USENIX Security '25

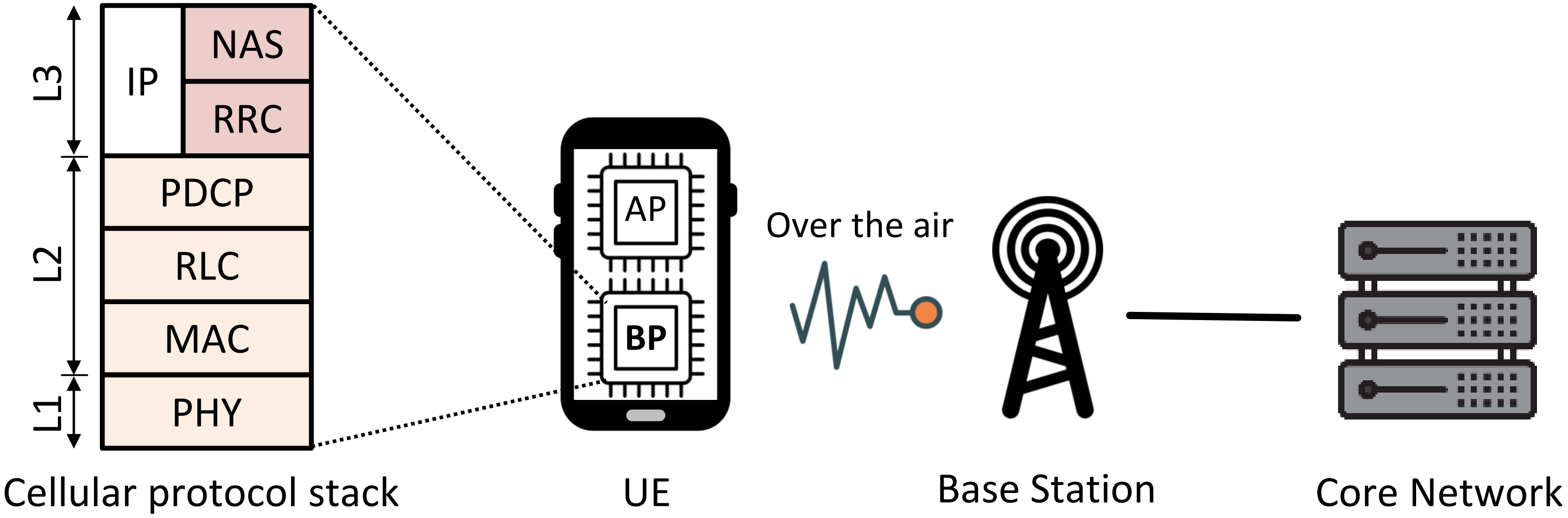
LLFuzz: An Over-the-Air Dynamic Testing Framework for Cellular Baseband Lower Layers

Tuan D. Hoang[1], Taekkyung Oh[1], CheolJun Park[2], Insu Yun[3], Yongdae Kim[1]
[1]SysSec Lab – KAIST, [2]SysSec Lab – Kyung Hee University, [3]Hacking Lab – KAIST

Aug. 15, 2025

Cellular Baseband

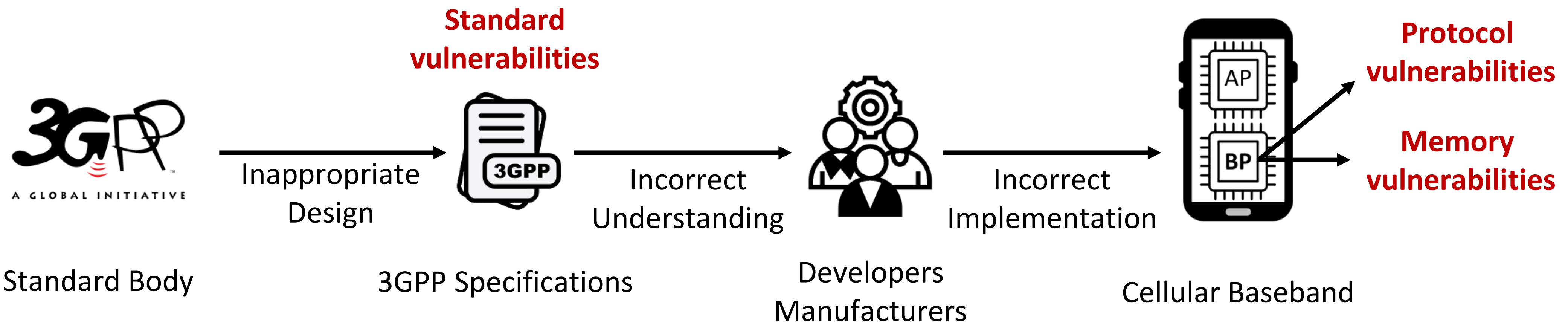
- Handles a wide range of tasks from low-level signal processing to high-level protocol management
- Implements multiple cellular generations, e.g. LTE, 5G
- Cellular protocol stack: 3 main layers
  - L1 – PHY
  - L2 includes three sub-layers: MAC, RLC, PDCP
  - L3 includes control-plane protocols: NAS, RRC



