Protocol-Aware Reactive LTE Signal Overshadowing and its Applications in DoS Attacks

Master Thesis

Author(s):
Erni, Simon

Publication date:
2020

Permanent link:
https://doi.org/10.3929/ethz-b-000455834

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Informatik
Computer Science

Protocol-Aware Reactive LTE Signal Overshadowing and its Applications in DoS Attacks

Master Thesis

Simon Erni

9th December, 2020

Advisors:
Patrick Leu
Dr. Marc Röschlin
Prof. Dr. Srdjan Čapkun

Department of Computer Science, ETH Zürich

Abstract

Long Term Evolution (LTE) communication is relied upon more than ever before. As such, attacks on its availability deserve to be investigated thoroughly. Recent discoveries in LTE research have shown the feasibility of a new approach called signal overshadowing, where the attacker signal is sent with the same timing and slightly higher power as the legitimate signal of an LTE base station, thereby successfully impersonating it. In this thesis, we build upon this discovery and improve its power efficiency for a success rate of 100% at a power difference of 1.8dB. Additionally, we found that by reactively overshadowing key messages in procedures between the user equipment (UE) and the core network, a denial of service (DoS) can be caused in many of them. We demonstrate that the attacks, which we devised using our own software in conjunction with open source libraries and software defined radios (SDR), causes smartphones from a wide range of manufacturers to suffer a connection-loss of more than 12 hours. Finally, we make the case for further research on uplink overshadowing by developing a high-impact attack and demonstrating its feasibility in current live networks.

Acknowledgements

I want to thank my supervisors from the System Security Group, Patrick Leu and Dr Marc Röschlin, for their continued support, guidance and valuable feedback throughout the development of this thesis. I would also like to thank Prof. Dr Srdjan Čapkun for making this thesis possible, and also my colleague Martin Kotuliak for all his support during the project. Finally, special thanks go to my family and my significant other for their generous support and patience during all my studies.

Contents

Chapter 1

Introduction

1.1 Motivation

Cellular communication has become ubiquitous in our lives, especially the LTE standard, or more commonly known as *4G*. This was shown in a report presented by OpenSignal, where every single one of the 87 surveyed countries had 4G available, providing an average regional coverage of 80% [1]. The availability of this system is highly important, as a recent risk analysis done by the Swiss government has shown in [2].

While jamming generally does not introduce widespread outages, it is inherent to any wireless communication scheme, of which LTE is no exception. While traditional analog wideband jammers emitting noise are always possible given enough output power, this requires expensive and energy-intensive hardware and there are scenarios where this is not desired. If used legally, such as in a courtroom or prison setting, jammers with high output power are generally undesirable due to health concerns. For nefarious purposes, an attacker might wish to impact a range as large as possible, while also remaining hard to detect from any authorities.

Second to the energy requirement is the persistence of an attack - an immediate connection re-establishment after the attack is turned off or the victim is out of range might similarly be undesirable.

Therefore, the objective of this thesis is to identify more efficient and/or persistent attacks on the availability of LTE. If any such attack is found, naturally, the ultimate goal is to harden current and future versions of mobile communication systems to be resilient against them.

1.2 Research Question

When looking at the state of security research concerning availability in LTE, it only recently began to gain traction with the increasing prevalence of Software Defined Radio (SDR) technology, such as with the Ettus Research B210 [3], the LimeSDR [4] or the bladeRF 2.9 micro xA4 from Nuand [5]. SDR devices can be used together with open source software such as srsLTE [6] to create fake LTE base stations or, as recently discovered by Yang et al. in [7], to carry out synchronized attacks, hijacking actively running LTE connections. This implies that the attacker model of previous studies was too limited at times, as they underestimated the capabilities of an attacker. Future research must therefore get closer to a Dolev-Yao model, as is traditionally done with internet communication protocols such as the well-understood Transport Layer Security (TLS) protocol.

1.3 Contribution

This works shows that the approach by Yang et al. in [7] is feasible and augments the capabilities of an attacker even further. We contribute a near real-time downlink sniffer and signal generation for the whole protocol stack, enabling reactive downlink signal overshadowing attacks. The software architecture used allows extensions to overshadow all kinds of messages with minimal effort, enabling further research.

Furthermore, we show that attacks on higher-level procedures show a much higher impact, and provide working implementations for them. Our evaluations showed that all smartphone tested from a wide range of manufacturers are vulnerable to our attacks on high level attach procedures, leading to a loss of LTE communication capabilities for more than 12 hours.

Chapter 2

Background

To understand the following chapters about existing and novel vulnerabilities in LTE, a short primer on LTE and its basic components is necessary. This is done from the bottom-up, starting with the radio layer and ending with the high-level transactions executed between different entities in the real world. However, this is still only designed as a high-level overview. Whenever more detail is necessary to follow, they are introduced at their respective point in the thesis.

2.1 Components of LTE

An LTE network is operated by a Mobile Network Operator (MNO). LTE is comprised of two main components which are shorty explained here. Refer to figure 2.1 for an overview of them and their relationships.

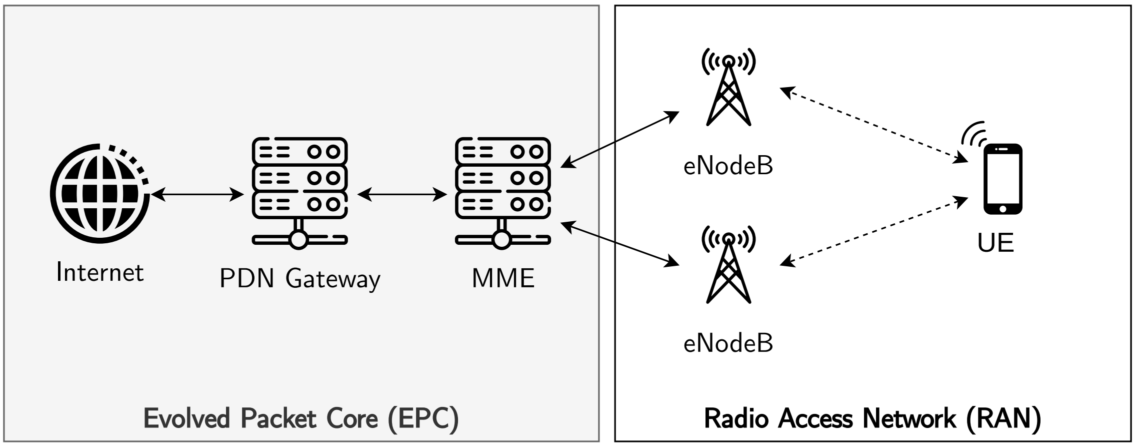


Figure 2.1: LTE Architecture Overview

2.1.1 Radio Access Network

First, there is the E-UTRAN Node B (eNodeB), or base station, placed in locations all around the country. Second, there is the User Equipment (UE), which is in most cases a phone, which interacts with the eNodeB over a wireless channel. As the UEs of the customers of a MNO may change their location, a lot of eNodeBs around the country are necessary to handover the UE to the most suitable eNodeB in their vicinity.

2.1.2 Evolved Packet Core

Connecting the eNodeBs together and providing crucial services for telephony and internet access is the Evolved Packet Core (EPC). It consists, at the minimum, of a Mobility Management Entity (MME), which connects to a PDN Gateway, which will offer further connectivity to the internet.

2.2 Physical Layer

2.2.1 Modulation

The basis of any wireless communication system lies in its scheme of how to encode bits of information onto the wireless channel, a process called modulation. LTE uses many different modulation schemes, each carefully selected to yield the most performance or the most reliability, depending on the use case. This modulation scheme is then applied onto a carrier signal at a specific frequency. The most used modulations in LTE are forms of Quadrature amplitude modulation (QAM), dynamically selected depending on the current channel quality.

2.2.2 Resource Sharing

As LTE is not a broadcast application, there must be a way to divide the available time and frequency resources such that each user will be able to receive and send data from the eNodeB.

Duplexing

The way you organize the available resources to allow both sending and receiving is called duplexing. LTE has two different duplexing mechanisms, Frequency Division Duplex (FDD) and Time Division Duplexing (TDD). FDD works by allocating one frequency range for downlink traffic and one for uplink traffic only. TDD works with the UE and eNodeB taking turns in who sends data when. In the following, we will examine FDD only, as it is the only variation used in Switzerland for LTE.

Interestingly, in the case of FDD, the UE always gets assigned the lower frequencies, as they allow for a transmission with a larger range for the same energy input, conserving energy in the UE.

Time Multiplexing

The time multiplexing elements in LTE for both down- and uplink are called frames and subframes. Each frame has a duration of 10ms and is divided up into 10 subframes, each taking 1ms. The structure is illustrated in figure 2.2. This multiplexing applies both to down- and uplink.

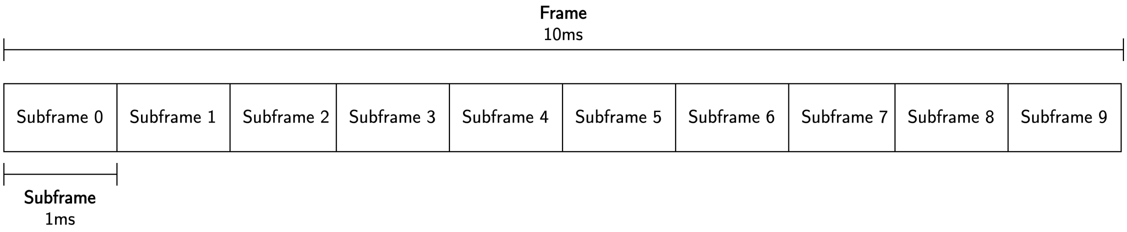


Figure 2.2: Frame and Subframe Structure

Frequency Resource Sharing

Downlink - Orthogonal Frequency Division Multiple Access (OFDMA) As, in the case of LTE, a large bandwidth of up to 20MHz can be used and there may be many users on a cell waiting to be served, LTE uses many different carrier signals spread out over its bandwidth. Each of those sub-carriers carries different data, possibly destined for different users. However with modulation applied, the carrier signal no longer remains at a single specified frequency, but rather spreads to other frequencies in the vicinity. To eliminate this intercarrier interference, the carrier signals are placed in such a way that when modulated, a sub-carrier signal will spread only between the adjacent sub-carriers, and will remain exactly zero at the frequency of other sub-carriers. This placement is called *orthogonal*, hence why this process in LTE is called OFDMA.

Uplink - Single-carrier Frequency Division Multiple Access (SC-FDMA) In the uplink LTE chose to use SC-FDMA, which uses a single carrier for each transmitting UE, eliminating the drawback of OFDMAs large Peak to Average Power Ratio (PAPR), which is beneficial for power-constrained devices.

2.2.3 Reference Signals

Because the QAM modulation is very susceptible to the current channel properties, reference signals are included in the downlink to properly estimate the channel and demodulate the signal. These reference signals are encoded with the cell identity differing among neighbouring cells, which enables communication even with an overlap of multiple cells on the same frequency. As is seen later for our attacks, these reference signals serve an important purpose.

2.2.4 Capture Effect

The modulation scheme of LTE in the down- and uplink is QAM. As has been seen in [8], [9], modulation schemes such as QAM exhibit in the presence of multiple, colliding, transmitters the so called capture effect. This capture effect allows that a receiver may, in the presence of 2 concurrent, colliding transmissions, to decode the one with a stronger signal. According to Lee et al. in [9], this succeeds even with the stronger signal being only around 2dB more powerful.

2.3 Identifiers

The UE has several different identifiers, each used for a different purpose.

2.3.1 International Mobile Subscriber Identity (IMSI)

The International Mobile Subscriber Identity (IMSI) is the most permanent identifier and is bound to the Universal Subscriber Identity Module (USIM) in the UE. It is sent, in the clear, on a very first connection attempt with a network.

2.3.2 Temporary Mobile Subscriber Identity (TMSI)

To attempt protecting the privacy of a user, the IMSI is only used for this first connection attempt. After this, it is replaced by the Temporary Mobile Subscriber Identity (TMSI), although in cases where the network or the UE has lost state, the IMSI will be used again.

2.3.3 Radio Network Temporary Identifier (RNTI)

The Radio Network Temporary Identifier (RNTI) is used to distinguish between different messages on a physical layer and is assigned by the eNodeB to the UE. It is only used during a single session, so when the UE enters idle mode and re-associates again, it may receive a different RNTI.

2.4 Channels and Layers

LTE is comprised of many different channels and layers, making it difficult to comprehend the flow of data. This is why provide the reader with a short overview of the dataflow for both down- and uplink, and explain each part at a high level.

In figure 2.3, all for our purposes important downlink channels and layers are depicted, beginning at the bottom with the grey physical channels, up to the yellow transport channel and then to the green logical channels. In figure 2.4, a very similar figure is drawn for the uplink, although there is the PDDCH channel included, which is necessary for the uplink to work.

2.4.1 Downlink - Physical Downlink Control Channel (PDCCH)

Because data of many users may be carried in a single subframe of 1ms, there must be a way to distinguish their intended recipient. To this end, LTE has designated the Physical Downlink Control Channel (PDCCH), containing Downlink Control Information (DCI) messages.

A DCIs job is to serve as a pointer towards the correct region where the Physical Downlink Shared Channel (PDSCH) data for the user lies. Each DCI is sent just at the beginning of each subframe and points to the correct region for decoding the PDSCH. To distinguish the DCIs themselves, they have a CRC appended to them, which is then XOR-ed with the RNTI of the target. Finally, the DCI is put at one of several locations depending on the RNTI, which the UE will attempt to decode in a procedure called *blind search*.

2.4.2 Downlink - Physical Downlink Shared Channel (PDSCH)

Once the UE has decoded a set of candidate DCIs, it will attempt do demodulate the corresponding PDSCH channel into a set of DL-SCH transport blocks. Apart from demodulation, there are other steps in this process, but all have been simplified into the *Demodulation* box for brevity. Depending on the RNTI the DL-SCH transport block was decoded with, it is either an SIB, a paging message, or some user data. In the case of it being an SIB or paging message, it may be directly decoded and no further processing is necessary. Otherwise, it is passed on to the MAC layer.

2.4.3 Downlink - Paging

The purpose of paging messages is to inform the idle UE of any available data/SMS/voice call or change in the system. When no paging message is received, and the user doesn't wish to send data, the UE may remain in a power-saving idle state. To know that the paging message is intended for the UE, it will search for its TMSI in the paging message. If it receives a paging message with an IMSI, it knows that the network must have lost state and will re-attach.

2.4.4 Downlink - System Information Block (SIB)

When connecting to a cell, many configuration parameters are necessary to be installed in the UE as LTE allows for many regional and operator-specific configurations. The very first configuration object is the Master Information Block (MIB), after which the different System Information Block (SIB)s may be decoded. All of these configuration objects are sent in broadcast and are not cryptographically protected.

2.4.5 Down- and Uplink - MAC

The MAC layer is responsible for de/multiplexing data destined belonging to a UE into control and user data channels. Each one of those channels is identified with a Logical Channel ID (LCID). A message passing through the MAC layer may contain multiple messages, destined for different LCIDs.

Common Control Channel (CCCH)

The LCID 0 identifies messages for the Common Control Channel (CCCH), which are initial messages used to set up a connection with an eNodeB, e.g. the `RRC Connection Setup` message. Messages on the CCCH are always sent unencrypted.

Dedicated Control Channel (DCCH)

After an RRC connection has been established with the eNodeB, it instructs the UE with the `RRC Connection Setup` message to establish another bearer, identified with LCID 1, the Dedicated Control Channel (DCCH). This carries messages that may be encrypted and/or integrity protected and used to connect with components in the core network. As visible in figure 2.3, 2-3 layers are involved in decoding the messages.

Dedicated Traffic Channel (DTCH)

As indicated by the n in figure 2.3 for the LCID, there may exist multiple such bearers for the user data, each with a different Quality of Service and/or purpose. One could be used for web-browsing, while the other transmits voice data. This way, one can look something up on the internet while still being in a phone call.

Hybrid Automatic Repeat Request (HARQ)

To know that the data the UE has sent on the uplink has arrived on the network, an acknowledgement is sent back down via the Hybrid Automatic Repeat Request (HARQ), without relying on any of the higher-level processes discussed later. Because the ACK/NACK is always sent exactly 4ms after the eNodeB has received the data, the system can react very fast to any channel problems and may thus increase overall throughput. Multiple such HARQ processes can run in parallel, so the MAC layer may send data from multiple bearers concurrently.

2.4.6 Down- and Uplink - Radio Link Control (RLC)

The purpose of the RLC layer is comparable to the TCP/UDP layer 4 of the internet. Its main responsibilities are acknowledging, re-ordering and re-assembling messages if desired. If not, it may also operate in a mode where it doesn't require acknowledges, but still re-assembles messages, or in a fully transparent mode, comparable with UDP.

2.4.7 Down- and Uplink - Packet Data Convergence Protocol (PDCP)

The Packet Data Convergence Protocol (PDCP) layer is there to de/compress the header of the data it passes through and maintain its integrity and confidentiality cryptographically.

However, the first messages that pass through during connection initialization are not protected until the keys have been installed and then activated through the `RRC Security Mode Command` message.

2.4.8 Down- and Uplink - Non-Access Stratum (NAS)

After the message has passed through the PDCP layer, it is decoded by the relevant RRC decoder. It may be, however, that this message is a `DL/UL Information Transfer` message, which has embedded in it a NAS message. This NAS messages are used to communicate between the UE and the MME and are also cryptographically protected and have their own, special encoding.

