



Advanced Network Security

Lecture 8: Attacks and Mobile Networks

Harald Vranken, Katharina Kohls

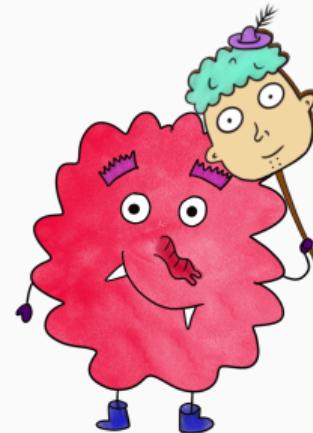
April 19th, 2021

Open University Nijmegen

Radboud University Nijmegen

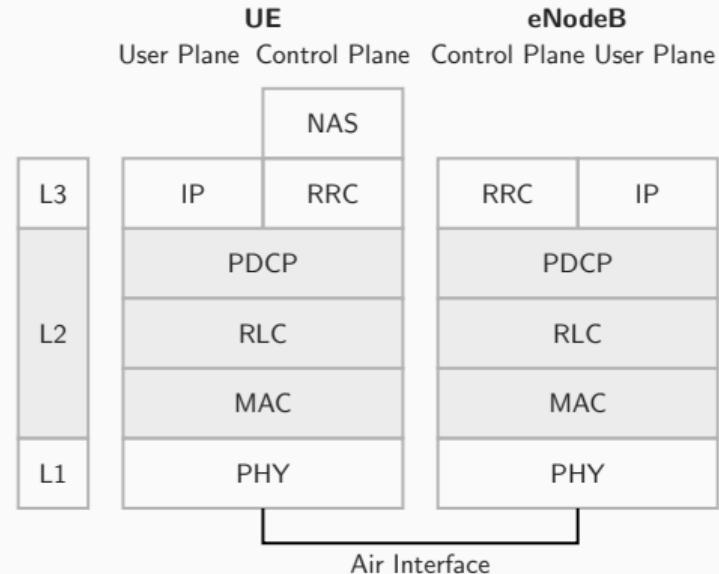
Last time...

- ▶ Why mobile network security is important
- ▶ Basics of mobile networks
 - Generic network setup
 - Long Term Evolution LTE
- ▶ Basic security goals
 - Confidentiality
 - Integrity
 - Availability
 - Privacy
- ▶ Mobile evolution



What was L2 again?

- ▶ **PDCP**: Transport of data with ciphering and integrity protection (RRC) and transport of IP packets.
- ▶ **RLC**: Transport PDCP data in different modes.
- ▶ **MAC**: Logical channels for RLC for multiplexing into the physical transmission. Scheduling of within and between UEs.



Breaking LTE on Layer Two

Breaking LTE on Layer Two

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

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

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

Christina Paeppe
New York University Abu Dhabi
christina.paeppe@nyu.edu

Abstract—Long Term Evolution (LTE) is the latest mobile communication standard and has a pivotal role in our information society. LTE complies with the goal of providing security and service quality as well as critical infrastructure and public safety communications. Both scenarios are demanding towards a resilient and secure specification and implementation. However, the design and analysis of LTE potentially lead to severe risks. Previous work on LTE protocol security identified crucial attack vectors for both the physical layer one and network layer three: Data Link layer (Layer Two). In this paper, however, remain a blind spot in existing LTE research.

In this paper, we present a comprehensive layer two security analysis. We identify three attack vectors. These attacks impact the confidentiality and integrity of LTE communication. More specifically, we first present a passive identity mapping attack that matches volatile radio identities to longer lasting network identities. This allows an adversary to track a user within a cell as a stepping stone for follow-up attacks. Second, we demonstrate how a passive attacker can abuse the resource allocation as a side channel to perform website fingerprinting that enables the identification of visited websites. Third, we present and present the ALTER attack that exploits the fact that LTE user data is encrypted in counter mode (AES-CTR) but integrity protection, which allows us to modify the message payload. As a proof-of-concept demonstration, we show how an active attacker can redirect DNS requests and then perform a DNS spoofing attack. As a result, the user is redirected to a malicious website. Our experimental evaluation demonstrates the real-world applicability of all three attacks and emphasizes the threat of open attack vectors on LTE layer two protocols.

I. INTRODUCTION

The latest mobile communication standard LTE represents the dominant mobile communication technology billion users in the world and has a pivotal role in our information society. LTE is designed to combine performance goals such as high transmission rates and low latency with a series of security features like formally proven mutual authentication, well-established encryption algorithms such as AES, and separated security domains. Besides causal use cases, LTE also has an emerging relevance for critical infrastructures and public safety communications [1]. Both scenarios are demanding towards a resilient and secure specification and implementation of LTE, as outages and open attack vectors potentially lead to severe risks. While the LTE specification considers a diverse set of security features, it can hardly predict all potential attacks, and it is even harder to cover sets of restrictions in real-world implementations.

Consequently, recent academic and non-academic work identified various potential vulnerabilities on different layers

of the LTE protocol stack. On the network layer (layer three), passive or active attackers can either localize a user or deny the service and thus downgrade the phone to the insecure GSM network [2]–[4]. On the physical layer (layer one), LTE can be attacked via signal jamming and signal alteration [5]–[8]. As a matter of fact, the previous research efforts focused only on layer one or layer three protocols and—to the best of our knowledge—no security analysis of data link layer (layer two) protocols exists to date. This leads to a situation of uncertainty about potential security and privacy threats that arise from the specification or implementation flaws of the data link layer and its three protocols: Medium Access Control (MAC), Radio Link Control (RLC), and Packet Data Convergence Protocol (PDCP).

In this paper, we perform a security analysis of LTE on layer two and analyze these protocols for potential vulnerabilities. As a result, we introduce two passive attacks and one active attack that impair the confidentiality and privacy of LTE communication. Table I shows an overview of the attacks and their properties. We first focus on a passive attack that allows an adversary to map the user's temporary network identity (TMSI) to the temporary radio identity (RNTI). More specifically, we demonstrate how an attacker can precisely localize and identify a user within the cell, distinguish multiple transmission streams, and use this information as a stepping stone for subsequent attacks. We then introduce a passive attack that performs website fingerprinting attack. Website fingerprinting is known from other contexts like Tor [9], where traffic analysis reveals the browsing behavior of users despite Tor's onion encryption. In the context of LTE, we demonstrate a comparable information leak in the resource allocation: even though transmissions are encrypted, we can access plaintext information up to the PDCP and learn the transmission characteristics for individual users. That information is sufficient to distinguish access to websites that do not have their own encryption that is applied to the user due to encryption. Both attacks already harm user privacy, but they can be combined to an even stronger version of website fingerprinting, while solely depending on passive (downlink) sniffing.

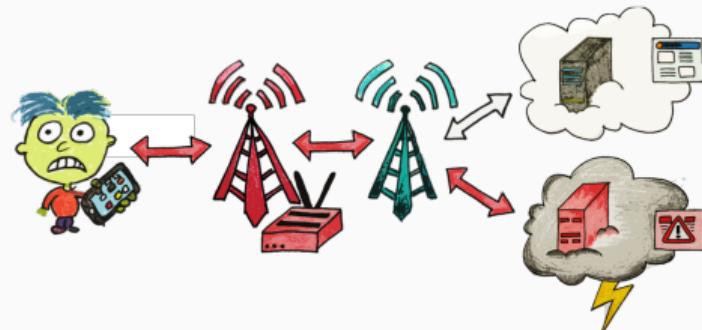
We further introduce an active attack called ALTER that exploits the missing integrity protection of LTE user data to perform a chosen-ciphertext attack. Our attack is based on the

Today: Attacks and what you need to understand them

<https://alter-attack.net>

Three attacks against LTE L2:

