LTrack: Stealthy Tracking of Mobile Phones in LTE

Martin Kotuliak, Simon Erni, Patrick Leu, Marc Röschlin,
and Srdjan Čapkun, *ETH Zurich*

https://www.usenix.org/conference/usenixsecurity22/presentation/kotuliak

This paper is included in the Proceedings of the
31st USENIX Security Symposium.

August 10–12, 2022 • Boston, MA, USA

978-1-939133-31-1

LTrack: Stealthy Tracking of Mobile Phones in LTE

Martin Kotuliak, Simon Erni, Patrick Leu, Marc Röschlin, and Srdjan Čapkun
ETH Zurich

Abstract

We introduce LTrack, a new tracking attack on LTE that allows an attacker to stealthily extract user devices' locations and permanent identifiers (IMSI). To remain stealthy, the localization of devices in LTrack is fully passive, relying on our new uplink/downlink sniffer. Our sniffer records both the times of arrival of LTE messages and the contents of the Timing Advance Commands, based on which LTrack calculates locations. LTrack is the first to show the feasibility of a passive localization in LTE through implementation on software-defined radio.

Passive localization attacks reveal a user's location traces but can at best link these traces to a device's pseudonymous temporary identifier (TMSI), making tracking in dense areas or over a long time-period challenging. LTrack overcomes this challenge by introducing and implementing a new type of IMSI Catcher named IMSI Extractor. It extracts a device's IMSI and binds it to its current TMSI. Instead of relying on fake base stations like existing IMSI Catchers, which are detectable due to their continuous transmission, IMSI Extractor relies on our uplink/downlink sniffer enhanced with surgical message overshadowing. This makes our IMSI Extractor the stealthiest IMSI Catcher to date.

We evaluate LTrack through a series of experiments and show that in line-of-sight conditions, the attacker can estimate the location of a phone with less than 6m error in 90% of the cases. We successfully tested our IMSI Extractor against a set of 17 modern smartphones connected to our industry-grade LTE testbed. We further validated our uplink/downlink sniffer and IMSI Extractor in a test facility of an operator.

1 Introduction

LTE is one of the most widely deployed and used cellular technologies. It was designed to not only enable communication but also to protect the security and privacy of users by encrypting communication between a user equipment (UE) and a base station (eNodeB). Unlike the user's data, LTE physical and MAC layer control messages are transmitted in plain-text, with subscriber identifiers (IMSI) replaced with temporary identifiers (TMSI) to protect users' privacy.

LTE security and specifically the security and privacy on the wireless link between base stations and UEs is an active area of research. Broadly, attacks against LTE can be classified as active or passive, where active attacks (e.g., IMSI Catcher [19, 36]) typically rely on fake base stations to which victim UEs connect. Recently, message overshadowing emerged as a new active, but stealthier manipulation technique [13, 45].

On the other hand, passive attacks rely on custom-built sniffers. In [8, 23], it was shown that an attacker can build a passive downlink traffic sniffer (from the eNodeB to the UE) using software-defined radios. Downlink sniffers were then used as tools for localization [32], to break the encryption of phone calls [34], and to allow traffic fingerprinting [20]. The idea of passive uplink and downlink sniffing was further proposed for user localization [32] but was not implemented. Unlike downlink sniffing, so far, uplink sniffing was implemented only using active techniques and relied on fake base stations [33, 37].

In this work, we focus on large-scale, stealthy UE tracking. To be successful in such an attack, the adversary needs to: (i) determine the location of the UE, (ii) obtain a UE's identifier that links observed locations into a trace, and (iii) avoid detection. Until now, no attack fulfills all of the above at the same time. Passive localization alone could leak UE traces in some low-density areas, but in urban areas with a high density of UEs, this task will be harder without the identifier that binds the observed locations together [38].

IMSI Catchers, which are used to leak a UE's IMSI to the adversary and therefore identify the UE, rely exclusively on fake base stations. However, to get the UE to connect to the fake base station (a requirement of the attack), the attacker needs to transmit continuously at a high power and can therefore be detected by law enforcement and operators [24, 26, 30].

This paper addresses the above and shows that stealthy localization and identification (and therefore tracking) of UEs in LTE is indeed possible. We present LTRACK, a new tracking attack on LTE which combines passive and stealthy active attacks. For passive localization, we use LTEPROBE, our uplink/downlink sniffer, and for binding the collected traces to an IMSI, we use our active but stealthy IMSI Extractor.

Our work focuses on the recovery of users' long-term mobility traces. How this information is then further used by adversaries is well studied and out of scope of our work. Prior research showed that traces can be used to deanonymize users through transportation routines [25], mobility patterns [14,29,43,46], home addresses [15,18,21], co-locations with other users [27,40], or online geo-tagged media [17].

In summary, we make the following contributions:

- We demonstrate the feasibility of a fully passive adversarial localization of UEs in an LTE network. We show that, in line-of-sight conditions, the attacker can estimate the location of a phone with an error of less than 6m in 90% of the cases.

- We propose a new type of IMSI Catcher, named IMSI Extractor. Our IMSI Extractor does not rely on fake base stations but instead uses a combination of low-power surgical message overshadowing and uplink/downlink sniffing. Even if our catcher injects a message, it does so in line with LTE protocol specification, making it hard to detect with existing IMSI Catcher detection techniques. We discuss the techniques that would be needed to detect this attack. We successfully tested our IMSI Extractor on 17 smartphones connecting to an industry-grade eNodeB.

- We combine our passive localization and our IMSI Extractor into a UE tracking system that we name LTRACK, which enables simultaneous identification and localization of UEs, allowing an attacker to track users more persistently and with higher accuracy than in prior attacks. LTRACK does this by cross-checking IMSI-TMSI pairs obtained with our IMSI Extractor, with the location data identified by the TMSI obtained from our localization attacks.

- We implement the first white-box uplink and downlink LTE sniffer, called LTEPROBE. So far, only downlink sniffers were presented in open research. This sniffer is one of the core components of LTRACK. Our sniffer records both protocol level information, e.g., synchronization parameters or phone model specific messages, and physical layer timings of messages.

- Using our sniffer, we implement mobile phone fingerprinting, which allows the attacker to identify the make and the model of the phone. This allows us, in some scenarios, to further increase the accuracy of phone localization and tracking by as much as 20 meters.

2 Background

2.1 LTE

The radio access network in LTE is managed by base stations (eNodeB). eNodeBs route the traffic over a secure channel to the network core, which handles most mobile network functions. Our sniffer captures and analyzes the communication between a base station and a mobile phone (UE): downlink from eNodeB to UE, and uplink from UE to eNodeB. Most providers implement uplink and downlink separation using FDD-LTE (Frequency Division Duplex). In FDD, uplink and downlink use two separate RF carriers, one for each direction. Multiplexing is implemented using OFDMA in downlink and SC-FDMA in uplink.

Physical layer data transmission is scheduled in 10ms long frames for both downlink and uplink [3]. Frames are indexed from 0 to 1023 and split into ten subframes, each with a duration of 1ms. Each subframe consists of two slots. By default, a slot is made up of 7 OFDM symbols with one cyclic prefix per symbol.

In both OFDMA and SC-FDMA, data is modulated onto orthogonal subcarriers. Modulated data values are called frequency samples. Using inverse fast Fourier transformation, frequency samples are transformed into a time signal and transmitted over the radio. An LTE receiver samples the incoming signal into time domain samples. Fast Fourier transform over the time samples outputs the frequency samples. The smallest indexed element is a resource block [3] which spans 12 subcarriers and lasts one slot.

Physical Layer Channels. Data on the physical layer is sent over different channels [3]. Each channel occupies predefined resource blocks. Physical shared channels are used for data transmission, and control channels manage flow and access to them. The Physical Random Access Channel is used to establish new UE connections.

All resource allocations of resource blocks are communicated to the UE in Downlink Control Information (DCI) elements transmitted over the downlink control channel. A 16-bit RNTI number addresses each DCI and specifies the recipient of the message. Depending on the function, the RNTI number specifies one UE or multiple UEs. The format of the DCI determines its function.

DCI Format 0 allocates resource blocks on the uplink to UEs. A UE can transmit on the uplink shared channel only if it receives a corresponding resource allocation. The DCI Format 0 also specifies parameters to be used for the message encoding, such as modulation schemes.

DCI Format 1 or 2 defines which resource blocks a UE should decode and which parameters it should use to decode the messages on the downlink shared channel. The downlink shared channel carries user data and other system information, such as the configuration of the base station.

Connection Establishment. A UE uses two numbers for identifying to the network: IMSI, a unique, persistent identifier, and TMSI, a temporary identifier. Each UE connection starts with an RRC Connection Request containing the UE TMSI. If the TMSI is not available, the UE samples a random value and includes it instead of the TMSI.

There are two ways how a UE requests the service from the network. If the UE connects for the first time after losing a state (e.g., restarting), it initiates an attachment procedure by sending an Attach Request, containing the TMSI if one has been assigned previously, or the IMSI otherwise. If the network does not recognize the TMSI, it will ask the UE to provide its IMSI in an identification procedure. At the end of the attachment procedure, after the security context has been set up, the network assigns the UE a new TMSI. The TMSI is at this point both ciphered and integrity protected.

If the UE is already attached to the network but idle, going from idle state to connected state, it enters the service request procedure by sending an integrity protected Service Request, after which the connectivity is immediately restored.

2.2 Relevant Attacks

Localization Attacks. By observing paging messages alone, an attacker can learn if a victim is currently in the same tracking area or the same cell (if smart paging is deployed), as shown in [36].

With the victim connected to the same base station as the attacker, more advanced attacks can be executed. As proposed in [32], an attacker can observe control messages on the MAC layer that contain propagation delay correction information. This information alone constrains the location of the victim to a 78 meters wide ring around the eNodeB with its perimeter defined by the propagation delay correction.

Localization attacks based on fake base stations [19] are even more accurate. However, we do not consider them stealthy enough to be used in a large-scale tracking attack.