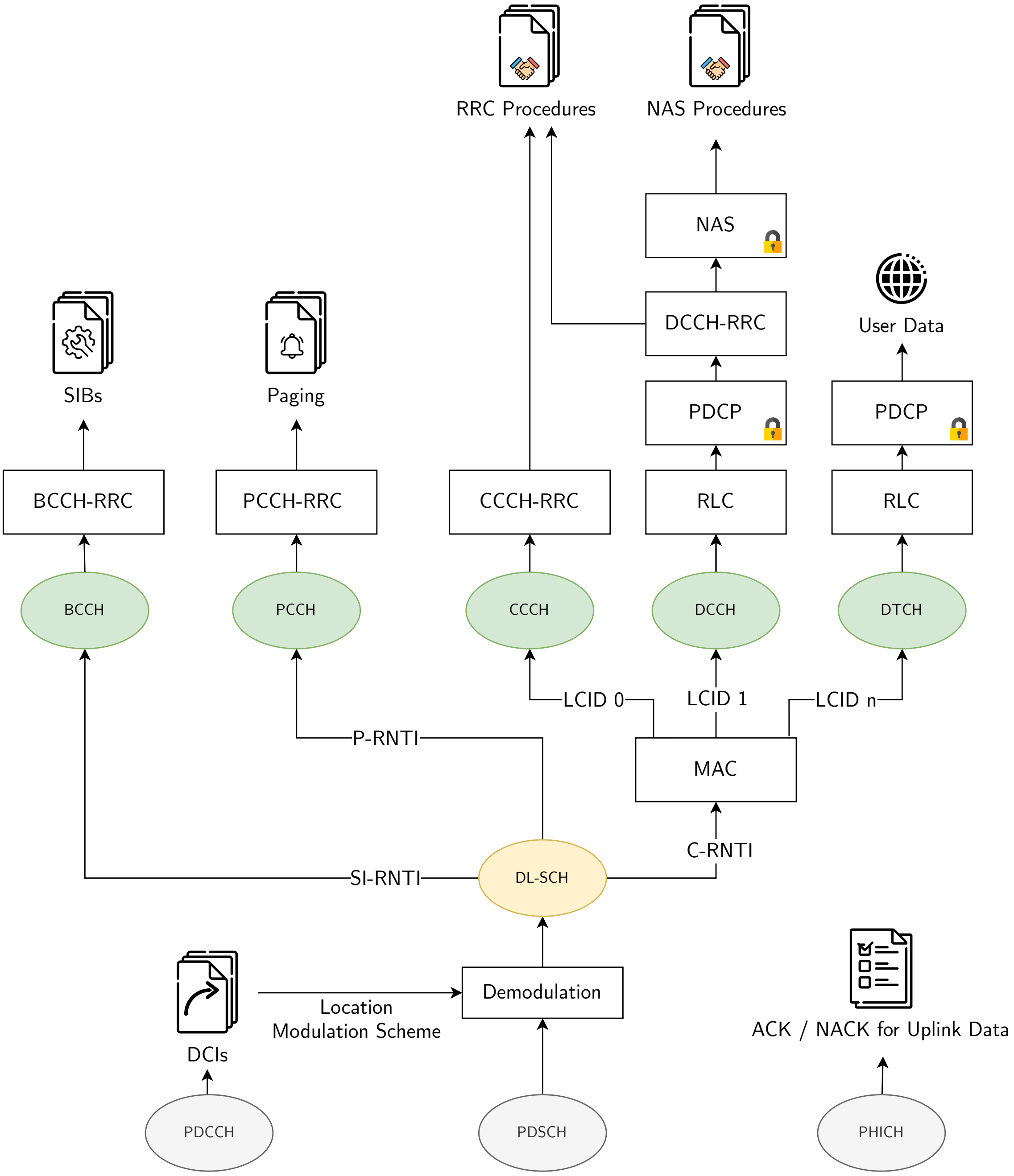


Figure 2.3: Overview over Downlink Channels and Layers

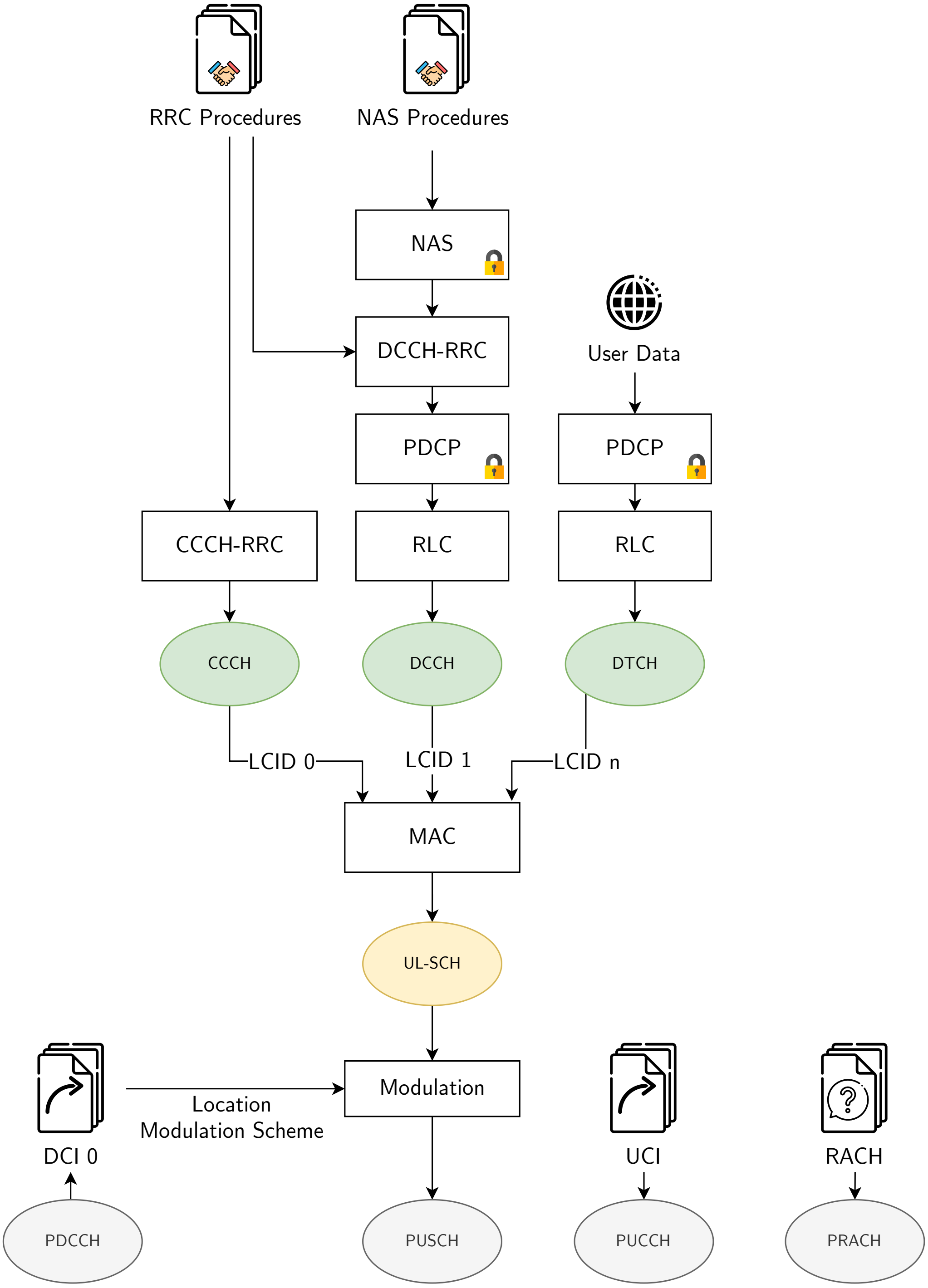


Figure 2.4: Overview over Uplink Channels and Layers

Chapter 3

Existing Attacks

A literature survey was conducted to identify pre-existing attacks against the availability of LTE. In each part, these attacks were analyzed according to the following metrics:

1. **Persistence** - How fast can legitimate communication resume after the jammer is turned off? Does it require human intervention (e.g. reboot of the cellphone)?
2. **Power Required** - How much peak power does the attacker need with respect to the original signal strength for the attack to succeed?
3. **Synchronization Effort Required** - Does the attacker require to synchronize with a legitimate eNodeB or UE?
4. **Type of Exploit** - Is this flaw an oversight of the operator of the network or manufacturer of the UE or is it something that by design cannot be avoided?
5. **Selectivity** - Can the attack, either by itself or by some additional measure, target specific UEs?

3.1 Blanket Jamming

These unsynchronized and non-protocol aware attacks typically only require low-cost equipment, capable of generating noise in the desired frequency bands, as explored by Krenz et al. [10]. There, they explored narrowband jamming, and reached the conclusion that the jamming signal must be at least 3dB stronger at the UE. Improvements have been suggested by Romero et al. [11] in 2019, where they swept through the frequency spectrum with a sweeping period of several microseconds. They surprisingly reached a large Error Vector Magnitude (EVM) of around 15% with a jamming signal 20 dB weaker than the legitimate signal.

3.1.1 Analysis

The upside of such attacks is that they are easily implemented and verified, no synchronization or protocol awareness is needed, other than choosing the correct frequency spectrum. The attacks suffer from a low persistence; however, since as soon as the jamming device is turned off, communication is possible again.

Furthermore, in areas with a large overlap of different cells, the jammer must jam on an even larger spectrum to block all communication, as the UE will otherwise choose a different cell. This translates directly to having to use a large, expensive amplifier, requiring a large amount of energy, or having to re-tune the amplifier to allow hopping or sweeping over the frequency bands.

Because the attacker disables all communication, also a targeted attack on a single UE is not directly possible. One could think of a scenario, however, where the attacker would use high-gain directional antennas, pointing at a stationary victim UE in their line-of-sight and thereby only jam the communication of this UE.

3.2 Jamming of Physical Channels

In 2014, Kakar et al. [12] explored the jamming of the Physical Control Format Indicator Channel (PCFICH) channel, where the size of how the DCI is transmitted. As this is a very sparse channel, it doesn't require much more power (2dB) to jam it. In the paper of Lichtman et al. [13] from 2016, they analyze all the physical channels of LTE, down- and uplink, and analyze the theoretical amount of effort required to jam these channels. Rao et al. analyzed in 2017 [14] this in an experimental setting and found that by overlaying the Cell-Specific Reference Signals (CRS) with a fake one, a jammer is most likely to induce a high error rate at the UE, even with a low J/S of -26dB (by their definition of J/S).

3.2.1 Analysis

The good thing with these kinds of attacks is that they require lower output power than a simple Radio Frequency (RF) jamming as seen before. However, the implementation effort is greater, as the jammer requires tight synchronization to the base station, and the persistence again is very low. The attacker also needs to do this on all cells in the vicinity to prevent communication effectively, as seen before, requiring multiple devices or hopping between the available frequency bands. As with the blanket jamming attack, there is no selectivity on the UE level possible, without resorting to directional antennas.

3.3 Fake Base Station Attacks

A Fake Base Station (FBS) attack is more complex to execute than a simple RF jamming attack but still simpler than jamming specific physical channels, because the attacker does not need to synchronize to any existing base station and emulate its properties at all. The attacker just needs some Open Source software to run the base station, as well as a SDR. A UE will try to connect to this base station, provided the Mobile Network Code (MNC) matches its expectation, but will eventually fail the MME attachment process, as the fake base station isn't really connected to the core network at all.

3.3.1 Messages impersonating the MME

Reject Messages

A number of papers have been exploring on how to set up such a fake base station ([15], [16]) and what to do with it. Considering availability, there are several different reject messages from different procedures that can be sent from the fake base station, each with a very similar result. Three of those procedures and their corresponding reject messages have been examined in the papers found.

Each of those procedures operates on the `NAS` level, meaning that the fake base station also imitates parts of the core network which normally takes part of such procedures. Upon reception of such a reject message, the UE will enter the `EU3 Roaming not allowed` state, meaning that any cellular services are disabled. In older UEs, this attack is persistent and the UE will not try to re-attach again by itself, only if the SIM card is re-inserted or the UE is restarted. However, the 3GPP standard 24.301 [17] has since been updated to address this, and newer UEs should still try to re-attach - up to a UE specific amount of times.

In 2016, Jover [18] sends a UE that connects to the fake base station an `ATTACH REJECT` message. In 2017, Shaik et al. [19] set up a fake base station with a different tracking area code, allowing them to send the `TRACKING AREA UPDATE REJECT` message. Finally, in 2018 Hussain et al. discover [20] that the `AUTHENTICATION REJECT` message has a similar effect.

Network Initiated Detach Request

Furthermore, in the same paper [20], they claim that a network initiated `DETACH REQUEST` sent to the UE will, depending on the detach type sent, result in the UE requiring manual restart or SIM reinsertion to receive service again. In the standard, on p 166 [17], it is indeed mentioned that in some cases, user interaction is necessary to re-activate the Evolved Packet System (EPS) bearers, hinting that this might not be as persistent on some UEs. Unfortunately, however, the EPS Mobility Management (EMM) cause value is not explored, as this could a high impact on the reaction of the UE. As per the standard, for some cause values, the timer T3245 is started upon reception of the detach request message, and upon its expiry, a re-attach procedure is started again.

Furthermore, the authors claim that these messages are - per the standard - not cryptographically protected. This couldn't be verified in the standard, as in p.53 of [17], this detach request message is not listed as an exception of requiring integrity protection at all times. As such, it remains doubtful if this attack could really work on modern UEs. The authors unfortunately fail to give a list of precise UE models used.

3.3.2 Analysis

The benefit of such an approach is that the attacker doesn't need to synchronize with any existing base station and must only modify the behaviour of a base station that he operates. Executing attacks that target a higher layer of the LTE protocol are therefore much easier to execute in this context. This is highly beneficial, as the attacks on this level have the potential to be much more persistent compared to a simple analog jammer. It is also possible to target such an attack to a specific UE, as the attacker learns the IMSI and optionally the International Mobile Equipment Identity (IMEI) before the attack is launched.

However, this does not necessarily apply to newer UEs, as the message sent is not integrity protected. Furthermore, such a fake base station must send with comparatively high power, otherwise, the UE will not attach to it in the first place.

3.3.3 Possible Countermeasures

Fake base stations have a number of negative impacts, not only availability as explored, but also regarding privacy as they allow the usage of IMSI catchers as shown in [15]. There is a lot of research around exploring how to detect these fake base stations acting as ISMI catchers, as the problem is also present in 2G and 4G networks, see the following: [21], [22], [23], [24], [25], [26].

In the literature, the detectors work by listening in on the configuration messages broadcasted by the eNodeBs, and evaluating their properties over time. These pieces of information are either collected through SDRs, or crowd-sourced by Android applications. Detectors by operators also use the reports provided by UEs to determine any base stations in the vicinity that shouldn't be there. As a side note - we aren't evaluating IMSI attacks in this thesis, but using the methods described in the following, it is possible to build such an IMSI catcher without triggering any of those detection mechanisms at all.

Hussain et al. argued in [27] that the root of these problems lies in the unauthenticated broadcast configuration messages (MIB & SIB) as well as the initial setup between the UE and the base station. According to him, the solution would be to authenticate all those messages. In a Dolev–Yao adversary model, some attacks on such a broadcast system will always be possible, including forms of fake base station attacks. Notwithstanding, it also does not prevent attacks targeting an active connection between a UE and a correct eNodeB, as explored in the next section.

3.4 Downlink Overshadowing

These kinds of attacks have become relevant only recently, as they are more complex to implement and require tight synchronization between a real eNodeB, UE and the attacker. However, compared to a FBS attack it has the big benefit that the attacker does not need to force the UE to connect to his eNodeB and may therefore use comparatively lower signal power. It has been explored in 2019 by Yang et al. [7] that the received signal strength at the UE must be around 3dB higher than the legitimate one. An even lower number also been shown in our evaluation, see section 7.2.

3.4.1 Paging Channel

Hussain et al. were the first in 2018 [20] to discover that they may hijack downlink transport channels on LTE. They identified the paging channel as the first vulnerable one and overshadowed this channel with empty messages, causing the UE to miss all incoming calls. The UE is still able to connect by itself, though. They also could inject paging messages containing the IMSI of the victim, causing the UE of the victim to drop all current connections and reconnect again. This was also examined in 2019 by Yang et al. [7]. Yang et al. also discovered that upon injection of a paging message of type Circuit Switched (CS), the UE will downgrade to 3G. This is because 4G is Packet Switched (PS) only, but a phone call from a UE supporting CS only must still work.

3.4.2 System Information Channel

Yang et al. observed in their SigOver [7] paper that they may also overshadow SIB messages on the downlink. They identified that the SIB2 message carries a `BarringFactor` parameter, which when set to 1.0, no UE is allowed to connect to this cell at all. This is accompanied by a `BarringTime`, which specifies how long the UE may not initiate another connection with this cell, making the attack rather permanent. When accompanied by a paging message carrying the `System Information Change` flag, active UEs will drop their connection as well.

3.4.3 RRC Connection Procedure

In a talk given in 2019 at 36c3 [28], they showed that older UEs will accept any `RRC Connection Release` message sent by the network, even if they aren't integrity protected, thereby releasing their current connection and initiating a reconnection procedure.

3.4.4 Analysis

The hijacking of an active downlink connection is rather complex to implement, as many of the channel properties of the correct eNodeB need to be duplicated for the attack to work. In turn, however, they achieved high persistence with the SIB2 Barring attack. The other downlink attacks mainly caused the UE to drop its current connection, but it will try to reconnect straight away. In spite of this achievement, also the SIB2 Barring attack could only attack one single cell. In reality, the UE could switch to a different available cell, causing the attacker to require one active radio for each cell in the vicinity. In [7], it was stated that to overshadow the signal successfully, the power of the attacker signal must be around 3dB higher than the legitimate one.

Regarding selectivity, the paging attacks are per their very nature, selective, as they require the UEs identifier to be executed. The SIB Barring attack, however, is not selective and will disable the communication to all UEs in the range of the attacker.

3.5 Rogue UE

The interaction between UE and MME was studied in 2019 by Kim et al. in [29], Hussain et al. in 2018 [20], and [30]. However, they did not hijack the active connection on the radio layer as has been done by the SigOver paper by Yang et al., but rather used a rogue UE to carry out their attacks. In their papers, they mainly uncovered implementation issues of their chosen operators and UEs.

3.5.1 eNodeB Resource Exhaustion by RRC connection requests

When acquiring a lot of connections with a eNodeB, it cannot make way for legitimate ones. Because this initial Radio Resource Control (RRC) connection has not set up any security, this is easily possible and very hard to prevent. In [29], they only acquire 20 connections per second, which is enough to drain their test eNodeB of all available resources. This can be easily improved by using a more efficient jammer architecture. Then again, the effect will be the same as that of an RF layer noise jammer on the uplink frequency spectrum.