AP: Application Processor, BP: Baseband Processor

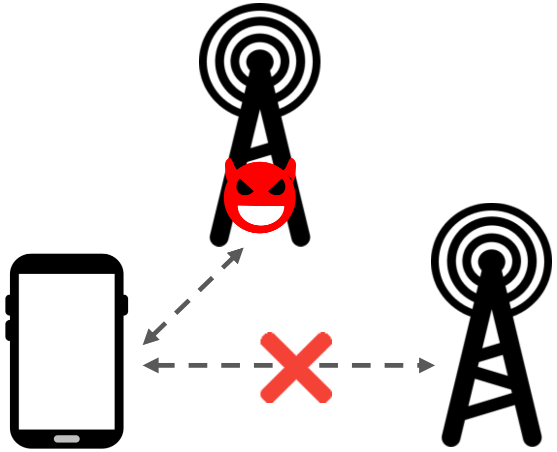
Vulnerabilities in Cellular Basebands

- Different types of vulnerabilities in basebands
  - Specification: vulnerabilities and attacks in design (e.g. [Oh24], [Karim21], [Yang19], [Hussain18,19], …)
  - Implementation
    - Protocol: Non-compliance with specifications (e.g. [Bitsikas23], [Chen23], [Park21], [Rupprecht16], [Kim19], …)
    - Memory: Low-level memory safety issues in C/C++ (e.g. [Shang24], [Hernandez22], [Maier20], [Kim21], …)



Memory Corruptions in Cellular Basebands

- Many functions across layers are used to process downlink packets from base stations
  - C/C++ code base
  - Shared memory architecture
- Potentially lead to severe consequences
  - DoS, remote code execution, information leakage
  - Can be exploited over the air
- A topic of great interest to both academia and industry



Attack model: FBS

E2E exploit on Huawei Smartphone
(Black Hat USA 2018)

0-click RCE on Tesla via a cellular modem
(Pwn2Own Automotive 2024)

FBS: Fake Base Station

Previous Works

- Mainly focus on Layer 3
  - Three main techniques: reversing, emulation-based fuzzing, and over-the-air fuzzing
- Only a few studies targeted Layer 2
  - Goos et al.:
    - Extended FirmWire to support L2 of GPRS
  - 5Ghoul:
    - Random mutation guided by grammar

Study	Gen.	Approach	Layers
Breaking Band (Comsecuris 16)	GSM-LTE	Reversing	L3
Nico Golde (Comsecuris 18)	GSM-LTE	Reversing	L3
Marco Grassi (Offensive 20)	GSM-LTE	Reversing	L3
Amat cama (OffensivCon 23)	GSM-LTE	Reversing	L3
BaseSpec (NDSS 21)	LTE	Reversing	L3
BaseSAFE (WiSec 20)	LTE	Emulation	L3
FirmWire (NDSS 22)	GSM, LTE	Emulation	L3
BaseBridge (SP 25)	LTE	Emulation	L3
LORIS (SP 25)	LTE, 5G	Emulation	L3
Goos et al. (BlackHat 24)	GPRS	Emulation	L2, L3
5Ghoul, U-Fuzz (ICST 24)	5G	OTA	L2, L3
LLFuzz	**LTE**	**OTA**	**L1, L2**

Previous Works

- Mainly focus on Layer 3
  - Three main techniques: reversing, emulation-based fuzzing, and over-the-air fuzzing
- Only a few studies targeted Layer 2
  - Goos et al.:
    - Extended FirmWire to support L2 of GPRS
    - (–) LTE/5G lower layers are not supported in the state-of-the-art emulator
  - 5Ghoul:
    - Random mutation guided by grammar
    - (–) Only focuses on the pre-authentication state
    - (–) Mutate elementary messages generated by open-source base station

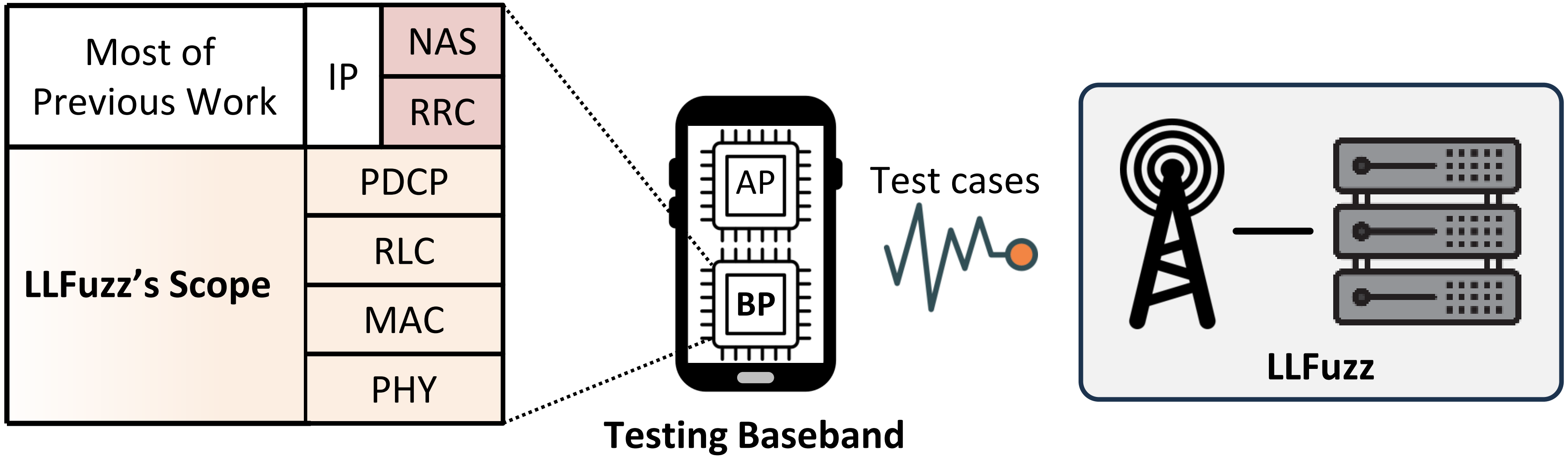
Study	Gen.	Approach	Layers
Breaking Band (Comsecuris 16)	GSM-LTE	Reversing	L3
Nico Golde (Comsecuris 18)	GSM-LTE	Reversing	L3
Marco Grassi (Offensive 20)	GSM-LTE	Reversing	L3
Amat cama (OffensivCon 23)	GSM-LTE	Reversing	L3
BaseSpec (NDSS 21)	LTE	Reversing	L3
BaseSAFE (WiSec 20)	LTE	Emulation	L3
FirmWire (NDSS 22)	GSM, LTE	Emulation	L3
BaseBridge (SP 25)	LTE	Emulation	L3
LORIS (SP 25)	LTE, 5G	Emulation	L3
Goos et al. (BlackHat 24)	GPRS	Emulation	L2, L3
5Ghoul, U-Fuzz (ICST 24)	5G	OTA	L2, L3
LLFuzz	**LTE**	**OTA**	**L1, L2**