IMSI Catchers. As mentioned in Section 1, for areas with a high density of UEs, the attacker needs to be able to obtain the identity of the victims in order to track them. The most potent attacks in this area are IMSI Catchers [19, 36], which reveal the unique IMSI number to the attacker. However, these attacks all rely on fake base stations.

3 LTEPROBE

The key component to make the stealthy tracking possible is the implementation of a combined uplink/downlink LTE sniffer that we name LTEPROBE. In what follows, we describe LTEPROBE and its abilities. As already discussed, downlink sniffing (see, e.g., [8, 23]) allows the attacker to record unencrypted Downlink Control Information and control elements on MAC layer. With an uplink sniffer, however, the attack surface increases substantially. Unencrypted messages, such as the initialization messages, can be used by the attacker for the leakage of users' identifiers. All uplink messages, even encrypted ones, can be used for precise time of arrival measurements.

3.1 System Architecture

We designed LTEPROBE to be a fully passive device and therefore virtually undetectable. LTEPROBE receives RF samples on both uplink and downlink. It records all communication between mobile phones and base stations but does not break encryption. LTEPROBE has a stable clock and synchronizes its reception to the base station. The clock drift between the base station's and LTEPROBE's clock is negligible, because both devices use GPS synchronized clocks.

LTEPROBE consists of two components: DOWNLINKPROBE, the downlink sniffer, and UPLINKPROBE, the uplink sniffer. DOWNLINKPROBE works as a standalone analyzer for downlink, but UPLINKPROBE requires scheduling information shared by DOWNLINKPROBE. The uplink and downlink sniffing is feasible due to the unencrypted DCI messages carried over the downlink control channel.

DOWNLINKPROBE first synchronizes to the base station and records identifiers of connected UEs. The LTE protocol specifies temporary RNTI numbers for the identification of UEs on the physical layer for the duration of the connection. With the RNTI, DOWNLINKPROBE finds and decodes messages intended for the victim UEs.

On the physical layer, when a UE connects to an eNodeB, the eNodeB replies with a Random Access Response. Inside the Random Access Response, the eNodeB specifies a new RNTI for the UE. Because the Random Access Response is sent in a plain-text, it is visible to DOWNLINKPROBE. However, this method can be used only for new connections. For already connected UEs, the assigned RNTI is not exchanged in plain-text, but coded into the CRC of DCI messages. The work in [23] describes a method for extracting the RNTIs from the DCIs, however, for our use-cases this is not necessary.

To decode downlink channels, DOWNLINKPROBE computes inverse OFDMA transformation to receive frequency samples. It performs channel correction and frequency offset correction. It then goes over all possible locations of DCIs for the set of recorded RNTIs and tries to decode them. Depending on the format of the decoded DCI, DOWNLINKPROBE either uses it to decode the PDSCH message or shares it with UPLINKPROBE. Finally, DOWNLINKPROBE parses PDSCH messages to get higher-layer messages, e.g., NAS-layer messages containing dedicated UE configuration. Figure 1 shows how DOWNLINKPROBE receives RNTIs in

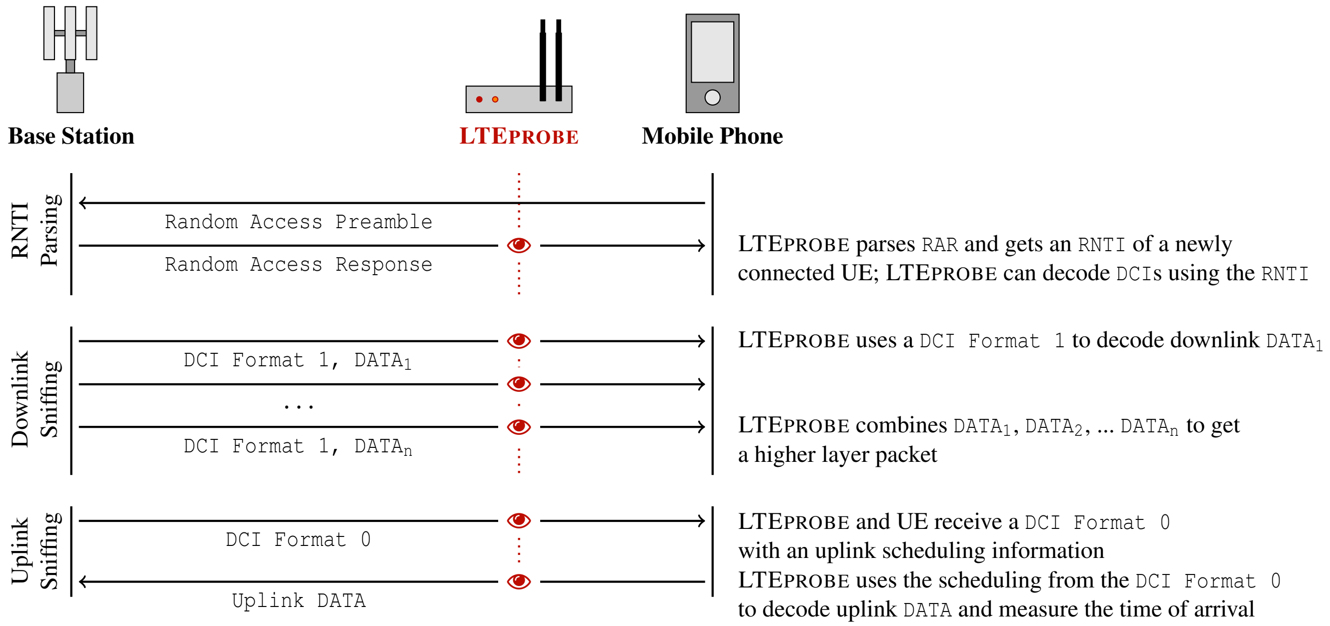

Base Station
LTEPROBE
Mobile Phone
RNTI Parsing
Random Access Preamble
Random Access Response
LTEPROBE parses RAR and gets an RNTI of a newly connected UE; LTEPROBE can decode DCIs using the RNTI
Downlink Sniffing
DCI Format 1, DATA1
...
DCI Format 1, DATAn
LTEPROBE uses a DCI Format 1 to decode downlink DATA1
LTEPROBE combines DATA1, DATA2, ... DATAn to get a higher layer packet
Uplink Sniffing
DCI Format 0
Uplink DATA
LTEPROBE and UE receive a DCI Format 0 with an uplink scheduling information
LTEPROBE uses the scheduling from the DCI Format 0 to decode uplink DATA and measure the time of arrival

Figure 1: Decoding of uplink and downlink channels by LTEPROBE. First LTEPROBE records an RNTI of a UE. Then it uses it to decode DCIs. DCIs either specify resource blocks containing downlink or uplink data.

Random Access Response and then obtains shared channel data.

UPLINKPROBE receives samples transmitted from multiple UEs. Similar to the eNodeB, it demodulates them and applies channel correction. Afterward, it tries to decode uplink shared channels and control channels.

The physical uplink shared channel is decoded according to the scheduling information. eNodeB controls the scheduling in the LTE protocol, so it knows the scheduled resource allocations. In our case, UPLINKPROBE has to obtain the scheduling information from the DCI messages the same way the UE receives them. UPLINKPROBE uses the passed DCI Format 0 messages from the downlink sniffer containing the scheduling information. Because DCI Format 0 messages carry the resource allocations for future uplink transmissions, without a downlink sniffer, UPLINKPROBE would not be able to decode uplink channels. Figure 1 visualizes the procedure of UPLINKPROBE.

To correctly decode uplink shared and control channels, UPLINKPROBE has to apply a dedicated UE configuration sent via a RRC-layer downlink message. UPLINKPROBE again uses information recorded by DOWNLINKPROBE. Similar to DOWNLINKPROBE, physical layer messages are parsed to receive higher layer messages.

LTEPROBE Implementation

We base our implementation on srsLTE [16], an open-source library for the LTE protocol. The two main components, DOWNLINKPROBE and UPLINKPROBE, run on two separate co-located USRP devices. The two components run as two threads of a parent LTEPROBE program.

Both downlink and uplink subframes are scheduled at the same time. Regular UEs learn the timings of the subframe from the synchronization signals transmitted by the eNodeB. Similarly, our LTEPROBE synchronizes to the eNodeB by observing the synchronization signals. However, only DOWNLINKPROBE is receiving them. Therefore, the uplink and downlink threads of LTEPROBE need to share timings and subframe numbers; otherwise, UPLINKPROBE would not be able to receive the uplink subframes at the correct time. To synchronize precisely, the two USRPs need to have the same time reference. This can be solved by using a GPSDO on both USRPs to have the same GPS clock or by using an Octoclock, a clock distribution module.

For each subframe, both UPLINKPROBE and DOWNLINKPROBE record the subframe index and the exact time they received it. If the timestamps for the same subframe index do not match, UPLINKPROBE has to adjust its reception time by discarding time samples. A perfect synchronization of the two components is then achieved.

Unless LTEPROBE is co-located with the eNodeB, the uplink messages sent by UEs will not be perfectly time synchronized to the frames at LTEPROBE location (due to different propagation delays). We tested the robustness of the LTEPROBE under such misalignment of the attacker in Figure 12 in the Appendix. Our results show that the attacker can still decode the messages under $< 4\mu s$ misalignment

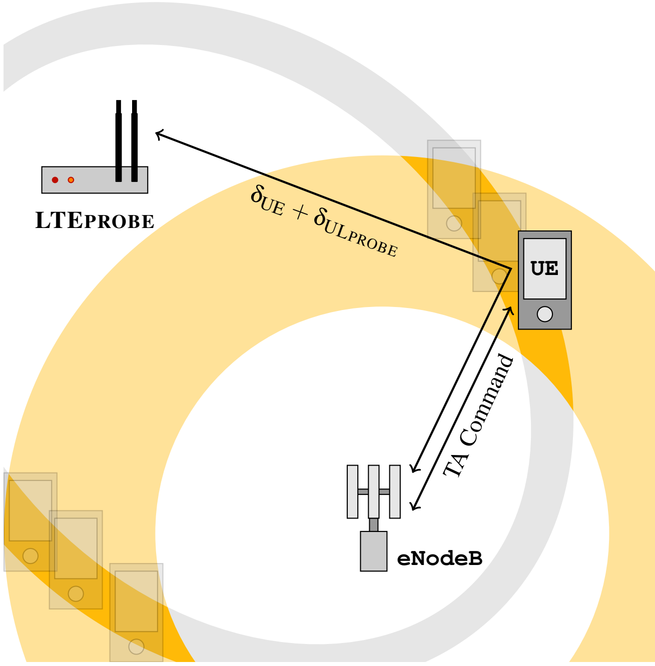


Figure 2: Passive localization attack using a single sniffer. The yellow ring is defined by the received Timing Advance Command, and the grey ellipse is defined by the time of arrival measured by LTEPROBE. The intersection of the two rings defines possible locations of the mobile phone.

