑ set up the connection to the plaintext generation server. Exploiting the unreachable reflection mechanism is part of both steps.

3) Unreachable Reflection: For triggering the unreachable reflection, the TCP proxy injects a TCP packet with a known plaintext and known length (1). The network accepts the packet and replaces the IP and TCP port according to the NAT rules. The network encrypts the packet for the radio transmission to the UE. On the radio layer, the malicious relay intercepts the packet based on the known size and alters the IP protocol field (2) to forward it to the UE. The UE decrypts the packet and forwards it to the operating system. Due to the usage of an unsupported protocol, the packet is reflected with an ICMP message including the received IP packet. In the next step, the relay receives the reflected packet. From this point, we differentiate between the information retrieval step Ⓐ and the connection establishment to the plaintext generation server Ⓑ.

Ⓐ Retrieve Internal Information. In the information retrieval step, we derive three pieces of information that we later need in the attack: (i) the internal IP address and (ii) the TCP port behind the NAT, and (iii) the Time To Live (TTL) of the TCP proxy. While the internal IP address is required in each attack step, i. e., set as the source or destination address, the TCP port and the TTL are just used in the connection setup with the plaintext generation server that conducts the plaintext prediction.

We can derive this information from the first injected TCP packet at the end of the original TCP connection. As described above, the injected TCP packet gets reflected by the UE's IP stack *including* the original downlink IP packet. The reflected original IP packet provides the required information and is routed to the decryption server where we can access the information. During our experiments in a commercial network, we found that the provider's firewall does not allow ICMP packets with the type "destination unreachable/protocol not supported". This problem can be easily addressed: the relay changes the ICMP type to *echo reply* to pass the firewall.

Ⓑ Plaintext Server Connection Establishment. For establishing a connection to the plaintext generation server, we send a crafted packet towards our plaintext generation server. This packet creates a new NAT/firewall rule and thus opens it for the plaintext connection. We use this connection for the generation of new plaintext and thus the encryption of packets. In our case, the plaintext generation server is a simple UDP server.

Again, we make use of the reflection mechanism and inject another TCP packet with a known plaintext and a known length, as described above. This time, when the packet is reflected and sent back, the malicious relay can predict the payload and the exact keystream. This is possible, as we now know *all* internal parameters from step Ⓐ, i. e., the IP address, TCP port, and TTL. The relay can transform the incoming packet by extracting the keystream through xor-ing the predicted plaintext to then encrypt it with its own content by xor-ing the keystream again. Consequently, the relay can send its own encrypted UDP packet in uplink direction. Finally, the firewall/NAT establishes a rule allowing all incoming packets from this UDP tuple. The plaintext generation server receives the incoming packet and can send packets in downlink direction to the UE.

C. Attack Phase: Uplink IMP4GT

The uplink IMP4GT allows a full impersonation of a user towards an arbitrary IP service (cf. Fig. 3), e. g., HTTP server. To this end, the attacker must be able to *encrypt* packets in uplink direction for establishing the connection and requesting the content. Furthermore, the attacker must *decrypt* packets in downlink direction to access the content sent from the server. For both cases, we construct an oracle that exploits the IP reflection mechanism and the missing integrity protection. In the following, we describe the detailed attack phase for the uplink encryption and downlink decryption.

1) Uplink Encryption: The uplink encryption enables the attacker to *create and encrypt* legitimate IP packets for sending them to the target HTTP server (cf. Figure 7). To do so, the attacker must learn the valid keystream for a PDCP frame

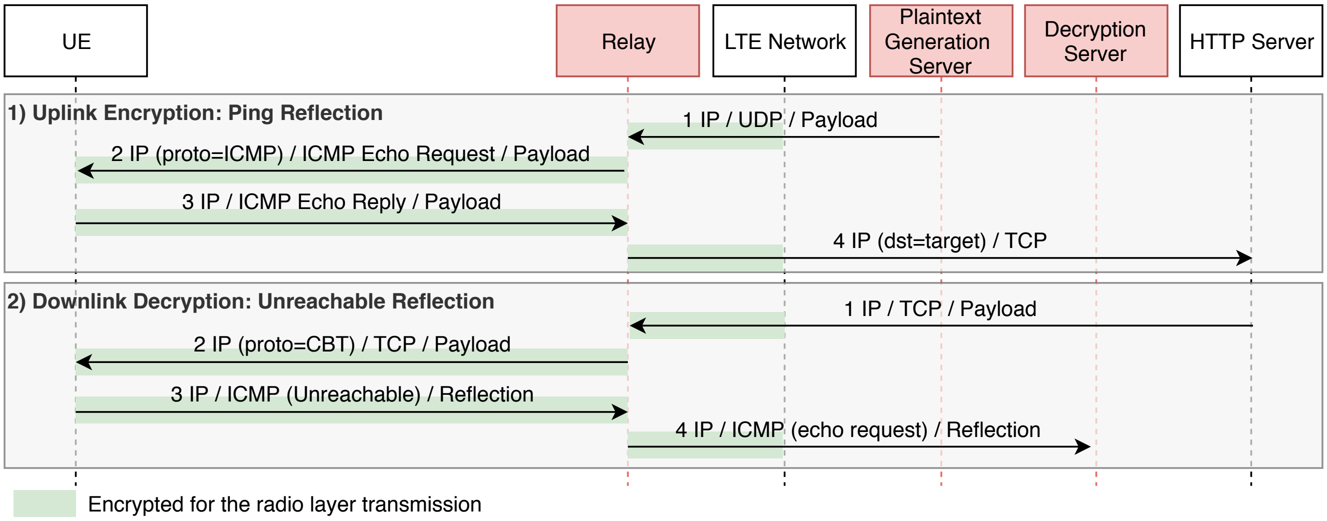
UE
Relay
LTE Network
Plaintext Generation Server
Decryption Server
HTTP Server
1) Uplink Encryption: Ping Reflection
1 IP / UDP / Payload
2 IP (proto=ICMP) / ICMP Echo Request / Payload
3 IP / ICMP Echo Reply / Payload
4 IP (dst=target) / TCP
2) Downlink Decryption: Unreachable Reflection
1 IP / TCP / Payload
2 IP (proto=CBT) / TCP / Payload
3 IP / ICMP (Unreachable) / Reflection
4 IP / ICMP (echo request) / Reflection
🟩 Encrypted for the radio layer transmission

Fig. 7. The uplink IMP4GT attack consists of uplink encryption exploiting the ping reflection and the downlink decryption based on the unreachable reflection.

and apply this to a packet sent in uplink direction. The core idea for an oracle with these abilities is to inject a packet in downlink direction, let the ICMP stack of the UE reflect this packet, and then use the uplink packet as keystream. For the uplink encryption, we can exploit the unlimited *ping reflection*, as the plaintext is known to the attacker and the correct ICMP checksum can be calculated. In the following, we first describe the general procedure of the uplink encryption and the ping reflection. Then, we go into more detail and discuss the technical challenges of predicting the uplink plaintext.