- (1) Website Fingerprinting
- (2) Identity mapping
- (3) User Data Redirection



Attack 1: Website Fingerprinting

Attack 2: Identity Mapping

Attack 3: User Data Redirection

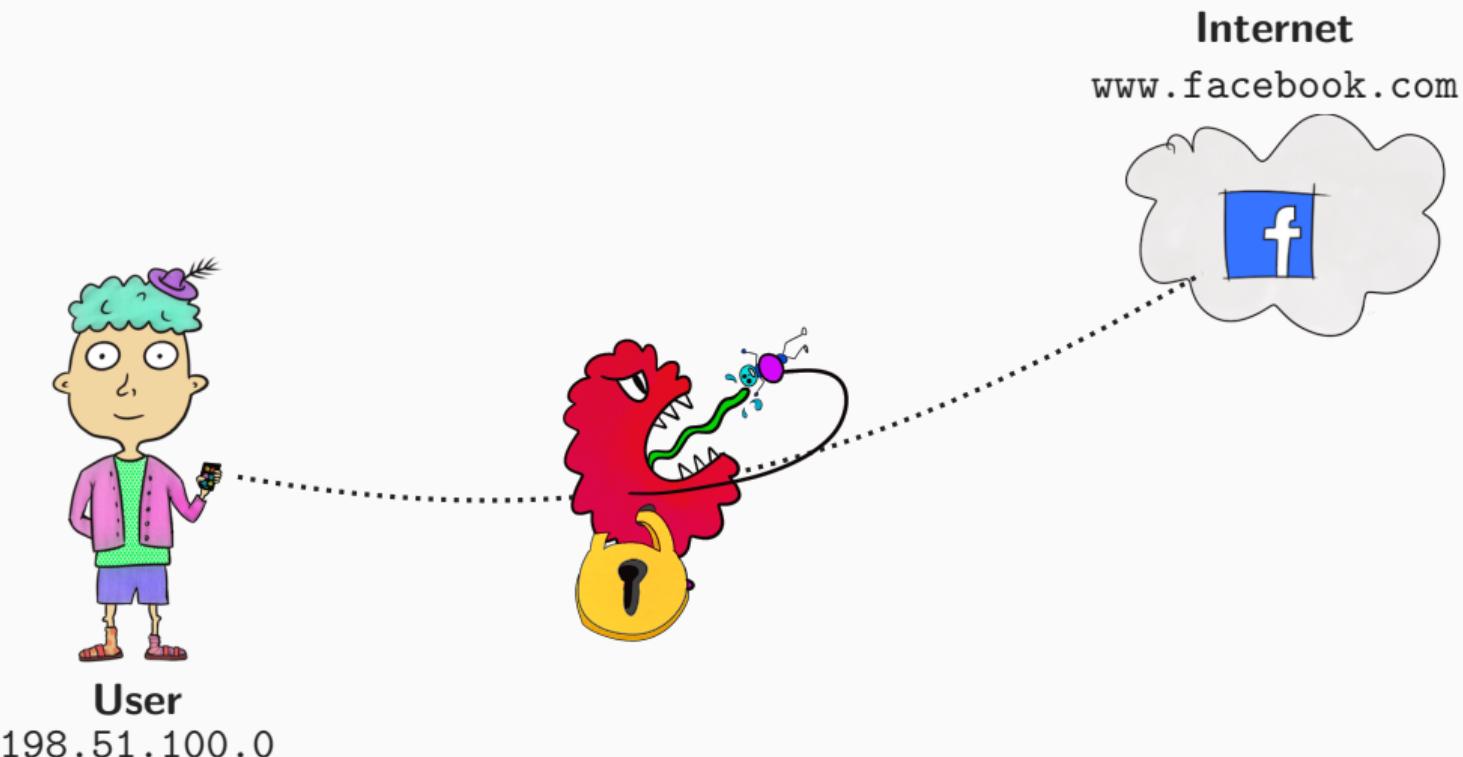
Summary

Attack 1: Website Fingerprinting

Required Background

- ▶ General concept of website fingerprinting (WF)
- ▶ Internet connection through LTE
- ▶ LTE metadata
- ▶ Basic attack setup and trace inspection

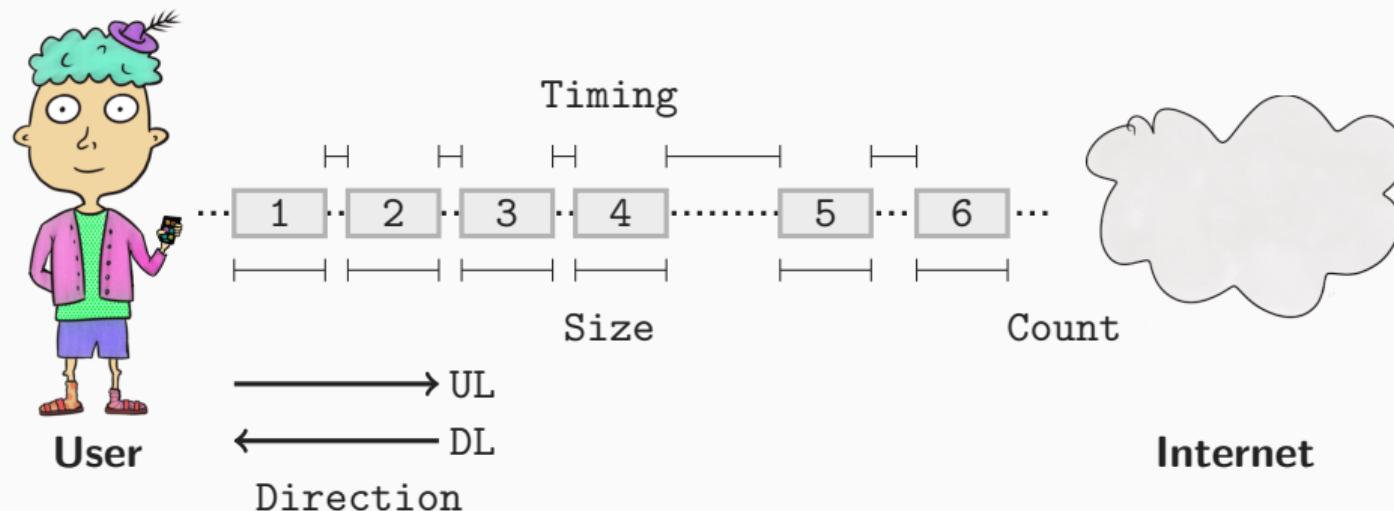
Website Fingerprinting: The Concept



Standard Internet Connection:

- ▶ User connects to a website
- ▶ IP address of user is sensitive
- ▶ Together with website they reveal Internet usage
- ▶ Attacker can monitor and learn sensitive data
- ▶ **Protection:** Encrypt transmissions

Metadata of Encrypted Traffic



Encryption protects the content. Transmissions still reveal metadata:

- ▶ Measure the timing between packets
- ▶ Measure the sizes of packets
- ▶ Count packets
- ▶ Check the transmission direction

How do we get this metadata?

- ▶ Can either be measured (timing, packet counts)
- ▶ Is part of the header information (size)
- ▶ Or is visible in the connection (direction)

**The amount of metadata depends on the
protocols, physical link, and optional security measures**

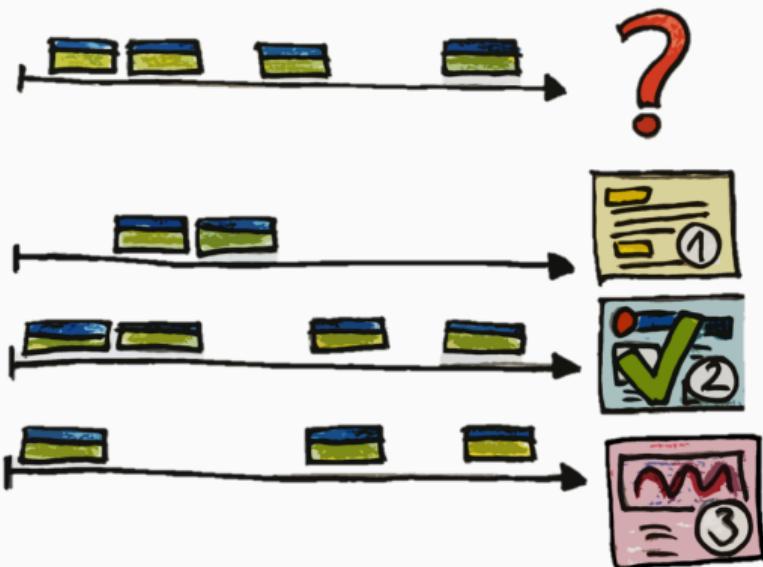
Classification Attack

Preparation

- ▶ Attacker pre-records requests and responses to n websites
- ▶ Repeats each website several times
- ▶ Results in a labeled data set

Classification Attack

- ▶ Record traffic of victim
- ▶ Compare trace with pre-recorded data
- ▶ Highest similarity → guess