(which corresponds to 1.2*km* distance). However, even if this misalignment would be larger, because UPLINKPROBE and DOWNLINKPROBE are independent devices, the attacker can apply time correction to the uplink messages and still correctly decode uplink messages.

4 Passive Localization Attack

Passive localization of UEs in LTE networks was proposed in a number of prior works [23, 32, 36]. Most notably, Roth et al. [32] proposed a passive localization attack that leverages synchronization parameters sent on the MAC layer (Timing Advance Command) and times of arrival of uplink and downlink messages.

Specifically, the attack proposed in [32] works by observing the Timing Advance Command containing propagation delay correction information. Because of the coarse granularity of the Timing Advance Command, the attack constrains the location of the victim to a 78 meters wide ring around the eNodeB. Furthermore, in LTE-Advanced, the UE has an option to connect to multiple cells at once. Multiple delay correction information then constrains the victim's location to the intersection of the rings. Finally, Roth et al. [32] proposes an idea of localizing a UE based on times of arrival of uplink messages, which would allow the attacker to constrain the victim's location to an additional 78 meters wide ring around the attacking device. However, the authors do not provide details, simulations, or implementation of this proposal.The LTE Positioning Protocol [2] has been standardized/implemented for localization in LTE. One supported technology is estimating the location from the UE with the observed time difference of arrival (OTDOA) of downlink transmission from multiple base station in the vicinity. The mechanism of the OTDOA method is the same as the one proposed by Roth et al.

Our passive localization attack also exploits unciphered Timing Advance Command and time of arrival of uplink messages. However, contrary to their work, we transform the geometry of the problem from a circle to an ellipse. This transformation enables us to remove the systematic error due to the course-grained Timing Advance Command. Instead of having an additional 78m wide ring around the sniffer, we have a precise ellipse with focal points at the base station and the sniffer, as drawn in Figure 2.

The most significant contribution of our attack is the actual implementation and its evaluation in Section 7. Our implementation revealed the imprecision of the hardware inside the mobile phones. To solve this problem, we have transformed the active fingerprinting attack introduced in [37] into an entirely passive attack in Subsection 4.4. Knowing the model of mobile phones can increase the precision of the localization by as much as 20 meters. In our work, we do not study the effects of the radio environment (e.g., multi-path propagation or shadow-fading) as these topics are orthogonal to our research. For performance reasons, any positioning system has to account for such conditions. The work in [41] provides an example how one can use the error budget to compute the precision of localization system under different channel conditions.

In this section, we develop this passive localization attack, provide its mathematical basis and describe its implementation. The attacker can constrain the victim's location to two possible areas as shown in Figure 2 using just one sniffing device and a base station. The two possible areas are the intersection of a wide ring defined by the Timing Advance Command and an ellipse defined by the time of arrival of uplink messages. Using two or more sniffing devices results in the attacker learning the location of the victim. Alternatively, the adversary can rule out possible locations by cross-checking with, e.g, a detailed map of the area.

4.1 Timing Advance Command

Multiple UEs connect to eNodeB at the same time. Each UE is at a different distance. Due to a propagation delay, without any corrective mechanism, uplink messages would be received with a different delay. Thus, the eNodeB needs to help correct each UE's timing to ensure alignment of all uplink messages within the resource blocks as observed by the eNodeB.

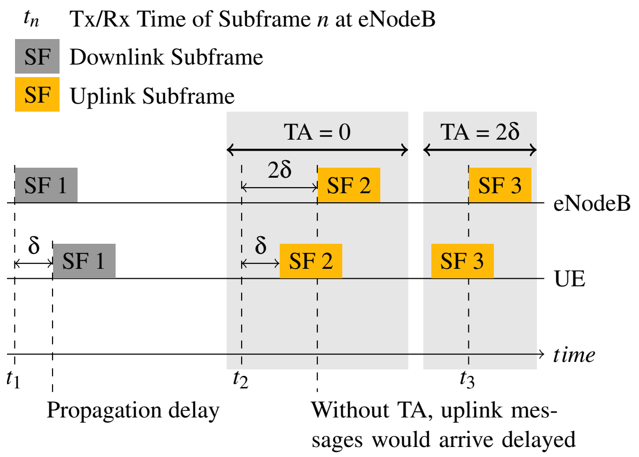


Figure 3: Timing Advance is used to align uplink transmissions. Transmission and reception at eNodeB are synchronized.

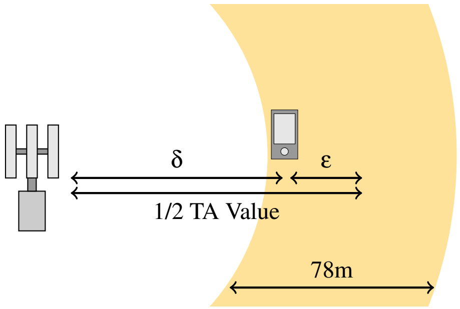


Figure 4: The propagation delay between the eNodeB and UE is δ, but the received Timing Advance value corresponds to a distance in the middle of yellow ring. The difference between these two values is a systematic error ε.

Figure 3 shows a situation where the propagation delay between the UE and eNodeB is δ. Due to the propagation delay of the downlink message, the frame synchronization of the UE is being shifted by δ from the eNodeB's time. The propagation delay of an uplink message is again δ. Therefore, the uplink message arrives at the eNodeB with a delay of 2δ. The eNodeB measures the delay and signals it to the UE with a Timing Advance (TA) Command.

The LTE specification [4] defines that the Timing Advance value is expressed as $T_A \times 16 \times T_S$, where $T_S = 1/30720ms$. T_A is the value signalled by the eNodeB. The T_A value is sent as a part of the MAC control element. It is sent unciphered by the eNodeB on the MAC layer.

The granularity of the TA is therefore $T_S \times 16 = 0.5208\mu s$. The UE does not receive a more precise value for the propagation delay δ. Given that the propagation speed is the speed of light, the UE can estimate its distance from the eNodeB in a range of $78.07m$ ($156.14m$ divided by 2 because of the round-trip). Figure 4 visualizes the difference between the actual distance of the UE and the eNodeB and the distance the UE computes from the TA Command.

4.2 Times of Arrival of Uplink and Downlink Messages

Localization attacks based on the time difference of arrival of a victim's messages constrain the victim's location to the intersection of multiple hyperbolas. The attacker can use the time difference of arrival between uplink and downlink message to define a hyperbola between the attacker and the base station. In the case of LTE, due to the systematic error introduced in the Timing Advance value, the attacker using this classical approach ends up with a 78m error. However, we show how the attacker can formulate the problem using ellipses and cancel the systematic error. We define the following variables to explain the unique localization problem in LTE:

t_n the time of the transmission of the downlink subframe n by the eNodeB. Tge UE tries to send the uplink subframe n such that it arrives at eNodeB at time t_n.

δ_{UE} propagation delay between the eNodeB and the UE.

δ_{DLPROBE} propagation delay between the eNodeB and LTEPROBE. We assume this value is known to the attacker since it knows the location of both the eNodeB and LTEPROBE.

δ_{ULPROBE} propagation delay between LTEPROBE and the UE.

δ_{TA} time corresponding to the TA value received in the Timing Advance Command.

ε systematic error TA value introduces due to discretization of the propagation delay. It is the difference between the propagation delay and the TA value shown in Figure 4 and its value ranges from $-0.1302\mu s$ to $0.1302\mu s$. We know that $\delta_{\text{TA}} = 2\delta_{\text{UE}} + 2\varepsilon$.

The attacker measures the time of arrival of downlink and uplink messages using LTEPROBE with subsample precision. The attacker uses the reference signals sent with each transmission for the timing estimation. Therefore, the attacker can collect independent measurements for each transmission and use a rolling average to smooth out any inconsistencies.

Downlink Message.

Tx at eNodeB	t_n
Rx at UE	$t_n + \delta_{\text{UE}}$
Rx at LTEPROBE	$t_n + \delta_{\text{DLPROBE}}$

Since δ_{DLPROBE} is known to the attacker, it can compute t_n from the reception time of the downlink message. The attacker can infer the times of transmission of the subsequent subframes as $t_{n+k} = t_n + k$, since the subframe length is 1ms.