In the first step, the relay requests the plaintext generation server to generate a UDP packet of a certain length n. The plaintext generation server sends the UDP packet via the established plaintext connection to the network, which performs all necessary steps, including the radio layer encryption, and forwards it to the UE (1). The relay intercepts the packet and alters the IP protocol field to ICMP. Further, it changes the ICMP field to *echo request* and sets the correct checksum for the foreknown payload (2). When the baseband of the UE receives the LTE frame, it decrypts it and forwards the contained IP packet to the OS. The ICMP echo request triggers the echo mechanism of the ICMP stack and the payload is reflected due to the ping reflection mechanism. The resulting IP/ICMP packet is encrypted and sent on layer two to the adversarial relay (3), where it can predict the whole plaintext. By that, the relay can derive the complete keystream by xor-ing the predicted plaintext on the received PDCP frame. The relay then uses the keystream to encrypt the target uplink packet, also by xor-ing the keystream on the target uplink packet. The relay sends the frame towards the commercial network, which then decrypts the frame and forwards it to the Internet (4). In this way, we can build an encryption oracle for uplink packets, which can be used for sending arbitrary packets to the Internet on behalf of the victim.

Plaintext Prediction. One important feature for the encryption of a crafted packet is the ability of predicting the exact plaintext, as otherwise the relay cannot encrypt the packet with a valid keystream. While the sent plaintext is known when it is sent by the plaintext generation server, the header information, e. g., IP address or port, change until the relay receives the packet. The general idea behind the plaintext prediction is to keep track of all changes until they arrive at the relay.

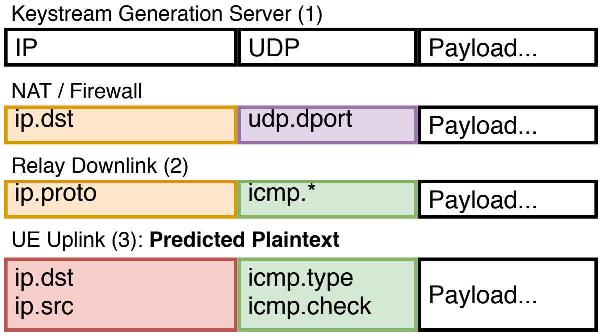

Fig. 8. Overview of changes of the reflected packet for uplink encryption.

Figure 8 depicts an overview of these changes to the downlink packet until the relay receives it as a reflected uplink packet. The payload generated by the plaintext generation server itself remains unchanged. However, the IP header and the UDP header underlie constant changes when passing through the network. First, the NAT/firewall maps the external connection to the internal addresses by changing the destination IP and port. Those changes need to be taken into account when the relay transforms the packet into an ICMP echo request. Therefore, the relay changes the protocol type to ICMP and sets the ICMP header accordingly, including the correct ICMP checksum (2.). The UE reflects the ICMP packet and creates, therefore, a new IP header, i. e., by swapping the source with destination IP and by changing the ICMP type (3.). When this packet arrives at the relay, the relay can deterministically replicate all changes from above, thus can predict the exact plaintext, and subsequently extract the exact keystream. The relay possesses now a valid keystream and can encrypt its crafted packet.

2) Downlink Decryption: A bi-directional IP connection also requires an attacker to decrypt the packet sent by the target server to either maintain the connection or to access information sent in response. For the downlink decryption, we make use of the *unreachable reflection*, as the attacker has no knowledge about the plaintext.

When the targeted HTTP server sends a downlink IP packet to the alleged UE (cf. Figure 7), this packet is encrypted by the commercial eNodeB and sent to the relay (1). The relay intercepts the frame, alters the IP protocol header field, and forwards the frame to the UE (2). Again, the UE reflects the packet and sets the destination IP address to the targeted HTTP server (3). In the uplink direction, the relay modifies the destination address to the attacker's decryption server. Additionally, the relay changes the ICMP type to an *echo request* (4), and forwards the frame to the commercial eNodeB. The ICMP packet (containing the original TCP/IP downlink packet) is decrypted and routed to the attacker's decryption server. In this way, the attacker is able to learn the content of the downlink IP packet.

3) Recovering the Downlink Plaintext: Triggering the unreachable reflection requires to modify the content of the packet, which in turn requires the attacker to recover the exact downlink plaintext. When the relay receives the downlink packet (1), it needs to change the protocol field for triggering the unreachable reflection. To compensate for this change, i. e., to not invalidate IP checksum, the relay can modify the total length field. Unfortunately, this leads to a situation in which one byte of the payload is missing. However, this information can be recovered based on the TCP checksum as follows: the original TCP checksum was calculated by the HTTP server and was not changed after the NAT, therefore, it still contains information about the missing byte. The decryption server recovers the byte by calculating the TCP checksum over the received TCP data. In a second step, it subtracts the calculated checksum from the original TCP checksum and obtains the missing byte. Finally, the attacker can reconstruct the full downlink TCP/IP packet.

To summarize, we explained how sequently combining the uplink encryption with downlink decryption allows an attacker to establish a fully-functional TCP/IP connection to any server on the Internet with the victim's identity.

D. Attack Phase: Downlink IMP4GT

The downlink impersonation allows an attacker to establish a TCP/IP connection to the phone and thus to bypass any firewall mechanism implemented in a given LTE network. This attack can be compared with an attacker that is located in the same local network: usually, local networks allow direct IP access to all link-local devices. For a bi-directional TCP communication, we must consider two cases: First, the attacker must be able to encrypt TCP packets towards the UE and, second, she must decrypt uplink traffic sent by the UE. Figure 9 depicts both cases. Note that the downlink variant by itself does *not* exploit the IP stack's reflection mechanism.

1) Downlink Encryption: For encrypting a downlink packet, the relay requests the plaintext generation server to generate a UDP packet of a certain length, which is sent to the UE via LTE (1). The relay intercepts the packet based on the length and xors the known plaintext to the intercepted packet to extract the keystream. The relay reuses the keystream to encrypt its injected TCP packet by simply xor-ing the packet to the keystream (2). For this, the relay needs to consider the modifications made by the NAT or the routing process, similar to those described in the first step of the Section III-C1. In particular, the NAT changes the destination address and port, and the routers change the TTL.