Trace Data Set

abc.net.au	bootstrapcdn.com	dawn.net	flightware.com	ietf.org	men	1-492-client.csv	1-987-exit.csv	2-422-client.csv	3-110-client.csv	3-53-client.csv	3-977-client.csv
aboutads.info	boston.com	debian.org	flipkart.com	ifeng.com	mes	1-492-exit.csv	1-988-client.csv	2-422-exit.csv	3-110-exit.csv	3-53-exit.csv	3-977-exit.csv
abs-cbn.com	brainly.com	dell.com	forbes.com	ign.com	met	1-493-client.csv	1-988-exit.csv	2-423-client.csv	3-111-client.csv	3-540-client.csv	3-978-client.csv
academia.edu	breitbart.com	denverpost.com	fortune.com	ikea.com	met	1-493-exit.csv	1-989-client.csv	2-423-exit.csv	3-111-exit.csv	3-540-exit.csv	3-978-exit.csv
addthis.com	britannica.com	detik.com	foxnews.com	lovepdf.com	mgf	1-494-client.csv	1-989-exit.csv	2-424-client.csv	3-112-client.csv	3-541-client.csv	3-979-client.csv
addtoany.com	businessinsider.com	deviantart.com	free.fr	imdb.com	mia	1-494-exit.csv	1-980-client.csv	2-424-exit.csv	3-112-exit.csv	3-541-exit.csv	3-97-client.csv
adp.com	bustle.com	dictionary.com	freepik.com	ingur.com	mic	1-495-client.csv	1-980-exit.csv	2-425-client.csv	3-113-client.csv	3-542-client.csv	3-97-exit.csv
adsrvr.org	buzzfeed.com	digg.com	ft.com	imore.com	mir	1-495-exit.csv	1-910-client.csv	2-425-exit.csv	3-113-exit.csv	3-542-exit.csv	3-980-client.csv
akismet.com	buzzfeednews.com	digicert.com	gamespot.com	indeed.com	mit	1-496-client.csv	1-910-exit.csv	2-426-client.csv	3-114-client.csv	3-543-client.csv	3-980-exit.csv
alibaba.com	cafemon.com	discordapp.com	genius.com	independent.co.uk	mlb	1-496-exit.csv	1-911-client.csv	2-426-exit.csv	3-114-exit.csv	3-543-exit.csv	3-981-client.csv
aliexpress.com	ca.gov	dol.org	getpocket.com	instagram.com	moa	1-497-client.csv	1-911-exit.csv	2-427-client.csv	3-115-client.csv	3-544-client.csv	3-981-exit.csv
allegro.pl	cam.ac.uk	domainmarket.com	gfycat.com	instructables.com	mon	1-497-exit.csv	1-912-client.csv	2-427-exit.csv	3-115-exit.csv	3-544-exit.csv	3-982-client.csv
altervista.org	cambridge.org	dotwtf.com	giphy.com	instucture.com	moz	1-498-client.csv	1-912-client.csv	2-428-client.csv	3-116-client.csv	3-545-client.csv	3-982-exit.csv
amazonaws.com	canva.com	dochub.com	github.com	intel.com	msn	1-498-exit.csv	1-913-client.csv	2-428-exit.csv	3-116-exit.csv	3-545-exit.csv	3-983-client.csv
amazon.com	carfax.com	doubleclick.net	github.io	intuit.com	mys	1-499-client.csv	1-913-exit.csv	2-429-client.csv	3-117-client.csv	3-546-client.csv	3-983-exit.csv
ameblo.jp	cars.com	doubleverify.com	gizmodo.com	iso.org	mys	1-499-exit.csv	1-914-client.csv	2-429-exit.csv	3-117-exit.csv	3-547-client.csv	3-984-client.csv
americanexpress.com	casalemedia.com	douyu.com	glassdoor.com	issuu.com	mys	1-499-client.csv	1-915-client.csv	2-42-client.csv	3-118-client.csv	3-547-exit.csv	3-984-exit.csv
ampproject.org	cbc.ca	dribbble.com	globe.com	jianshu.com	nam	1-499-exit.csv	1-915-exit.csv	2-42-exit.csv	3-118-exit.csv	3-548-client.csv	3-985-client.csv
androidcentral.com	cbslocal.com	dropbox.com	gmail.com	jimdo.com	nas	1-494-client.csv	1-916-client.csv	2-430-client.csv	3-119-client.csv	3-548-exit.csv	3-985-exit.csv
android.com	cbsnews.com	dropcatch.com	gmw.cn	jquery.com	nat	1-4-exit.csv	1-916-exit.csv	2-430-exit.csv	3-119-exit.csv	3-549-client.csv	3-986-client.csv
answers.com	cbssports.com	drudgereport.com	gnu.org	kayak.com	nat	1-500-client.csv	1-917-client.csv	2-431-client.csv	3-11-client.csv	3-549-exit.csv	3-986-exit.csv
apache.org	cdc.gov	drugs.com	go.com	khanacademy.org	nbc	1-500-exit.csv	1-917-exit.csv	2-431-exit.csv	3-11-exit.csv	3-54-client.csv	3-987-client.csv
aparat.com	change.org	duckduckgo.com	godaddy.com	kickstarter.com	nbc	1-501-client.csv	1-918-client.csv	2-432-client.csv	3-120-client.csv	3-54-exit.csv	3-987-exit.csv
apnews.com	chase.com	dw.com	gofundme.com	kompas.com	ndt	1-501-exit.csv	1-918-exit.csv	2-432-exit.csv	3-120-exit.csv	3-550-client.csv	3-988-client.csv
apple.com	chaturbate.com	eater.com	goodreads.com	ladbible.com	nes	1-502-client.csv	1-919-client.csv	2-433-client.csv	3-121-client.csv	3-550-exit.csv	3-988-exit.csv
archive.org	cheatsheet.com	ebay.com	google.com	laratini.net	net	1-502-exit.csv	1-919-exit.csv	2-433-exit.csv	3-121-exit.csv	3-551-client.csv	3-989-client.csv
arnebrachhold.de	chicagotribune.com	ebay-kleinanzeigen.de	grammarly.com	launchpad.net	new	1-503-client.csv	1-91-client.csv	2-434-client.csv	3-122-client.csv	3-551-client.csv	3-989-exit.csv
arstechnica.com	china.com.cn	economist.com	gravatar.com	legacy.com	new	1-503-exit.csv	1-91-exit.csv	2-434-exit.csv	3-122-exit.csv	3-552-client.csv	3-98-client.csv
ask.com	chourtv.ma	ed.gov	grid.id	lenovo.com	new	1-504-client.csv	1-920-client.csv	2-435-client.csv	3-123-client.csv	3-553-client.csv	3-98-exit.csv
asus.com	chron.com	sepurl.com	guardian.co.uk	letsencrypt.org	nfl	1-504-exit.csv	1-920-exit.csv	2-435-exit.csv	3-123-exit.csv	3-553-exit.csv	3-990-client.csv
att.com	citi.com	elegantthemes.com	hao123.com	linkedin.com	ngi	1-505-client.csv	1-921-client.csv	2-436-client.csv	3-124-client.csv	3-554-client.csv	3-990-exit.csv
autodesk.com	cloudflare.com	elpais.com	harvard.edu	liputan6.com	ngi	1-505-exit.csv	1-921-exit.csv	2-436-exit.csv	3-124-exit.csv	3-555-client.csv	3-991-client.csv
avast.com	cmu.edu	elsevier.com	hbr.org	littletings.com	nje	1-506-client.csv	1-922-client.csv	2-437-client.csv	3-125-client.csv	3-555-exit.csv	3-991-exit.csv
avito.ru	cnet.com	entrepreneur.com	hdfcbank.com	live.com	nic	1-506-exit.csv	1-922-exit.csv	2-437-exit.csv	3-125-exit.csv	3-556-client.csv	3-992-client.csv
bandcamp.com	cnn.com	online.com	healthgrades.com	livejasmin.com	nih	1-507-client.csv	1-923-client.csv	2-438-client.csv	3-126-client.csv	3-556-exit.csv	3-992-exit.csv
bankoffamerica.com	columbia.edu	epa.gov	healthline.com	livewjournal.com	nin	1-507-exit.csv	1-923-exit.csv	2-438-exit.csv	3-126-exit.csv	3-557-client.csv	3-993-client.csv
barnesandnoble.com	comicbook.com	epicgames.com	history101.com	livescience.com	noa	1-508-client.csv	1-924-client.csv	2-439-client.csv	3-127-client.csv	3-557-exit.csv	3-993-exit.csv
battle.net	constantcontact.com	espn.com	holas.org	loc.gov	npr	1-508-exit.csv	1-924-exit.csv	2-439-exit.csv	3-127-exit.csv	3-558-client.csv	3-994-client.csv
bbb.org	consumerreports.org	etsy.com	hollywoodreporter.com	lonelyplanet.com	nps	1-509-client.csv	1-925-client.csv	2-43-client.csv	3-128-client.csv	3-558-exit.csv	3-994-exit.csv
bbc.com	coolimba.com	ettoday.net	homedept.com	looper.com	ntp	1-509-exit.csv	1-925-exit.csv	2-43-exit.csv	3-128-exit.csv	3-559-client.csv	3-995-client.csv
bbc.co.uk	cornell.edu	europa.eu	homestalk.com	ltn.com.tw	nyp	1-50-client.csv	1-926-client.csv	2-440-client.csv	3-129-client.csv	3-559-exit.csv	3-995-exit.csv
berkeley.edu	cosmopolitan.com	eventbrite.com	hootsuite.com	magiquizx.com	nyt	1-50-exit.csv	1-926-exit.csv	2-440-exit.csv	3-129-exit.csv	3-55-client.csv	3-996-client.csv
bet365.com	coursera.org	evernote.com	hotels.com	mailchimp.com	off	1-510-client.csv	1-927-client.csv	2-441-client.csv	3-12-client.csv	3-55-exit.csv	3-996-exit.csv
betbjpa.com	cpninel.com	exalator.com	hoststar.com	mail.ru	oke	1-510-exit.csv	1-927-exit.csv	2-441-exit.csv	3-12-exit.csv	3-560-client.csv	3-997-client.csv
bidswitch.net	craigslist.org	facebook.com	houstnworks.com	manoramaonline.com	ok	1-511-client.csv	1-928-client.csv	2-442-client.csv	3-130-client.csv	3-560-exit.csv	3-997-exit.csv
bilibili.com	crashlytics.com	fandom.com	howtogeek.com	mapquest.com	onl	1-511-exit.csv	1-928-exit.csv	2-442-exit.csv	3-130-exit.csv	3-561-client.csv	3-998-exit.csv
bing.com	creativecommons.org	fantasypros.com	hp.com	mashable.com	onl	1-512-client.csv	1-929-client.csv	2-443-client.csv	3-131-client.csv	3-562-client.csv	3-999-client.csv
bitly.com	criepto.com	fastcompany.com	huanqiu.com	mathtag.com	ope	1-512-exit.csv	1-929-exit.csv	2-443-exit.csv	3-131-exit.csv	3-562-exit.csv	3-99-client.csv
blackboard.com	csdn.net	fastly.net	hubspot.com	mayoclinic.org	ope	1-513-client.csv	1-92-client.csv	2-444-client.csv	3-132-client.csv	3-563-client.csv	3-99-exit.csv
blogger.com	custhelp.com	fc2.com	huffpost.com	mediabfire.com	ora	1-513-exit.csv	1-92-exit.csv	2-444-exit.csv	3-132-exit.csv	3-563-exit.csv	3-9-client.csv
bloomberg.com	dailycaller.com	fedex.com	hugedomains.com	medicalnewstoday.com	ora	1-514-client.csv	1-930-client.csv	2-445-client.csv	3-133-client.csv	3-564-client.csv	3-9-exit.csv
bushidoout.com	dailydot.com	finance101.com	hulu.com	medicinenet.com	otv	1-514-exit.csv	1-930-exit.csv	2-445-exit.csv	3-133-exit.csv	3-564-exit.csv	3-9-exit.csv
bonappetit.com	dailykos.com	findagrave.com	ibm.com	medium.com	oup	1-515-client.csv	1-931-client.csv	2-446-client.csv	3-134-client.csv	3-565-client.csv	