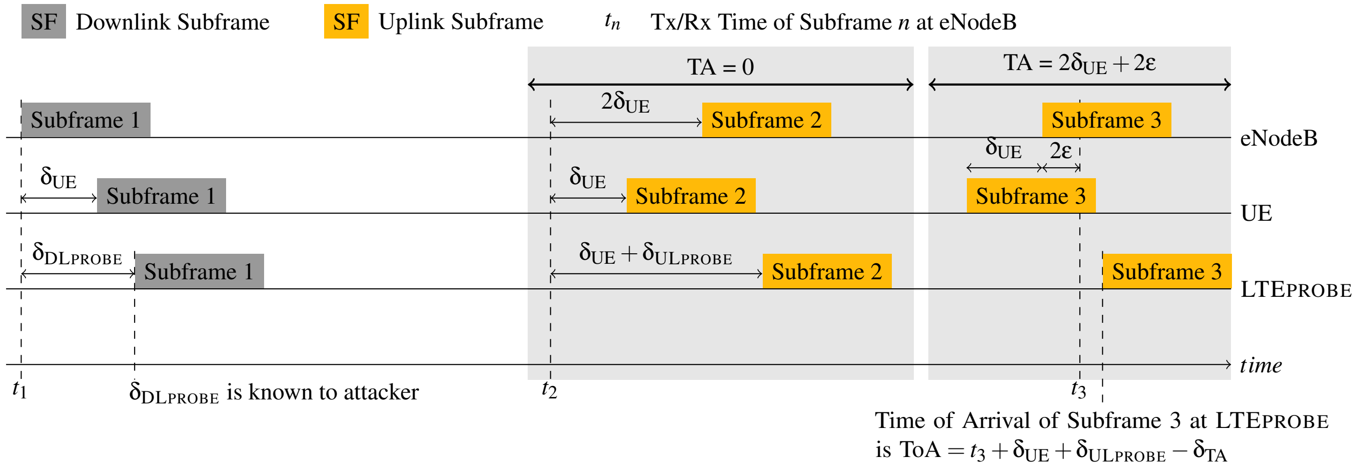

SF Downlink Subframe
SF Uplink Subframe
t_n Tx/Rx Time of Subframe n at eNodeB
TA = 0
TA = $2\delta_{UE} + 2\varepsilon$
$2\delta_{UE}$
Subframe 1
Subframe 2
Subframe 3
eNodeB
δ_{UE}
2ε
UE
$\delta_{DLPROBE}$
$\delta_{UE} + \delta_{ULPROBE}$
LTEProbe
time
t_1
$\delta_{DLPROBE}$ is known to attacker
t_2
t_3
Time of Arrival of Subframe 3 at LTEProbe
is ToA $= t_3 + \delta_{UE} + \delta_{ULPROBE} - \delta_{TA}$

Figure 5: Visualization of times of arrival and delays of uplink and downlink messages.

Uplink Message without TA Command.

Tx at UE	$t_n + \delta_{UE}$
Rx at eNodeB	$t_n + 2\delta_{UE}$
Rx at LTEProbe	$t_n + \delta_{UE} + \delta_{ULPROBE}$

The UE receives all the downlink messages delayed with δ_{UE}, therefore its synchronization is shifted as explained in Subsection 4.1. It will transmit uplink messages at time $t_n + \delta_{UE}$ instead of t_n. The Attacker computes t_n from the downlink message. It can measure $\delta_{UE} + \delta_{ULPROBE}$ from the reception time of the uplink message by subtracting t_n.

Uplink Message with TA Command.

Tx at UE	$t_n + \delta_{UE} - \delta_{TA} = t_n - \delta_{UE} - 2\varepsilon$
Rx at eNodeB	$t_n + 2\delta_{UE} - \delta_{TA} = t_n - 2\varepsilon$
Rx at LTEProbe	$t_n + \delta_{UE} + \delta_{ULPROBE} - \delta_{TA} =$
	$= t_n - \delta_{UE} + \delta_{ULPROBE} - 2\varepsilon$

The Attacker can no longer precisely compute $\delta_{UE} + \delta_{ULPROBE}$ by subtracting t_n because of the error 2ε which can range from $-0.2604\mu s$ up to $0.2604\mu s$.

4.3 Localization

In the previous two subsections, we saw two sets of information that the attacker can use to localize a victim: Timing Advance Command sent by the base station on MAC layer and the times of arrival of uplink and downlink messages at LTEProbe. Figure 2 visualizes the attack and possible locations of the victim's phone in the environment.

The simple localization attack works by sniffing TA Commands since they are transmitted unciphered on the MAC layer of LTE protocol. Therefore, TA Command can be recorded by our DownlinkProbe. Because of the coarse granularity due to discretization of the TA value, TA Command localization constricts possible location to a ring around the downlink sniffer with a width of 78m (yellow ring in Figure 2).

We saw in Subsection 4.2 how the attacker learns times of transmission of subframes t_n from the time of arrival of downlink messages. When LTEProbe receives a Downlink Control Information with scheduling for uplink transmission of the victim, it decodes the uplink message and measures its time of arrival. The time of arrival of the uplink message to LTEProbe is:

$$\text{ToA} = t_n - \delta_{UE} + \delta_{ULPROBE} - 2\varepsilon$$

By subtracting the subframe transmission time t_n, the attacker gets a time difference of arrivals of the uplink and the downlink message. The attacker is able to then define a hyperbola of possible locations with an error 2ε. In our approach we instead subtract the subframe time t_n and add the value leaked from the TA Command to learn the sum of distances:

$$\delta_{UE} + \delta_{ULPROBE} = \text{ToA} - t_n + \delta_{TA}$$

Therefore, we are able to completely cancel out the systematic error ε from the equation. The measured sum of the two propagation delays δ_{UE} and $\delta_{ULPROBE}$ constraints a set of possible locations of the victim's UE as:

$$d_{UE} + d_{ULPROBE} = c \times (\delta_{UE} + \delta_{ULPROBE})$$

, where d_{UE} is the distance between UE and eNodeB, $d_{ULPROBE}$ is the distance between UE and LTEProbe, and c is the speed of light in the air. This constraint defines an ellipsis with two focal points: LTEProbe and the base station.

The location is now constricted to the intersection of a ring and an ellipsis shown in Figure 2. Using just one sniffer, the attacker gets two narrow location areas.

The attacker can significantly improve the precision of TA Attack by employing multiple LTEProbes in different locations. The final UE location lies at the intersection of multiple precise ellipses. However, it introduces extra complexity and increases the cost of the attack.

4.4 Passive Fingerprinting Attack

Hardware Error. There are four hardware devices in the system: eNodeB, DOWNLINKProbe, UPLINKProbe, and the victim's UE. All four add a slight timing error due to the circuit design, length of the cables, antennas, etc. We assume the hardware error is constant for the specific model of the device. We have not observed the error changing during the experimental evaluation of the attack in Section 7. Software-defined radios in LTEProbe are chosen by the attacker and the base station's hardware is selected by the operator but visible to the outside world. The only device the attacker cannot foresee in the system is the victim's UE. However, the attacker can build a database with various phones and corresponding hardware errors. If it can then identify the phone type of the victim, it can look up the corresponding hardware error.

Passive Fingerprinting. To learn the hardware error introduced by the phone model, we modify and extend the attack by Shaik et al. [37]. This attack analyzes the uplink traffic and classifies the baseband modem of the phones connected to the cell. A baseband modem is the chip responsible for mobile network communication. The attack in [37] uses a relay base station to decode uplink information; therefore, this is an active attack. We instead use LTEProbe to receive uplink information. Our improvement makes the attack entirely passive, and we show how it can be used for modem and phone type fingerprinting using the decision tree model.

As a feature vector used in fingerprinting, we use the core capabilities sent in plain-text with the Attach Request. Each phone has different capabilities implemented; therefore, these messages differ significantly.

Figure 6 shows a PCA decomposition of the feature vector for all the tested phones, excluding iPhones. We can see that phones with the same modem manufacturer have similar core capabilities (see Table 2 for list of phones and their corresponding modem). We do not include iPhones in the visualisation for clarity reasons, as iPhones are clustered together far away from other phones. We can see four clusters in Figure 6. Green are the phones with Huawei modem, yellow with Samsung modem, and the blue and purple clusters are Qualcomm phones. Blue phones are older models of phones, whereas purple ones correspond to recent models. The only exception is the OnePlus 7T which was clustered with old Qualcomm phone models. Four models of phones pictured in the Figure 6 have the same modem: Xiaomi Mi9, Xiaomi MiX 3 Google Pixel 4, and OnePlus 7T. OnePlus 7T is an outlier; however, the other three phones still do not have the same feature vector as they are only clustered closely together. Therefore, the capability object depends both on the modem and phone model. Thus, the attacker can learn the exact fingerprint of each phone model.

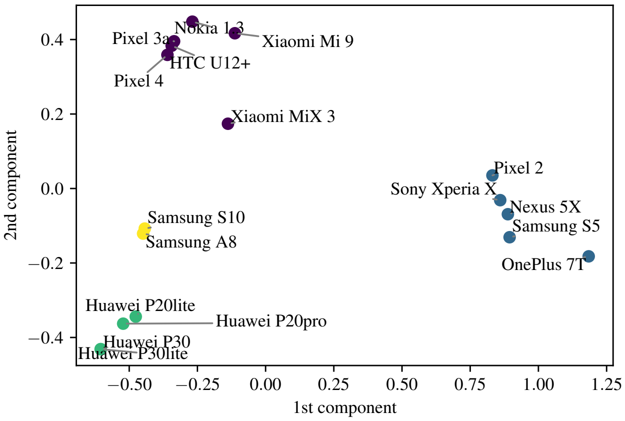


Figure 6: First two components of PCA decomposition of the feature vector.

5 IMSI Extractor

To associate UEs with a unique key and therefore facilitate their tracing, we propose a new identification attack based on message overshadowing and LTEProbe. For message overshadowing, we use AdaptOver [13], a recently proposed LTE overshadowing attack. In AdaptOver, the attacker sends a message perfectly aligned with the base station's message timing and frequency, but with up to 3dB higher power, thus replacing the original with the attacker's message. To the UE, the attacker's messages are indistinguishable from legitimate messages.

In this section, we show that by combining sniffing on the uplink and AdaptOver injecting just one adversarial message, we can get the UE to leak the IMSI. Since each SIM card has a unique, persistent IMSI number, the attacker perfectly distinguishes the victim with this attack. Even though the attack is active, the attacker can choose the granularity of when it wants to perform the attack as well as only target specific UEs. Our attack is triggered when the eNodeB sends a RRC Connection Setup, as pictured in Figure 7. This happens when the UE goes from an off or idle state to a connected state (e.g., the phone receives a paging message or needs to transmit data).