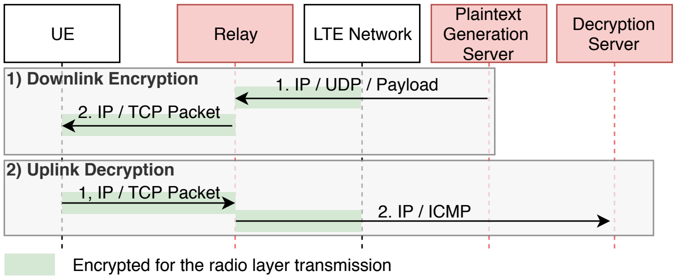

Fig. 9. The downlink IMP4GT attack that combines downlink encryption and uplink decryption.

2) Uplink Decryption: The UE responds to the downlink TCP packet and sends an uplink TCP packet (1), which needs to be decrypted. The relay cannot forward the packet as it is, as the provider's firewall is not aware of the TCP connection and would drop the packet. Therefore, the relay changes the IP protocol field to ICMP and sets the ICMP type to *echo request*. Changing the protocol field again requires compensation, as otherwise the IP checksum is invalid and the packet would be dropped. The relay can compensate the protocol change by modifying the type of service, as this can be predicted for TCP connections. By changing the IP protocol to ICMP, the packet passes the firewall (2) and is routed to the decryption server.

To summarize, we showed how an attacker can establish a fully-functional TCP/IP connection to the UE by combining the downlink encryption followed by the uplink decryption.

IV. Preliminary Experiments

We conduct several experiments to verify the IP stack's reflection mechanism and to investigate the openness of the providers' firewall for ICMP messages. These preliminary experiments influence parameters of the real-world IMP4GT attack that we discuss in Section V.

A. Reflection Mechanism

The full impersonation depends on the ability to encrypt and decrypt packets, which we achieve by exploiting the reflection mechanism of the UE. While the RFCs [2], [20] specify the reflection mechanisms, it is unclear how operating systems implement them within their IP stack. We investigate how the reflection is implemented by Android and iOS, as those two operating systems have the most significant market share. We explore the behavior for the ping and unreachable reflection mechanisms and both IP versions (IPv4 and IPv6). We determine two parameters: (a) reflected packet size and (b) the reflection rate that is the ratio between the packets sent to the device and the packets sent as a response. Both parameters may influence the performance, i. e., data-rate of the attack. Table I gives an overview of the resulting parameters for both reflection mechanisms.

TABLE I. OVERVIEW OF THE ICMP REFLECTION ON ANDROID AND IOS

Method	Feature	Android	iOS
Ping	Size IPv4	1452 byte (MTU)	1452 byte (MTU)
	Size IPv6	1452 byte (MTU)	1452 byte (MTU)
	Response Rate IPv4	100 %	100 %
	Response Rate IPv6	100 %	100 %
Unreachable	Size IPv4	548 byte	0 byte
	Size IPv6	1236 byte	1236 byte
	Response Rate IPv4	rate limiting	0 %
	Response Rate IPv6	rate limiting	100 %

Approach. We determine both parameters with a code review and verify them with practical experiments, using a Wi-Fi connection. For this, we assume that the OS uses the same settings for Wi-Fi as for mobile connections. Exemplarily, we determine the refelection rates for three Android devices (Huawei P10 (Android 7.0), Samsung S6 (Android 7.0) and LG Nexus 5 (Android 5.1)) and for one iPhone XR (iOS 12.2).

For testing the *unreachable* reflection rate, we use *scapy* [36] to send a packet with an unsupported transport-layer protocol (CHAOS (0x10)) to the device. As payload, we use a running sequence number followed by a 500 B string. We sniff the incoming (reflected) packets and determine the rate by matching the sequence numbers of the packets we sent in comparison to the received reflected packets. We use a fixed delay between outgoing packets of 10 ms. This delay resembles the maximum rate for incoming packets and was chosen based on a realistic round trip time for an Internet connection [47].

Android. All Android devices show the same behavior, as they use the same Linux IP stack with no rate or size limiting for the *ping* reflection[1]. This differs for the *unreachable* reflection, where we found that the ICMP stack reflects the minimum Maximum Transmission Unit (MTU). In those cases, the IPv4 stack reflects 548 B and the IPv6 stack 1236 B of the original packets (excluding corresponding headers).

Further, Linux limits the rate for outgoing ICMP messages for the *unreachable* reflection with a *global* ratelimit and an additional *peer* ratelimit; where a peer is defined by the source IP address. While the global limit allows one message per 1 ms (default), the peer ratelimit triggers earlier and only allows one outgoing message each 1000 ms for one particular peer. This is with one exception; the peer rate limiting allows a burst for the first *six* messages. As the peer ratelimit is the stricter limit, we distinguish two cases in our experiments. In the first test, we keep the same source IP for all unsupported protocol messages. In the second case, we change the source address for every six packets. Figure 10 shows the results of the ICMPv4 response rate for both cases. The first case shows that the rate limiting is triggered after six packets and only one packet is reflected each 1000 ms (constant source IP). In contrast, alternating the source IP address does not trigger the peer ratelimit and the reflection remains stable at 100 % over time. As the *multi-peer reflection* allows to perform full-rate encryption, we continue using it in the following experiments.

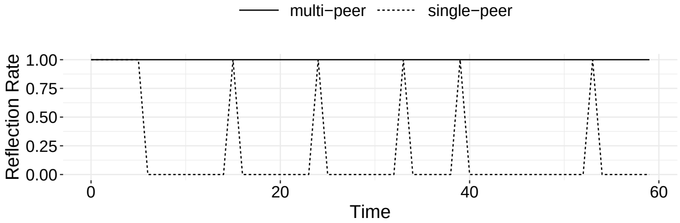

Fig. 10. *Unreachable* reflection rate in case of a fix source IP address (single-peer) and for alternating source IP addresses (multi-peer).

iOS. Apple's mobile operating system uses the Darwin kernel and again, the *ping* reflection is neither limited by size nor rate. However, iOS does not support the *unreachable* reflection for IPv4 packets. For IPv6, iOS reflects the minimum MTU, resulting in 1236 B of payload without any rate limiting. This means that the IMP4GT attack on iOS is not possible for IPv4, but can be conducted without limitation for IPv6.