State of the Art

Attack Techniques

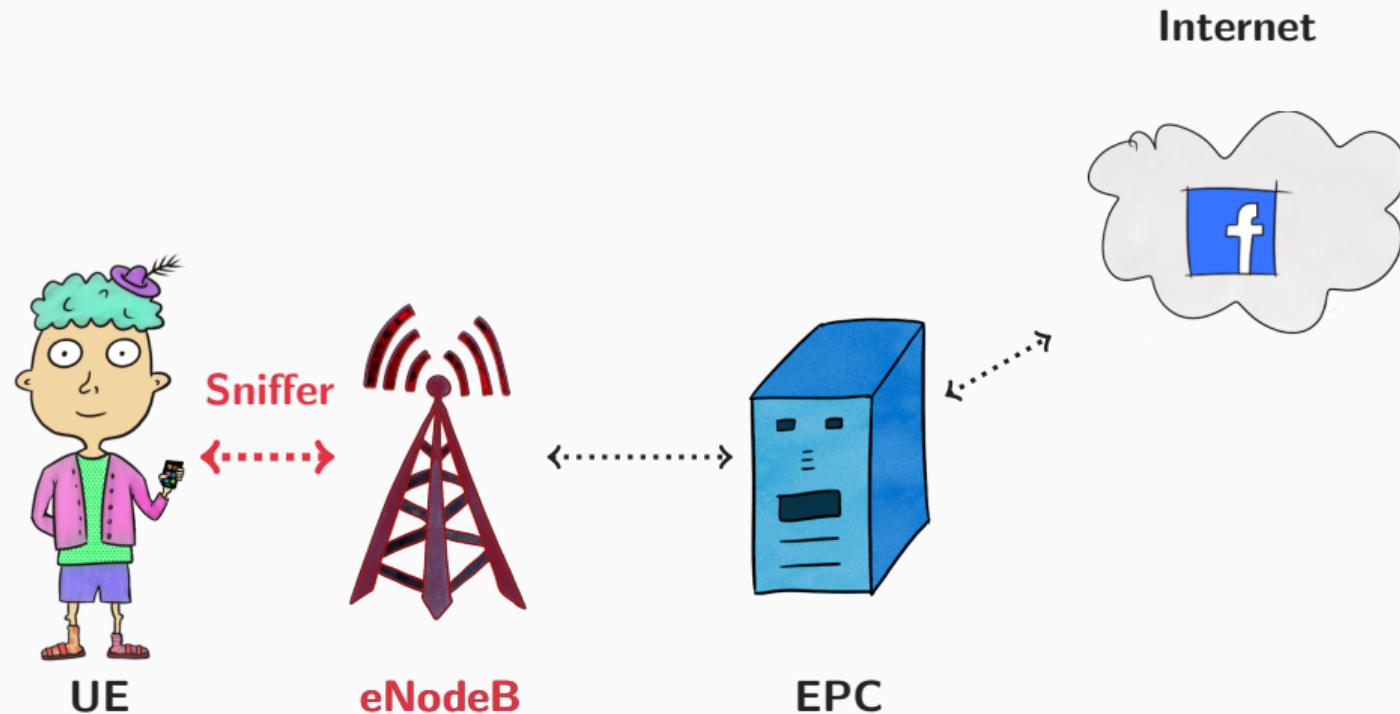
- ▶ Machine learning:
Make sense of metadata
 - ▶ Deep learning:
Automatic feature generation

Evaluation

- ▶ Pre-recorded data sets are always too small
 - ▶ Scientific evaluation is unrealistic

Not relevant for the exam!

Website Fingerprinting on LTE



Mobile Data Connection

- ▶ Radio connection between UE and eNodeB
- ▶ eNodeB connects to core network
- ▶ Forwards website request

How do we get the attack traffic?

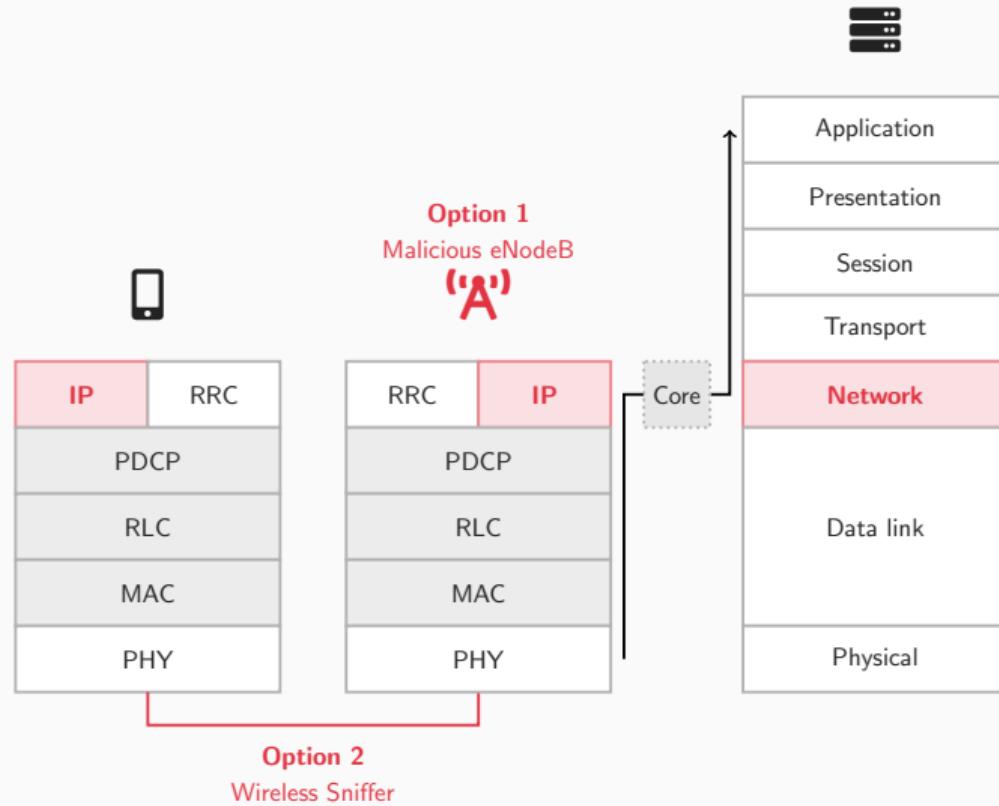
- ▶ **Option 1:** Malicious eNodeB records all traffic
What happens when the eNodeB is malicious?
- ▶ **Option 2:** Wireless sniffer monitors radio connection
What's the difference to wire tapping?

Option 1: eNodeB

- ▶ Access to L1-L3
- ▶ LTE encryption

Option 2: Sniffer

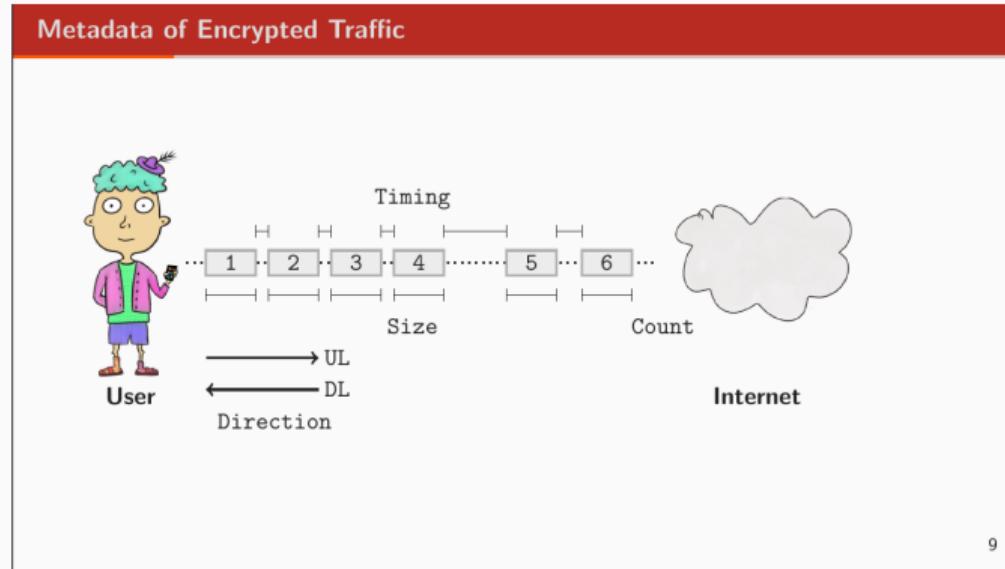
- ▶ Access to air interface
- ▶ Only transmissions



Recall: Metadata

- ▶ timing
- ▶ count
- ▶ direction
- ▶ ...

Metadata in LTE?

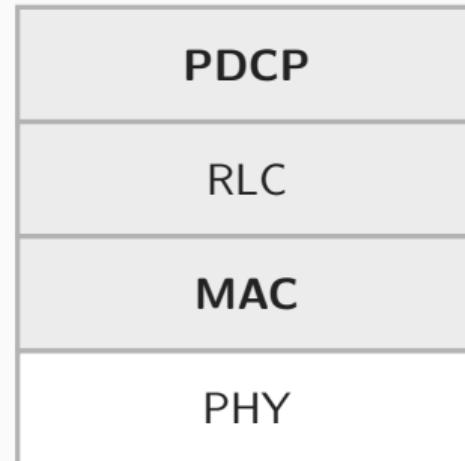


Where do we get the metadata?

- ▶ The PDCP sub-layer gives us the *data*
- ▶ The MAC sub-layer gives us *identifiers*

Challenge:

- ▶ Physical transmission applies *encoding*
- ▶ Option 1: We directly get decoded information in the eNodeB
- ▶ Option 2: We must decode the recordings first



Raw Information

Uncompressed information from traffic

f_1 rnti RNTI

f_2 seq PDCP Sequence Number

f_3 len PDCP Packet Length

f_4 abs Absolute Timestamp

f_5 rel Relative Timestamp

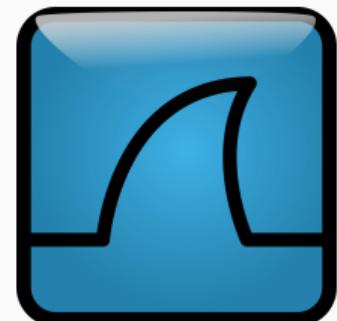
f_4	2.3	2.4	2.5	2.6	2.8	3.0
f_2	11	12	13	14	15	16
f_5	0.1	0.2	0.3	0.4	0.6	0.8
f_3	—	—	—	—	—	—

Requirements to repeat this at home:

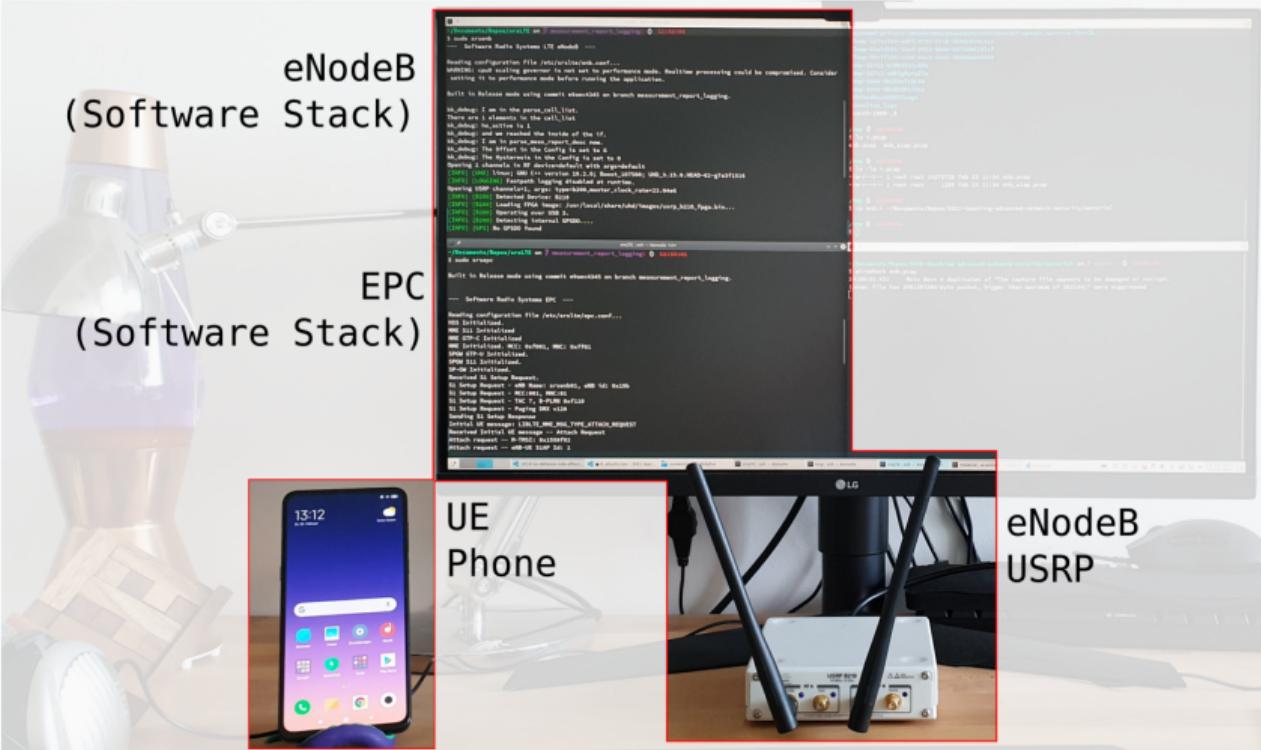
- ▶ Install Wireshark
- ▶ Download the PCAP example from Brightspace
- ▶ Open in wireshark

What we will look at:

- (1) MAC-LTE → RNTI
- (2) PDCP transmissions
- (3) Metadata features in PCAP



Demo: Measurement Setup



Demo: Finding the RNTI

The screenshot shows the Wireshark interface with a packet list and a detailed analysis pane.

Packet List:

No.	Time	Sour	Dest	Protocol	Length	Info
1	0.000000			MAC-LTE	27	RAR (RA-RNTI=2, SFN=296 , SF=6) (RAPID=44: TA=1, UL-Grant=52236, Temp C-RNTI=70)
2	0.010134			LTE RRC UL_CCCH	26	RRConnectionRequest
3	0.022164			LTE RRC DL_CCCH	51	RRConnectionSetup

Analysis Pane:

- > Frame 1: 27 bytes on wire (216 bits), 27 bytes captured (216 bits)
 - | DLT: 147, Payload: mac-lte-framed (mac-lte-framed)
- └ MAC-LTE RAR (RA-RNTI=2, SFN=296 , SF=6) (RAPID=44: TA=1, UL-Grant=52236, Temp C-RNTI=70)
 - > [Context (RNTI=2)]
 - > RAR Headers: (1 RARs)
 - └ RAR Body: (RAPID=44: TA=1, UL-Grant=52236, Temp C-RNTI=70)
 - 0... = Reserved: 0x0
 - > .000 0000 0001 = Timing Advance: 1
 - > 0000 1100 1100 0000 1100 = UL Grant: 52236
 - Temporary C-RNTI: 70
 - Padding data: 00
 - [Padding length: 1]

Hex View:

0000	01 01 02 02 00 02 03 00 00 04 12 86 07 01 0a 00
0010	0f 00 01 6c 00 10 cc 0c 00 46 00	...l.... F.

Bottom Status Bar:

Packets: 2171 · Displayed: 2171 (100.0%) · Profile: Default

RNTI

stands for Radio Network *Temporary* Identifier. They are used to differentiate between multiple connected UEs.

Why do we need the RNTI?

- ▶ MAC sub-layer manages active radio connections
- ▶ Every active connection has its own RNTI
- ▶ There are many different types of RNTI
- ▶ For now we just treat this as a unique and temporary identifier

Demo: Finding PDCP Traffic

enb.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Sou	Dest	Protocol	Length	Info
58	1.247005	8...	172...	DNS	124	[DL] [UM] DRB:1 sn=3 [93-bytes]
59	1.274259	17...	8.8...	TCP	176	[UL] [UM] DRB:1 sn=3 [54-bytes]
60	1.291235	17...	8.8...	TCP	176	[UL] [UM] DRB:1 sn=4 ..15-bytes]
61	1.292084	17...	8.8...	TLSv1.2	104	[UL] [UM] DRB:1 sn=5 ..65-bytes]

Frame 59: 176 bytes on wire (1408 bits), 176 bytes captured (1408 bits)

DLT: 147, Payload: mac-lte-framed (mac-lte-framed)

MAC-LTE UL-SCH: (SFN=423 , SF=6) UEId=0 (Short BSR) (Power Headroom Report) (3:remainder)

 [Context (RNTI=70)]

 MAC PDU Header (Short BSR) (Power Headroom Report) (3:remainder) [3 subheaders]

 Short BSR (lcgid=2 200 < BS <= 234)

 Power Headroom Report (32 <= PH < 33)

 RLC-LTE [UL] [UM] DRB:1 sn=3 [54-bytes] [54-bytes] [39-bytes..]

 [Context]

 UM header sn=3 (2 extensions)

 PDCP-LTE (SN=2)(52 bytes data)

 PDCP-LTE (SN=3)(52 bytes data)

 UM Data: 800445000034185840004006664aac100002080808089bee0355fbca395604e224788010...

0000 01 00 03 02 00 46 03 00 00 04 1a 76 07 01 0a 00 F V

What did you recognize?

- ▶ With the right decoding we see TCP packets.
To make life easier the encryption is disabled.
- ▶ Context (RNTI=70)
This is the same RNTI as in the initial MAC packet.
- ▶ PDCP-LTE (SN=2) (52 bytes data)
PDCP-LTE (SN=3) (52 bytes data)
There are multiple PDCP packets inside one TCP packet.

PDCP

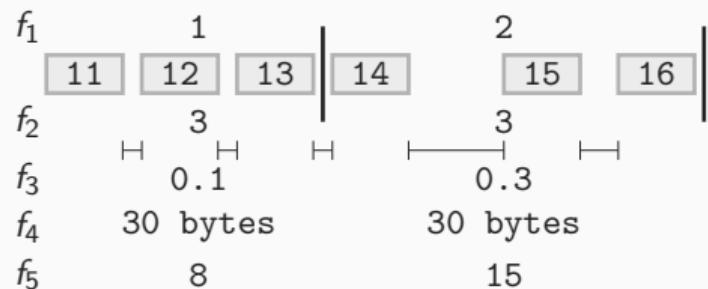
transport the control plane and user plane data and can apply features like header compression, ciphering, or integrity protection.

Why do we look at PDCP packets?

- ▶ They transport the main data
- ▶ It's the data we see on the air interface or in the eNodeB
- ▶ We can derive several traffic features that relate to the transmission

Compressed Information

Aggregated in windows of 500ms length



f_1	win	Window Index
f_2	cnt	Num. Packets in Window
f_3	iat	Avg. Inter-Arrival Time
f_4	byt	Tot. Data in Window
f_5	seq	Avg. Sequence Number

Demo: Malicious eNodeB

If you want to follow along or repeat this later:

- ▶ With SDR: Ettus USRP B2x0/B205mini/X3x0, LimeSDR, bladeRF
- ▶ Clone, install dependencies, build
- ▶ Without SDR: Docker version, integrated channel model

Setting this up is annoying!

Detailed steps can be found in the work sheet, if you get stuck just ask.

Requirements and Preparations

```
# clone, install dependencies, build  
# plugin SDR, antennas, ...  
sudo srsepc # start EPC in terminal 1  
sudo srsenb # start eNodeB in terminal 2
```

- ▶ **Problem:** Traffic is encrypted but metadata leaks information.
- ▶ **Metadata:** Timings, frequencies, sizes, directions, ...
- ▶ **WF:** Classification attack where pre-recorded data set is compared to attack trace.
- ▶ **WF on LTE:** Monitor traffic in eNodeB or at air interface
- ▶ **Features:** RNTI, PDCP, timing.
- ▶ **Demos:** Finding information in PCAP traces, try this at home!

Website Fingerprinting: Exam Examples

- ▶ Recall the protocol stack. Where is the air interface? Why is there an IP-layer in the stack? What's the difference between the UE stack and the eNodeB stack? What is the control plane, what is the user plane?
- ▶ What protocols are part of the second layer in the LTE stack?
- ▶ Name examples of LTE metadata information, this can be either raw information or compressed information.
- ▶ What is the tool we used to take a closer look at PCAP traces?
- ▶ In the context of WF, what is the RNTI used for? What kind of RNTI do we see in the MAC packets of the eNodeB trace?

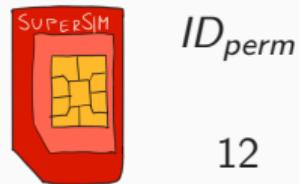
We'll learn more about this in the next part.