Identification of a Victim. As shown in the Figure 7, the UE sends an initial RRC Connection Request containing its TMSI number. However, because the LTE network can change this identifier at any time, the attacker does not have any assurance about UE's long-term identity. Instead, the IMSI number satisfies this, but the UE does not transmit IMSI in

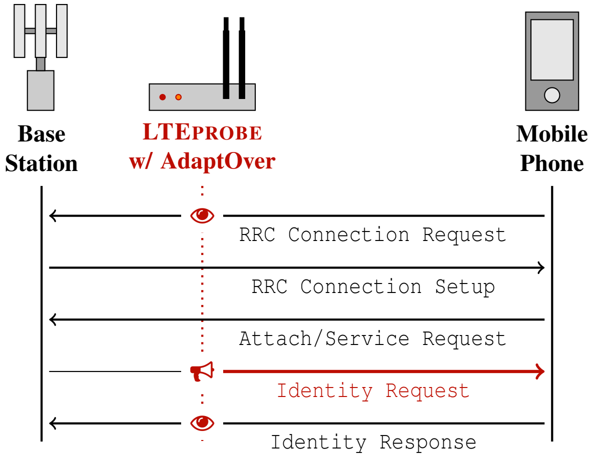


Figure 7: The attacker sniffs a Connection Request containing the UE's TMSI. After receiving a Connection Setup, the attacker overshadows the message sent by the base station with an Identity Request message. The attacker then sniffs the Identity Response from the UE and learns its IMSI. The attacker is able to link the temporary identifier TMSI to the unique persistent IMSI.

plain-text in the usual behavior of the protocol. Nevertheless, the LTE protocol allows the core of the network to request the IMSI number at any time (e.g., when the network loses the TMSI number) by sending an Identity Request.

5.1 Overshadowing with Identity Request

Specified in [5], an Identity Request for the IMSI number can be sent by the eNodeB without any integrity protection before the security context is created. Since the security context is not set up before the Service or Attach Request, the attacker can inject an Identity Request as a response to those requests. The UE will respond to the Identity Request with an Identity Response message containing its unique IMSI number, which LTEPROBE receives. Figure 7 shows the message exchange. Even though the legitimate base station proceeds with a connection procedure, AdaptOver sends a message with a higher power, overshadowing it. Thus, the UE only decodes the Identity Request that is sent by the attacker. The base station does not receive the Identity Response sent by the UE, because AdaptOver also modifies the uplink allocation during the attack. Overall the attack requires a limited number of transmissions by the attacker, with only a slightly higher power than the base station.

It is essential to point out that using Identity Request is just one concrete approach to how IMSI Extractor can operate. However, the attacker is not constrained by this and can create other protocol compliant communication traces, which trigger IMSI transmission in plain-text by the UE (e.g., Service Reject with cause 9, "UE identity cannot be derived by the network").

We present the first attack that combines the overshadowing attack with an uplink sniffing to violate user privacy. Earlier overshadowing attacks like SigOver [45] and AdaptOver [13] focused on denial of service.

Stealthiness of Our Attack. To a UE, the message exchange with a spoofed Identity Request looks benign. According to the LTE specification [5], a network can start an identification procedure at any time, even right after it received an Attach Request or Service Request. Therefore, from the protocol-level point of view at the UE, our attack does not raise any alarms. The base station also does not notice any problems. From the perspective of the eNodeB, the connection with the UE halted (e.g., due to bad reception at the UE). For both the UE and the base station, the traces generated by the attacker's messages are therefore compliant with the protocol.

Current detection mechanisms against IMSI Catchers work by detecting fake base stations [7, 9, 24, 26, 30]. These frameworks either work by comparing open-sourced locations of base stations to measured reports by users or special devices, or by detecting anomalies in the behavior of base stations by UEs. In case of our attack, these techniques do not work since a UE connects to a real base station. Therefore, to UEs, the behavior and location of the cell are legitimate. As proposed in [12], a signature based anomaly detector with a signature: "if Identity Request, then attack", is successful in the detection of our attack. However, because Identity Requests are also sent during a legitimate protocol flow, such a solution will inherently report false positives during legitimate identification procedures. Moreover, the attacker is not constrained to sending Identity Requests to perform IMSI Extractor, as mentioned above.

As explained in [7, 9], other features (e.g., number of neighbouring cells) are considered in most of the anomaly-based IMSI Catcher catching apps. Evaluating them, our catcher is not classified as an IMSI Catcher. Therefore, we consider our attack to be stealthy, at least with respect to existing deployed and proposed techniques.

6 LTRACK

In this section, we discuss how the techniques that we introduced in Section 4 and Section 5 can be put together to support large-scale tracking attacks, similar to those described e.g., in [38]. The goal of such a tracking attacks is to obtain traces of all users while staying as stealthy as possible.

Figure 8 visualizes a city setting where the attacker tries to localize users. The attacker uses the passive localization attack to locate individual users during their connections to base stations. However, without identification, all the UEs look the same, and after each reconnection, UEs might anonymize themselves with a new TMSI. If the user moves along less

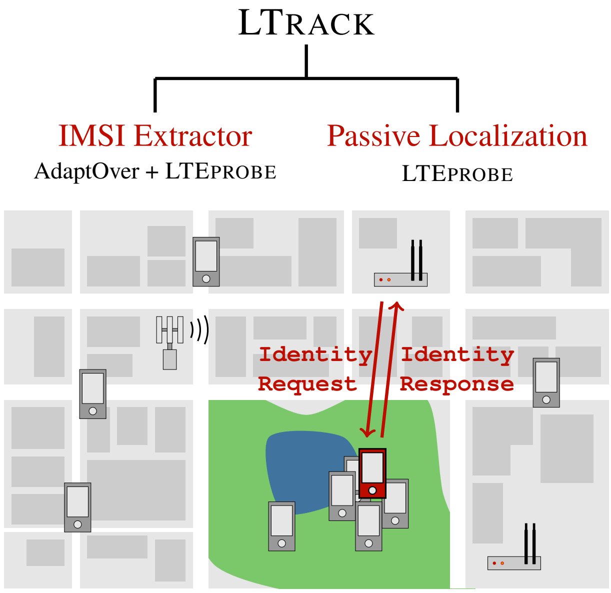


Figure 8: Visualization of LTrack, tracking attack based on passive localization and IMSI Extractor. When the attacker loses track of a victim in a natural mix-zone, the attacker uses the IMSI Extractor to distinguish the victim from other UEs in the area.

frequented areas when the TMSI gets updated, the attacker could still link the two temporary identifiers based on their locations. However, as an UE enters an area with many other UEs, this area will act as a natural mix zone. LTrack solves this problem by using a combination of passive tracking and IMSI Extractor, allowing the attacker to distinguish UEs.

In order to launch our attack on a large-scale, the attacker needs to deploy at least one, preferably two, LTEProbes for each base station the attacker decides to monitor. LTEProbes are placed away from the base stations such that the attacker can perform the localization attack pictured in Figure 2. We do not put any restrictions on the attacker in terms of available funds or access to the locations. The attacker can place its devices at high vantage points (e.g., skyscrapers or communication towers). Building such a network of devices is feasible. Competitor service providers often already have devices (base stations) at preferred locations, which they can transform into sniffers. Our attack works in the following four stages:

(i) Communication Recording. The attacker uses LTEProbes to passively record all the traffic on the set of base stations it monitors. All the uplink and downlink communication with corresponding arrival times from all LTEProbes is stored in the attacker's database. All messages during a UE's connection to the eNodeB are addressed on the physical layer by a unique RNTI value as explained in Section 3, linking the messages together. Moreover, each connection of a UE to the network starts with an RRC Connection Request containing the TMSI of the user. The attacker stores a list of TMSIs observed during the attack's execution and links them to the corresponding connections. To link connections of the same user, whose TMSI changed during the execution, the attacker runs IMSI Extractor, described below.

(ii) IMSI Extractor. IMSI Extractor extracts and stores TMSI-IMSI pairs of the users, linking observed communication to the unique, persistent identifier of the UE (IMSI) as explained in Section 5. Once LTEProbe registers RRC Connection Request, the attacker checks whether it already knows the corresponding IMSI to the enclosed TMSI inside the RRC Connection Request. If the TMSI-IMSI pair exists in the database, the attacker does not engage, passively records the communication, and links the communication to the stored pair. However, if the TMSI has not been seen before, it runs the IMSI Extractor to learn the IMSI number and stores the new TMSI-IMSI pair in the database.

(iii) Passive Localization. Finally, the attacker has recordings of all the users at multiple base stations under different TMSI-IMSI pairs. The attacker uses recorded data to get each UE uplink message's time of arrival and Timing Advance Commands sent by the base station. As shown in Figure 2, each uplink message measurement constrains possible locations of the user.

Moreover, the attacker runs a passive fingerprinting attack on the saved recordings of Attach Requests to learn the phone model. With the model of the phone, the attacker can increase the precision of the localization attack. Furthermore, since the attacker stores all the recorded communication, it can retroactively compensate hardware error to increase the precision of measured times of arrival of an uplink messages for that user. Therefore, even if the user's TMSI changes, we will update it in the database during the subsequent Service/Attach Request.

We can further improve the precision of localization by choosing more likely locations, e.g., the user probably moves along the street, not through the walls of buildings. Altogether, the attacker can visualize the movement of the victim. Finally, the attacker builds a whole trace of a user's movement.

(iv) Special Cases. Under certain conditions (i.e., handover), a UE stays connected to the network but changes the serving cell to a cell with a stronger signal. Then, the UE disconnects from the old cell and performs a random access procedure with the new cell. Since it is still connected to the network, there is no need for a Service Request message. Thus, the attacker can observe new random access without a Service Request, and it can match it to a connection that halted at a neighboring cell. If the localization attack is in place, the attacker can improve the matching between the old and new connections based on the location of the UE.

Even if the attacker loses track of a UE, the attacker observes the UE again during the next Service Request it performs. For example, for an inactivity timer of 10 seconds, on

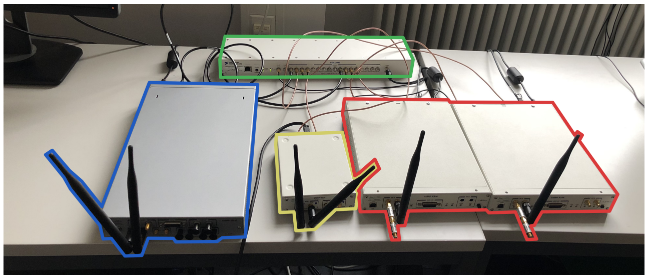


Figure 9: Our setup used for the evaluation of the passive localization and the IMSI Extractor.

average, a UE connects to the network more than once per minute under background traffic (i.e., a user does not actively use a phone) [1], which is a usual scenario during a movement of a person. The attacker can also force a reconnection using a paging message or a call.