No systematic approach for testing modern lower layers (LTE/5G)

SYSSEC KAIST

LLFuzz

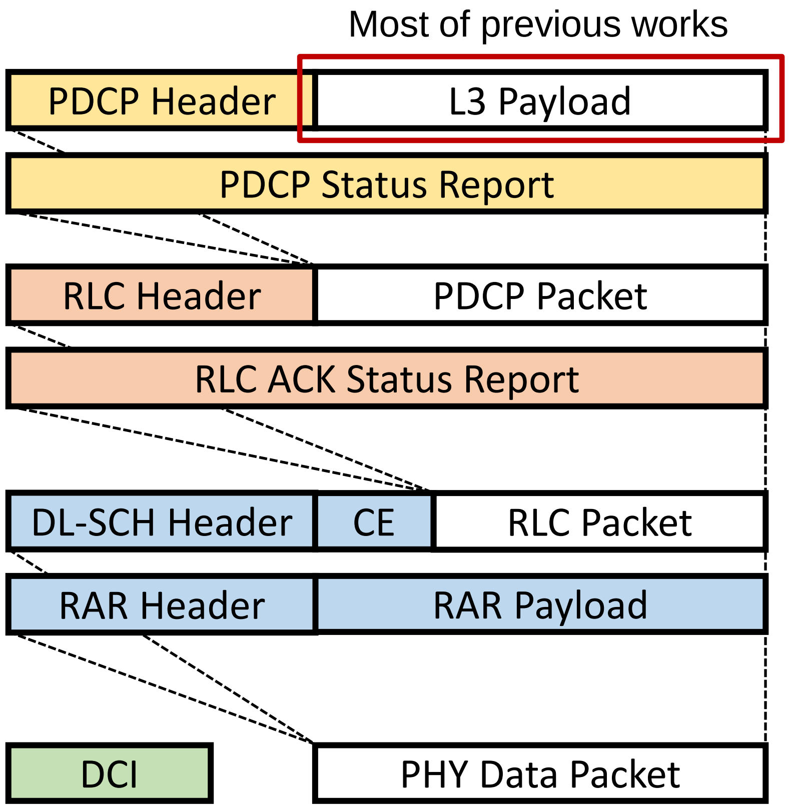
- ❖ Lower layers remain underexplored despite having no protection (MAC-I, encryption)
- ❖ Develop a systematic approach to detect memory corruptions in lower layers
  - – Layer 1 – PHY
  - – Layer 2 – MAC, RLC, PDCP
- ❖ Leverage over-the-air (OTA) fuzzing
  - – Can fuzz commercial basebands from any vendor
  - – Stateful testing