B. ICMP Firewall/NAT Rules

IMP4GT requires that the provider's firewall/NAT allows sending certain ICMP messages to the attacker's decryption server. We exemplary examine the local providers' firewall/NAT policy. In particular, we test whether the firewall is open for three outgoing ICMP messages; ICMP protocol unsupported ($type = 3/code = 2$), ICMP echo reply ($type = 0/code = 0$), and ICMP echo request ($type = 8/code = 0$). Again, we test this for IPv4 and IPv6, i. e., ICMPv4 and ICMPv6. We again use *scapy* to craft a message and send it to our server, where we monitor the incoming packets.

We tested three providers in western Europe. The results indicate that none of the providers allow the ICMP message *protocol unreachable*, but all providers allow the echo request and echo reply message. This behavior influences the IMP4GT attack as follows: When the IP stack reflects the packet with the ICMP protocol unreachable message, this packet is dropped at the firewall. However, when the attacker updates the protocol type of the ICMP message to an echo request or echo reply, it passes the firewall. We already considered this firewall behavior when describing the IMP4GT attack in Section III. In particular, we change the ICMP message in step Ⓑ (4) of the preparation phase, in step (4) during the downlink decryption for the uplink variant, and in step (2) for the uplink decryption in the downlink variant.

C. Conclusion

While the ping reflection is not limited in size nor rate for Android or iOS, the unreachable reflection is limited. During our tests, we found that Android limits the unreachable reflection in length and rate, influencing the performance of the downlink decryption (uplink IMP4GT). In particular, downlink packets are not allowed to exceed the minimum MTU. Android also limits the reflection rate for a specific peer; the multi-peer reflection technique enables us to decrypt downlink packets with the full rate. iOS does not support the IPv4 unreachable reflection but supports the IPv6 reflection with full-rate. By now, major operators in the USA and Japan deploy IPv6 in their mobile network [26], allowing an impersonation with iOS in those networks.

[1] Note that the MTU of the transmitting interface limited the experiment. The MTU was set to 1500 B, which allows 1452 B of ICMP echo request payload *without* IP fragmentation.

V. End-to-end IMP4GT Attack

We demonstrate the practical feasibility of the IMP4GT attacks by conducting full end-to-end uplink and downlink impersonations in a commercial network.

For the uplink impersonation, we show that an attacker can access a service site[2] of the provider without any user interaction. We choose this targeted website, as it demonstrates the possible consequences of an impersonation attack. In the usual case, those service websites are only accessible by the user and contain personalized content such as phone number or consumed data volume. Further, such service sites allow users to manage their account, access the used data volume, and book new data plans or TV streaming. Accessing such service website can be, therefore, a privacy threat and can also have fraud implications. For the downlink impersonation, we demonstrate that an attacker can establish a direct TCP connection to an app running on the victim's phone. By doing so, we show that an attacker can bypass the provider's firewall mechanism, and the phone is open to any incoming connection. Such an attack is a stepping stone for further attacks, such as malware deployment.

We first describe an additional implementation tweak that allows us to conduct the attack smoothly *without* any modification of the UE. Later we present our experimental setup and the results for the uplink and downlink impersonation.

A. Filtering Background Traffic

We demonstrate the attack without any user interaction; i. e., we begin the attack by redirecting the first DNS request of the Android Internet connectivity check. If this connectivity check is successful, plenty of other Android services use the Internet connection to connect to their home server. Those connections run in parallel to the attack and hence interfere with the state machine of our implementation. We, therefore, implemented a filter mechanism at the relay that drops all unexpected packets during the attack. The filtering mechanism is solely based on the packet lengths and match the assumption of our attacker model. Accordingly, the filter terminates all connections running in parallel to the attack, and the attack is conducted free of any background noise. After the attack, the filtering is switched off, and a regular Internet connection is guaranteed.

B. Setup

We use the following components for our experiments:

UE: We use an unmodified LG Nexus 5 running Android 5.1 with a commercial SIM card. For a stable radio connection to the relay, we place the phone in a shielding box and enable flight mode. Finally, we connect the phone to the PC for controlling it, extract the session key, and record traces with ADB (Android Debug Bridge) and SCAT [45]. For the downlink impersonation, we further implement an app that listens for TCP connections and prints the contents of incoming messages.

Malicious Relay: Our malicious relay consists of two Ettus USRP B210 (about 2600 $) with relay software based on the srsLTE 18.03 stack [17], [1]. One USRP sets up a fake eNodeB towards the UE, while the other USRP emulates the UE towards the commercial network. The relay implements a virtual interface on the operating systems such that the attacker can use any IP-based application. For the uplink impersonation, we use `curl` on top of the virtual interface to access the service site. For the downlink impersonation, we use `netcat` on top of the virtual interface to establish a connection to the installed App.

Commercial eNodeB and Network: We connect to a commercial network using a SIM card.

Attacker's Entities: We use a virtual Ubuntu 16.04 server in the AWS cloud running the attacker's Internet entities. For the DNS server, we use a modified version of dnsmasq [32]. We build all other entities with Python, including the TCP proxy, the plaintext generation server, and the decryption server. For the TCP proxy, we point the IP address to the domain `connectivity.android.com`. The plaintext generation server is based on the UDP socket class of Python. The decryption server is built with *scapy* [36] and listens permanently for ICMP packets. It is reachable via two IP addresses: one IP address is solely used for the attack traffic and matches the requirements of the IP address of the aLTEr attack [40]. The second IP address is used for the control connection between the Internet entities and LTE relay.

Target HTTP Server: To demonstrate that the attacker can access a website on behalf of the victim, we choose to access a service site of the local provider that is only accessible by the victim. On this service site, the user can manage his/her account, for example, to see the used data volume or select a new data plan.

C. Results

We conduct the uplink and downlink impersonation in two separate experiments. As the procedure and results do not differ until the attack phase, we first describe the common preparation phase and later both attacks separately.

We instruct the UE to disable flight mode, which triggers the LTE attach procedure. Then, the UE connects to our relay up to layer two. The relay relays all messages above layer two; in particular, the control data, including the AKA procedure. When the LTE attach procedure is finished and the data connection is activated, the baseband notifies the OS about an existing Internet connection. To check the Internet connection, Android automatically connects to the connectivity service that triggers two DNS requests (AAAA for IPv6 and an A for IPv4) for the domain `connectivity.android.com`. The relay intercepts both requests and redirects them to the malicious DNS server. The DNS server performs the DNS spoofing attack and thus redirects the following HTTP connection to the TCP proxy. In this phase, the relay and the proxy forward all data, until the TCP connection is closed. A first injected packet starts the internal information retrieval, and the decryption server receives the internal information. A second injected packet introduces the establishment to the connection keystream generation server. Until this point, the uplink and downlink impersonation are similar in their procedure. We now describe the results of the uplink impersonation and later of the downlink variant.