In this work we do not address user deanonymization. Research in this area is already quite extensive and the attacker can use multiple existing techniques to obtain true user identities. User traces reconstructed by LTRACK can be used to identify users [22, 44], for example, based on transportation routines [25], mobility traces [14, 29, 43, 46], home addresses [15, 18, 21], who they meet [27, 40], or online geotagged media [17]. [10, 11] show that even coarse spatial and temporal traces deanonymize the users based on their unique mobility patterns.

7 Experimental Evaluation

7.1 Experimental Setup

For the experimental evaluation of our attack we used the setup pictured in Figure 9. It consists of:

eNodeB running on software defined radio USRP N310, highlighted in the blue color in Figure 9. Alternatively, we use an entry-grade base station AMARI Callbox Mini [6] for the evaluation of IMSI Extractor attack. However, due to its lower grade clock, the timing is inaccurate. Thus, we do not use it for the localization attack, where the accuracy is necessary.

LTEPROBE running on two USRP X310 SDRs, highlighted in the red color in Figure 9. One X310 is used as a DOWNLINKPROBE and the other as UPLINKPROBE. There is no antenna connected to the Tx port of the radios confirming it is a passive device. Both devices are connected to the Octoclock to share the same clock.

Octoclock model CDA-2990, highlighted in the green color distributes the same clocking signal to all connected devices. It takes the GPS signal as input. All connected devices have the same sense of time. The two sniffing USRPs are always connected to Octoclock.

AdaptOver running on software defined radio USRP B210, highlighted with in the yellow color in Figure 9.

UE During the experimental evaluation, we use multiple mobile phones as UEs. The full list of UEs is recorded in Table 2 and Table 1 in the Appendix.

7.2 Passive Localization Attack

For the experimental evaluation of our localization attack, we collocated the eNodeB and LTEPROBE, and we varied the location of UEs. Instead of the location, we estimated the distance of the LTEPROBE from a UE. In our experiment, we learn the measurement error of LTEPROBE, which we can use to quantify the localization error under various dilutions of precision. Since eNodeB and LTEPROBE are at the same location, the distance of the UE from LTEPROBE is:

$$d_{\text{ULPROBE}} = c \times (\delta_{\text{UE}} + \delta_{\text{ULPROBE}})/2$$

We conducted the experiment with five different UEs: USRP B210 with srsUE, Huawei P20 Pro, Huawei P30, iPhone X, and iPhone 8. We positioned the UE in line-of-sight at six different distances in a long corridor indoors: $0m$, $7.5m$, $15m$, $30m$, $45m$, and $60m$. For each distance and UE, we reconnected six times to measure the distance over multiple connections. For each distance measurement and UE, we restarted LTEPROBE at least once to reset the synchronization errors.

The accuracy of the internal clock without a GPS is ±2.5ppm and ±0.1ppm for the USRP X310 and USRP N310 respectively. This accuracy improves to ±0.01ppb for both types of devices when the GPS lock is acquired. Since we could not acquire a GPS lock in our environment, we instead had both the eNodeB and the LTEPROBE connected to Octoclock, which we use as a proxy to the GPS acquisition in a real world scenario. Assuming Octoclock provides perfect accuracy (0ppb), using a GPS locked clock instead of Octoclock introduces a synchronization error of merely ±0.02ppb. This error would translate to an error of 40nm while performing distance measurement over 1 km. We therefore consider the improvement of using Octoclock negligible. We did not use Octoclock to synchronize the attacker's devices and the eNodeB.

In our experiment, one data point corresponds to the median distance measurements during one connection of the UE to our eNodeB. We do not consider connections for which we have less than ten measurements. For each UE, there is a constant hardware error that comes from properties of UE modem, LTEPROBE radios, and eNodeB radio. We estimate constant hardware error as a mean difference between estimated distances and actual distances. Before plotting, we

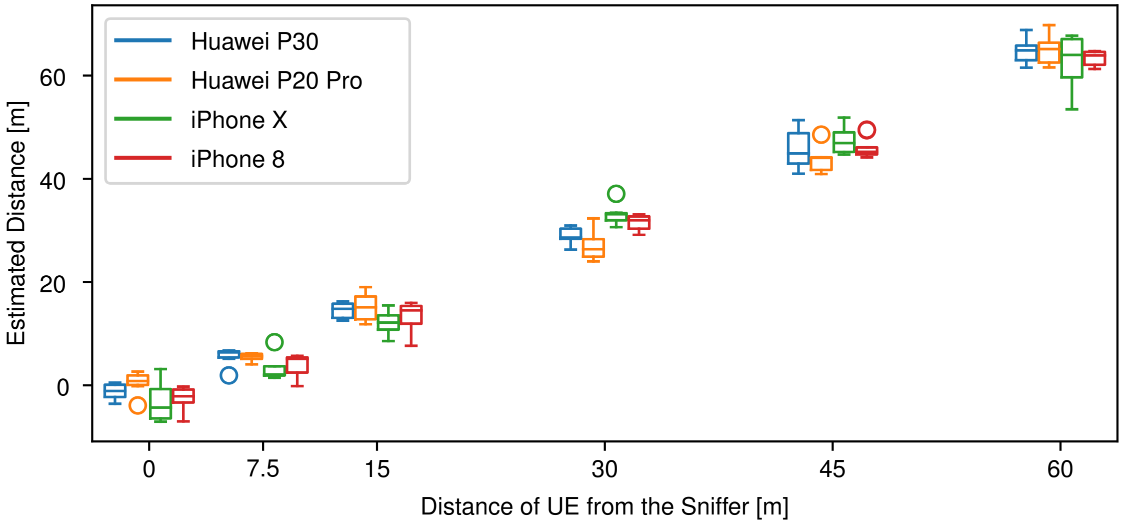


Figure 10: Distance measurements for four different phone at six distances.

remove hardware error from these distance estimations. Finally, we visualize data points with boxplots for each UE in Figure 10.

In Table 2, we quantify the constant hardware error for all the test phones as well. We estimate the hardware error with a single distance measurement at $0m$. We observe that the hardware error is the same for all UEs with the same LTE baseband modem. Moreover, all Intel modems have the same error.

To quantify distance estimation error, we compute errors between estimated variables (with corrected constant hardware error) and actual distances. The distance estimation error can be directly translated into localization error under the ideal dilution of precision. We observe that for all mobile phones the 90th percentile error is $\sim 6m$. Concretely, the 90th percentile of the errors are: 5.659m for Huawei P20 Pro, 5.214m for Huawei P30, 7.238m for iPhone X, and 4.672m for iPhone 8. For USRP B210 the 90th percentile is 10.474, however, the performance of B210's clock is limited without a GPS lock. Obviously, for lower percentile, the values get significantly better. Median error is $\sim 2m$ for phones and $\sim 7m$ for B210.

One of the problems we observed was an error arising from the UE not receiving the TA Command. If the UE does not receive the TA Command, the eNodeB resends it. However, LTEPROBE receives it twice and applies the command again, resulting in a mismatch. Since the N310 is not a professionally graded eNodeB device, its Tx power is lower. We can expect better performance in the real world. A possible fix in the future would be to monitor ACKs sent by the UE. LTEPROBE would then only apply TA Commands that the UE acknowledged. We removed connection outliers that were more than ten times the interquartile range away from the median point. Out of 186 connections, we removed 4 data points.

7.3 IMSI Extractor

Since IMSI Extractor is a protocol-level attack, we evaluate it using an industry-grade base station software by Amarisoft on the AMARI Callbox Mini hardware [6]. The base station, the AdaptOver USRP and the Octoclock used GPS clock.

We ran the attack against 17 modern phones for both Attach Request and Service Request messages. For all 17 phones, we obtained the IMSI number as a response to the Attach Request. As a response to the Service Request, we were successful for all but one mobile phone, iPhone 7. After transmitting the Identity Response, the UEs successfully connected to the network. To the user, the attack was not noticeable. The comprehensive list of phones used in the evaluation and an example packet capture file from our attack can be found in Appendix in Table 1 and Figure 11.

Finally, we confirmed our attack and the capabilities of LTEPROBE against a live network of a national operator. The setup consisted of a real-world Ericsson eNodeB, connected to the operator's production core network, with its antennas and our attacker devices installed inside a 5×6m Faraday cage. Therefore, we could run tests against the same configuration as found in outside cells, without influencing real users.

8 Countermeasures

As shown in [31], it is impossible to mitigate location leakage attacks presented in this work unless messages and their transmission/reception times are fully randomized. Due to the highly synchronized operation of LTE, these requirements are not feasible to be implemented.

Instead, we propose a solution that only requires changes on UEs and is compatible with the current LTE protocol. In our countermeasure, UE sends the initial Random Access message with a random offset. Since UE knows the offset, it modifies the received Timing Advance Command by adding the applied random offset. The recorded Timing Advance value by LTEPROBE is therefore not relevant and using it in the localization attack results in wrong location estimates. Our proposal does not mitigate the localization attack, but increases its complexity and cost. The attacker can employ more sniffers and infer the random offset UE applies.

We propose three types of countermeasures against our IMSI Extractor: (i) UE-based countermeasures are deployed on the UEs and work by observing Identity Requests for the IMSI number. UEs notify users about incoming Identity Requests or report to the network an unusual number of Identity Requests. Reporting to the operator requires trust in the UEs that they report the numbers honestly. (ii) Network-based countermeasures use a large number of eavesdroppers in the covered area. They compare the eavesdropped Identity Requests with the ones sent by the base stations. Since the operators deploy the eavesdroppers, they have access to all the transmitted Identity Requests. Neither UE-based nor network-based countermeasures prevent IMSI Extractor but merely detect it. (iii) Finally, Protocol-based countermeasures are the most robust and work even against IMSI Extractor based on other procedures; however, they require the most extensive changes to LTE, likely unfeasible to retrofit for existing devices. In 5G, IMSI catching is no longer possible since IMSI is encrypted using the network's public key. Thus, the attacker cannot decode the IMSI.