SYSSEC KAIST

Background on Lower layers

❖ Many packet structures to support each layer's functionalities

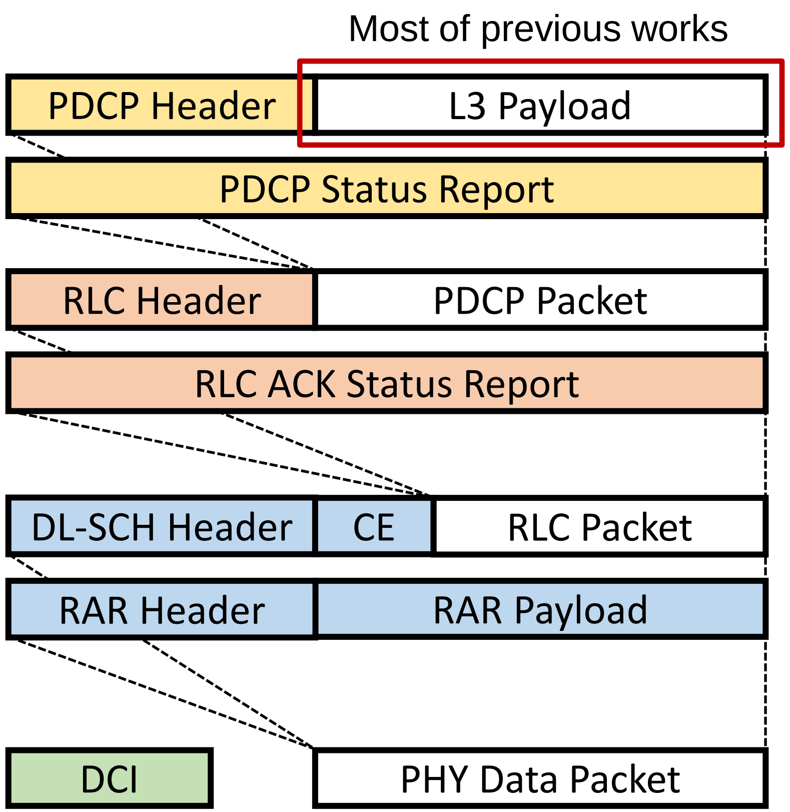


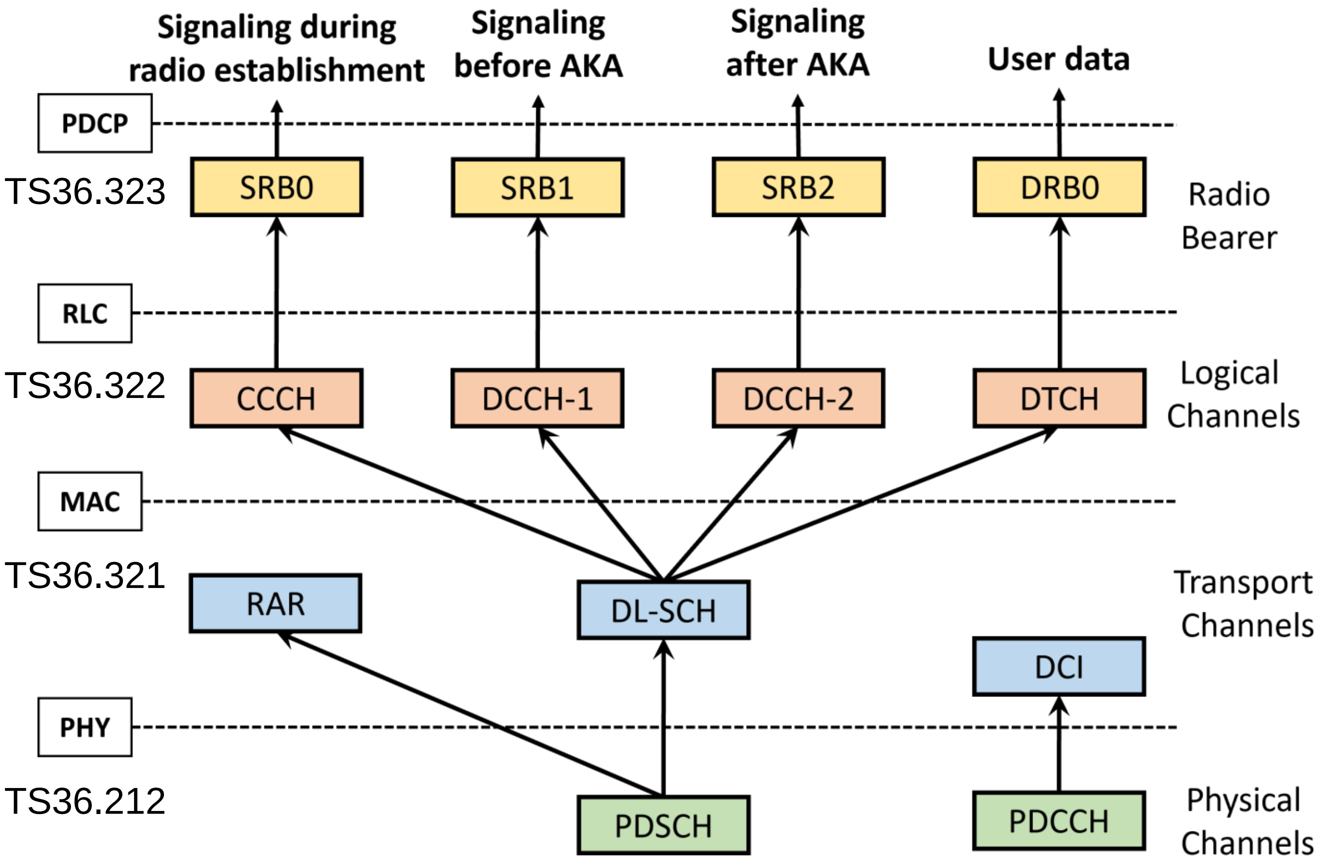
PDCP	Encryption, Integrity Protection Header compression
RLC	Segmentation Layer 2 ACK, Retransmission
MAC	Random Access, Multiplexing Dedicated control information (Control Element – CE)
PHY	Wireless signal decoding

AKA: Authentication and Key Agreement; DCI: Downlink Control Information; RAR: Random Access Response

Background on Lower layers

- ❖ Many packet structures to support each layer's functionalities
- ❖ Packets are mapped to different **logical channels** based on data types





AKA: Authentication and Key Agreement, DCI: Downlink Control Information, CE: Control Element