[2] `pass.telekom.de` for T-Mobile Germany or `ma.web2go.com` for T-Mobile USA

1) Uplink IMP4GT*:* As soon as the plaintext generation server receives the first packet, we start with the attack phase of the uplink impersonation. The first packet is an uplink TCP SYN packet addressed to the server of the service site. Therefore, the plaintext generation server sends a known-plaintext packet, and the relay modifies this packet for the encryption of the uplink TCP SYN packet. The TCP handshake and the HTTP request and response follow. A downlink packet strictly follows an uplink packet, because we disabled the TCP scale option and the limiting of TCP window to $500\,\mathrm{B}$ on the attacker's relay. This option requires the relay to acknowledge each TCP downlink packet before the target HTTP server sends a new downlink packet. In total, we sent 18 TCP uplink packets and received 16 packets to download the plain HTML site ($5.6\,\mathrm{kB}$), which took $4\,\mathrm{sec}$ with a bitrate of $11.2\,\mathrm{kbit/sec}$. The latency between one uplink packet and receiving the corresponding downlink packet is in average $0.183\,\mathrm{sec}$. *Finally, we access the service site without any user interaction and fully impersonate the victim against the network.*

2) Downlink IMP4GT*:* Again, we start with downlink impersonation, immediately after the preparation phase. The first downlink packet is changed to the TCP SYN packet and followed by an uplink TCP SYN-ACK packet which is decrypted. As payload, we send a character array of $22\,\mathrm{B}$ to the self-written App, which is successfully displayed. Additionally, we changed the source address to `8.8.8.8` and thus hid the identity of the actual server. In total, we sent downlink five packets and received the acknowledgments accordingly. Consequently, we demonstrated that an attacker can bypass any firewall mechanism of the provider, guaranteeing direct network access to the victim's phone.

VI. Discussion

During the attach procedure, LTE establishes mutual authentication with a provably secure AKA protocol. By itself, IMP4GT does *not* attack this AKA protocol and when the AKA is performed, both communication parties are authenticated on the control plane. Even relaying the messages transparently with the relay would not be problematic if the chosen security measures were secure against manipulation. However, this is not the case for the user plane due to the lack of integrity protection. Consequently, IMP4GT exploits the lack of integrity protection in combination with the IP stack behavior, effectively enabling an attacker to impersonate the respective party. IMP4GT breaks mutual authentication *only* on the user plane. In this section, we first discuss the implications of our attack for providers, juridical entities, and users. We assess its real-world applicability, present possible countermeasures, and describe the state of integrity protection in the 5G specification.

A. Implications

Providers rely on mutual authentication for several use cases, such as billing and authorization. One essential requirement for providers is the correct billing of the customers. Furthermore, certain services are only accessible by the authenticated identity, like service websites. Such authentication is performed through header enrichment, which uses only the IP address. Additionally, some providers support third-party PDN networks that are only accessible with APN settings and the correct authentication. IMP4GT undermines user authentication and thereby puts the provider's business model at risk. For example, IMP4GT allows for draining the data volume or accessing the service site with a victim's identity. Providers are required to analyze their risk for each case in which they rely on user authentication.

Additionally, law enforcement agencies have an interest in identifying the correct person during a prosecution. Lawful interception is one requirement that allows targeted wiretapping. Another method to identify a possible perpetrator of Internet crime is to request the identity of a user for a particular public IP address from the provider based on a lawful disclosure request. With IMP4GT, an attacker can forge *any* traffic to the Internet. For example, an attacker can upload prosecution relevant material with the identity of the victim to the Internet. In those cases, the traces from an interception activity can show anomalies such as repetitive UDP packets and a high amount of ICMP packets. However, an attacker can imitate legitimate traffic by simulating DNS traffic for the UDP connection and normal ICMP echo/reply traffic for the ICMP packets. When the agency requests the identity solely from the public IP address, any defects such as traffic anomalies are missing. In both cases, the law enforcement agencies cannot rely on mutual authentication and need to consider the possibility of a IMP4GT attack while investigating the case.

Users are affected by all points that apply for the provider and law enforcement agencies. For example, the provider charges the user's bank account when additional packages are bought, or a law enforcement agency initiates an investigation based on the false assumption of mutual authentication. In those cases, the user has no means to prove his/her innocence. Additionally, the downlink impersonation is an attack directed against the user's phone and can be a stepping stone for further attacks. An attacker can exploit vulnerabilities of network applications, e. g., IoT applications or the operating system. In the light, of zero-day attacks discovered in the wild, IMP4GT can be an additional stepping stone of such an attack. Our attack shows that the user cannot rely on the provider's firewall and they need to harden their device.

B. Real-World Considerations

We have demonstrated the feasibility of the uplink *and* downlink IMP4GT attack with an unmodified phone in a commercial network. Nevertheless, the attack implementation in its current form has limitations regarding stealth, performance, and real-world applicability.

1) Stealth: In our experiments, we filtered unwanted traffic at the relay by dropping packets with an unexpected length. During the attack, we also terminate legitimate connections but restore the regular Internet connectivity after the attack. Additionally, we conduct the attack without any user interaction, which makes it independent from an active usage of the phone. Therefore, we need to consider two cases for reviewing the stealth of the attack. If the victim is actively using the phone, she/he will notice a short time of Internet connection loss. In the case of accessing the local service site, the time of Internet loss amounts to $4\,\mathrm{sec}$, which is justifiable for the attack. With some engineering effort, the filtering can be improved such that the Internet connection remains intact for the user. In case the victim is idle, the loss of Internet connection remains unnoticed.

2) Performance: The attack performance depends on the reflection mechanism because it is one central component of the attack. The *only* reflection that is limited is the unreachable reflection. Because the uplink impersonation builds upon this reflection, it has performance impairments. In particular, the unreachable reflection limits the downlink decryption. We only need to consider the limited length, as the reflection can be triggered with full-rate due to the multi-peer mechanism (see Section IV). We discuss the performance impairments due to the length limitations of the unreachable reflection.