9 Related Work

The first paper to implement a downlink control channel sniffer was by Kumar et al. [23]. The follow up work by Bui et al. [8] implements a downlink control channel sniffer with the open-source library srsLTE [16]. We improve on these two papers with a downlink sniffer that decodes data channels and reconstructs higher layer datagrams. This allows us to receive TA Commands on the MAC layer or get a UE dedicated configuration for the uplink channel on the RRC layer. However, neither of these works implements an uplink sniffer functionality, which is paramount for LTRACK.

Three commercial sniffers are available. Airscope [39] is a downlink-only sniffer, whereas Wavejudge [35] and thinkRF [42] cover both uplink and downlink sniffer functionality. These products are high price and closed source, so we could not compare our sniffer to these products nor use them to mount our attacks.

In terms of user tracking and localization, Shaik et al. [36] show how an adversary may sniff on paging messages at different eNodeBs. An operator will first broadcast a paging message for a particular user from the last used eNodeB. From this, the attacker learns a coarse location of the UE.

In LTEye [23], the authors extend a synthetic aperture radar to capture the shortest and the most direct path of the radio signal from the User Equipment. The users' location is estimated at the intersection of the direct paths, estimated by multiple radars at different locations.

The closest work to ours (in the context of UE localization) is the work of [32]. [32] also proposes the use of both Timing Advance Command from eNodeB and times of arrival of uplink messages to approximate the geolocation of the UE. However, [32] does not provide details regarding the measurement of the time of arrival from uplink messages, does not implement the attack, nor does it bind the obtained location with the UE identity as we do in this paper. In particular, in our work, we also increase the localization accuracy by fingerprinting the phone model and correcting its hardware error. [32] opted for approximating transmission time of UE from Timing Advance Command, which introduces a significant error. We transform the geometry of the problem into an ellipse with two focal points, which cancels the large systematic error introduced by the Timing Advance Command. [32] further highlights how their work is successful in a setting where the UE is in the vicinity of multiple eNodeBs. Having a single eNodeB close to the victim with a deployed sniffing device is sufficient in our attack.

[28] show real-world localization attack based on Timing Advance Commands against WiMax networks. They used a commercial device, WaveJudge 4900A [35] to perform the attack. They improve the distance estimation of mobile phones from the base station by using time of arrival measurement. However, compared to that work, our time of arrival estimation offers subsample precision. [28] further didn't evaluate modern smartphones and their corresponding hardware errors.

So far, the primary tool for UE identification were IMSI Catchers, which rely on fake base stations [19, 36] and are therefore detectable. Other approaches included triggering reconnections by forcing the victim's UE to act, e.g., by sending a WhatsApp or Facebook message [20, 36]. When UE reconnects to the network, the attacker can infer the model and make of the device and compare it to the victim's UE [37]. However, such attacks are primarily targeted against specific UEs and are not sufficient for large-scale tracking.

10 Conclusion

In this work, we proposed and showed the feasibility of large-scale tracking of users in an LTE network. Furthermore, we built LTEPROBE, a robust uplink and downlink sniffer based on components of srsLTE. The implementation of LTEPROBE is white-box and does not depend on any costly or proprietary modules other than off-the-shelf software-defined radios. Using our sniffer, we were able to devise a tracking attack that we call LTRACK. LTRACK improves on the state-of-the-art by combining Timing Advance Command sniffing and measuring the times of arrival of both LTE downlink and uplink messages. LTRACK also contains a purpose-built IMSI Catcher that does not rely on a fake base station but rather overshadows packages with surgical precision and very little energy. This work is the first to explore UE tracking in a practical setting and with affordable hardware.

Acknowledgment

This project has received funding from the Swiss National Science Foundation under NCCR Automation, grant agreement 51NF40_180545.

References

[1] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Radio Access Network;LTE Radio Access Network (RAN) enhancements for diverse data applications. Technical Specification (TS) 36.822, 3rd Generation Partnership Project (3GPP), 2020.

[2] 3GPP. Evolved universal terrestrial radio access (e-utra); LTE positioning protocol (LPP). Technical specification (TS) 36.355, 3rd Generation Partnership Project (3GPP), July 2020.

[3] 3GPP. Evolved universal terrestrial radio access (e-utra); physical channels and modulation. Technical specification (TS) 36.211, 3rd Generation Partnership Project (3GPP), October 2020.

[4] 3GPP. Evolved universal terrestrial radio access (e-utra); physical layer; measurements. Technical specification (TS) 36.214, 3rd Generation Partnership Project (3GPP), July 2020.

[5] 3GPP. Universal Mobile Telecommunications System (UMTS); LTE; 5G; Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS);. Technical Specification (TS) 24.301, 3rd Generation Partnership Project (3GPP), 2020.

[6] Amarisoft. AMARI Callbox Series.

[7] Ravishankar Borgaonkar, Andrew Martin, Shinjo Park, Altaf Shaik, Jean-Pierre Seifert, and TU Berlin. White-Stingray: Evaluating IMSI Catchers Detection Applications. In *In11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17) 2017*, page 12, 2017.

[8] Nicola Bui and Joerg Widmer. OWL: a reliable online watcher for LTE control channel measurements. In *Proceedings of the 5th Workshop on All Things Cellular Operations, Applications and Challenges - ATC '16*, pages 25–30, New York City, New York, 2016. ACM Press.

[9] Adrian Dabrowski, Nicola Pianta, Thomas Klepp, Martin Mulazzani, and Edgar Weippl. IMSI-catch me if you can: IMSI-catcher-catchers. In *Proceedings of the 30th Annual Computer Security Applications Conference*, ACSAC '14, pages 246–255, New York, NY, USA, December 2014. Association for Computing Machinery.

[10] Yves-Alexandre de Montjoye, César A. Hidalgo, Michel Verleysen, and Vincent D. Blondel. Unique in the Crowd: The privacy bounds of human mobility. *Scientific Reports*, 3(1):1376, March 2013. Number: 1 Publisher: Nature Publishing Group.

[11] Yoni De Mulder, George Danezis, Lejla Batina, and Bart Preneel. Identification via location-profiling in GSM networks. In *Proceedings of the 7th ACM workshop on Privacy in the electronic society*, WPES '08, pages 23–32, New York, NY, USA, October 2008. Association for Computing Machinery.

[12] Mitziu Echeverria, Zeeshan Ahmed, Bincheng Wang, M. Fareed Arif, Syed Rafiul Hussain, and Omar Chowdhury. PHOENIX: Device-Centric Cellular Network Protocol Monitoring using Runtime Verification. *arXiv:2101.00328 [cs]*, January 2021. arXiv: 2101.00328.

[13] Simon Erni, Marc Röschlin, Patrick Leu, Martin Kotuliak, and Srdjan Capkun. AdaptOver: Adaptive Overshadowing of LTE signals. *arXiv*, August 2021.

[14] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. De-anonymization attack on geolocated data. *Journal of Computer and System Sciences*, 80(8):1597–1614, December 2014.

[15] Philippe Golle and Kurt Partridge. On the Anonymity of Home/Work Location Pairs. In Hideyuki Tokuda, Michael Beigl, Adrian Friday, A. J. Bernheim Brush, and Yoshito Tobe, editors, *Pervasive Computing*, volume 5538, pages 390–397. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. Series Title: Lecture Notes in Computer Science.

[16] Ismael Gomez-Miguelez, Andres Garcia-Saavedra, Paul D. Sutton, Pablo Serrano, Cristina Cano, and Doug J. Leith. srsLTE: an open-source platform for LTE evolution and experimentation. In *Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization - WiNTECH '16*, pages 25–32, New York City, New York, 2016. ACM Press.

[17] Benjamin Henne, Christian Szongott, and Matthew Smith. SnapMe if you can: privacy threats of other peoples' geo-tagged media and what we can do about it. In *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks*, WiSec '13, pages 95–106, New York, NY, USA, April 2013. Association for Computing Machinery.

[18] Baik Hoh, M. Gruteser, Hui Xiong, and A. Alrabady. Enhancing Security and Privacy in Traffic-Monitoring Systems. *IEEE Pervasive Computing*, 5(4):38–46, October 2006. Conference Name: IEEE Pervasive Computing.

[19] Roger Piqueras Jover. LTE security, protocol exploits and location tracking experimentation with low-cost software radio. *arXiv:1607.05171 [cs]*, July 2016. arXiv: 1607.05171.

[20] Katharina Kohls, David Rupprecht, Thorsten Holz, and Christina Pöpper. Lost traffic encryption: fingerprinting LTE/4G traffic on layer two. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, pages 249–260, Miami Florida, May 2019. ACM.

[21] John Krumm. Inference Attacks on Location Tracks. In Anthony LaMarca, Marc Langheinrich, and Khai N. Truong, editors, *Pervasive Computing*, volume 4480, pages 127–143.

Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. Series Title: Lecture Notes in Computer Science.

[22] John Krumm. A survey of computational location privacy. *Personal and Ubiquitous Computing*, 13(6):391–399, August 2009.

[23] Swarun Kumar, Ezzeldin Hamed, Dina Katabi, and Li Erran Li. LTE radio analytics made easy and accessible. In *Proceedings of the 2014 ACM conference on SIGCOMM - SIGCOMM '14*, pages 211–222, Chicago, Illinois, USA, 2014. ACM Press.

[24] Zhenhua Li, Weiwei Wang, Christo Wilson, Jian Chen, Chen Qian, Taeho Jung, Lan Zhang, Kebin Liu, Xiangyang Li, and Yunhao Liu. FBS-Radar: Uncovering Fake Base Stations at Scale in the Wild. In *Proceedings 2017 Network and Distributed System Security Symposium*, San Diego, CA, 2017. Internet Society.

[25] Lin Liao, Donald J. Patterson, Dieter Fox, and Henry Kautz. Learning and inferring transportation routines. *Artificial Intelligence*, 171(5-6):311–331, April 2007.