Attack 2: Identity Mapping

Required Background

- ▶ Identifiers
- ▶ Connection establishment
- ▶ Uplink and downlink sniffer

Identity Mapping Idea



Match IDs

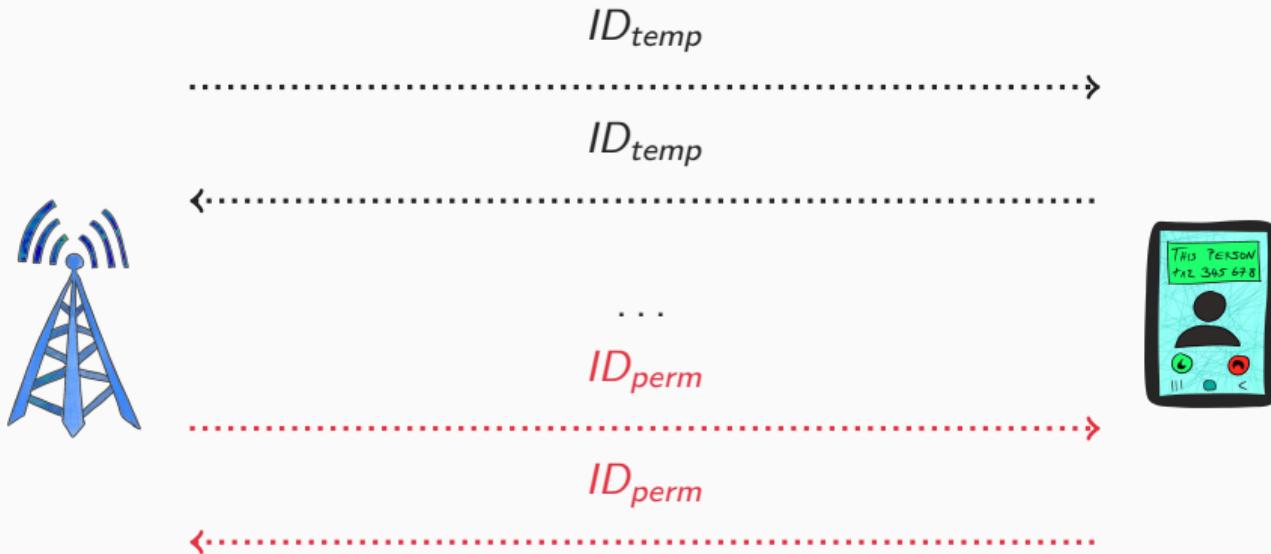
$ID_{temp} \rightarrow ID_{perm}$

From permanent to temporary... or from critical to uncritical

- ▶ IMSI: Lifelong identifier, does not reset
- ▶ TMSI: Semi-permanent random ID, can be reset if needed
- ▶ RNTI: Temporary ID, updated with every new session

We'll see more details about IMSIs, TMSIs, and RNTIs later in this lecture. For now this is enough.

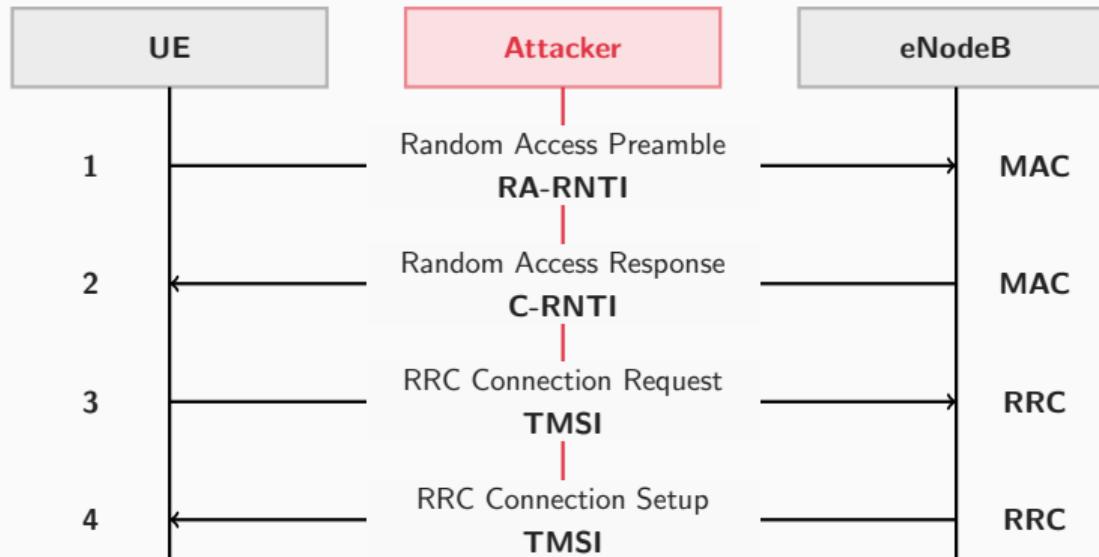
Attack Concept



Monitor Communication

- ▶ Connection establishment exchanges messages between the UE and the eNodeB
- ▶ They're first addressed using the RNTI
- ▶ Later when everything is in place, they can switch to the TMSI
- ▶ Recording both, the $ID_{temp} = RNTI$ and the $ID_{perm} = TMSI$, is the goal
- ▶ **Allows to match the identifiers!**

Identity Mapping Attack



Different types of RNTI exist:

- ▶ **RA-RNTI**: Random Access RNTI. Used for PRACH Response.
- ▶ **C-RNTI**: Cell RNTI. Used for the transmission to a specific UE after RACH.
- ▶ P-RNTI: Paging RNTI. Used for Paging Message.
- ▶ SI-RNTI: System Information RNTI. Used for transmission of SIB messages
- ▶ T-CRNTI: Temporary C-RNTI. Mainly used during RACH
- ▶ SPS-C-RNTI, TPC-PUCCH-RNTI, TPC-PUSCH-RNTI, M-RNTI, CC-RNTI, G-RNTI, SC-RNTI, SL-RNTI, SC-N-RNTI, eIMTA-RNTI



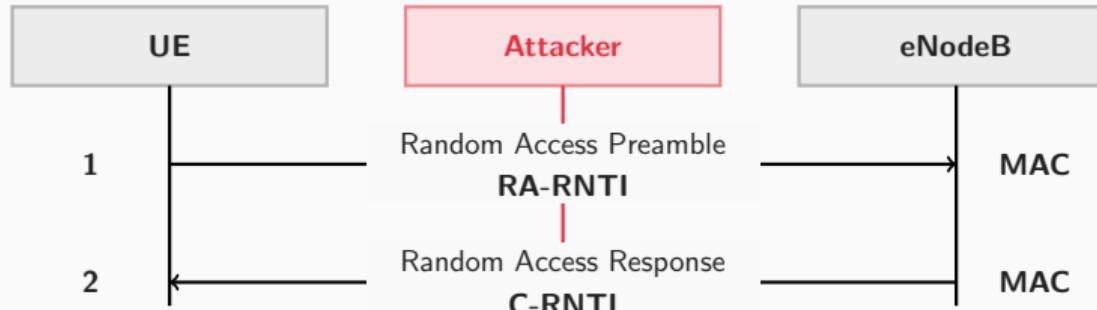
Random Access Preamble

- ▶ UE determines the value of the RA-RNTI
- ▶ $RA-RNTI = 1 + t_{id} + 10 * f_{id}$
- ▶ t_{id} is the index of the first subframe of the specified PRACH
- ▶ f_{id} is the index of the specified PRACH
- ▶ Physical Random Access Channel: UE requests uplink resources from eNodeB



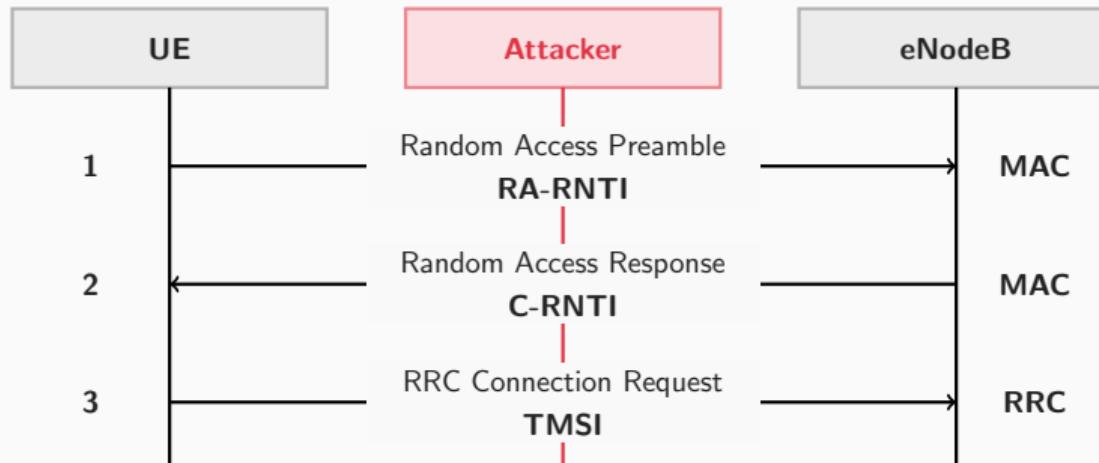
Random Access Preamble (Simplified)

- ▶ UE determines the value of the RA-RNTI
- ▶ **There are only ten possible RA-RNTIs**
- ▶ $RA - RNTI \in [1..10]$



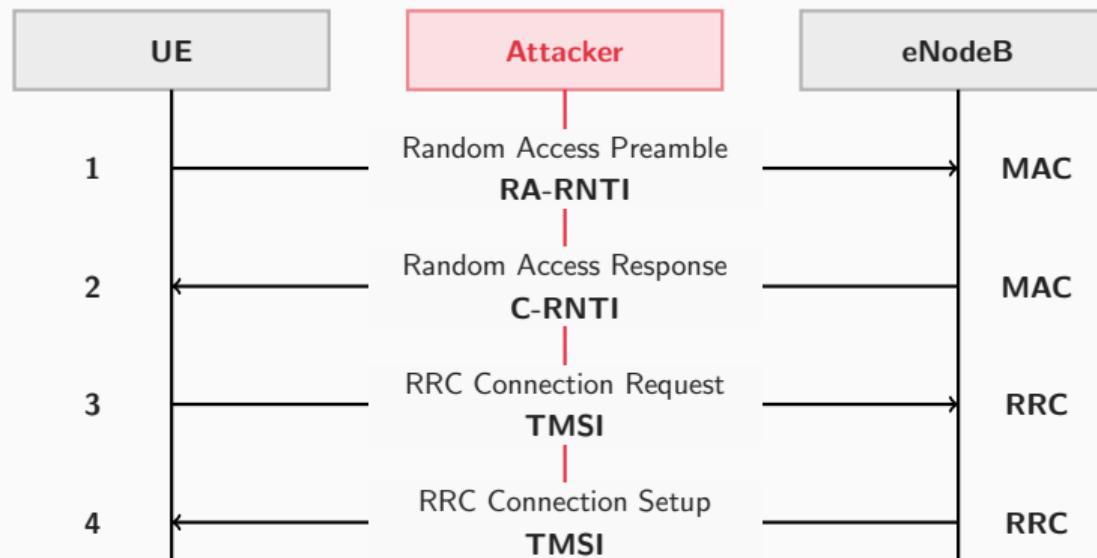
Random Access Response

- ▶ UE determines the value of the RA-RNTI
- ▶ **There are only ten possible RA-RNTIs**
- ▶ $RA - RNTI \in [0..9]$



RRC Connection Request

- ▶ UE requests the connection and sends its TMSI
- ▶ Match between C-RNTI and TMSI



RRC Connection Setup

- ▶ eNodeB setups the connection
- ▶ Match between C-RNTI and TMSI

TMSI

Temporary Mobile Subscriber Identity, randomly assigned temporary identity. For security reasons, the TMSI is a placeholder for the unique IMSI of a user. It can be updated after a certain time period.