3.5.2 Establishing a 2nd RRC Connection with the same Identifier

In [29], They acquired the Serving Temporary Mobile Subscriber Identity (S-TMSI) identifier of their victim UE and subsequently attached to the eNodeB with the same. They observed that some operators subsequently released the RRC connection with the victim and did not deliver any due paging message as long as the rogue connection stayed active. The attack suffered however from low persistence, as within an average of 0.5s the victim was connected again.

3.5.3 Spoofing NAS Attach/Detach Messages

These attacks have been covered in [29] and [30]. By establishing a connection with the eNodeB and subsequently sending Non-Access Stratum (NAS) messages that are either plaintext, replayed or fail the integrity check, they manage to disconnect the connection with the correctly attached UE. This is, however, an implementation failure of the operators because in the standard [17] these cases have been exhaustively covered.

3.5.4 Authentication Synchronization Failure Attack

This attack, covered in [20], exploits the design of LTE by breaking the sequence number sanity check. The eNodeB is expected to use a sequence number with a certain range for the SIM card to successfully authenticate the `AUTHENTICATION REQUEST` message sent from the MME. However, by connecting as a rogue UE to the MME and sending a lot of `NAS ATTACH REQUEST` messages, the sequence numbers between the UE and the MME get out of sync, as the sequence number at the MME increases for each `AUTHENTICATION REQUEST` received. When the legitimate UE tries to connect again, it will fail, requiring the execution of the re-synchronization protocol. This requires the attacker to constantly launch the attack.

3.5.5 Analysis

Compared to the downlink attack, no active uplink connection was hijacked by the authors of the 2 papers analyzed. Rather, they modified an existing open source implementation of a UE to behave as they liked. Because they impersonate a UE, the power required is not higher than any normal UE and thus one of the most efficient attacks in this regard. However, they could not identify a high persistence DoS attack, as all participants recovered quite fast from the incident after the jammer turned off. Despite this, the attacks they considered were highly selective, as they required obtaining and using identifiers of the UE for their respective attacks. This also allowed them to verify their attacks in the real world.

Chapter 4

Static Attacks

As has been established in the previous chapters, the availability of SDR technology has enabled new attacks, mainly concerning the hijacking of existing communications. To facilitate further research on this topic, an attack infrastructure has been developed to evaluate the feasibility and impact of these attacks rapidly. While developing such infrastructure, it makes sense to replicate existing state-of-the-art attacks from Yang et al. from [7]. They have been implemented and described in this chapter.

4.1 PDSCH Overshadowing

All attacks that are described in the following sections and chapters use this attack as their basis. Thus, it makes sense to explore this first. The PDSCH channel is one of the most important physical channels in LTE, as all of the user and most of the control data is encoded on this channel. Overshadowing such a channel means that an attacker can exchange the content of this downlink channel from a real eNodeB with any other content he/she likes. To achieve this, the attacker signal must be sent with on same frequency, with the same timing and with at least 1.8dB higher power (see section 7.2) than the real eNodeB. Figure 4.1 illustrates this - the bottom signal is the attacker signal, sending in sync with the real eNodeB signal on top.

To successfully overshadow a legitimate connection, one must thus overshadow the PDCCH with DCI messages of higher power. The PDSCH data then can be placed at any location in

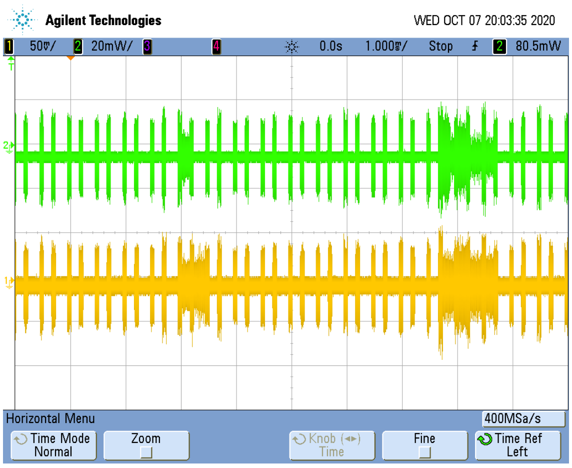


Figure 4.1: Trace from Oscilloscope - Top: eNodeB signal - Bottom: Attacker signal

the resource grid, where it may or may not overshadow other legitimate data. To that end, when talking about PDSCH overshadowing, it always means PDCCH overshadowing as well.

4.2 IMSI Paging

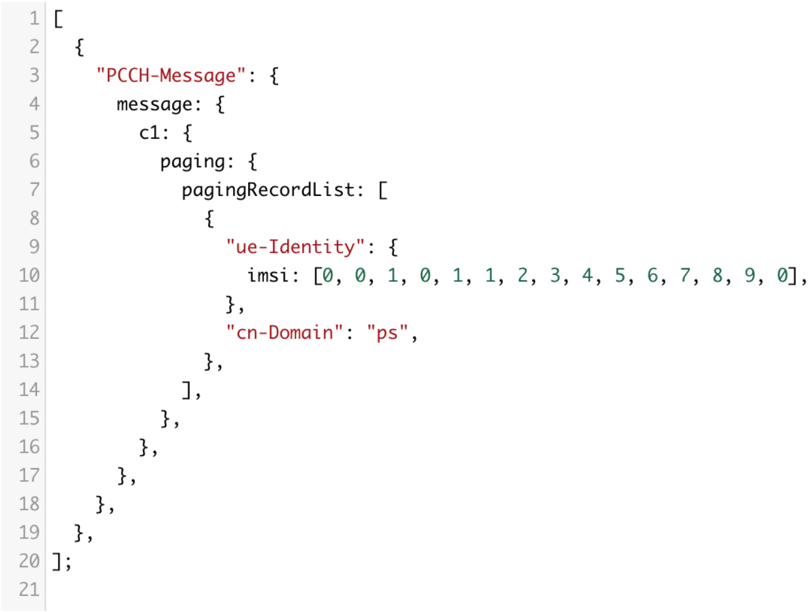
One of the most basic attacks is to overshadow the paging control channel, also known as PCCH, with a IMSI paging message. According to the 3GPP technical specification 24.301 [17] in section 5.6.2.2.2, upon reception of a paging message containing its IMSI, the UE shall detach from the EPS and delete any key material associated with it. The UE shall then initiate an attachment procedure again. Thus, if an attacker can inject such a paging message continuously on the downlink of a legitimate cell, the victim associated with this cell will suffer from a Denial of Service (DoS) attack.

This attack was discovered by Yang et al. [7]. The first drawback is that the attacker needs to be in posession of the IMSI of the victim. While one might argue that the acquisition of an IMSI is possible with an IMSI catcher, it still requires either the setup of a fake base station, with comparatively very high power requirement. Second, it requires a continuous transmission, as the UE will immediately reconnect. Still, it is a relatively simple method of disconnecting a UE temporarily and forcing it to re-acquire its EPS context, while being a highly selective attack.

As it is is still power-intensive listening to and decoding everything sent on the downlink channel, the UE will listen for paging messages only on a small number of specific subframes, based on a few parameters. Among the parameters of the is the IMSI of the UE, as well as its Discontinuous Reception (DRX) cycle specification, which is sent in an RRC Reconfiguration Message by the eNodeB after a secure connection has been established.

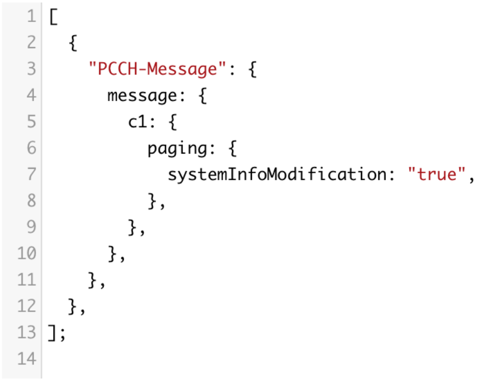
Thus to precisely overshadow only the subframes that the UE will listen on, these two parameters must be known. As the IMSI is, by definition of this attack, known to the attacker, only the DRX parameter remains. However, these parameters are sent in a `RRC Connection Reconfiguration` message after the RRC security context has been set up. Thus, as an attacker, it is only possible to guess when the UE will actually listen to the paging message. The easiest solution is thus to send the paging on every subframe.

The paging message sent is encoded using Abstract Syntax Notation One (ASN.1) with the schema defined by 3GPP. A generated paging message for a test IMSI is shown as JSON in figure 4.2.



```
[
  {
    "PCCH-Message": {
      message: {
        c1: {
          paging: {
            pagingRecordList: [
              {
                "ue-Identity": {
                  imsi: [0, 0, 1, 0, 1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0],
                },
                "cn-Domain": "ps",
              },
            ],
          },
        },
      },
    },
  },
];
```

Figure 4.2: IMSI Paging Message sent for IMSI 001011234567890 decoded as JSON



```
[
  {
    "PCCH-Message": {
      message: {
        c1: {
          paging: {
            systemInfoModification: "true",
          },
        },
      },
    },
  },
];

```

Figure 4.3: Paging message with `systemInfoModification` set to true

4.3 SIB Cell Barring

The SIB1, one of the broadcast configuration objects, contains among other parameters the boolean flag `Cell Barred`. If this flag is set to true, then the UE will not try to associate with this cell at all and will try connecting to a different one in the vicinity. Overshadowing this flag is thus a potent attack, although still limited in persistency. It is not possible to attack UEs selectively with this attack.

As the SIB messages are sent on the PDSCH channel as well, the same approach from the previous section can be used. However, a different RNTI needs to be used, in this case, `0xffff`, or also known as the System Information RNTI (SI-RNTI). All SIB messages are sent using the SI-RNTI. Regarding scheduling, it is simpler than for the paging messages, as the SIB1 message is always sent on subframe 5 at every even frame.

To further help a victim UE to decode this SIB, the reference signals are generated by the attacker and are sent on all subframes, not just the ones carrying the SIB1. This is to give the channel estimator of the UE time to estimate the channel properties of the attackers signal. Without these reference signals, the message is often not decoded when the eNodeB and attacker channel differ a lot.

When constructing the fake SIB to send out, the real SIB sent by the cell must serve as a template, because the attack is otherwise easily detectable due to the changing parameters, very similar to fake base station attacks. Therefore, the simplest approach is first to decode the correct SIB1 as sent by the real cell, modify the `cellBarred` parameter to `barred`, and then send it out again. To encode the message, ASN.1 is used with the `BCCH-DL-SCH` schema provided by 3GPP.

The attack, as stated, would work on UEs that are not yet associated with the cell attacked. However, already associated UEs do not decode SIB messages all the time, only when there is a notification of system information modification delivered through a paging message. Sending this is done with the same procedure as in section4.2, only with a different paging message, which is shown in figure 4.3 as JSON. Together, this attack now effectively denies any UE the use of a cell.

Chapter 5

Reactive Downlink Overshadowing

As we have seen in the previous chapter, static overshadowing attacks can be powerful DoS attacks for a single cell. However, in the real world, most of the time there are multiple cells to choose from, making the impact of the attacks seen rather limited. Furthermore, even with the presence of a single cell, the attacks suffer from low persistence. This means that if the UE gets outside of the attackers range or the attack is stopped, the connection is regained quickly, as evaluated in section 7.3 and 7.4.

Thus, the aim of this chapter is to establish attacks that use the same amount of power, but are more persistent and impactful in a real network. Furthermore, as all the attacks are reactive, they require the use of a downlink decoder. For acquiring the ability to selectively deny UEs service with the attacks presented, additionally an uplink decoder would be necessary to decode the request messages.

5.1 RRC Connection Reject

This first attack targets the RRC connection establishment procedure and results in the UE eventually choosing a different cell, but could also serve as a preliminary step for further attacks.

The attack procedure is depicted in figure 5.2 while the regular contention-based RRC connection setup procedure is shown in figure 5.1. Before associating or during re-association with a cell, the UE will perform a random access procedure. During normal interaction, the eNodeB will allocate via a DCI 0 when the UE is allowed to send data on the uplink. However, at first, the UE must make itself heard. This is achieved via a message sent at a pseudo-random time, containing a Physical Random Access Channel (PRACH) preamble chosen by the UE. The purpose of this preamble is to distinguish between different UEs connecting at the same time.

Next, the eNodeB will decode and acknowledge the reception of the PRACH preamble and allocate uplink resources to the UE via the PRACH response message. This is sent on the PDSCH channel with a RNTI between `0x0001` and `0x000A`. Which one will be used depends on the timing of the first message from the UE. An attacker decoding the downlink can decode this message as well, triggering the attack. In addition to the uplink allocation resource, the RNTI allocated to the UE is contained therein, which will be used for all further messages between the eNodeB and the UE.

Whenever the `PRACH response` message is received, the attacker now sends out a `RRC Connection Reject` message to the UE with the RNTI indicated, but must do this faster than the real eNodeB can. In practice, it has been observed that there is around 9ms between the PRACH response and the legitimate `RRC Connection Setup` message.

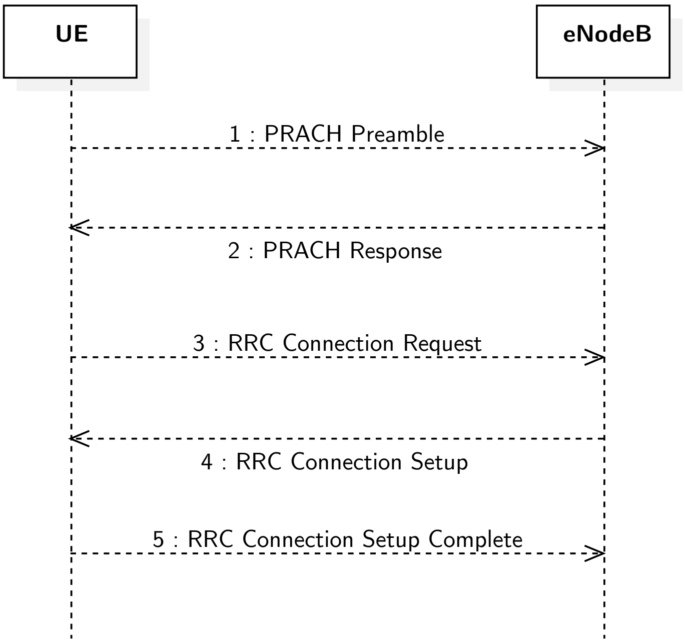


Figure 5.1: RRC Connection Setup Procedure (Contention Based)

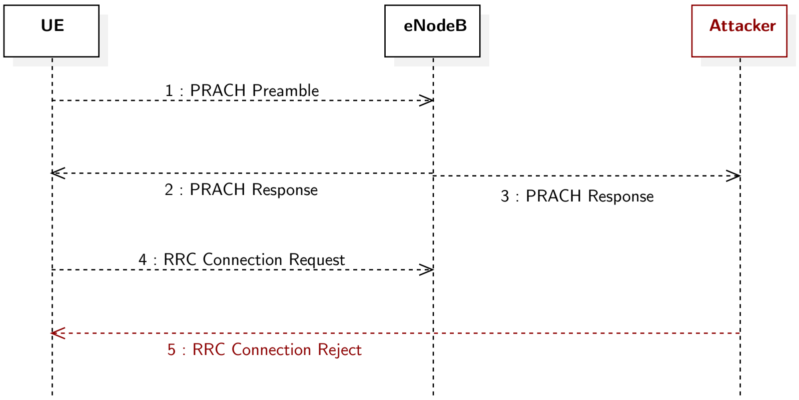


Figure 5.2: RRC Connection Reject Attack Procedure

The `RRC Connection Reject` message doesn't contain much information, only a retry parameter, indicating when the UE should try to connect to this cell again. This can be set to at most 16s, after which the UE will try to associate again. In the meantime, it will not try to associate with any other cell. This could make it a better attack than the simple SIB1 overshadowing, especially in an urban environment. With many cells available, the UE will choose the next available cell if one is attacked with the SIB1 barring attack.

With a corresponding uplink sniffer, it would be possible to target this attack to a set of UEs, based on the TMSI identifier in the `RRC Connection Request` message.

5.2 NAS Attach Reject

5.2.1 Overview

The impact of previous attacks is limited to a single cell. If there are multiple cells in the vicinity of the UE, the attacker must target all cells simultaneously, increasing cost and complexity of the attack. Therefore, attacks targeting the interaction with not only the eNodeB, but the MME component of the EPC, should have a much larger impact on the UE.

The attack procedure is shown in figure 5.3. When analyzing the attachment procedure after the RRC Connection Setup Complete, the next message sent by the UE is, if it has not previously established bearers it may use, a NAS Attach Request message. After the Attach Request message has been sent, the MME can decide to deny this request right away, ask for more information from the UE or enter the authentication procedure. However, in this attack, where we are pretending to be the MME, we are sending a non-integrity protected NAS Attach Reject message and overshadow all other messages that may be sent by the MME.

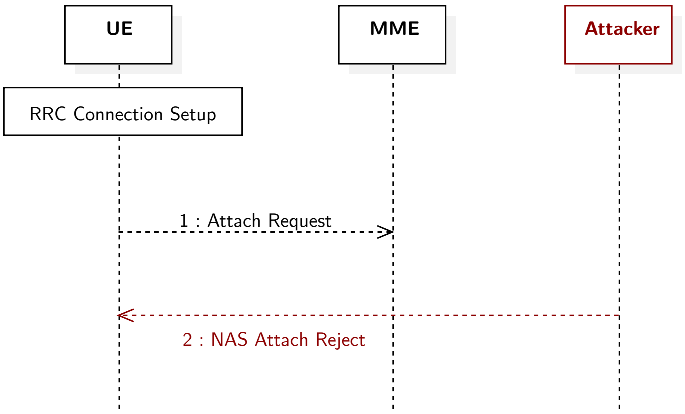


Figure 5.3: NAS Attach Reject Attack

The NAS Attach Reject also carries a cause value. The action taken by the UE upon reception of the reject message depends on that value - the most drastic action is for the value 3, standing for Illegal UE. In 3GPP 24.301 [17], it is stated that upon reception of an Attach Reject message with this cause value, the UE shall consider this Public Land Mobile Network (PLMN) as forbidden and shall not try to reconnect again, unless some timer with a large timeout expires. The UE may, however, reconnect with a cell from a different tracking area.

Our evaluations in section 7.6 confirm these findings.