[26] Nakarmi Prajwol Kumar, Noamen Ben Henda, and Vlasios Tsiatsis. 3GPP Release 15 and the battle against false base stations, January 2019. Last Modified: 2020-04-27T11:21:46+00:00.

[27] Alexandra-Mihaela Olteanu, Kévin Huguenin, Reza Shokri, and Jean-Pierre Hubaux. Quantifying the Effect of Co-location Information on Location Privacy. In Emiliano De Cristofaro and Steven J. Murdoch, editors, *Privacy Enhancing Technologies*, Lecture Notes in Computer Science, pages 184–203, Cham, 2014. Springer International Publishing.

[28] Benjamin A Pimentel. *Passive Geolocation in a 4G WIMAX Single Base Station Scenario*. PhD thesis, Naval Postgraduate School, Monterey, California, 2013.

[29] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. Knock Knock, Who's There? Membership Inference on Aggregate Location Data. In *Proceedings 2018 Network and Distributed System Security Symposium*, San Diego, CA, 2018. Internet Society.

[30] Cooper Quintin. *Detecting Fake 4G Base Stations in Real Time*. DEF CON, 2020.

[31] Kasper Bonne Rasmussen and Srdjan Čapkun. Location privacy of distance bounding protocols. In *Proceedings of the 15th ACM conference on Computer and communications security*, CCS '08, pages 149–160, New York, NY, USA, October 2008. Association for Computing Machinery.

[32] John D. Roth, Murali Tummala, John C. McEachen, and James W. Scrofani. On Location Privacy in LTE Networks. *IEEE Transactions on Information Forensics and Security*, 12(6):1358–1368, June 2017. Conference Name: IEEE Transactions on Information Forensics and Security.

[33] David Rupprecht, Katharina Kohls, Thorsten Holz, and Christina Pöpper. Breaking LTE on Layer Two. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1121–1136, May 2019. ISSN: 2375-1207.

[34] David Rupprecht, Katharina Kohls, Christina Pöpper, and Thorsten Holz. Eavesdropping Encrypted LTE Calls With REVOLTE. In *Proceedings of the 29th USENIX Conference on Security Symposium*, page 17, 2020.

[35] Sanjole. WaveJudge 5000 Wireless test system for LTE and WiMAX.

[36] Altaf Shaik, Ravishankar Borgaonkar, N. Asokan, Valtteri Niemi, and Jean-Pierre Seifert. Practical Attacks Against Privacy and Availability in 4G/LTE Mobile Communication Systems. *arXiv:1510.07563 [cs]*, August 2017. arXiv: 1510.07563.

[37] Altaf Shaik, Ravishankar Borgaonkar, Shinjo Park, and Jean-Pierre Seifert. New vulnerabilities in 4G and 5G cellular access network protocols: exposing device capabilities. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, pages 221–231, Miami Florida, May 2019. ACM.

[38] Reza Shokri, George Theodorakopoulos, Jean-Yves Le Boudec, and Jean-Pierre Hubaux. Quantifying Location Privacy. In *2011 IEEE Symposium on Security and Privacy*, pages 247–262, Oakland, CA, USA, May 2011. IEEE.

[39] Software Radio Systems. Products | SRS.

[40] Mudhakar Srivatsa and Mike Hicks. Deanonymizing mobility traces: using social network as a side-channel. In *Proceedings of the 2012 ACM conference on Computer and communications security*, CCS '12, pages 628–637, New York, NY, USA, October 2012. Association for Computing Machinery.

[41] Sven Fischer. Observed Time Difference Of Arrival (OTDOA) Positioning in 3GPP LTE.

[42] ThinkRF. The Leader in Software Defined Spectrum Analysis.

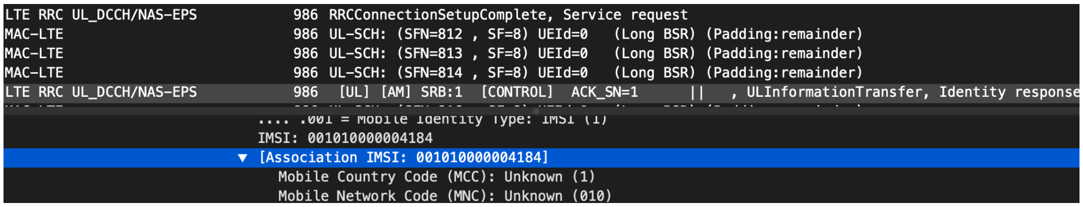
[43] Huandong Wang, Chen Gao, Yong Li, Gang Wang, Depeng Jin, and Jingbo Sun. De-anonymization of Mobility Trajectories: Dissecting the Gaps between Theory and Practice. In *Proceedings 2018 Network and Distributed System Security Symposium*, San Diego, CA, 2018. Internet Society.

[44] Marius Wernke, Pavel Skvortsov, Frank Dürr, and Kurt Rothermel. A classification of location privacy attacks and approaches. *Personal and Ubiquitous Computing*, 18(1):163–175, January 2014.

[45] Hojoon Yang, Sangwook Bae, Mincheol Son, Hongil Kim, Song Min Kim, and Yongdae Kim. Hiding in Plain Signal: Physical Signal Overshadowing Attack on LTE. In *Proceedings of the 28th USENIX Conference on Security Symposium*, SEC'19, page 19, 2019.

[46] Hui Zang and Jean Bolot. Anonymization of location data does not work: a large-scale measurement study. In *Proceedings of the 17th annual international conference on Mobile computing and networking*, MobiCom '11, pages 145–156, New York, NY, USA, September 2011. Association for Computing Machinery.

A Appendix



```
LTE RRC UL_DCCH/NAS-EPS            986 RRCConnectionSetupComplete, Service request
MAC-LTE                            986 UL-SCH: (SFN=812 , SF=8) UEId=0   (Long BSR) (Padding:remainder)
MAC-LTE                            986 UL-SCH: (SFN=813 , SF=8) UEId=0   (Long BSR) (Padding:remainder)
MAC-LTE                            986 UL-SCH: (SFN=814 , SF=8) UEId=0   (Long BSR) (Padding:remainder)
LTE RRC UL_DCCH/NAS-EPS            986  [UL] [AM] SRB:1  [CONTROL]  ACK_SN=1     ||   , ULInformationTransfer, Identity response
                          .... .001 = Mobile Identity Type: IMSI (1)
                          IMSI: 001010000004184
                        ▼ [Association IMSI: 001010000004184]
                            Mobile Country Code (MCC): Unknown (1)
                            Mobile Network Code (MNC): Unknown (010)
```

Figure 11: Packet capture file from IMSI Extractor.

UE model	Identification Attach Request	Identification Service Request
Samsung Galaxy s10	yes	yes
Samsung Galaxy a8	yes	yes
Huawei P20 Pro	yes	yes
Huawei P30 Lite	yes	yes
Huawei P30	yes	yes
Xiaomi Mi9	yes	yes
Xiaomi MiX 3	yes	yes
Google Nexus 5X	yes	yes
Google Pixel 2	yes	yes
Google Pixel 3a	yes	yes
HTC U12+	yes	yes
OnePlus 7T	yes	yes
iPhone 6s	yes	yes
iPhone 7	yes	no
iPhone 8	yes	yes
iPhone X	yes	yes
iPhone 11	yes	yes
iPhone 11 Pro	yes	yes

Table 1: Mobile phones used in the IMSI Extractor experiments.

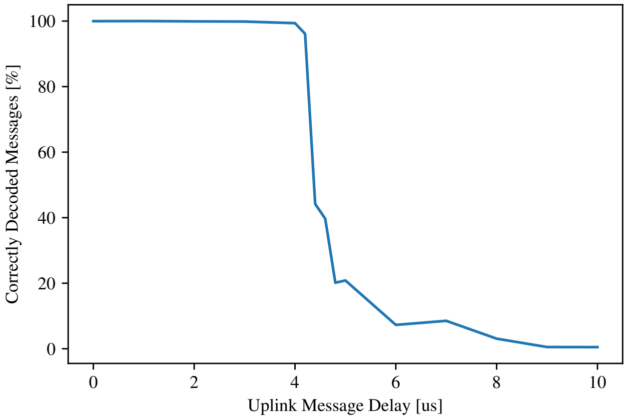


Figure 12: Percentage of correctly decoded uplink messages by our sniffer as a function of the time delay from the start of a frame.

UE model	Modem	Hardware Error [m]	std [m]
Samsung Galaxy s10	Exynos 9820	11.29	7.22
Samsung Galaxy a8	Exynos 7885	-26.62	4.77
Samsung Galaxy s5	Qcom. Gobi 4G	-	-
Huawei P20 Lite	Kirin 659	-24.47	2.13
Huawei P20 Pro	Kirin 970	-9.34	2.90
Huawei P30 Lite	Kirin 710	-10.27	0.98
Huawei P30	Kirin 980	-24.51	1.49
Xiaomi Mi9	Qcom. X24 LTE	10.44	2.20
Xiaomi MiX 3	Qcom. X24 LTE	11.57	1.60
Nokia 1.3	Qcom. X5 LTE	-	-
Sony Xperia X	Qcom. X8 LTE	-11.20	4.78
Google Nexus 5X	Qcom. X10 LTE	5.08	2.51
Google Pixel 2	Qcom. X16 LTE	-13.52	2.32
Google Pixel 3a	Qcom. X12 LTE	4.46	2.14
Google Pixel 4	Qcom. X24 LTE	12.88	1.67
HTC U12+	Qcom. X20 LTE	-13.66	1.55
OnePlus 7T	Qcom. X24 LTE	12.66	1.42
iPhone 7	Intel XMM7360	-23.86	0.88
iPhone 8	Intel XMM 7480	-23.65	2.28
iPhone X	Intel XMM7480	-25.64	3.75
iPhone 11	Intel XMM 7660	-23.19	2.49
iPhone 11 Pro	Intel XMM 7660	-25.35	2.46

Table 2: Mobile phones used in the localization and fingerprinting experiments.