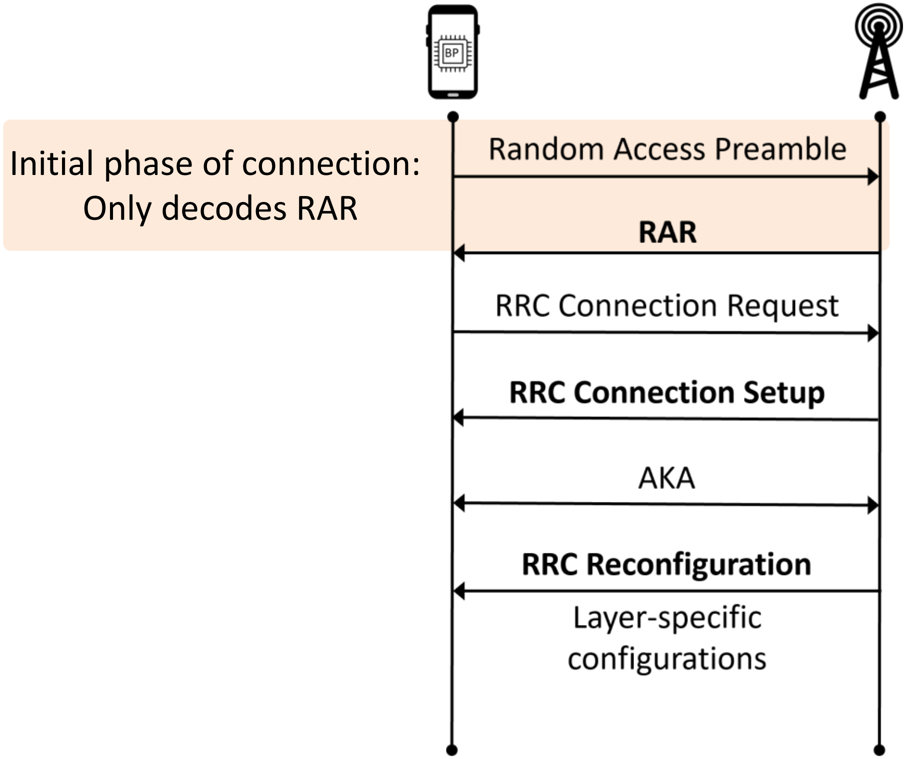
Challenges – Approaches (1)

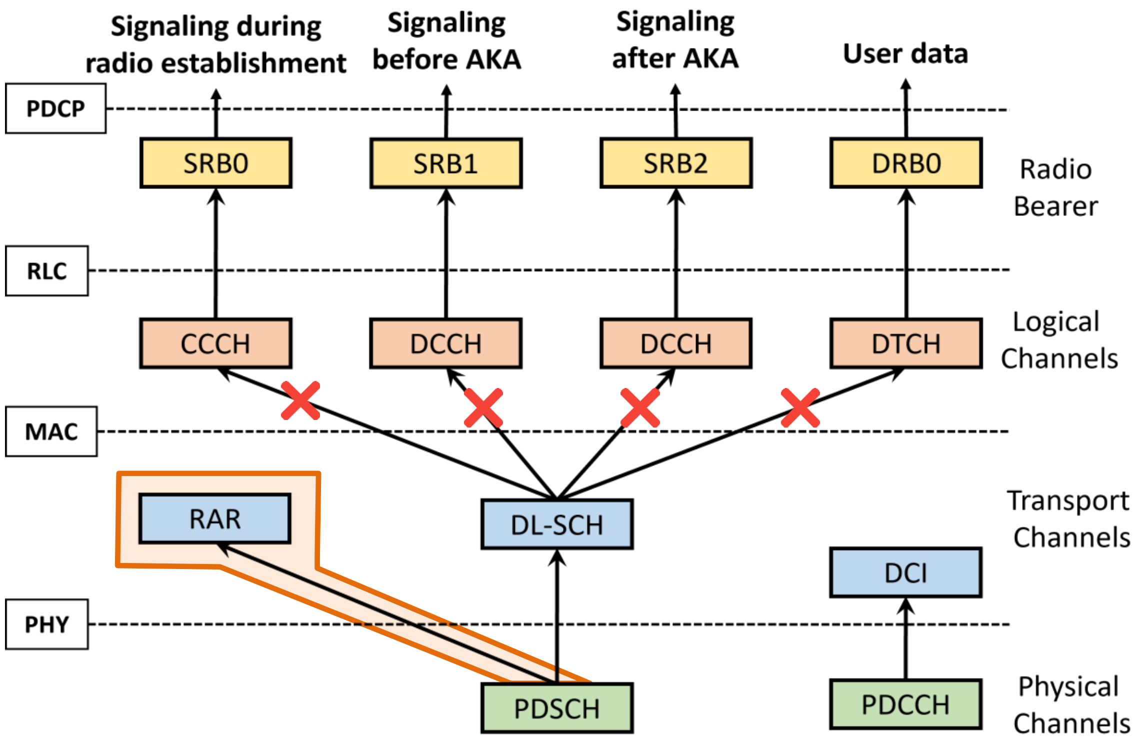
- Challenge 1: Complex packet structures for OTA testing
  - Many packet structures defined in specifications
    - Many of them are rarely used in commercial and open-source networks
    - Slow OTA testing speed
  - Commercial basebands are black-box
    - Coverage-guided fuzzing cannot be applied

- Approach 1: Specification-guided test case generation
  - Generate diverse standard-compliant packets
  - Structure-preserving field mutation (length-respecting)
  - Useful for root cause analysis when bugs are found

Challenges – Approaches (2)