5.2.2 Integrity Protection

We are sending the reject message without any integrity protection. However, newer versions of the standard have an explicit section about the special treatment of non-integrity protected messages. They state that, upon the reception of a non-integrity protected NAS Attach Reject message, the UE may retry the connection attempt. Although the amount of retries is limited, it is up to the UE how large this number is.

5.3 NAS Service Reject

Compared to the NAS Attach Reject attack presented in section 5.2, this targets UEs that are already associated with the core network. As UEs typically stay attached unless turned off or enter flight mode, this attack can target the majority of the UEs. This is because a service request is sent whenever the UE would like to establish some connectivity - e.g. connecting to the internet, making a call, reacting to a paging message or similar.

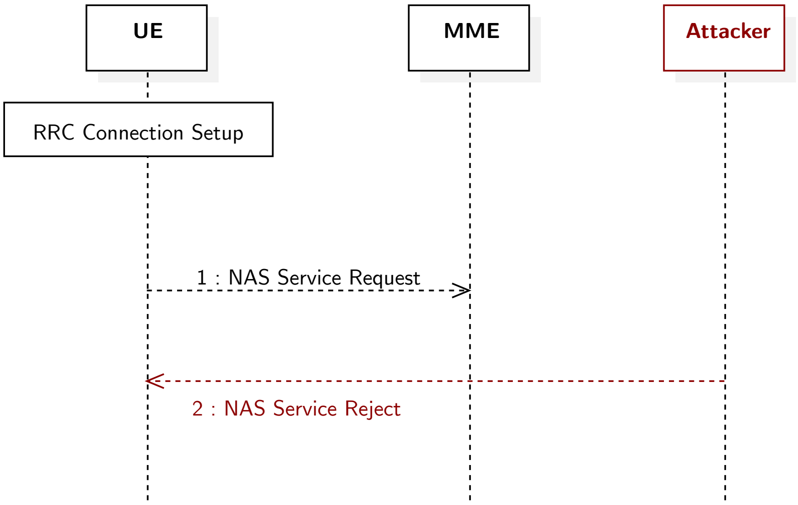


Figure 5.4: NAS Service Reject Attack

The attack flow is depicted in figure 5.4 and is very straightforward. For brevity, the eNodeB as the middle component has been left out. After having attached with the eNodeB and set up a RRC connection, the `NAS Service Request` message is sent to the eNodeB, which forwards it to the MME. The MMEs response, however, is overshadowed by the attacker and replaced by a `Service Reject` message with EMM cause value 3. According to the 3GPP TS 24.301 [17], this now puts the UE in the state where it will not try to re-attach with any cell of this PLMN again, unless some timer with a large value chosen by the UE expires.

The advantage for this kind of attack is that the attacker will be able to deny service to a UE that has otherwise an active connection, which is now just idling. It would thus be suited for targeting many active UEs in the attacker's vicinity.

See section 7.7 for a detailed evaluation of the effect of this attack, confirming its large impact on availability.

5.4 NAS Authentication Reject

5.4.1 Overview

To authenticate a UE to the network and derive the session keys, the NAS authentication procedure is used, as depicted in figure 5.5, taken from [17]. Should the authentication fail, however, the network will send a `Authentication Reject` message back to the UE. The attack is therefore to send an `Authentication Reject` message to the UE after it has received a valid `Authentication Request` by the network and sent back the `Authentication Response`, see figure 5.6. In the process of doing so, any other messages coming from the MME are overshadowed.

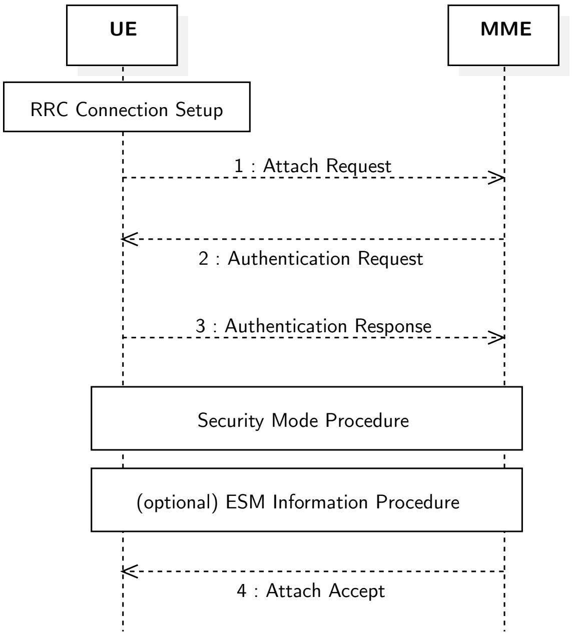

UE
MME
RRC Connection Setup
1 : Attach Request
2 : Authentication Request
3 : Authentication Response
Security Mode Procedure
(optional) ESM Information Procedure
4 : Attach Accept

Figure 5.5: NAS Authentication Procedure

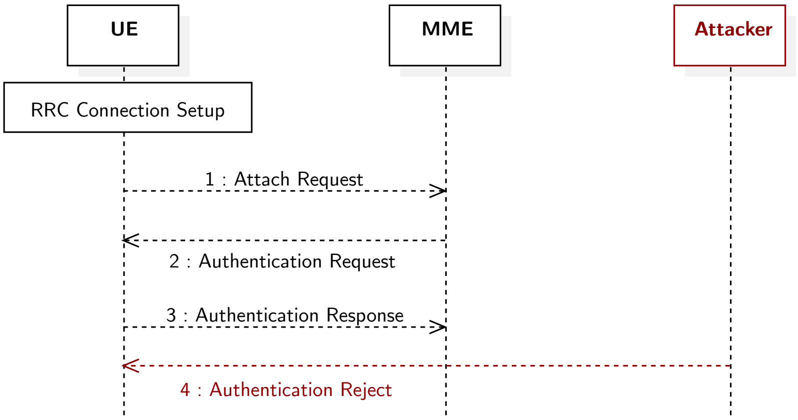

UE
MME
Attacker
RRC Connection Setup
1 : Attach Request
2 : Authentication Request
3 : Authentication Response
4 : Authentication Reject

Figure 5.6: NAS Authentication Reject Attack

5.4.2 Integrity Protection

We can only send this `Authentication Reject` without integrity protection. As described in section 5.4.2.5 of 3GPP TS 24.301 [17], if this message is received without integrity protection, the UE will wait at least 30-60mins before trying to re-attach again. In the course of doing so, it will search for a cell in another tracking area, depending on the implementation of a counter for `SIM/USIM considered invalid for GPRS services`.

Retrying often in this scenario makes sense for the UE, as it could successfully verify the integrity of the first `Authentication Request` message, then why should the MME now suddenly be unable to send the `Authentication Reject` message with integrity protection?

Aside from the low persistence, a further downside to this attack is that this authentication procedure is only done when the UE has lost all security context with the MME, thus it is only rarely applicable.

5.5 NAS Authentication Request Forgery

5.5.1 Overview

From section 5.4, we established that maybe it isn't such a good idea to let the authentication procedure proceed that far before intervening as an attacker. Thus, we devised the following attack, targeting an earlier message - the `Authentication Request` from the network. The procedure is outlined in figure 5.7.

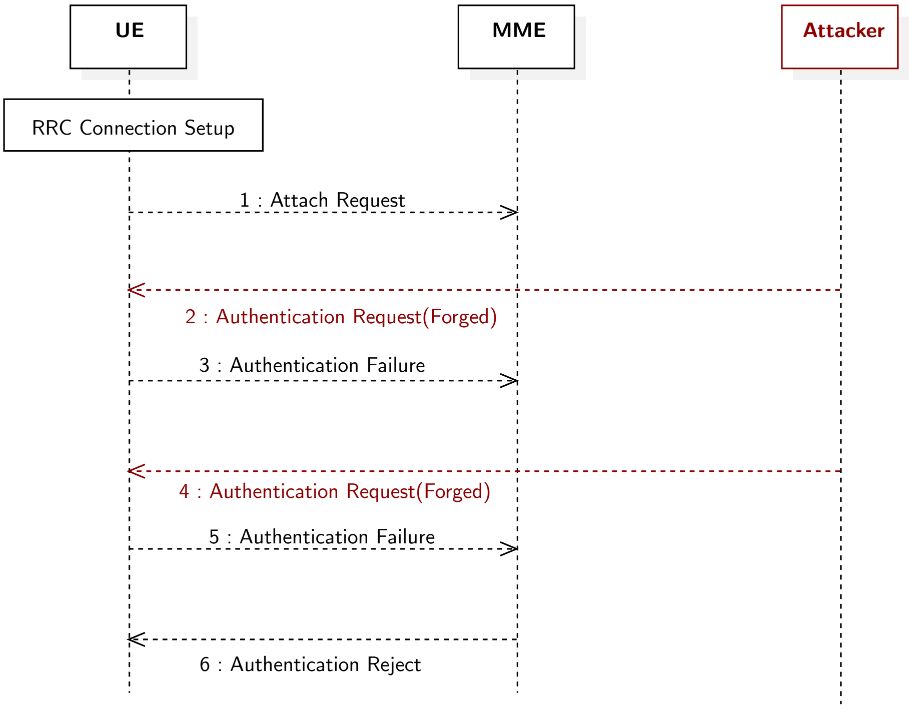

UE
MME
Attacker
RRC Connection Setup
1 : Attach Request
2 : Authentication Request(Forged)
3 : Authentication Failure
4 : Authentication Request(Forged)
5 : Authentication Failure
6 : Authentication Reject

Figure 5.7: Authentication Request Forgery Procedure

This attack assumes that the UE has either not associated with the network before, or the keys from the last session have been expired. Thus, the network decides to re-authenticate the UE via a `NAS Authentication Request` message. In a normal procedure as outlined in figure 5.5, the UE will respond to any valid `Authentication Request` message with a `Authentication Response`. It can verify the validity of the `Authentication Request` message by inspecting the `AUTN` parameter of the message, which is a MAC over the `RAND` parameter. The `RAND` parameter is then used to derive the various keys for this session.

The idea of this attack is now to overshadow the `Authentication Request` message to the UE with a faulty one. As the UE cannot verify the integrity of the `RAND` parameter, it will respond with a `Authentication Failure` message, with a cause of *wrong authentication*. The network, as stated in TS 24.301, [17], may retry, but again, this message would be overshadowed and the UE sends a failure message back to the network. According to [17], when the UE sends 2 `Authentication Failure` messages in a row, it will cause the network to send a `Authentication Reject` message. In practice, however, one such `Authentication Failure` message is enough to cause the MME to send the desired `Authentication Reject` message, see the evaluation in section 7.8.

Chapter 6

Reactive Uplink Overshadowing

We have seen that reactive downlink overshadowing enables attacks that are both very power efficient and have a high persistence, which is unprecedented in recent literature. However, it is still possible to improve things further. The main disadvantage now for an attacker is the fact that the physical range of overshadowing attacks is still limited, covering a large area such as an entire cell is still not easily possible.

This is where reactive uplink overshadowing comes into play. The main idea is to reactively attack higher-level protocols to get a higher impact on the UE, but only use downlink decoding and uplink overshadowing to do so. This means that the attacker may be anywhere within the cell coverage and still be able to target any other UE within the same cell, without having to invest in a lot of output power and equipment that would be necessary with downlink overshadowing only.

Although the uplink overshadowing has not been implemented, we are very much convinced of its technical feasibility. Still, the following is an example of an attack utilizing uplink overshadowing, combining low energy requirements, maximum range, and high persistence.

6.1 Forged Attach Request

6.1.1 Overview

It is assumed that the attacker is able to listen on the downlink and therefore on the RRC Connection Setup message, serving as a trigger. After receiving this message, the attacker overshadows the combined `RRC Setup Complete / NAS Attach Request` message of the UE with his own. In his `NAS Attach Request` message, he may put whatever he wants. If we limit ourselves to the IMSI field of the attachment procedure, we can identify 2 main cases. How real MNOs react to those cases has been evaluated in section **??**, while the evaluation of the impact of the authentication reject is found in section 7.8.

6.1.2 Rogue IMSI Variant

In this case, the attacker sends an attach request message with a known-good IMSI of the network, that belongs to another user. Such an IMSI can be obtained either live by sniffing the uplink and waiting for such a message, or buying a SIM card from the MNO (which would not be advisable if the attacker wished to remain stealthy). In either case, the network will proceed with the authentication procedure. But, as the key material is for the wrong user, the `Authentication Request` will fail at the UE, which will respond with an `Authentication Failure` message. The network may try again to authenticate the UE, which will result in the

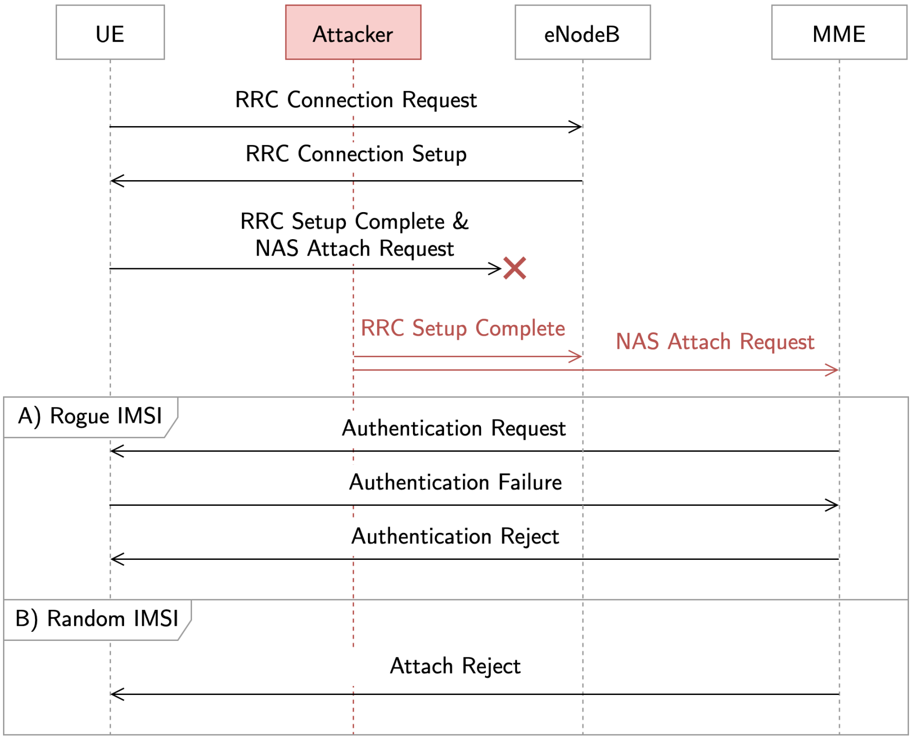


Figure 6.1: Attach Request Forgery Procedure

same outcome, or abort the procedure with a `Authentication Reject` message. This will in turn put the UE in a state where it will not try to re-attach to the network again by itself.

6.1.3 Random IMSI Variant

In the case that the attacker has no known-good IMSI which to put into the `Attach Request` message, he may also resort to a random IMSI. In this case, the network responds with a `Attach Reject` message with a cause value chosen by the MNO implementation. It may be that, depending on the cause value, that the UE will continuously try to re-attach again, as defined in 3GPP 24.301 [17].

Chapter 7

Evaluation

7.1 Lab Setup

The hardware used for most of the experiments is depicted in figure 7.1. It consists of an Amarisoft Callbox Mini as eNodeB and 2 Ettus Research USRP B210 SDRs, one of them acting as the attacker and the other as the decoder. Not all of these components are used in every experiment, and any variations are outlined in their relative sections.

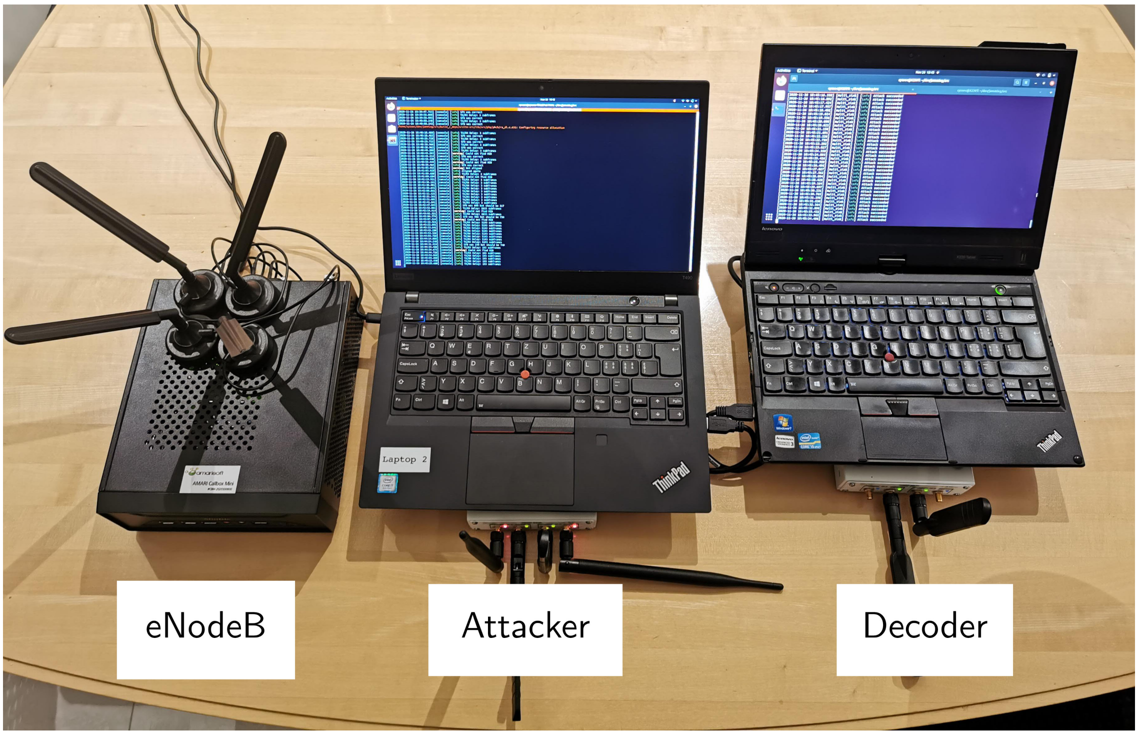


Figure 7.1: Components used in the Lab for Evaluation

7.2 J/S Analysis

A central number in analyzing different jamming system is the ratio between jamming and real signal strength, J/S. Yang et al. reported [7] that their approach has a success rate of 98% for a J/S of 3dB. For traditional blanket jamming, numbers of 3dB J/S have been reported by Krenz et al. in [10]. It is important for the J/S to be not too high, as it translates directly into lower output power requirements and longer attack range. To get an idea of how PDSCH Overshadowing compares with other approaches and verify the numbers seen in other papers, we measured the J/S on our own.