Android and iOS (IPv6 only) reflect only the minimum MTU of the incoming packet, which restricts the length of the to-be-decrypted packet. The attacker cannot *directly* limit the downlink packet length sent by the target server. However, the attacker can *indirectly* force the TCP implementation of the target server to send shorter packets, i. e., by setting the TCP window size to the minimum MTU for the TCP connection. The disadvantage of this option is that each downlink TCP packet needs to be acknowledged. In turn, this limits *only* the throughput of the connection for the uplink impersonation.

Consequently, the uplink IMP4GT attack may not be suitable for low-latency and high data-rate applications, e. g., video streaming, but sufficient to access a website. The downlink IMP4GT remains unaffected of any performance impairments and can be used in full-rate to establish a connection to a UE.

3) Real-World Applicability: For our experiments, we use a shielding box to prevent interference with the licensed spectrum and unwanted UEs with the relay. In a real-world setting, the attacker needs to consider interference and multiple UEs on all layers for building a relay. However, from the UE's perspective, such a relay attack is comparable to fake base station attacks, which are already conducted in the real-world [41], [35]. Nevertheless, we need to consider an attacker with strong domain knowledge along with several resources to implement such a MitM relay and carry out the IMP4GT attack.

Despite all limitations, we demonstrate the feasibility of the IMP4GT attack in a commercial network with an unmodified UE. Thus, it represents a threat for all users and stakeholders that rely on mutual authentication in LTE.

C. Potential Countermeasures

IMP4GT exploits the specification flaw of missing integrity protection *along* with the RFC conform reflection behavior of the IP stack. We first discuss possible mitigations on the higher layers. Then we discuss the opportunity of mitigation in the IP stack, but will argue that the only sustainable countermeasure is mandatory integrity protection.

One possibility is to protect against the initial DNS spoofing attack with DNSSec, DNS over TLS, or DNS over HTTPs. However, IMP4GT does not necessarily need the initial DNS spoofing attack. As soon as the attacker knows the IP address of an outgoing TCP connection, she can directly redirect the TCP connection with the ALTER attack and, thus, hijack the connection for continuing the preparation phase of IMP4GT. An example of outgoing TCP connections are the connections of the email client that connects periodically to pre-known IP addresses. Another possibility would be to secure all TCP connection with TLS such that the client can detect a redirection based on mismatching certificates. However, the TCP proxy transparently relays the TLS connections, and thus the redirection remains undetected. Also, a VPN connection has only limited mitigation properties as not all connections can use the VPN connection and are therefore protected. For example, the connectivity check of Android connects to a service before the OS notifies other applications about the Internet, such as VPN applications. Those connections remain attackable by IMP4GT.

One mitigation is to disable the IP reflection mechanism at the UE, as IMP4GT relies on it. However, any modification would invalidate the RFC conformity of the IP stack and harm interoperability. For example, the ICMP echo request (ping) is often used to check if the device is reachable and disabling the echo responses would break the ping protocol. Consequently, it would be impossible to check if the device is reachable on the IP layer. Thus, any modification of the IP stack might work, but comes at the cost of interoperability.

The main reason for IMP4GT is the lack of integrity protection and thus the possibility of user data manipulation. Mandatory integrity protection was neglected due to the additional overhead on the radio layer. The retrospective specification and deployment of integrity protection in LTE requires financial and logistic efforts, as all UEs and eNodeBs must be updated to be secured against IMP4GT. Despite these efforts, this paper should be read as a reminder of the urgency for mandatory integrity protection on the user plane in LTE.

D. Integrity Protection in 5G

While LTE is already used for nearly a decade, the currently deployed 5G specification comes with different states regarding user-plane integrity protection. We discuss the state of integrity protection for the two deployment phases.

Non-standalone (NSA) with dual connectivity is the first phase, in which the phone connects via 4G for all control data, but uses 5G for user data. The 3GPP 5G Security working group stated: "Although integrity protection for UP data is supported in 5G networks, it will not be used in dual connectivity case." [7]. Thus, the early 5G deployments cannot prevent IMP4GT attacks.

The second phase will be the standalone (SA) phase, in which the UE has a control connection to the 5G core network along with the 5G radio layer. At the time of publication, this phase was still under development; its current state is as follows: First, user-data integrity protection is optional to use and the provider can decide to enable it. Second, the phone can implement integrity protection within full-rate or only up to $64\,\mathrm{kbit/s}$, whereas only the latter option is demanded in the specification. Most data connections exceed this data rate, as 5G promises high data rates up to $20\,\mathrm{Gbit/s}$ and, thus, the user-plane integrity protection cannot be applied [14]. Obviously, the requirement for high-data rates contradicts the requirement for security and the attack vector remains exploitable in 5G [4], [5]. **We emphasize the requirement for a mandatory *and* full-rate integrity protection for all 5G data connections** to prevent IMP4GT.

E. Disclosure Process

We have informed providers and vendors about the attacks through the GSMA CVD process [18]. The GSMA notified the 3rd Generation Partnership Project (3GPP) with a liaison statement [19]. In response, the 3GPP RAN group has confirmed that LTE specifications do not support any integrity protection. For 5G, the 3GPP RAN group points out that integrity protection up to 64 kbit/s is mandatory to support but optional to configure [3], which provides insufficient protection. However, we hope that our findings emphasize the demand for a mandatory and full-rate integrity protection.

F. Ethical Considerations

At all time, we ensured that no real-world users were harmed during the commercial network experiments. We used a shielding box that prevents nearby users from connecting to our relay. Furthermore, the relay's UE component conforms the specification, which ensured a correct behavior towards the commercial network in the up- and downlink.

VII. Related Work

In the following, we discuss related work in the context of mobile networks with a focus on the security of LTE.

The aLTEr attack relies on the same weaknesses and attacker model as IMP4GT, but it follows different attack aims. The aLTEr attack aims to redirect DNS requests and leads a victim to a malicious website. In contrast, the IMP4GT attacks aim to impersonate one of the communication partners and thereby break the mutual authentication for the user plane.

Until now, impersonation attacks in LTE only exploited implementation flaws or misconfiguration. Examples of such flaws are the work by Rupprecht et al. [39], where the authors demonstrated that a UE accepts null security algorithms due to misimplementation. Chlosta et al. [9] found similar issues in the configuration of commercial networks. Both cases allow an impersonation of the respective communication partner, but are fixable with a firmware or configuration update. In contrast, IMP4GT exploits a specification flaw and the requirements of the IP stacks, which are only fixable by a specification change.