IMSI

International Mobile Subscriber Identity, uniquely identifies every mobile user. It is *not* the identifier of the SIM card, but still part of the profile.

The TMSI is used for security reasons!

It can be reset if compromised. The IMSI cannot be reset.

- ▶ **Challenge:** Learn the identifier of a specific user.
- ▶ **Problem:** C-RNTI is different in every new session
- ▶ **Solution:** Try to learn the TMSI! It's temporary but is rarely updated.
- ▶ **Uplink:** Monitor the RRC Connection Request.
- ▶ **Downlink:** Monitor the RRC Connection Setup
- ▶ **Result:** Match C-RNTI to TMSI → Identity!

Identification Attack: Exam Examples

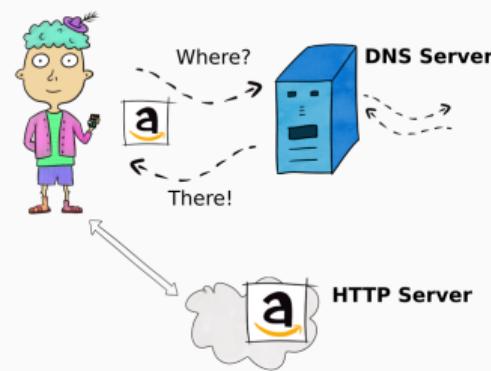
- ▶ Sketch the principle of the Identity Mapping Attack (draw the protocol, know everything in bold font)
- ▶ What is the difference between the RA-RNTI and the C-RNTI?
- ▶ What is the difference between the C-RNTI and the TMSI?
- ▶ Explain what the TMSI is. Why is the TMSI used instead of the IMSI?
- ▶ Explain what the IMSI is.

Attack 3: User Data Redirection

User Data Redirection?

DNS requests simplified:

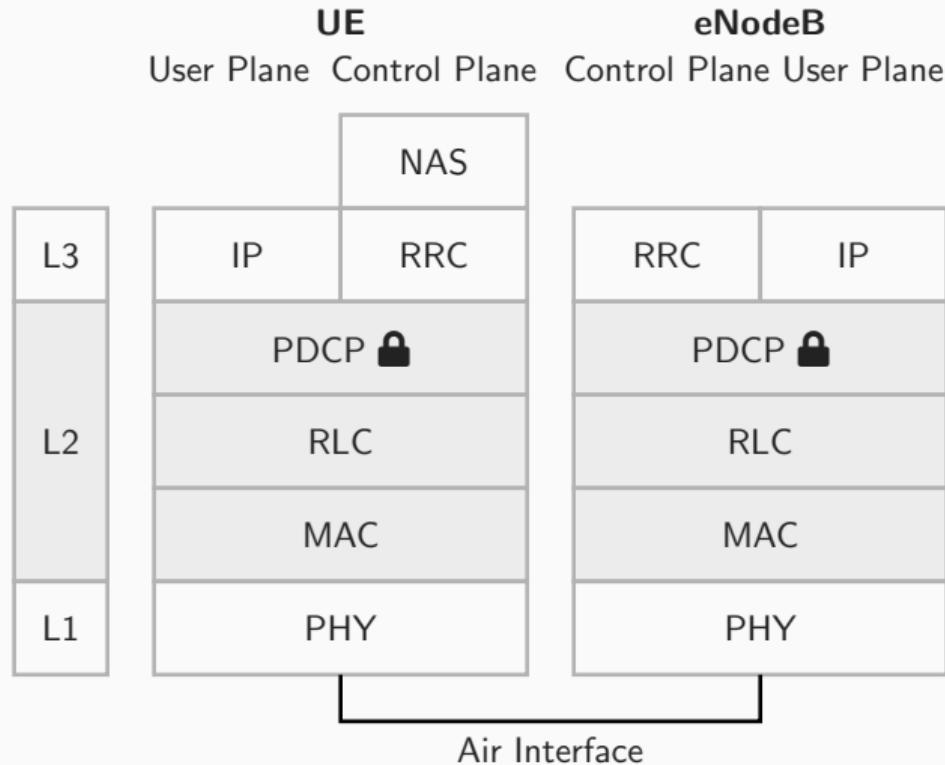
- ▶ User wants to visit a site
- ▶ Asks the DNS Server for directions
- ▶ DNS server looks around
- ▶ Responds
- ▶ User contacts HTTP Server



Three Attack Components

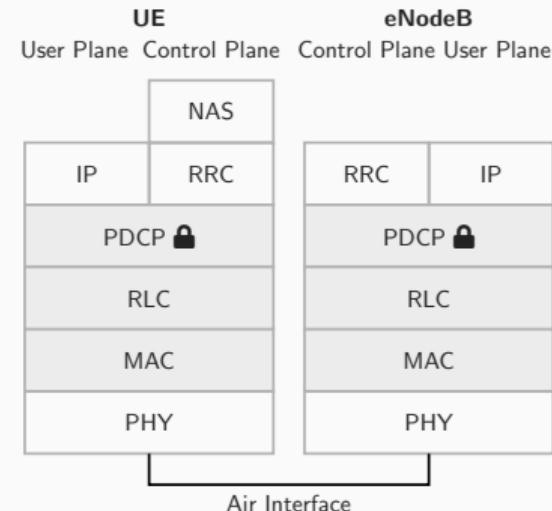
- (1) **Plaintext Modification**
- (2) DNS Spoofing
- (3) Man-in-the-Middle

Plaintext Modification



Plaintext Modification

Feature	Control Plane	User Plane
		
Encryption	✓	✓
Integrity Protection	✓	✗

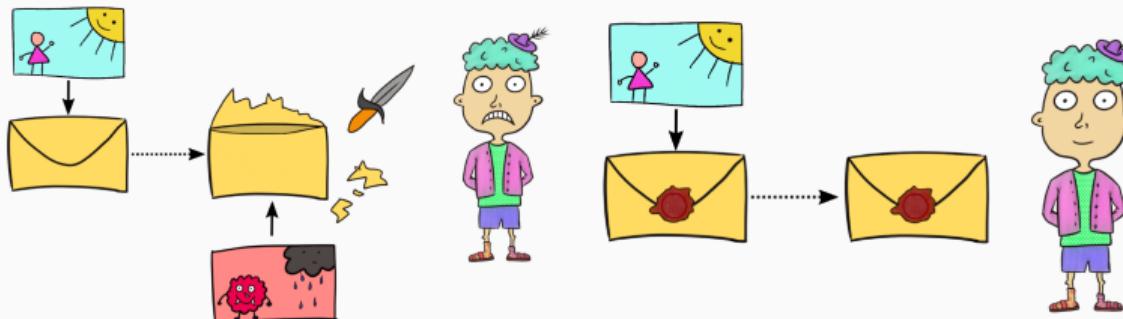


There is no integrity protection for user plane traffic!

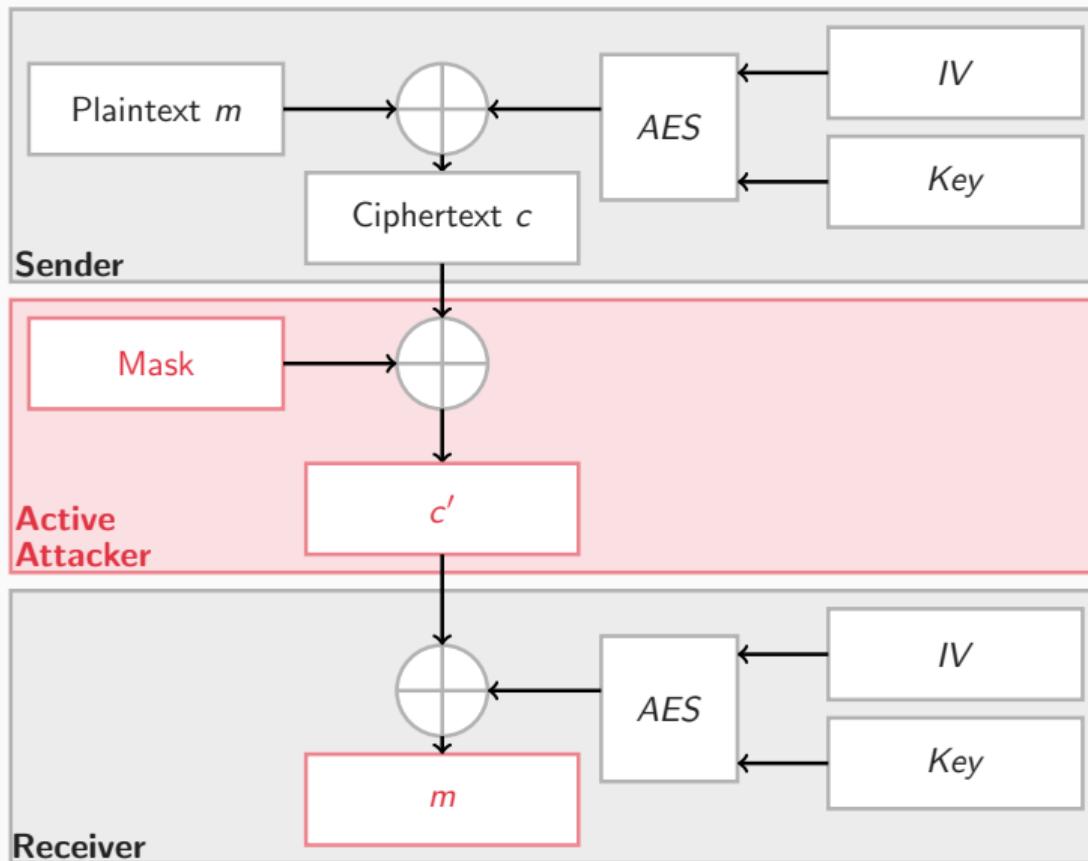
Data Integrity

Nobody fiddled with the data:

- ▶ Original message arrives at the recipient
- ▶ Not changed along the way



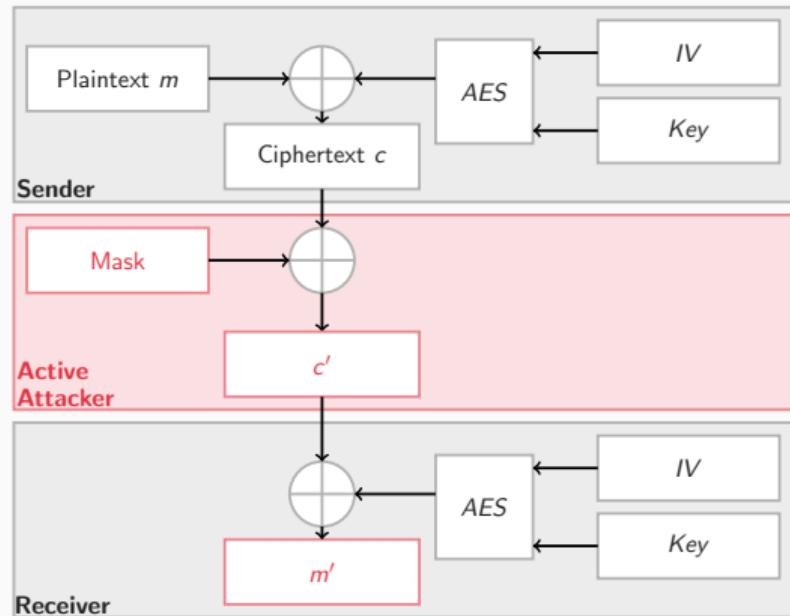
Known-Plaintext Attack



Known-Plaintext Attack

- ▶ PDCP encrypts IP packet
- ▶ Stream cipher: AES in counter mode
- ▶ XOR manipulation mask m
- ▶ Deterministic manipulation
- ▶ Manipulation remains undetected...
But why?

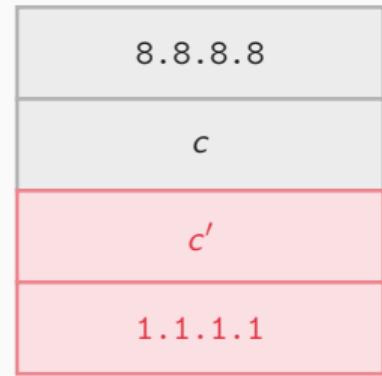
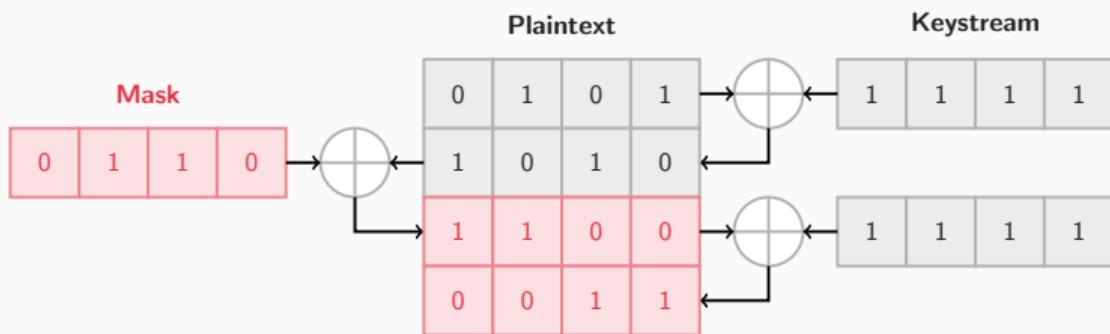
No user plane integrity protection 



Three Attack Components

- (1) Plaintext Modification ✓
- (2) **DNS Spoofing**
- (3) Man-in-the-Middle

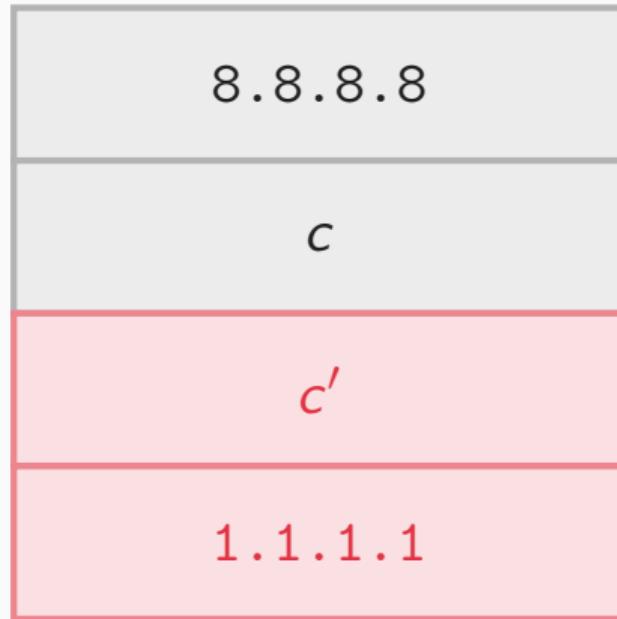
Manipulation Mask



**Example: DNS
Spoofing**

Spoofing DNS Requests

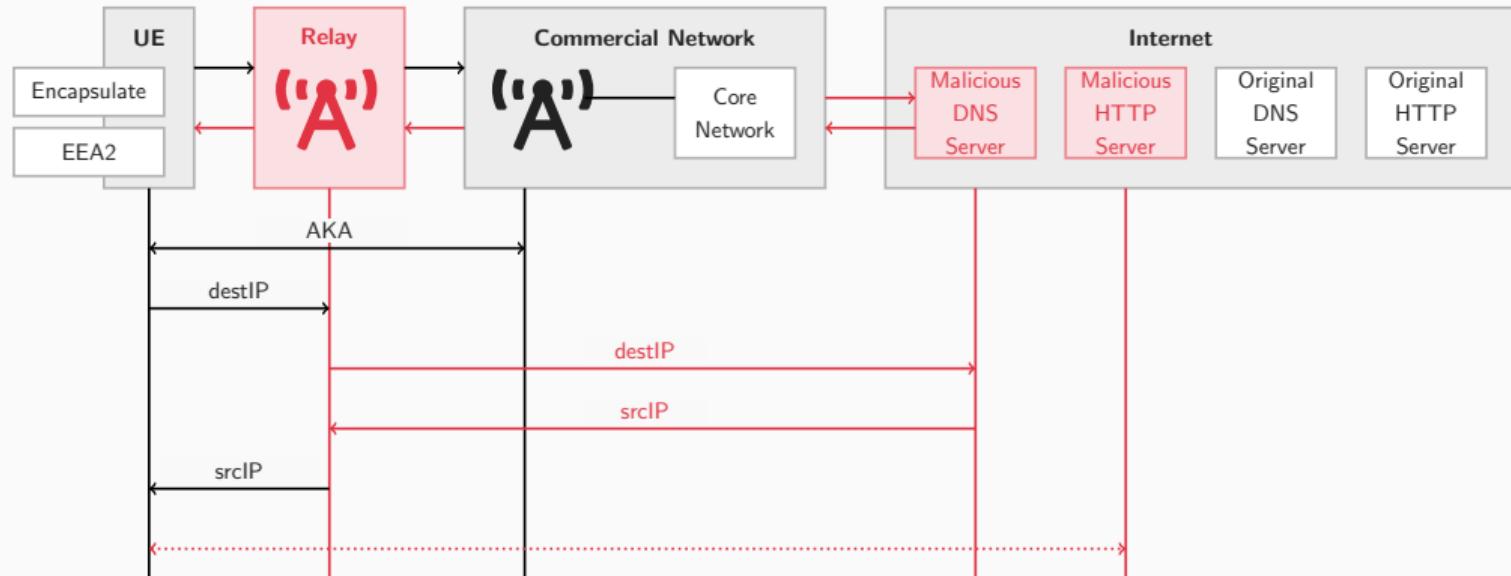
- ▶ Why do we know the plaintext?
Providers have standard DNS resolvers!
- ▶ Prepare a mask that flips bits like we need it
- ▶ Add the mask to create the manipulated c'
- ▶ Receiver recovers plaintext $m' \neq m$



Three Attack Components

- (1) Plaintext Modification ✓
- (2) DNS Spoofing ✓
- (3) **Man-in-the-Middle**

The aLTEr attack



Bringing it all together:

- (1) UE  and eNodeB **"A"** conduct AKA (authentication and key agreement) → Connection is established and ready to use
- (2) UE sends website request including the destIP of the Original DNS Server
- (3) Malicious eNodeB **"A"** recognizes the request and replaces it with a new **destIP** of the **Malicious DNS Server**
- (4) **Malicious DNS server** responds with address of **Malicious HTTP Server**
- (5) **Malicious eNodeB "A"** recognizes response and replaces the malicious **srcIP** with the one of the intended DNS server **srcIP**
- (6) UE  now has spoofed response and sends website request to the **Malicious HTTP Server**. **Unrecognized because of missing integrity protection!**

Three Attack Components

- (1) Plaintext Modification ✓
- (2) DNS Spoofing ✓
- (3) Man-in-the-Middle ✓

Summary

Three L2 Attacks

(1) Website Fingerprinting

- Metadata information in LTE
- Classification attack

(2) Identity Mapping

- Temporary and permanent identifiers
- Matching them by passive sniffing

(3) User Data Redirection

- Known-plaintext attack
- Man-in-the-middle
- DNS spoofing