- Challenge 2: Diverse packet structures across multiple channels
  - Logical channels decide which packet structures should be used
  - Not all channels are active at all times

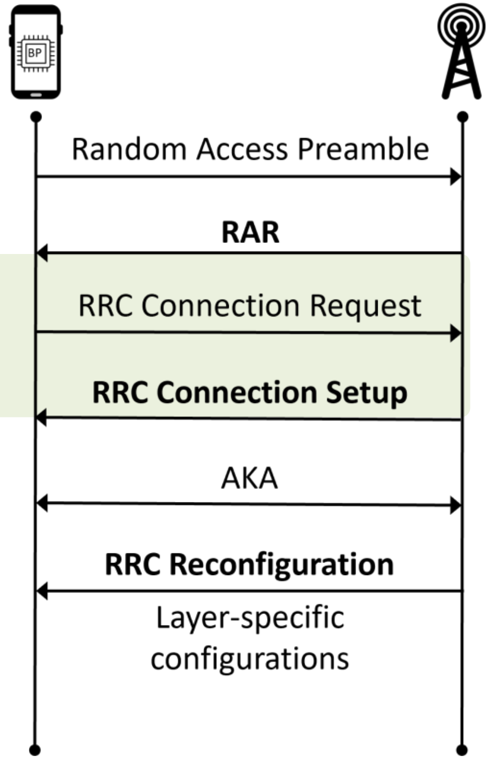




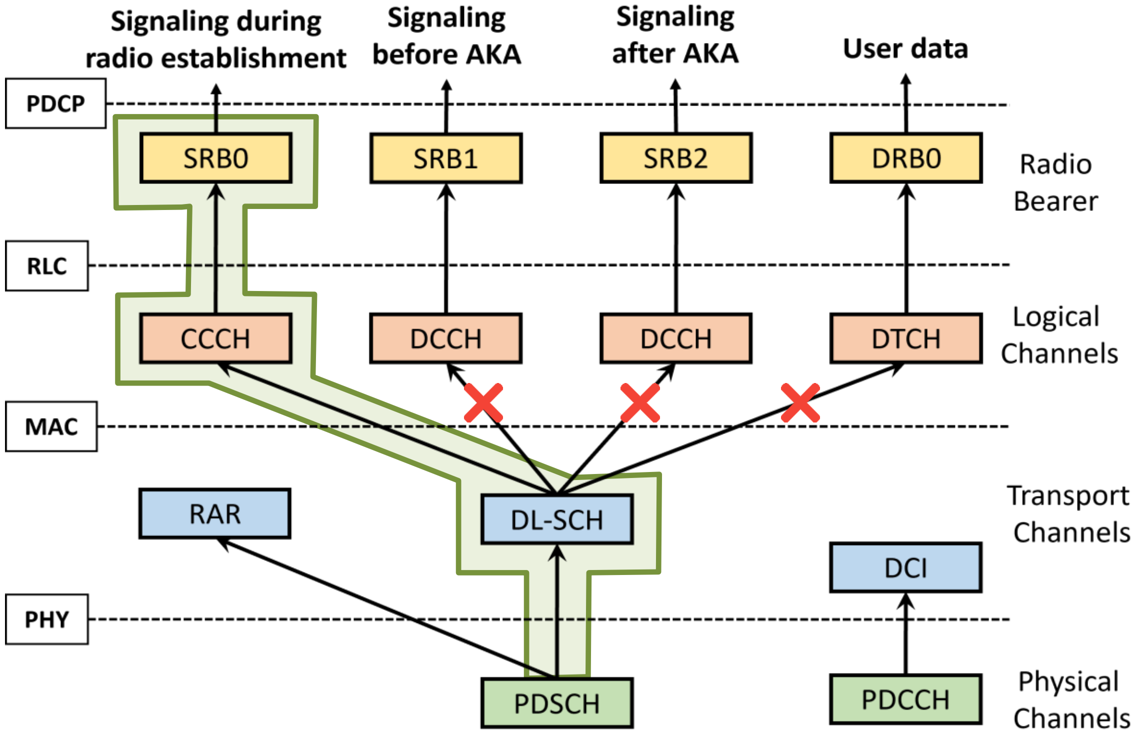
RAR: Random Access Response message; CCCH: Common Control Channel

Challenges – Approaches (2)

- Challenge 2: Diverse packet structures across multiple channels
  - Logical channels decide which packet structures should be used
  - Not all channels are active at all times