7.2.1 Experimental Setup

We run this experiment in a single-channel configuration (SISO, 1 TX, 1 RX antenna) and used the equipment as is depicted in figure 7.1, as well as different smartphones as UEs.

Relative Gain

Each experiment is run with a USRP B210 running as an attacker with different output gains applied. However, the B210 and Amarisoft Callbox are not factory calibrated to any absolute output power. As such, simply comparing their configured output gain numbers would yield no useful information. The attacker signal is generated with GNU Radio [31].

Additionally, as the property of a signal sent by the attacker for different attacks varies significantly, we compare each on their own. After starting the eNodeB and the attacker, we connected the TX output of both using a cable with the same type and length to the DSO6104A oscilloscope from Agilent. Using the `ratio` measurement tool on the oscilloscope, we could establish the relative power difference between the signal of the eNodeB and the attacker. The `ratio` measurement takes the Root Mean Square (RMS) of two signals and displays their ratio in dB. This was done for different output gains of the B210 acting as the attacker, with the RMS calculated over 1 whole frame of 10ms, and each measurement averaged over 100 consecutive frames.

Overshadowing Success Rate

The TX antennas of both the eNodeB and the attacker were placed within 5cm of each other. Then, the antenna of the USRP B210 acting as decoder was placed at exactly 1m distance from each of the antennas. This was verified by a laser distance measuring tool Bosch GLM 50 C. There were no obstructions between the antennas. Then, the attack was started for different gain levels. The decoder then counted how many SIB1s it received from the attacker vs from the real eNodeB until it counted 4’000 SIB1s in total. Thus, it did not count any SIB1s that could not be received due to interferences, which greatly helps to measure the impact of the attack accurately. This experiment was then conducted for different output gain levels of the attacker.

After an output gain level has been established where 100% of the messages arrive at the decoder, this is also verified by sending several IMSI paging message with the same output gain to a UE. It is then verified that the UE can decode the paging messages by listening in with QCSuper [32].

Wideband & Sweeping Jamming

As in the presence of interference, we fear our decoder might not perform as good as state-of-the-art smartphones, which is why we devised the experiment from the outset to include only regular smartphones.

1. Connect the UE to the eNodeB and perform a ping from the UE to an internal ip address in regular intervals
2. Start the attack for a given output gain level
3. **Observe** - do pings still arrive?
4. If pings still arrive, increase the output gain level by 1 dB and go to 3

First, simple blanket jamming is done over the whole bandwidth of the cell (5 MHz). Second, sweeping jamming according to [11] is done, with a sweeping interval of $10\mu s$. For the different UEs, the minimum amount of output gain for which no pings succeed for more than 30s is logged.

7.2.2 Results

Relative Gain

In figure 7.2, the J/S as measured on the oscilloscope is displayed for each attack. The error bars display the standard deviation. The measured J/S increases linearly, as expected. However, the energy on the channel for the noise emitting attacks is much higher for the same output level gain. This may be explained by two reasons. First, the overshadowing attack doesn't need to apply energy in places where there is no data to be overshadowed, gaining a first advantage there. Second, OFDMA has a relatively high PAPR, thus, to avoid clipping, the amplifier must be used with a large remaining power headroom. In contrast, the noise samples sent by GNURadio are using the full range and energy of the amplifier.

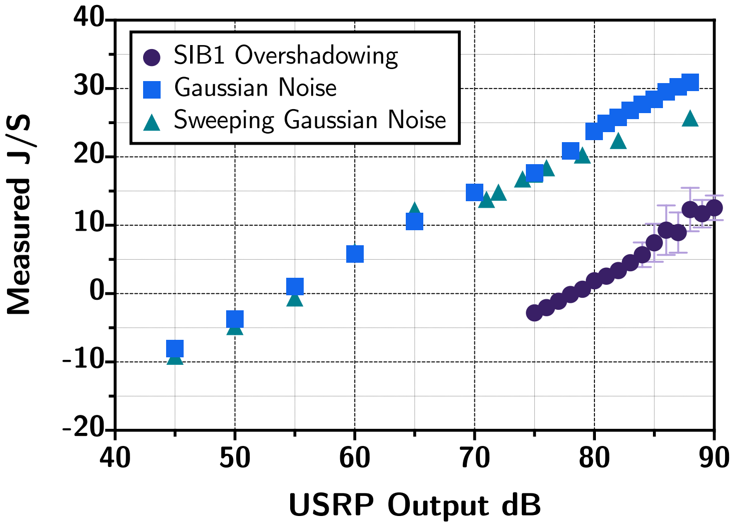


Figure 7.2: Correlation of USRP Output Gain with J/S for different attacks

Overshadowing Success Rate

In figure 7.3, the attack success rate is plotted in relation to the output gain, while the overshadowing attack results are summarized in table 7.1. We achieved a success rate of 100% at an output gain level of 80dB, corresponding to a J/S of 1.870 dB ($\sigma = 0.641$ dB), and 96.8% at a J/S of 0.639 dB ($\sigma = 0.619$ dB). For reaching 100%, Yang et al. in [7] reported a J/S of 5dB. This improvement can be attributed to the continuous overshadowing of all subframes and the inclusion of reference signals in all of them, thereby tuning the downlink channel estimator to our attacker channel.

To verify that this is not depending on our implementation with srsLTE, we sent out a few IMSI paging messages consecutively at the same power level, which was received reliably by the LG Nexus 5X and made visible through QCSuper.

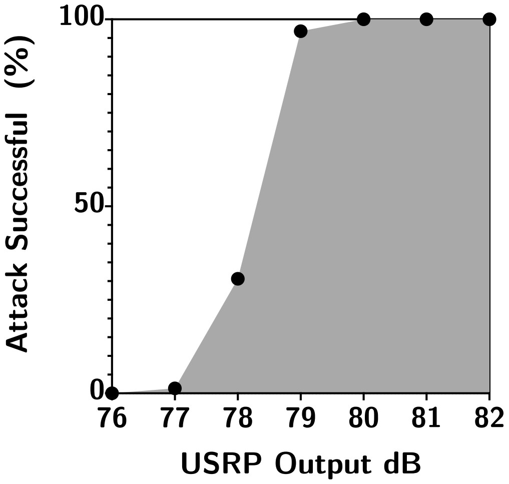


Figure 7.3: Overshadowing Success Rate

$\mu_{J/S}$	$\sigma_{J/S}$	Success Rate
-2.049 dB	0.627	0%
-1.1202 dB	0.675	1.325%
-0.117 dB	0.665	30.625%
0.639 dB	0.619	96.825%
1.870 dB	0.641	100%
2.559 dB	0.733	100%

Table 7.1: Summary of Overshadowing Success Rate and Resulting J/S

Wideband & Sweeping Jamming

In table 7.2, the results have been noted for each UE and the respective output gain for the USRP and the matched J/S value. As is visible from the table, the numbers vary greatly for each individual UE model. However, one smartphone stood out - the Xiaomi Mi Mix 3 5G. The Mi Mix 3 5G has a Qualcomm Snapdragon 855+ with the Snapdragon X24 LTE Modem and its robustness to these attacks is astonishing. For both Blanket Jamming and Sweeping Jamming, the maximum output gain of 88 dB, corresponding to a J/S of 30.3 dB and 20.3 dB, had to be used to block all pings.

UE	**Wideband** $\mu_{J/S}$	**Sweeping** $\mu_{J/S}$
Huawei P20 Lite	20.8 dB	13.8 dB
Huawei P30	17.7 dB	16.8 dB
Samsung Galaxy S10	20.8 dB	17.5 dB
Samsung Galaxy A8	29.5 dB	17.5 dB
iPhone 7	20.8 dB	17.5 dB
iPhone 8	20.8 dB	14.9 dB
iPhone 11 Pro	20.8 dB	20.3 dB
iPhone 11	20.8 dB	18.4 dB
Pixel 2	20.8 dB	22.4 dB
Nokia 1.3	27.7 dB	16.8 dB
Xiaomi Mi Mix 3 5G	30.9 dB	25.7 dB
Nexus 5X	30.9 dB	14.9 dB
Sony Xperia X	30.3 dB	20.3 dB

Table 7.2: Individual Results for Blanket and Sweeping Jamming

7.3 IMSI Paging

7.3.1 Experimental Setup

The goal of the experiment is to see the effect of the attack when the UE is currently in use and receives an IMSI paging message. To this end, the following procedure has been established:

1. Associate the UE with the cell
2. Start the Attack, send the paging message on all subframes for 1s to the UE
3. **Observe** - How long does it take for the UE to re-establish connection?

To observe the time between connection loss and re-establishment, the Android application `Net Monitor Lite` [33] was used to record the connection status of each UE.

The paging message is sent for 1s only, as this allows to correlate the disconnection to the reception of the actual IMSI paging message, rather than triggering a radio link failure by overloading the UE with non-decodable data. If the UE has not lost its connection during that time, it is assumed that the message was sent outside the paging interval, so it is tried again.

7.3.2 Results

The UEs all lost connection within the second in which the paging message was sent. Only very rarely there was a need for sending the message a second time for 1s. The time required to re-establish a connection varies a lot, ranging from a few seconds up to more than 15 minutes. For each of the UEs, the experiment was conducted at least 3 times. The column per UE shows the average over the runs, while its error-bar shows the standard deviation.

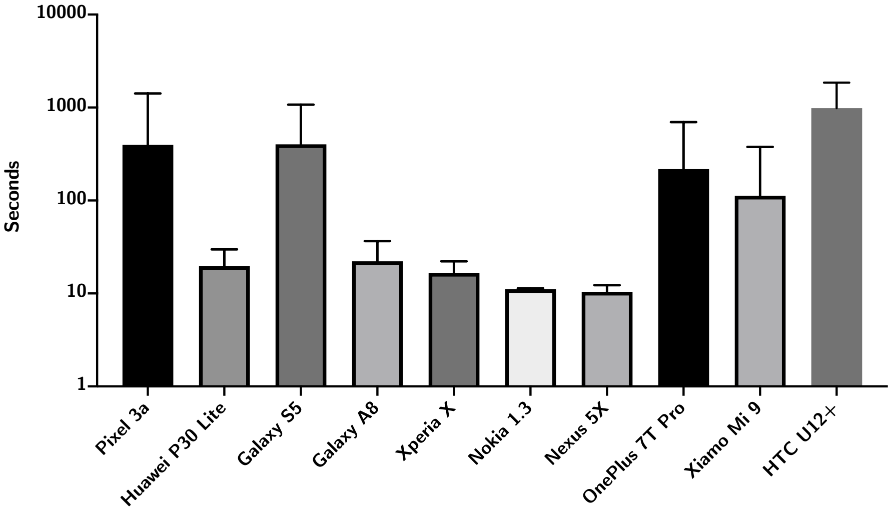


Figure 7.4: IMSI Paging Attack - Time until the UE restored connection with the cell

7.4 SIB Cell Barring

7.4.1 Experimental Setup

Without pre-established connection

This scenario emulates the process where a phone enters the range of a cell which is currently under a SIB Cell Barring attack. To this end, all available phones were put through the following procedure:

1. Forcefully disassociate the UE from the cell (activate flight mode)
2. Start the Attack
3. Try to associate the UE with the cell (de-activate flight mode)
4. **Observe** - can the UE connect?

This was tested using the Amarisoft Callbox Mini as the eNodeB, configured to serve 1 cell.

With pre-established connection

The second scenario concerns an attacker entering a cell and beginning to send spoofed SIB1 Cell Barred messages. As UEs do not constantly refresh the SIB messages, this was helped by sending along a paging message indicating the change of a SIB message. The following procedure was followed to evaluate this attack:

1. Associate the UE with the cell
2. Start the Attack
3. **Observe** - how long does it take for the UE to lose connection?
4. Once the UE has lost connection, Stop the Attack
5. **Observe** - how long does it take for the UE to reconnect after the attack is stopped?

7.4.2 Results

Without pre-established connection

The observation was that none of the UEs tested could successfully associate with the cell as long as the attack was running.

With pre-established connection

The results in figure 7.5 and 7.6 show that whilst the UEs lose the connection with the cell mostly within 10s, the time until they re-connect after the attack has stopped varies greatly between UEs. For each of the UEs, the experiment was conducted 3 times. The column per UE shows the average over the runs, while its error-bar shows the standard deviation.

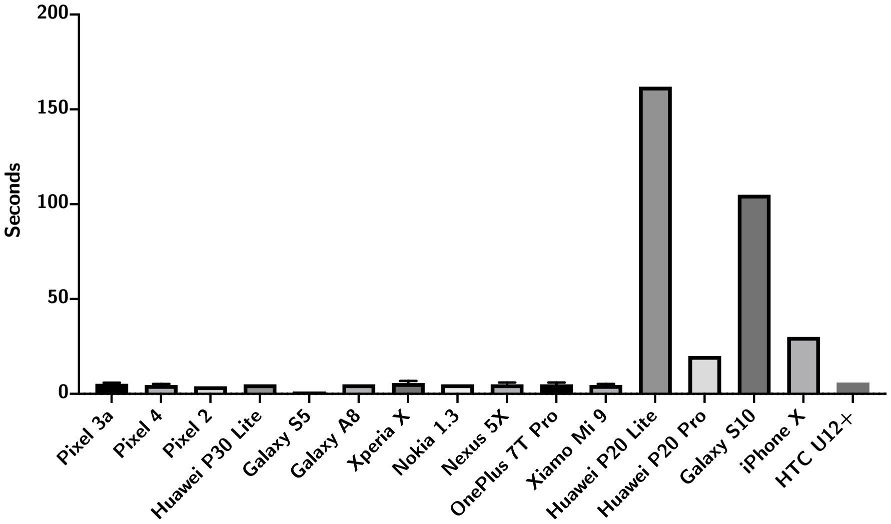


Figure 7.5: SIB Cell Barring Attack - Time until the UE lost connection with the cell

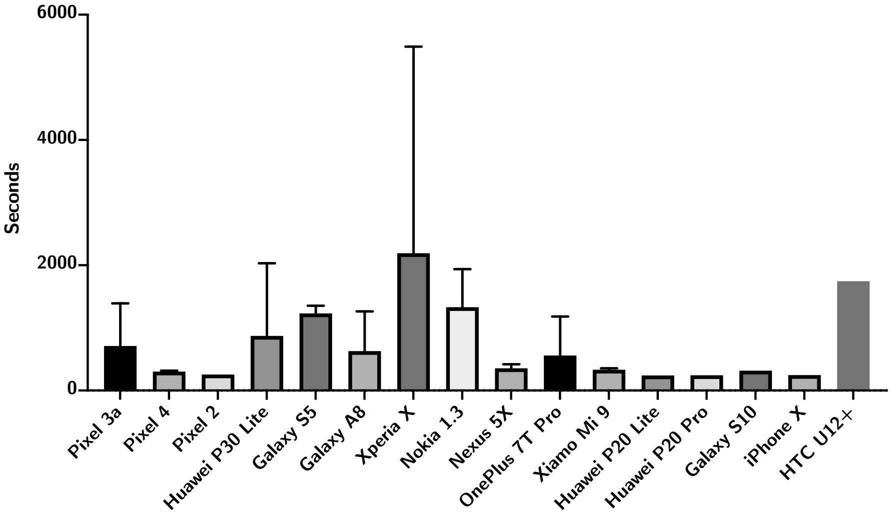


Figure 7.6: SIB Cell Barring Attack - Time until the UE restored connection with the cell

7.5 RRC Connection Reject

7.5.1 Experimental Setup

The benefit of an attack with a `RRC Connection Reject` would be to remain more stealthy than with a `SIB Barring` attack, as the attacker only needs to send reactively after the `PRACH Response` has been received on the downlink for a short time. Still, the effect of the attack remains confined to a single cell only. We devised a simplified experiment, as follows.

1. Put the UE into flightmode
2. Start the attacker
3. Try to attach the UE to the cell
4. **Observe** - Can the UE attach to the cell?

Furthermore, disrupting active connections was evaluated as described below.

1. Attach the UE to the cell, wait until it is in the `RRC Idle` state
2. Start the attacker
3. Interact with the UE
4. **Observe** - Can the UE attach to the cell?

7.5.2 Results

As seen in table 7.3, some of the smartphones tested could attach with the cell during the attack and most of them could re-activate the service if they were attached before. If the attack was stopped, immediate attachment or re-connection was, however, possible, as they tried to re-attach constantly after the specified wait time of 8s (the maximum) was up. This result led us to not investigate the attack in greater detail or increase its reliability. We believe the issue is that the UE will retry to attach constantly, and once the burst of `RRC Connection Reject` messages is over and the UE receives a retransmission from the eNodeB, re-connection is possible.

Phone	**Attachment Possible**	**Service activation possible**
iPhone 7	No	No
iPhone 8	No	Yes
iPhone 11	No	Yes
iPhone 11 Pro	No	No
Pixel 2	No	Yes
Huawei P20 Lite	No	Yes
Huawei P20 Pro	No	Yes
Huawei P30 Lite	No	Yes
Samsung Galaxy S5	No	No
Samsung Galaxy S10	No	No
LG Nexus 5X	Yes	Yes
Xiaomi Mi MIX 3 5G	No	Yes

Table 7.3: Attack Results for RRC Connection Reject

7.6 NAS Attach Reject

7.6.1 Experimental Setup

Baseline

To observe what happens to a UE who receives an `Attach Reject` message, we must be sure that the message is received and processed correctly by the UE. Otherwise, we cannot really interpret the results in any meaningful way. This could be done by attaching a debugger to the baseband chip of each UE, which would print out the messages received. However, this is a rather convoluted process, as it would involve gaining root access on all tested UEs, as well as using different diagnostic tools of questionable reliability for each baseband vendor. Therefore, the base station used from Amarisoft was configured to behave exactly the same as during an attack. Upon receiving an `Attach Request` message, it will reply with a non-integrity protected `Attach Reject` message with EMM cause value 3 and will not proceed further with the attach procedure. Because there is no attacker in this setup and having observed Amarisoft to work reliably, we can safely say that this `Attach Reject` message will be received and processed by the UE. The experiment is conducted as follows:

1. Forcefully disassociate the UE from the cell (activate flight mode)
2. Configure the cell to reject NAS attachments
3. Try to associate the UE with the cell (de-activate flight mode)
4. **Observe** - can the UE establish a connection?
5. **Observe** - are there any GUI indicators, other than *Emergency calls only*?
6. **Observe** - How long does it take for the UE to try and re-establish a connection by itself?
7. **Observe** - What manual interventions are necessary to re-establish connectivity?