The IMP4GT attacks do not break the mutual authentication that is established during the AKA. Previous work analyzed the AKA on different security properties. Alt et al. [6] prove the security of the LTE AKA against a MitM attacker by formal cryptographic analysis. Basin et al. and Cremers et al. [10], [8] analyze the 5G AKA that is similar to the LTE AKA using the protocol verification tool Tamarin. While these prior approaches focus on the AKA security itself in the presence of pre-defined attacker models, our work targets the user plane security mechanisms *following* the AKA that aim for mutual authentication.

For the presented IMP4GT attack, we exploited missing integrity protection which is a specification flaw. So far, previous studies found vulnerabilities in the specification ranging from privacy to denial of service attacks [38]. Privacy attacks can localize and track a user [43], [25], [44] or infer the visiting websites [30], [28]. Further, attacks can exploit protocol vulnerabilities of the phone [43], [24] or exhaust resources [46] for a denial of service. A special kind of denial of service are jamming attacks that disturb the physical communication [27], [31], [16]. Hussain et al. [24] present an authentication relay attack that allows eavesdropping on un-encrypted traffic. The attacks mentioned above exploit specification flaws of control traffic. For our analysis, we do not solely focus on specification flaws but consider the cross-layer interactions on the user plane. Besides specification flaws, LTE implementations offer a wide attack surface. Therefore, one building block of LTE security is the correct implementation and, hence, is target to different analysis methodologies. Kim et al. [29] introduce a semi-automatic tool for analyzing the behavior of equipment with the input of malicious data. By doing so, they discover vulnerabilities, including SMS spoofing attacks or an AKA bypass allowing to eavesdrop data sessions. Fang et al. [15] analyze the implementation security of mobile basebands permuting the input with the support of reinforcement learning. Hong et al. [23], [22] passively analyze the reallocation behavior of temporary identifiers and found that the reallocation is not sufficiently random, which allows the tracking of users. In our analysis, we focus on the cross-layer *specification* issues rather than the implementation security.

For conducting the IMP4GT attack, the relay emulates a fake base station on layer two. Normally, fake base stations in LTE exploit the control traffic sent before the security establishment and allow to track or localize a victim. Previous work targets the detection of fake base stations [12], [33], [11], [34], [42]. By looking for malicious control traffic, e. g., identity requests, the probability of an attack is calculated. In our work, we utilize a relay that acts as a fake base station that does *not* depend on the modification of control traffic. Consequently, the relay can be integrated into the commercial network without being recognized and, thus, circumvents existing detection methods. Recently, Hussain et al. [37] proposed a prevention mechanism against fake base stations based on a public key scheme together with distance bounding, allowing the detection of relayed control traffic. Our relay forwards control traffic and such mechanism can prevent it if specified and correctly implemented. In the context of the IPsec protocol, Degabriele et al. showed that the use of encryption-only mode with no integrity protection is vulnerable to decryption [13]. For the proposed attack, the authors also exploit the ICMP refection mechanism along with IPSec specific padding and are able to extract the plaintext. In contrast to our work, the authors focus on IPsec and perform a ciphertext-only attack.

VIII. Conclusion

Mutual authentication is one central and key security aim of LTE and is the basis for authorization, accounting, and lawful interception. In LTE, a provably secure AKA establishes mutual authentication and subsequent security mechanism ensure the confidentially of data. However, recent studies revealed that the security mechanism does not protect against the manipulation of user data.

In this paper, we presented the novel IMP4GT attacks that completely break the mutual authentication aim on the user plane. More specifically, IMP4GT allows an active radio attacker to establish arbitrary TCP/IP connections to *and* from the Internet through the victim's UE. IMP4GT exploits the lack of integrity protection along with ICMP reflection mechanisms. As a result, the attacker can circumvent *any* authorization, accounting, or firewall mechanism of the provider. We perform experiments to verify our assumptions and demonstrate the real-world feasibility of IMP4GT in a realistic setup. As a result, we can access a service site that should only be accessible by the user or circumvent the provider's firewall. The lack of integrity protection can break mutual authentication which is one fundamental security aim of LTE. Considering this fact, we demand to specify effective countermeasures for LTE and mandatory user-plane integrity protection for 5G.

Acknowledgment

This work was supported by German Federal Ministry of Education and Research (BMBF Grant 16KIS0664 SysKit) and the German Research Foundation (DFG) within the framework of the Excellence Strategy of the Federal Government and the States – EXC 2092 CASA – 39078197.

References

[1] "Ettus Research USRP B210," https://www.ettus.com/product/details/UB210-KIT, [Online; accessed 20-Feb-2020].

[2] "Internet Control Message Protocol," RFC 792, Sep. 1981. [Online]. Available: https://rfc-editor.org/rfc/rfc792.txt

[3] 3GPP, "Reply to LS on Impersonation Attacks in 4G Networks," https://www.3gpp.org/ftp/tsg_ran/WG2_RL2/TSGR2_107/LSout/R2-1911819.zip, [Online; accessed 20-Jan-2020].

[4] 3GPP, "NR; NR and NG-RAN Overall Description;," 3rd Generation Partnership Project (3GPP), TS TS38.300, 2018. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/38300.htm

[5] ——, "Security architecture and procedures for 5G System," 3rd Generation Partnership Project (3GPP), TS TS33.501, 2018. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/33501.htm

[6] 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.

[7] Anand R. Prasad, Alf Zugenmaier, Adrian Escott and Mirko Cano Soveri, "3GPP 5G Security," https://www.3gpp.org/news-events/1975-sec_5g, 08 2018, [Online; accessed 20-Jan-2020].

[8] D. Basin, J. Dreier, L. Hirschi, S. Radomirovic, R. Sasse, and V. Stettler, "A Formal Analysis of 5G Authentication," in *Conference on Computer and Communications Security (CCS)*. ACM, 2018, pp. 1383–1396.

[9] M. Chlosta, D. Rupprecht, T. Holz, and C. Pöpper, "LTE Security Disabled — Misconfiguration in Commercial Networks," in *Conference on Security & Privacy in Wireless and Mobile Networks (WiSec)*. ACM, 2019.

[10] C. Cremers and M. Dehnel-Wild, "Component-Based Formal Analysis of 5G-AKA: Channel Assumptions and Session Confusion," in *Symposium on Network and Distributed System Security (NDSS)*. ISOC, 2019.

[11] A. Dabrowski, G. Petzl, and E. R. Weippl, "The Messenger Shoots Back: Network Operator Based IMSI Catcher Detection," in *Recent Advances in Intrusion Detection (RAID)*. Springer, 2016.