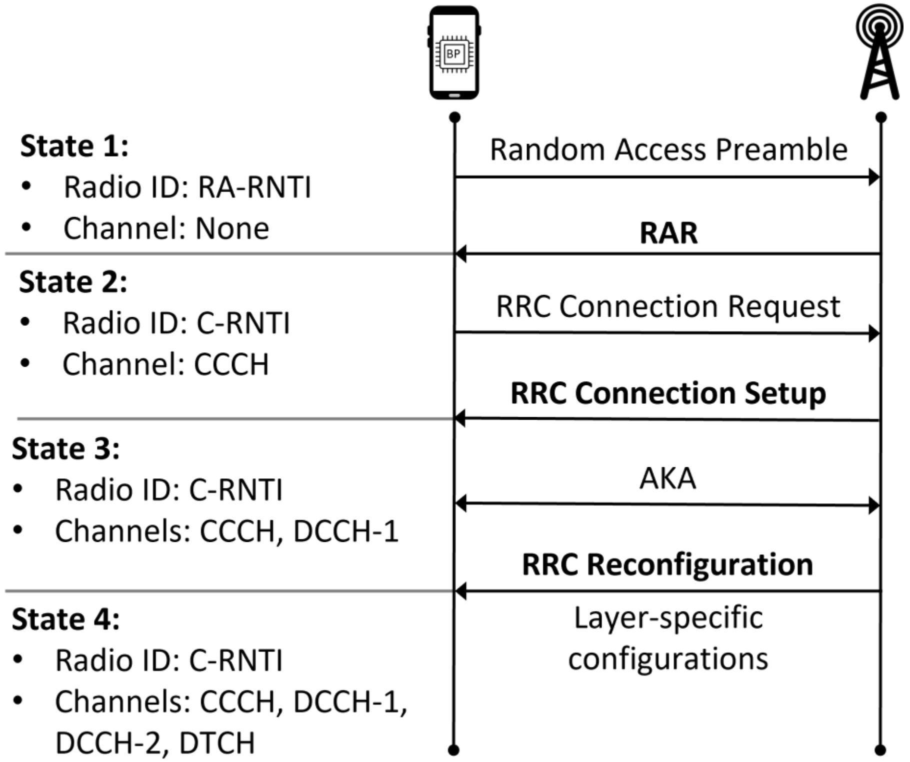
Enables CCCH channel
Others remain inactive

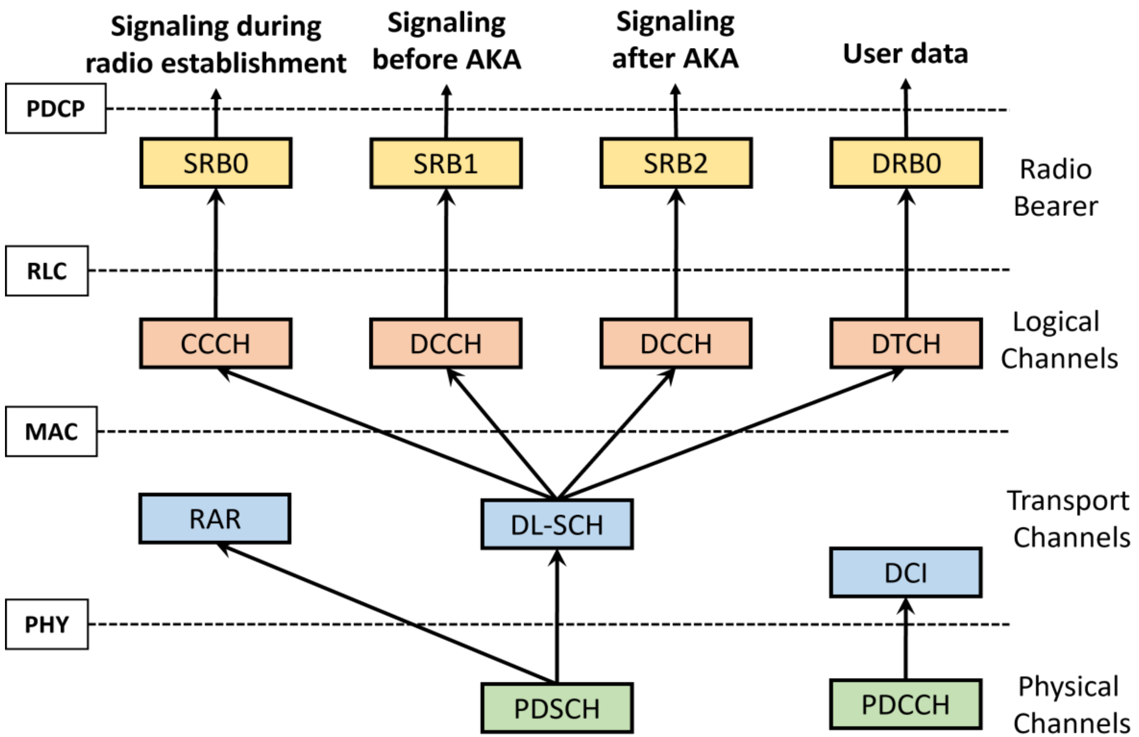


RAR: Random Access Response message; CCCH: Common Control Channel

Challenges – Approaches (2)

- Approach 2: Channel-driven stateful testing
  - Newly define 4 channel-oriented states based on the establishment of logical channels