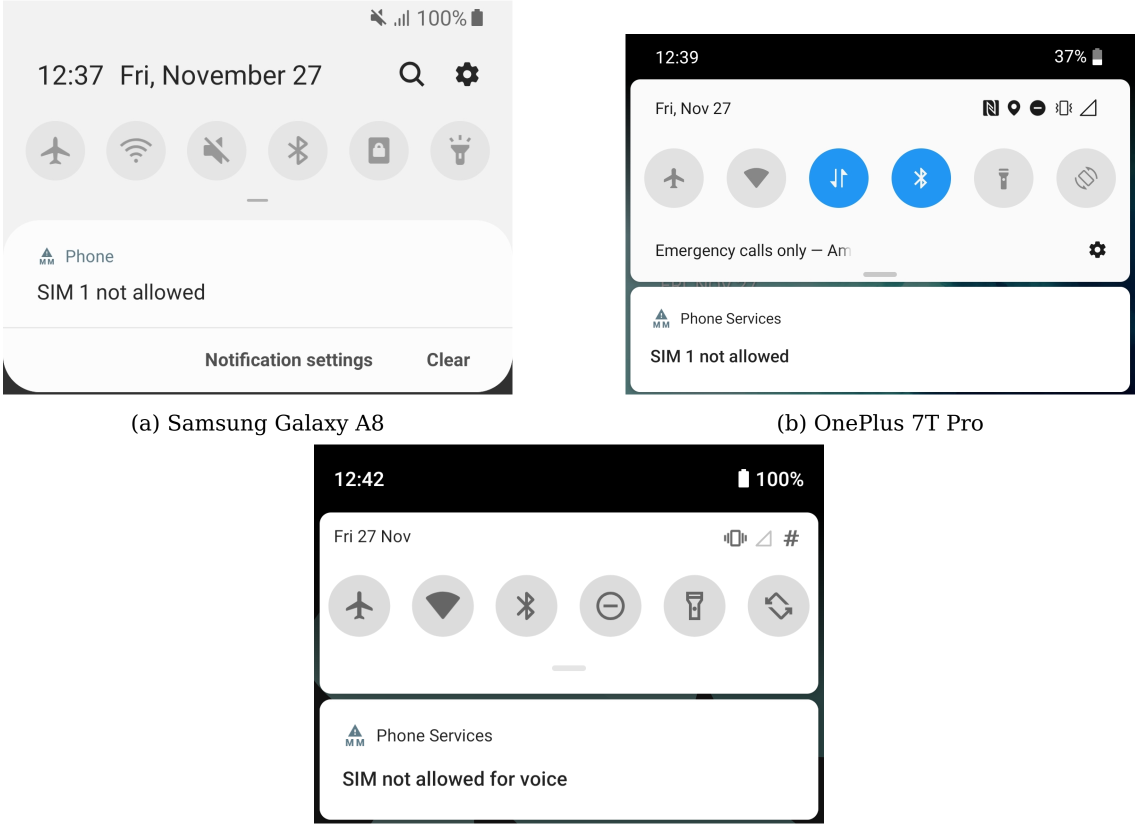
Attack

After the baseline of the UE behaviour has been established, roughly the same procedure is used to observe the effect of the attack. However, instead of re-configuring the cell, in this case, the attacker is turned on and left running for 5 minutes. If then the same behaviour of the UE can be monitored as with the baseline, it may be said that the message was correctly received and the attack conducted successfully.

7.6.2 Results

Baseline

After the `Attach Reject` was sent, the phones were left alone for 12h, while all connection attempts were monitored. As shown in table 7.4, except for the Google Pixel 4, the phones did not even once try to re-establish a connection within those 12 hours. The Google Pixel 4, however, tried 3 times after the first rejection and then stopped trying. Some of the phones indicated the reception of the `Attach Reject` message by a push notification as seen in figure 7.7 with a text *SIM not allowed* and similar variations thereof. This can be used as a reliable indicator that the attack works as intended.



Phone

SIM 1 not allowed

Emergency calls only — Am

Phone Services

SIM 1 not allowed

Phone Services

SIM not allowed for voice

(a) Samsung Galaxy A8

(b) OnePlus 7T Pro

(c) Samsung Galaxy S5

Figure 7.7: Notifications as a result of the Attach Reject attack

Phone	Reconnection Time	Actions Necessary	GUI Indicators
Pixel 2	> 12h	Toggle Flightmode	
Pixel 3a	> 12h	Restart phone	
Pixel 4	30s,60s,15min	Toggle Flightmode	
Huawei P20 Lite	> 12h	Toggle Flightmode	
Huawei P20 Pro	> 12h	Toggle Flightmode	
Huawei P30 Lite	> 12h	Toggle Flightmode	
Samsung Galaxy S5	> 12h	Toggle Flightmode	SIM not allowed
Samsung Galaxy A8	> 12h	Toggle Flightmode	SIM not allowed
Samsung Galaxy S10	> 12h	Toggle Flightmode	
Sony Xperia X	> 12h	Restart phone	
Nokia 1.3	> 12h	Toggle Flightmode	
LG Nexus 5X	> 12h	Restart phone	
iPhone X	> 12h	Toggle Flightmode	
OnePlus 7T Pro	> 12h	Toggle Flightmode	SIM not allowed
Xiaomi Mi 9	> 12h	Toggle Flightmode	

Table 7.4: Baseline Results for Attach Reject

Attack

After the attack was executed, the phones were left alone for more than 12 hours. During this time, the cell was up and running so they could establish a connection again. However, none of the phones chose to do so, as shown in table 7.5. Furthermore, the actions and GUI indicators were largely the same as with the baseline. Differences between the baseline and the attack behaviour have been marked in *italic*.

Phone	**Reconnection Time**	**Actions Necessary**	**GUI Indicators**
Pixel 2	> 12h	*Restart phone*	
Pixel 3a	> 20h	*Toggle Flightmode*	
Pixel 4	> 20h	Toggle Flightmode	
Huawei P20 Lite	> 12h	Toggle Flightmode	
Huawei P20 Pro	> 12h	Toggle Flightmode	
Huawei P30 Lite	> 20h	Toggle Flightmode	
Samsung Galaxy S5	> 20h	*Restart phone*	SIM not allowed
Samsung Galaxy A8	> 20h	Toggle Flightmode	SIM not allowed
Samsung Galaxy S10	> 12h	Toggle Flightmode	SIM not allowed
Sony Xperia X	> 20h	Restart phone	
Nokia 1.3	> 20h	Toggle Flightmode	
LG Nexus 5X	> 20h	Restart phone	
iPhone 7	> 12h	Toggle Flightmode	
iPhone 8	> 12h	Toggle Flightmode	
iPhone 11	> 12h	Toggle Flightmode	
iPhone 11 Pro	> 12h	Toggle Flightmode	
HTC U12+	> 20h	Toggle Flightmode	
OnePlus 7T Pro	> 20h	Toggle Flightmode	SIM not allowed
Xiaomi Mi 9	> 20h	Toggle Flightmode	
Xiaomi Mi Mix 3 5G	> 12h	Toggle Flightmode	

Table 7.5: Attack Results for Attach Reject

7.7 NAS Service Reject

7.7.1 Experimental Setup

This attack may target UEs that are already attached to the MME and are merely re-connecting with a cell, capturing the most common state of any UE in the real world. The following experimental procedure has been devised to evaluate the effectiveness of the attack:

1. Attach the UE with the cell
2. Start the attacker, which reacts on `RRC Connection Setup` messages from the eNodeB
3. Wait a few seconds to let the UE go into `RRC IDLE` state
4. Interact with the UE again to make it re-establish the connection

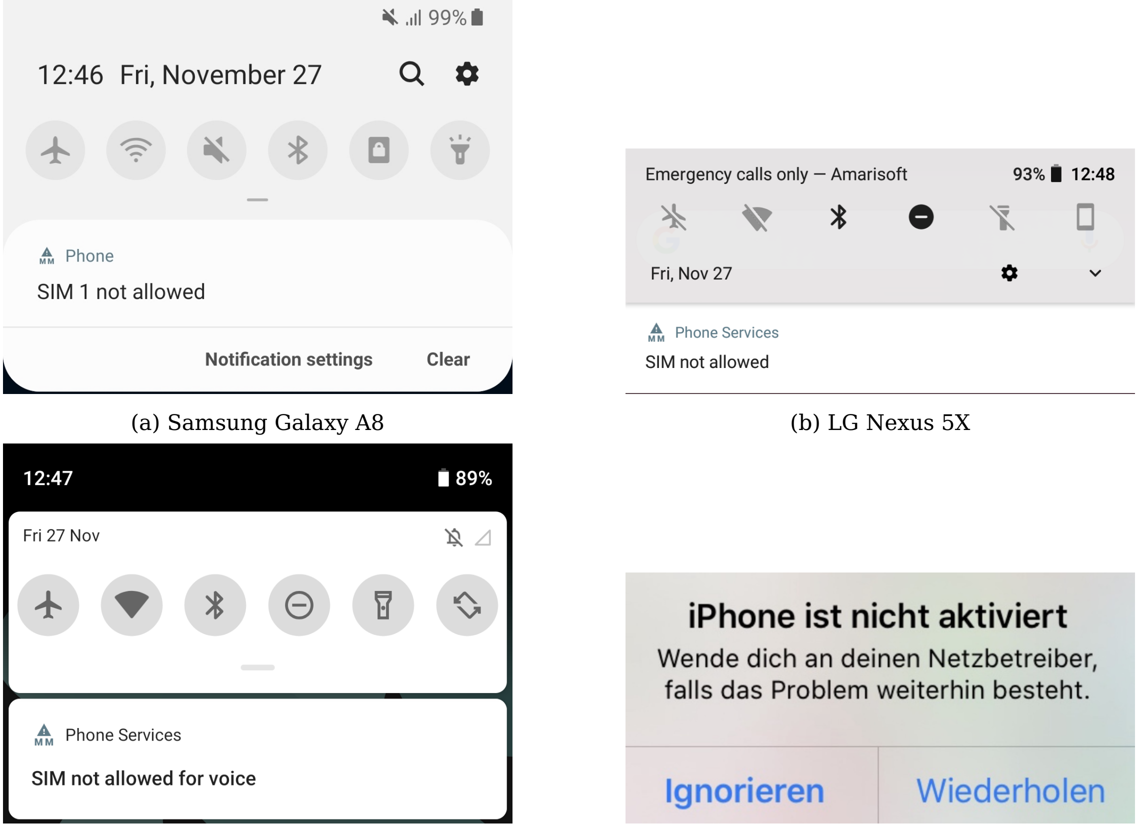
   The UE sends a `NAS Service Request` message and receives a `NAS Service Reject` from the attacker
5. **Observe** - can the UE establish a connection?
6. **Observe** - are there any GUI indicators, other than *Emergency calls only*?
7. **Observe** - How long does it take for the UE to try and re-establish a connection by itself?
8. **Observe** - What manual interventions are necessary to re-establish connectivity?

7.7.2 Results

After the attack was executed, the phones were left alone for more than 12 hours. During this time, the cell was up and running so they could establish a connection again. However, none of the phones chose to do so. Similarily to the attach reject attack, some of the UEs tested showed a notification after being subjected to the attack. These notifications are shown in figure 7.8. Additionally, it became clear that this is the most efficient attack tested thus far. Because the UEs will periodically enter a connected state, presumably to receive push-notifications or perform other background tasks that require internet connectivity, the attack can be left running and will eventually be able to target any UE in its vicinity without any user interaction.

Phone	Reconnection Time	Actions Necessary	GUI Indicators
iPhone 6S	> 12h	Restart phone	iPhone not activated
iPhone 7	> 12h	Toggle Flightmode	
iPhone 8	> 12h	Toggle Flightmode	
iPhone 11	> 12h	Toggle Flightmode	
iPhone 11 Pro	> 12h	Toggle Flightmode	
Pixel 2	> 12h	Restart phone	
Pixel 3a	> 12h	Toggle Flightmode	
Pixel 4	> 12h	Toggle Flightmode	
Huawei P20 Lite	> 12h	Toggle Flightmode	
Huawei P20 Pro	> 12h	Toggle Flightmode	
Huawei P30 Lite	> 12h	Toggle Flightmode	
Samsung Galaxy S5	> 12h	Toggle Flightmode	SIM not allowed
Samsung Galaxy A8	> 12h	Toggle Flightmode	SIM not allowed
Samsung Galaxy S10	> 12h	Toggle Flightmode	SIM not allowed
LG Nexus 5X	> 12h	SIM reinsertion	SIM not allowed
HTC U12+	> 12h	Toggle Flightmode	
OnePlus 7T Pro	> 12h	Toggle Flightmode	
Xiaomi Mi 9	> 12h	Toggle Flightmode	
Xiaomi Mi MIX 3 5G	> 12h	Toggle Flightmode	

Table 7.6: Attack Results for Service Reject



SIM 1 not allowed

SIM not allowed

SIM not allowed for voice

iPhone ist nicht aktiviert

Wende dich an deinen Netzbetreiber, falls das Problem weiterhin besteht.

(a) Samsung Galaxy A8

(b) LG Nexus 5X

(c) Samsung Galaxy S5

(d) iPhone 6S

Figure 7.8: Excerpt of Notifications as a result of the Service Reject attack

7.8 Authentication Reject

A lot of the attacks presented rely on the fact that when the UE receives an `Authentication Reject` message, it will not try to reconnect for a long time, at least as long as if it had received an `Attach Reject` or `Service Reject`. Especially implementing the Uplink version, which relies on overshadowing the `Authentication Response` message, will be complex to implement and debug, so it naturally makes sense to get an idea how the outcome will look like.

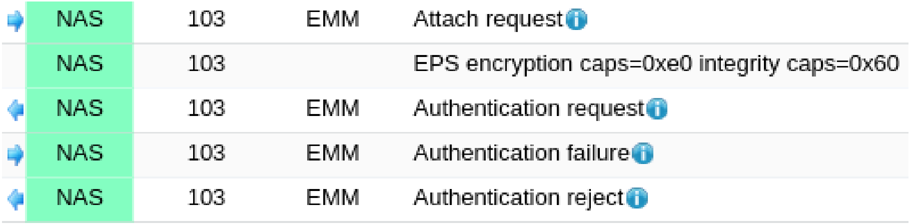
7.8.1 Experimental Setup

For this experiment, the Amarisoft Callbox was used and several UEs.

1. All of the UEs were regularly attached to the network.
2. The MME was reconfigured with wrong keys for all UEs
3. All of the UEs were sent an IMSI paging message containing their IMSI
4. The UEs try to re-attach, receiving a `Authentication Reject` message
5. **Observe** - How long does it take for each of the UEs to retry the attachment?

7.8.2 Results

See figure 7.9 for the resulting interaction between the UE and the eNodeB. The `Authentication Reject` message was not integrity protected. It is clear that this is a powerful and persistent DoS vector as well, as in table 7.7 is described, although 2 phones tried to attach again once within one minute, none of the phones after that tried to re-attach within 12 hours. There were no special GUI indicators other than `Emergency Calls only`.



➡	NAS	103	EMM	Attach request ℹ
	NAS	103		EPS encryption caps=0xe0 integrity caps=0x60
⬅	NAS	103	EMM	Authentication request ℹ
➡	NAS	103	EMM	Authentication failure ℹ
⬅	NAS	103	EMM	Authentication reject ℹ

Figure 7.9: Interaction between the UEa and the eNodeB during the experiment

Phone	# of tries	Reconnection Time	Actions Necessary	GUI Indicators
iPhone 7	1	> 12h	Toggle Flightmode	
iPhone 8	1	> 12h	Toggle Flightmode	
iPhone 11	1	> 12h	Toggle Flightmode	
iPhone 11 Pro	2	> 12h	Toggle Flightmode	
Pixel 2	2	> 12h	Restart phone	
Huawei P20 Lite	1	> 12h	Toggle Flightmode	
Huawei P30 Lite	1	> 12h	Toggle Flightmode	
Samsung Galaxy S10	1	> 12h	Toggle Flightmode	
LG Nexus 5X	1	> 12h	Restart phone	
Xiaomi Mi MIX 3 5G	1	> 12h	Restart phone	

Table 7.7: Results for Authentication Reject

Chapter 8

Comparison

8.1 Comparison Overview

In this thesis, we have identified 5 different classes of attacks on the availability of LTE. Each attack considered is visible in table 8.1 and has been summarized according to the metrics set forth in chapter 3. For persistence and power required, a range from 0-4 dots was established, corresponding roughly to their time to reconnect and J/S power requirements. The Type column refers to whether the issue is a design (D) or implementation (I) issue.

Blanket Jamming First, there is blanket jamming, which just puts noise over the physical channel layout, but it requires a lot of power and suffers from fast recovery times, while it's also an attack which is not selective to any UE. They exploit an inherent flaw of any wireless communication scheme, however, LTE was not particularly designed to be resistant to this type of jamming.

Fake Base Station The are easy to implement and can gain high persistence, but suffer from a high power requirement. Due to this, they are also easy to detect. Some operators already claim the ability to detect those with the measurement reports sent by the UEs, and as the problem has persisted for some years now, it has gotten some attention from researchers.

Rogue UE Next, there are the Rogue UE attacks, which use only as much power as a regular UE and exploit some implementation and design issues, but still suffer from low persistence.