[12] 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.

[13] J. P. Degabriele and K. G. Paterson, "Attacking the IPsec Standards in Encryption-only Configurations," in *IEEE Symposium on Security and Privacy (SP)*. IEEE, 2007.

[14] ETSI — European Telecommunications Standards Institut, "Why do we need 5G?" https://www.etsi.org/technologies/5g, [Online; accessed 20-Jan-2020].

[15] K. Fang and G. Yan, "Emulation-Instrumented Fuzz Testing of 4G/LTE Android Mobile Devices Guided by Reinforcement Learning," in *European Symposium on Research in Computer Security (ESORICS)*. Springer, 2018.

[16] F. Girke, F. Kurtz, N. Dorsch, and C. Wietfeld, "Towards Resilient 5G: Lessons Learned from Experimental Evaluations of LTE Uplink Jamming," *arXiv preprint arXiv:1903.10947*, 2019.

[17] I. Gomez-Miguelez, A. Garcia-Saavedra, P. D. Sutton, P. Serrano, C. Cano, and D. J. Leith, "srsLTE: An Open-source Platform for LTE Evolution and Experimentation," in *Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization*, ser. WiNTECH '16. ACM, 2016.

[18] GSMA, "GSMA Coordinated Vulnerability Disclosure Programme)," https://www.gsma.com/aboutus/workinggroups/working-groups/fraud-security-group/gsma-coordinated-vulnerability-disclosure-programme, [Online; accessed 20-Jan-2020].

[19] ——, "Liaison Statement: Impersonation Attacks in 4G Networks," http://www.3gpp.org/ftp/Inbox/LSs_from_external_bodies/GSMA_CVD/CVD%20Doc24_01%20LS%20to%203GPP.zip, [Online; accessed 20-Jan-2020].

[20] M. Gupta and A. Conta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification," RFC 4443, Mar. 2006. [Online]. Available: https://rfc-editor.org/rfc/rfc4443.txt

[21] M. Holdrege and P. Srisuresh, "IP Network Address Translator (NAT) Terminology and Considerations," RFC 2663, Aug. 1999. [Online]. Available: https://rfc-editor.org/rfc/rfc2663.txt

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

[23] 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.

[24] 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.

[25] S. R. Hussain, M. Echeverria, O. Chowdhury, N. Li, and E. Bertino, "Privacy Attacks to the 4G and 5G Cellular Paging Protocols Using Side Channel Information," in *Symposium on Network and Distributed System Security (NDSS)*. ISOC, 2019.

[26] Internet Society, "State of IPv6 Deployment 2018," https://www.internetsociety.org/wp-content/uploads/2018/06/2018-ISOC-Report-IPv6-Deployment.pdf, 06 2018, [Online; accessed 20-Jan-2020].

[27] 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.

[28] ——, "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

[29] H. Kim, J. Lee, E. Lee, and Y. Kim, "Touching the Untouchables: Dynamic Security Analysis of the LTE Control Plane," in *IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019.

[30] K. Kohls, D. Rupprecht, T. Holz, and C. Pöpper, "Lost Traffic Encryption: Fingerprinting LTE/4G Traffic on Layer Two," in *Conference on Security & Privacy in Wireless and Mobile Networks (WiSec)*. ACM, 2019.

[31] 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.

[32] N/A, "dnsmasq - A lightweight DHCP and caching DNS server." https://manpages.debian.org/stretch/dnsmasq-base/dnsmasq.8.en.html, [Online; accessed 20-Jan-2020].

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

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

[35] S. Park, A. Shaik, R. Borgaonkar, and J.-P. Seifert, "Anatomy of Commercial IMSI Catchers and Detectors," in *Workshop on Privacy in the Electronic Society (WPES)*. ACM, 2019.

[36] Philippe Biondi and the Scapy community, " Scapy Project: Packet crafting for Python2 and Python3." https://scapy.net/, [Online; accessed 20-Jan-2020].

[37] S. Rafiul Hussain, M. Echeverria, A. Singla, O. Chowdhury, and E. Bertino, "Insecure Connection Bootstrapping in Cellular Networks: The Root of All Evil," in *Conference on Security & Privacy in Wireless and Mobile Networks (WiSec)*. ACM, 2019.

[38] D. Rupprecht, A. Dabrowski, T. Holz, E. Weippl, and C. Pöpper, "On Security Research towards Future Mobile Network Generations," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2518–2542, 2018.

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

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

[41] Sam Biddle, "Long-Secret Stingray Manuals Detail How Police Can Spy on Phones," https://theintercept.com/2016/09/12/long-secret-stingray-manuals-detail-how-police-can-spy-on-phones, Sep. 2016.

[42] Security Research Labs, "SnoopSnitch - Mobile Network Security Tests," https://opensource.srlabs.de/projects/snoopsnitch, 2014, [Online; accessed 20-Feb-2020].

[43] 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.

[44] A. Shaik, R. Borgaonkar, S. Park, and J.-P. Seifert, "New Vulnerabilities in 4G and 5G Cellular Access Network protocols : Exposing Device Capabilities," in *Conference on Security & Privacy in Wireless and Mobile Networks (WiSec)*. ACM, 2019.

[45] The Computer Security Group at Berlin University of Technology, " SCAT: Signaling Collection and Analysis Tool." https://github.com/fgsect/scat, [Online; accessed 20-Jan-2020].

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

[47] WonderNetwork, "Global Ping Statistics —Ping times between WonderNetwork servers," https://wondernetwork.com/pings, [Online; accessed 20-Jan-2020].

Acronyms

3GPP	3rd Generation Partnership Project
AKA	Authentication and Key Agreement
eNodeB	Evolved NodeB
EPC	Evolved Packet Core
GSMA	GSM Association
GSM	Global System for Mobile Communications
HSS	Home Subscriber Server
ICMP	Internet Control Message Protocol
IMSI	International Mobile Subscriber Identity
LTE	Long Term Evolution
MitM	Man-in-the-Middle
MME	Mobility Management Entity
MTU	Maximum Transmission Unit
NAS	Non-Access Stratum
NAT	Network Address Translation
PDCP	Packet Data Convergence Protocol
PDN	Packet Data Network
P-GW	Packet Data Network Gateway
RRC	Radio Resource Control
S-GW	Serving Gateway
TMSI	Temporary Mobile Subscriber Identity
TTL	Time To Live
UE	User Equipment