Downlink Overshadowing These attacks have the unique advantage that they only modify real cell behaviour slightly, and can be carried out with a low power requirement. Thus, they can operate much more covertly than fake base station attacks, as all previous detection mechanisms investigated do not work against those attacks. However, until now, they suffered from low persistence and were carried out statically. With our new reactive overshadowing approach, we can implement attacks on various procedures resulting in a high persistence as well, making this a powerful attack class.

Uplink Overshadowing They are suitable for a unique attack scenario with hitherto unprecedented range, combined with the same high persistence as our new downlink overshadowing attacks. However, to target all UEs regardless of their current state, further work must be done, potentially chaining multiple attacks together, making it even more complex. Furthermore, this approach has only been theoretically validated to prove the impact of the attack.

Attack	Persistence	Power	Sync	Type	Selectivity	New
Blanket Jamming						
Wideband	□□□□	■■■■	○	D	○	
Sweeping	□□□□	■■■■	○	D	○	
PCFICH Jamming	□□□□	■■□□	●	D	○	
Fake Base Station						
Attach Reject	■■■■	■■■■	○	D	●	
TAU Reject	■■■■	■■■■	○	D	●	
Authentication Reject	■■■■	■■■■	○	D	●	
Rogue UE						
RRC Resource Exhaustion	?	■□□□	○	I	○	
RRC 2nd Connection	■□□□	■□□□	○	I	●	
NAS 2nd Connection	■□□□	■□□□	○	I	●	
Auth. De-Synchronization	■□□□	■□□□	○	D	●	
DL Overshadowing						
IMSI Paging	■□□□	■■□□	●	D	●	
CS Paging	■□□□	■■□□	●	D	●	
SIB Barring	■■■□	■■□□	●	D	○	
RRC Connection Release	■□□□	■■□□	●	I	○	
RRC Connection Reject	■□□□	■■□□	●	D	○	⋆
Attach Reject	■■■■	■■□□	●	D	○	⋆
Service Reject	■■■■	■■□□	●	D	○	⋆
Authentication Reject	■■■■	■■□□	●	D	○	⋆
Auth. Request Forgery	■■■■	■■□□	●	D	○	⋆
UL Overshadowing						
Forged Attach Request	■■■■	■□□□	●	D	○	⋆

Table 8.1: Comparison of Attacks Presented

Chapter 9

Attack Infrastructure

9.1 Requirements

To facilitate further research on the impact of overshadowing different messages of the LTE protocol, an infrastructure must be built to enable sending messages of different types and content at precisely the right times.

9.2 Baseline

A good starting point for this attack infrastructure would be the downlink, where existing attack code from Yang et al. from their SigOver paper [7] could be obtained, for which we are very thankful as their insight into how to get the attack to work proved of crucial importance time and again.

The code of both our and Yang et al. implementation is based on srsLTE [6], an open-source UE and eNodeB, including some rudimentary EPC. srsLTE is split into roughly 4 parts. There is a common C API, which takes care of most of the low-level pieces of work. Three packages depend on this, the implementations of a UE, the eNodeB and the EPC. These are mostly written in C++ and contain the higher-level logic, manage the state, procedures and interfacing with the IP stack for the user plane data.

The original code from Yang et al. is doing roughly the following tasks in order:

1. Put the data to be sent out on the PDSCH channel and generate the samples for 1 subframe (1ms)
2. Synchronize to the eNodeB
3. Send out the generated samples of the subframe at the right point in time of every frame (10ms) until turned off

To fully appreciate and understand their implementation, we wanted to re-build their approach, relying on their insight whenever possible. Doing so allowed us to extend the approach and use modern C++ features like smart pointers, trying to abstract the memory management of the C API of srsLTE as much as possible, as this is a source of many errors. It was decided early on that the new jamming infrastructure should try to *use* the srsLTE project as much as possible, rather than try to, e.g. modify the srsENB project to fit the new requirements. This way, there is some re-implementation effort, but because of the requirements between a real eNodeB and a reactive jammer are quite different, this was determined as the way-to-go.

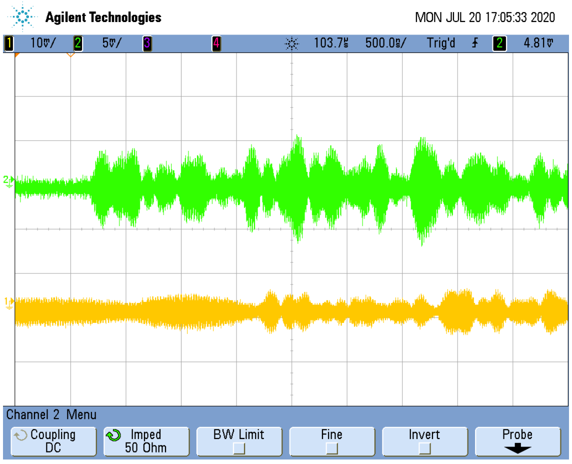


Figure 9.1: Oscilloscope Output, Top: Jamming Signal, Bottom: eNodeB signal, Current Offset: $-1.5\mu s$

We developed and tested our implementation with the aid with 3 Universal Software Radio Peripheral (USRP) B210 SDRs. One of the USRPs was used as the eNodeB running srsENB, one running srsUE to verify the reception of messages and one as the jammer. They were used in a cabled setup to avoid any outside interference.

The objective of the first implementation was to send an ISMI paging message, replicating the attack from the paper by Yang et al. In the course of doing so, naturally many flaws in our implementation arose and had to be tackled. The most time-consuming failures were the ones concerning time-synchronization and frequency offset correction.

The samples had to be sent at a slightly earlier time to the USRP, as was also visible in the code by Yang et al. This time offset arises from the fact that the samples take some time travelling through the DSP of the Field Programmable Gate Array (FPGA) inside the USRP before being released on the air. This offset is easily measurable by any oscilloscope, but during the lockdown, no such tool was readily available. Thus an improvised oscilloscope using Matlab and its Timescope component helped, but no very precise measurement could be taken. When finally the oscilloscope could be used, it was found that the clock of the USRP was drifting, see figure 9.1, yielding a synchronization offset oscillating between $\pm 2\mu s$. To combat this, the sending time was continuously synchronized and updated from the eNodeB with the help of the Primary Synchronization Signal (PSS) and Secondary Synchronization Signal (SSS) sent from the eNodeB.

However, the signal still could not be decoded by srsUE. Debugging with srsUE proved difficult, and a completely new decoder was created using Matlab and its LTE Toolbox. $40ms$ worth of samples were read in Matlab directly from the USRP and then tried to decode. The paging messages from srsENB could be decoded and the paging messages sent by the jammer if run alone. However, the combined signal could not be decoded. It was then further analyzed and established that the frequency of the jammer and the eNodeB must match within a very close range (around $\pm 200Hz$) to decode successfully. This was fixed by measuring the offset from the jammer to the eNodeB, and then applying this offset to the transmit frequency. This way, the jammer and eNodeB sent on precisely the same frequency ($\pm 20Hz$).

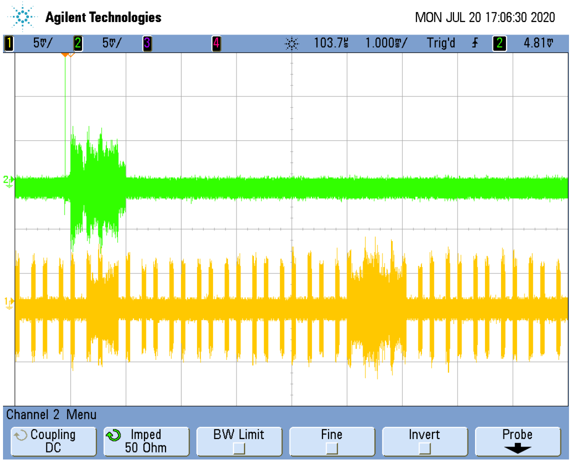


Figure 9.2: Oscilloscope Output, Top: Jamming Signal, Bottom: eNodeB signal

9.3 Extensions

Once these problems were resolved, paging messages could be overshadowed and were decoded correctly in the UE. However, there remained one problem as depicted in figure 9.2. Namely, there is a sharp peak in transmit power right at the beginning of the transmission, preventing from anything being decoded sent right at this moment. This was solved by Yang et al. by inserting zeroes at the beginning of the transmission and sending the intended data afterwards, compensating the delay by starting to send slightly earlier.

9.3.1 Overshadowing all Subframes

We quickly determined that we would like to be able to overshadow all subframes to successfully overshadow all data from the eNodeB, as would be necessary for the higher-layer attacks. To that end, it was first necessary to eliminate this power peak. It was discovered that the parameter to the sending function `rf_send_timed` always included flags for indicating if this is the first and/or last samples packet. Whenever the `first` flag is set to `true`, such an unwanted peak arises. However, it is necessary to set it `true` at least once, for only then the necessary timing information is transmitted to the USRP along with the samples. To eliminate this, we used an approach where we just didn't stop sending data at all. During the time where no data was to be sent, simply all zeroes were sent out, which perfectly eliminated the peak.

The data for the subframes was kept in a ring buffer, such that the generation of messages didn't interfere with loading them onto the USRP. Moreover, as there now was a way to control the data sent out at all subframes, this proved easy to extend to acquire the ability to overshadow all subframes with data.

9.3.2 Downlink Decoder

To enable further attacks on higher layers that can react to the activity of a UE, it was deemed essential to be able to not only synchronize to the base station for acquiring antenna and timing information, but also to develop the ability to decode downlink traffic, especially of UEs that are beginning to establish a connection with the eNodeB. This traffic is sent on the PDSCH channel, which required the decoding of the PDCCH channel containing DCIs first. But to decode the PDCCH channel, the C-RNTI of the UE must be known in advance. This is because the DCI information is put randomly at a few locations specific to a set of UEs.

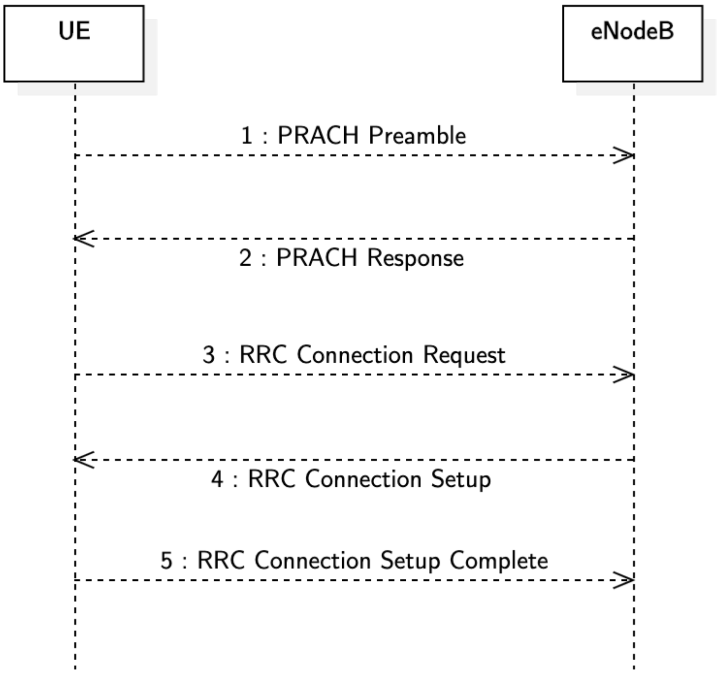


Figure 9.3: Contention based Random Access Procedure

Additionally, the UE may determine if the DCI belongs to it by calculating the CRC of the DCI message, XOR-ing this with its RNTI and checking if the result is zero.

Getting RNTIs

The eNodeB allocates the RNTIs during the course of the Random Access Procedure. Figure 9.3 shows the random access procedure based on section 10.1.5 of the 3GPP standard [34]. There are 2 different types of procedures, the contention-based and the non-contention based one. The contention-based is used in most situations when contention, or a clash of 2 or more UEs connecting at the same time, may occur. The procedure itself works as follows:

1. The UE sends a PRACH Preamble, at a time the UE chooses. The preamble is chosen from a set of preambles specified in the SIB2.

2. After decoding the preamble, the eNodeB will send the PRACH Response on the PDSCH back to the UE. As the UE does not know yet what RNTI to use, this response is sent on one of the specified RAR-RNTIs, which are 10 special RNTIs where only these PRACH responses are sent. The UE knows based on the system frame number and some formula, on which of those RNTIs to listen for the response. In the PRACH Response, as well as as the T-CRNTI, also an uplink grant is included for the UE to send to the next message on.

3. The UE sends the RRC Connection Request, along with its last allocated TMSI. If it doesn't have a TMSI allocated, it will use a random number.

4. As up until this point, multiple UEs could have done the very same process, the next message resolves this contention. The RRC Connection Setup message includes the identifier sent in the RRC Connection Request - if this is not the one the UE sent, it must

assume there was a collision in the random access procedure, and it will try again. The RRC Connection Setup message is critical, as it contains also dedicated configuration information for the UE, ranging from the physical layer to the setup of bearers for the setup of the higher-level connection with the MME.

5. If the contention resolution is successful, it will confirm this to the eNodeB by sending the RRC Connection Setup Complete message.

The decoder was implemented to listen on the downlink for messages 2 and 4 - the PRACH Response and the RRC Connection Setup message. Message 2 is necessary to get the RNTI and message 4 for the physical layer configuration, as without this information, decoding later messages will fail with CRC errors. Furthermore, as revealed later, it also is important to encode the messages from the higher layers correctly.

Multi Threading

The amount of time required to decode the PDCCH is linear with respect to the amount of RNTIs that are expected. Searching for a DCI of one RNTI takes around $50\mu s$, which is not much by itself, but given the fact that we need to decode every single subframe, we only have $1ms$, or $1000\mu s$, leaving space for around 20 RNTIs. Because we need to decode the SI-RNTI and the 10 RA-RNTIs, already 11 RNTIs need to be decoded. This led to constant buffer overflows. Therefore, we allocated a constant set of threads, one for each core available, and distributed the decoding task among them. To avoid copying samples around in memory, the following scheme was implemented. Each thread entered a queue when idle and was selected by a separate receiving thread, whose only purpose was to read samples from the USRP and perform some time alignment. The samples were put into a buffer belonging to the decoding thread, and when a full subframe was successfully received, the thread woke up and began decoding. This solved prevented sudden latency spikes from causing receive buffer overflows, leading to dropped samples.

9.3.3 Isolating RX and TX

When sending and receiving on the same frequency at the same time, some isolation between TX and RX ports are necessary, as otherwise no data can be decoded while the jammer is sending. This isolation was achieved by using directional antennas that point away from the jammers unidirectional antennas - preferably in the direction of the eNodeB. Another option would be to use a device called a *circulator*; however, this proved difficult to find in a ready-to-use fashion and not only as a small part designed to be mounted on a PCB. Moreover, the design with the directional antenna is usable with any frequency band and thus any cell. The *circulator* would be tuned to a specific frequency band only, limiting its use to a specific cell.

9.3.4 Multiple Antennas

To achieve the high throughput of modern versions of LTE, efficient usage of the available RF spectrum is impertinent. To this end, eNodeBs are equipped with a large number of antennas, sending in MIMO or Transmit Diversity fashion, depending on what is sent. To decode the data, at least 2 RX antennas are needed, oriented orthogonally. The decoding code has been updated to reflect this, as well as the sending part, which may also imitate sending in a configuration called *Transmit Diversity*, which will not use the multiple antennas to set up multiple transmit channels and increase throughput, but to achieve diversity gain. This is most often used in challenging RF situations or when the feedback channel with the UE hasn't been established yet. In our case, we do not have this closed-loop required with the UE, so we will always use transmit diversity. The type of pre-coding used is encoded in the DCI at the start of every subframe, which is also sent out by our jammer.

9.3.5 Reactive Jammer

Combining the receive and sending part together, we now have a reactive jammer that may send based on what is received. However, the generation of the data and subframes was mostly static, at least in the original architecture. It was then optimized such that the samples of each subframe may be generated *online*, that is, right before its sent out, the scheduled data for that frame/subframe is gathered and encoded to the PDSCH channel. Then its respective DCI is attached, reference signals and broadcast information (MIB) is included, and finally, samples are generated and sent out to the SDR. This has the huge benefit that the reaction time between receiving information on the downlink and sending the appropriate attack message is very low. It showed that this reaction time, from the end of the decoded downlink subframe and the beginning of the subframe of the response message, is around 6ms.

9.3.6 PDSCH Channel Parameters

As has been hinted to in section 9.3.2, the physical channel parameters of the PDSCH channel p_a and p_b are very important. They are sent on the `RRC Connection Setup` message on the downlink by the eNodeB. Any subsequent messages use this channel configuration, and the UE expects as much. This is why, when overshadowing messages that come later than the `RRC Connection Setup`, these parameters must be respected.

9.4 Higher Layer Messages

In section 5.2, the Attach Reject attack is described. In this section, we will use this attack as an example to describe what steps we had to undertake to successfully send messages from the NAS layer to the UE.

9.4.1 From NAS to PDSCH

In figure 2.3 from chapter 2, the whole stack from the NAS messages to the PDSCH channel is presented. First, after packing the `Attach Request` message with its header into an RRC message, this is ASN.1 encoded and sent to the PDCP layer, which will add another header with a sequence number to it. Optionally, another layer of encryption and/or integrity protection would be applied here, but as this is not applicable to messages this early in the attachment procedure, the messages leave the PDCP layer in plain-text. Finally, it passes through the Radio Link Control (RLC) layer, which will add yet another header to it and might segment it, according to the needs of the MAC layer. Because we are controlling the resource scheduling, there is no need for segmentation and the message leaves the RLC as one message and is put to the MAC layer. Here, it may be joined with other messages from other bearers, which may be user data, which is why it's identified by another header containing the LCID 1. Now the message is ready to be put on the PDSCH channel, together with an appropriate DCI.

During the implementation of this part, the availability of the srsLTE greatly helped. We developed a few integration tests using the Googletest framework [35], that verified this encoding is working as intended, before moving on to tackling the next problems.

9.4.2 HARQ & RLC ACK Process Emulation

In the attack, the UE is sending the `Attach Request` message to the MME, while the subsequent `Attach Accept` from the MME is overshadowed with our bogus `Attach Reject` reply. The issue is that we dont' know when the `Attach Request` message will be sent by the UE in the first place, as the eNodeB may wait for some time before allocating another DCI0 for the UE to send data.

First, one might be inclined to send the `Attach Reject` much earlier than anticipated. While this worked for some UEs, because they dropped their connection immediately after this. However, other UEs interpreted the later arriving messages from the MME and could connect successfully nonetheless.

The solution to this problem would be to send the `Attach Reject` on every subframe for a few seconds, overshadowing all the data possibly sent by the MME during this time, including all retries. But as we're sending this potentially even earlier than even the request is made by the UE, this means we also overshadow all the DCI0s, meaning that the UE will never even send the `Attach Request` in the first place.

Therefore, we opted to implement sending DCI0s during the time we sent the `Attach Reject` message as well. We sent one DCI0 during each frame for each UE the attack is currently running for. However, simply sending these is not enough, as the UE expects a confirmation of reception of the message on the Physical HARQ Indicator Channel (PHICH) channel 4ms after it has sent the data. Thus, we implemented sending these as well. This involved storing the sent DCI0s and converting their parameters to a proper acknowledgement. Luckily, they don't depend on having received any data from the UE at all, otherwise, this would have meant that an uplink decoder or guessing this parameter would've been necessary.

Now that we have opened up the uplink channel for the UE to send data on, we have inadvertently solved another problem by doing so. As the DCI0s we are sending do not correspond with anything the eNodeB might expect the UE to send data on, the eNodeB, and thus, the MME, will not receive any data from the UE and will thus also not try to retry any correct message they haven't received an acknowledgement for.

However, as visible in figure 2.3 from chapter 2, the HARQ is not the only acknowledgement process running in LTE. The RLC layer will also issue sequence numbers and expect acknowledgements for sent data. Without these, the UE will think that its `Attach Request` has not yet arrived and will ignore any `Attach Reject` it may have received. Upon inspection with several UEs from different manufacturers, it was revealed that the UEs also split the `Attach Request` message over different RLC messages - how many, depended entirely on the UE model. All of them used between 1 and 4 segments, however. While the idea arose that sending an acknowledgement of a large enough sequence number might suffice, the UEs simply dropped those.

Therefore, we had to send RLC acknowledgement messages for all sequence numbers in the range from 1 to 4. We sent them increasingly, so for the first 350ms, sequence number 1 is acknowledged, then sequence number 2 and so on. We combined them together with the `Attach Reject` messages in the MAC layer with a separate header on the logical channel 1. With those modifications, the messages were finally received reliably by all UEs tested.

9.4.3 Terminal Interface

To simplify working with the program during experimentation, live interaction with the system is necessary. Therefore, the program allows the input of commands, just like in a terminal. The following is an excerpt of the set of commands built in:

set <PLMN> <PCI>	Choose the cell with the given PLMN and PCI to decode
scan	Scan for nearby cells, printing out a list
g+ / g-	Increase or decrease the TX gain
rxg+ / rxg-	Increase or decrease the RX gain
s	Start/Stop all TX output
sib	Start/Stop the SIB Cell Barring attack
imsi <IMSI>	Send an IMSI paging message to the IMSI specified for 10s
rrc	Start/Stop the RRC Connection Reject attack
attach	Start/Stop the NAS Attach Reject attack
service	Start/Stop the NAS Service Reject attack
<empty>	Repeat the last entered command
q	Stop everything and exit

Chapter 10

Conclusion

We have seen that the signal overshadowing attack presented by Yang et al. is a new and extremely capable approach on its own. Combined with our near-realtime downlink decoder and signal generation, a reactive attacker was built and shown to be capable of highly power-efficient attacks combined with high persistence and low detectability. Of those attacks presented, the attach- and service reject attacks have proven to be the most impactful. They would be devastating if used in a real network because all of the UEs tested exhibited total loss of LTE communication for more than 12 hours after being subjected to the attack.

While a countermeasure is easy to implement, it all depends on device and baseband vendors to implement those changes. It is unclear if older models would ever be updated. Finally, as is seen in the future work section, it’s a building block for even more efficient and covert attacks also on other security goals, such as privacy.

10.1 Countermeasures

For the new attacks presented, the UE manufacturers and their baseband vendors can implement relatively straightforward countermeasures. Upon reception of any reject message without integrity protection, they must continue to try and attach to the network, without waiting excessively between re-tries. MNOs may evaluate, based on their current traffic patterns, just how much this will increase traffic that is rejected and may scale their core network accordingly, if at all necessary.

10.2 Future Work

10.2.1 Uplink Overshadowing Implementation

Because the impact of the Attach Request overshadowing attack has been explored and validated, its implementation is a logical next step to solidify the concept further. Further research must be done to find a way to target UEs that are issuing a mere `Service Request` as well, which would be facilitated by the availability of a working uplink overshadowing implementation. Every single message of the attachment procedure would then be able to be exchanged and faked, which will, with a high probability, uncover more weaknesses.

Cooperation with MNOs may, however, be necessary, to respect ethical and legal concerns when evaluating such attacks in the real world.

10.2.2 Combination with Uplink Sniffer

IMSI Catcher

The infrastructure presented to overshadow the Downlink can be easily extended to send any message on the Downlink, including imitating the MME messages, as explored in various attacks in this thesis. During attachment, the `Identity Request` message is used by operators to establish the identity of the connecting UE. Upon reception of this message, the UE sends back their IMSI in plaintext. An attacker can exploit this by sending the above-mentioned message in combination with a corresponding uplink sniffer designed to read the answer from the UE. By doing so, a highly covert IMSI Catcher is constructed. The resulting IMSI Catcher is capable of defeating all of the countermeasures in current literature, which were explored in section 3.3.3.

Selectivity

Generally speaking, combining the works of this thesis with an uplink sniffer makes sense, as it allows for the presented attacks to be targeted at specific UEs. This would work by reading out the first `Attach Request` message sent by the UE and deciding to overshadow the response from the MME, based on the identifiers submitted.

10.2.3 Synchronized Fake Base Station Attacks

During experimentation, we observed that the Amarisoft base station has a configuration option for aligning the beginning of frame #0 with a customizable GPS time. Thus, it would be possible to shift it to align exactly with a commercial eNodeB. Once the cell configuration parameters of the real cell are duplicated, and the gain is slightly higher as well, it is possible to have a fake base station that is continuously overshadowing the real cell with a very low detectability and an integrated uplink sniffer by design. Although the flexibility of the messages that can be sent is limited, the service, attach and authentication reject attacks are all possible, as well as a ready-made IMSI catcher, simply by configuring this commercially available product correctly.

Acronyms

ASN.1 Abstract Syntax Notation One

CRS Cell-Specific Reference Signals

CS Circuit Switched

DCI Downlink Control Information

DoS Denial of Service

DRX Discontinuous Reception

EMM EPS Mobility Management

eNodeB E-UTRAN Node B

EPC Evolved Packet Core

EPS Evolved Packet System

EVM Error Vector Magnitude

FBS Fake Base Station

FPGA Field Programmable Gate Array

GPS Global Positioning System

GPSDO GPS Disciplined Oscillator

IMSI International Mobile Subscriber Identity

LTE Long Term Evolution

MIB Master Information Block

MME Mobility Management Entity

MNC Mobile Network Code

NAS Non-Access Stratum

OFDM Orthogonal Frequency Division Multiplexing

PCFICH Physical Control Format Indicator Channel

PDSCH Physical Downlink Shared Channel

PLMN Public Land Mobile Network

PRACH Physical Random Access Channel

PS Packet Switched

PSS Primary Synchronization Signal

RF Radio Frequency

RMS Root Mean Square

RNTI Radio Network Temporary Identifier

RRC Radio Resource Control

S-TMSI Serving Temporary Mobile Subscriber Identity

SDR Software Defined Radio

SIB System Information Block

SSS Secondary Synchronization Signal

TLS Transport Layer Security

UE User Equipment

USRP Universal Software Radio Peripheral

Bibliography

[1] P. Boyland, "The State of Mobile Network Experience," May 2019, last accessed: 17.09.2020. [Online]. Available: https://www.opensignal.com/sites/opensignal-com/files/data/reports/global/data-2019-05/the_state_of_mobile_experience_may_2019_0.pdf

[2] S. Brem, M. Hohl, W. Möller, L. Blaser, and T. Schulze, "Bericht zur nationalen Risikoanalyse," Nov. 2020, last accessed: 07.12.2020. [Online]. Available: https://www.babs.admin.ch/content/babs-internet/de/aufgabenbabs/gefaehrdrisiken/natgefaehrdanalyse/_jcr_content/contentPar/tabs/items/fachunterlagen/tabPar/downloadlist/downloadItems/109_1604480153059.download/KNSRisikobericht2020-de.pdf

[3] Ettus Research, a National Instruments Brand, "USRP B210 USB Software Defined Radio (SDR)," last accessed: 07.12.2020. [Online]. Available: https://www.ettus.com/all-products/ub210-kit/

[4] Lime Microsystems, "LimeSDR," last accessed: 07.12.2020. [Online]. Available: https://limemicro.com/products/boards/limesdr/

[5] Nuand, "bladeRF 2.0 micro xA4," last accessed: 07.12.2020. [Online]. Available: https://www.nuand.com/product/bladerf-xa4/

[6] A. Puschmann, I. Gomez, P. Alvarez, X. Arteaga, F. Paisana, P. Sutton, and J. Tallon, "srsLTE/srsLTE," Apr. 2020, last accessed: 28.04.2020. [Online]. Available: https://github.com/srsLTE/srsLTE

[7] H. Yang, S. Bae, M. Son, H. Kim, S. M. Kim, and Y. Kim, "Hiding in plain signal: physical signal overshadowing attack on LTE," in *Proceedings of the 28th USENIX Conference on Security Symposium*, ser. SEC'19. USA: USENIX Association, Aug. 2019, pp. 55–72, last accessed: 16.10.2020.

[8] K. Whitehouse, A. Woo, F. Jiang, J. Polastre, and D. Culler, "Exploiting the capture effect for collision detection and recovery," in *The Second IEEE Workshop on Embedded Networked Sensors, 2005. EmNetS-II.*, May 2005, pp. 45–52.

[9] J. Lee, W. Kim, S.-J. Lee, D. Jo, J. Ryu, T. Kwon, and Y. Choi, "An experimental study on the capture effect in 802.11a networks," in *Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*, ser. WinTECH '07. New York, NY, USA: Association for Computing Machinery, Sep. 2007, pp. 19–26. [Online]. Available: https://doi.org/10.1145/1287767.1287772

[10] R. Krenz and S. Brahma, “Jamming LTE signals,” in *2015 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*. Constanta, Romania: IEEE, May 2015, pp. 72–76, last accessed: 27.07.2020. [Online]. Available: http://ieeexplore.ieee.org/document/7185089/

[11] G. Romero, V. Deniau, and O. Stienne, “LTE Physical Layer Vulnerability Test to Different Types of Jamming Signals,” in *2019 International Symposium on Electromagnetic Compatibility - EMC EUROPE*, Sep. 2019, pp. 1138–1143, iSSN: 2325-0364.

[12] J. Kakar, K. McDermott, V. Garg, M. Lichtman, V. Marojevic, and J. H. Reed, “Analysis and Mitigation of Interference to the LTE Physical Control Format Indicator Channel,” in *2014 IEEE Military Communications Conference*. Baltimore, MD, USA: IEEE, Oct. 2014, pp. 228–234, last accessed: 12.08.2020. [Online]. Available: http://ieeexplore.ieee.org/document/6956764/

[13] Y. Li, C. Peng, Z. Yuan, J. Li, H. Deng, and T. Wang, “Mobileinsight: extracting and analyzing cellular network information on smartphones,” in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, ser. MobiCom ’16. New York City, New York: Association for Computing Machinery, Oct. 2016, pp. 202–215, last accessed: 28.04.2020. [Online]. Available: https://doi.org/10.1145/2973750.2973751

[14] R. M. Rao, S. Ha, V. Marojevic, and J. H. Reed, “LTE PHY Layer Vulnerability Analysis and Testing Using Open-Source SDR Tools,” *arXiv:1708.05887 [cs]*, Sep. 2017, arXiv: 1708.05887, Last accessed: 01.10.2020. [Online]. Available: http://arxiv.org/abs/1708.05887

[15] S. F. Mjølsnes and R. F. Olimid, “Easy 4G/LTE IMSI Catchers for Non-Programmers,” in *Computer Network Security*, ser. Lecture Notes in Computer Science, J. Rak, J. Bay, I. Kotenko, L. Popyack, V. Skormin, and K. Szczypiorski, Eds. Cham: Springer International Publishing, 2017, pp. 235–246.

[16] T. Fei and W. Wang, “LTE Is Vulnerable: Implementing Identity Spoofing and Denial-of-Service Attacks in LTE Networks,” in *2019 IEEE Global Communications Conference (GLOBECOM)*. Waikoloa, HI, USA: IEEE, Dec. 2019, pp. 1–6, last accessed: 30.04.2020. [Online]. Available: https://ieeexplore.ieee.org/document/9013397/

[17] 3GPP, “3GPP TS 24.301,” Jul. 2020, last accessed: 21.09.2020. [Online]. Available: https://www.3gpp.org/ftp//Specs/archive/24_series/24.301/24301-g51.zip

[18] R. P. Jover, “LTE security, protocol exploits and location tracking experimentation with low-cost software radio,” *arXiv:1607.05171 [cs]*, Jul. 2016, arXiv: 1607.05171, Last accessed: 30.04.2020. [Online]. Available: http://arxiv.org/abs/1607.05171

[19] A. Shaik, R. Borgaonkar, N. Asokan, V. Niemi, and J.-P. Seifert, “Practical Attacks Against Privacy and Availability in 4G/LTE Mobile Communication Systems,” *arXiv:1510.07563 [cs]*, Aug. 2017, arXiv: 1510.07563, Last accessed: 30.04.2020. [Online]. Available: http://arxiv.org/abs/1510.07563

[20] S. R. Hussain, O. Chowdhury, S. Mehnaz, and E. Bertino, “LTEInspector: A Systematic Approach for Adversarial Testing of 4G LTE,” in *Proceedings 2018 Network and Distributed System Security Symposium*. San Diego, CA: Internet Society, 2018, last accessed: 30.04.2020. [Online]. Available: https://www.ndss-symposium.org/wp-content/uploads/2018/02/ndss2018_02A-3_Hussain_paper.pdf

[21] CellularPrivacy, “CellularPrivacy/Android-IMSI-Catcher-Detector,” Dec. 2020, last accessed: 07.12.2020. [Online]. Available: https://github.com/CellularPrivacy/Android-IMSI-Catcher-Detector

[22] C. Quintin, *Detecting Fake 4G Base Stations in Real Time*, 2020, published: DEF CON, Last accessed: 07.12.2020. [Online]. Available: https://doi.org/10.5446/49771

[23] A. Dabrowski, N. Pianta, T. Klepp, M. Mulazzani, and E. Weippl, “IMSI-catch me if you can: IMSI-catcher-catchers,” in *Proceedings of the 30th Annual Computer Security Applications Conference on - ACSAC ’14*. New Orleans, Louisiana: ACM Press, 2014, pp. 246–255. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2664243.2664272

[24] Z. Li, W. Wang, C. Wilson, J. Chen, C. Qian, T. Jung, L. Zhang, K. Liu, X. Li, and Y. Liu, “FBS-Radar: Uncovering Fake Base Stations at Scale in the Wild,” Jan. 2017.

[25] T. van Do, H. T. Nguyen, N. Momchil, and V. T. Do, “Detecting IMSI-Catcher Using Soft Computing,” in *Soft Computing in Data Science*, M. W. Berry, A. Mohamed, and B. W. Yap, Eds. Singapore: Springer Singapore, 2015, vol. 545, pp. 129–140, series Title: Communications in Computer and Information Science. [Online]. Available: http://link.springer.com/10.1007/978-981-287-936-3_13

[26] P. K. Nakarmi and K. Norrman, “Detecting false base stations in mobile networks,” Jun. 2018, last accessed: 07.12.2020. [Online]. Available: https://www.ericsson.com/en/blog/2018/6/detecting-false-base-stations-in-mobile-networks

[27] S. R. Hussain, M. Echeverria, A. Singla, O. Chowdhury, and E. Bertino, “Insecure connection bootstrapping in cellular networks: the root of all evil,” in *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, ser. WiSec ’19. Miami, Florida: Association for Computing Machinery, May 2019, pp. 1–11, last accessed: 30.04.2020. [Online]. Available: https://doi.org/10.1145/3317549.3323402

[28] C. Park and M. Son, “SigOver + alpha,” Dec. 2019, last accessed: 23.09.2020. [Online]. Available: https://media.ccc.de/v/36c3-10801-sigover_alpha

[29] H. Kim, J. Lee, E. Lee, and Y. Kim, “Touching the Untouchables: Dynamic Security Analysis of the LTE Control Plane,” in *2019 IEEE Symposium on Security and Privacy (SP)*, May 2019, pp. 1153–1168.

[30] M. T. Raza, F. M. Anwar, and S. Lu, “Exposing LTE Security Weaknesses at Protocol Inter-layer, and Inter-radio Interactions,” in *Security and Privacy in Communication Networks*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, X. Lin, A. Ghorbani, K. Ren, S. Zhu, and A. Zhang, Eds. Cham: Springer International Publishing, 2018, pp. 312–338.

[31] GNU Radio, “gnuradio/gnuradio,” Dec. 2020, last accessed: 06.12.2020. [Online]. Available: https://github.com/gnuradio/gnuradio

[32] M. Moulinier and B. Michau, “P1sec/QCSuper,” Nov. 2020, last accessed: 30.11.2020. [Online]. Available: https://github.com/P1sec/QCSuper

[33] V. V, “NetMonitor Cell Signal Logging,” Nov. 2020, last accessed: 30.11.2020. [Online]. Available: https://play.google.com/store/apps/details?id=ru.v_a_v.netmonitor&hl=de&gl=US

[34] 3GPP, “36.300 V16.3.0 Technical Specification 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2,” Feb. 2020, last accessed: 05.10.2020. [Online]. Available: https://www.3gpp.org/ftp//Specs/archive/36_series/36.300/36300-g30.zip

[35] Google, “google/googletest,” Dec. 2020, last accessed: 08.12.2020. [Online]. Available: https://github.com/google/googletest