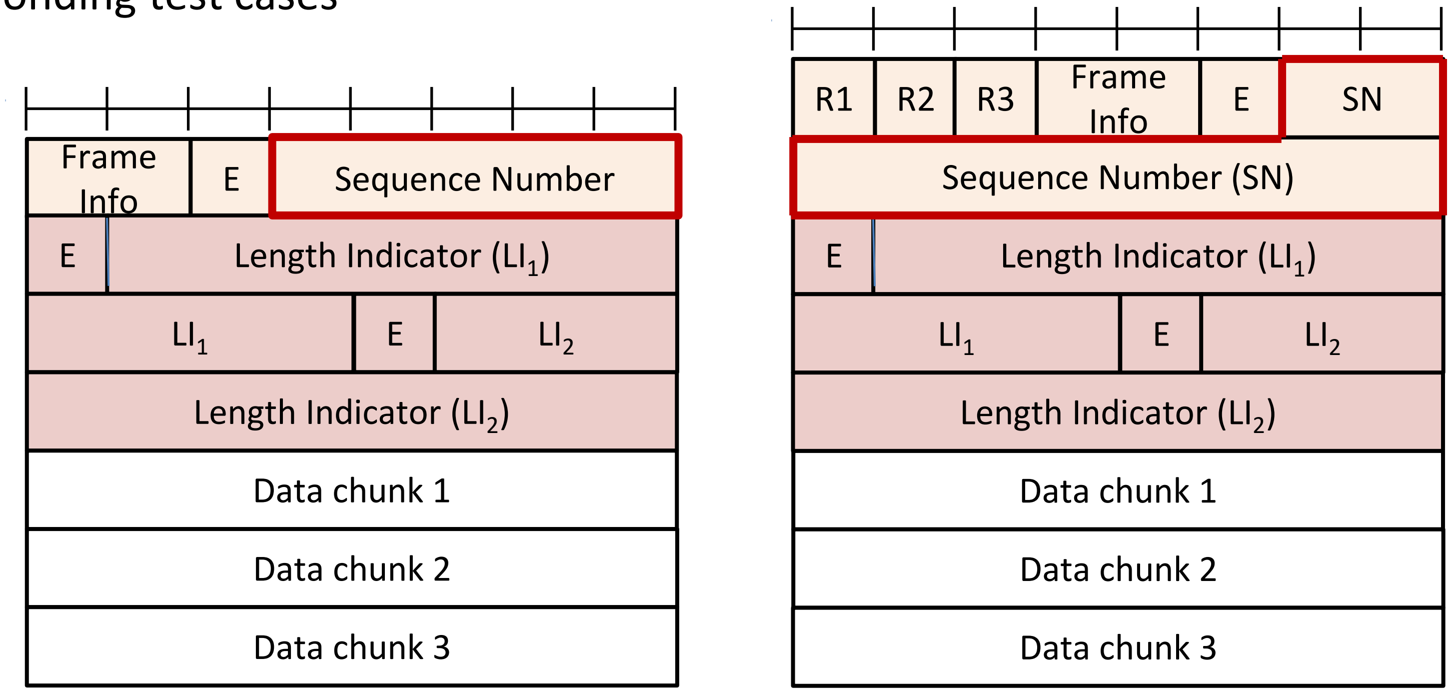
RAR: Random Access Response message; CCCH: Common Control Channel

Challenges – Approaches (3)

- Challenge 3: Configurable packet structures
  - RLC & PDCP packet structures mapped to the DTCH channel can be configured differently
- Approach 3: Configuration-aware testing
  - First, modify the **RRC Connection Reconfiguration** to deliver target configuration to UE
  - Then, generate and send corresponding test cases

```
rrcConnectionReconfiguration-r8
  radioResourceConfigDedicated
    drb-ToAddModList: 1 item
      Item 0
        DRB-ToAddMod
          eps-BearerIdentity: 5
          drb-Identity: 1
          pdcp-Config
          rlc-Config: um-Bi-Directional (1)
            um-Bi-Directional
            ul-UM-RLC
            dl-UM-RLC
              sn-FieldLength: size5 (0)
```

RRC Reconfiguration msg delivers 5-bit SN configuration

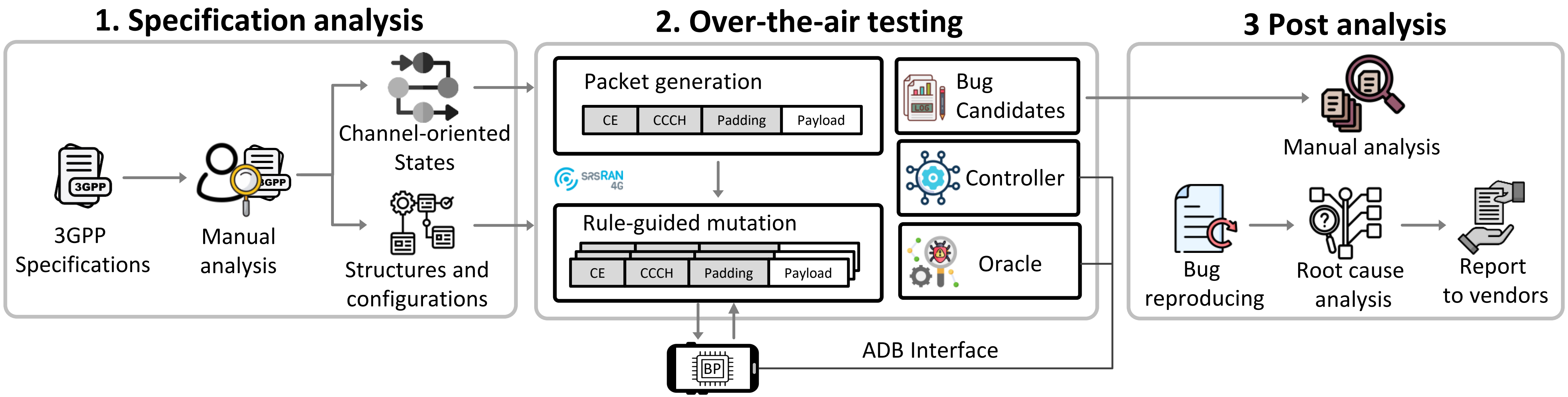


RLC UM Data Packets with 5-bit/10-bit Sequence Number (SN)

DTCH: Dedicated Traffic Channel; UM: Unacknowledged Mode in RLC layer

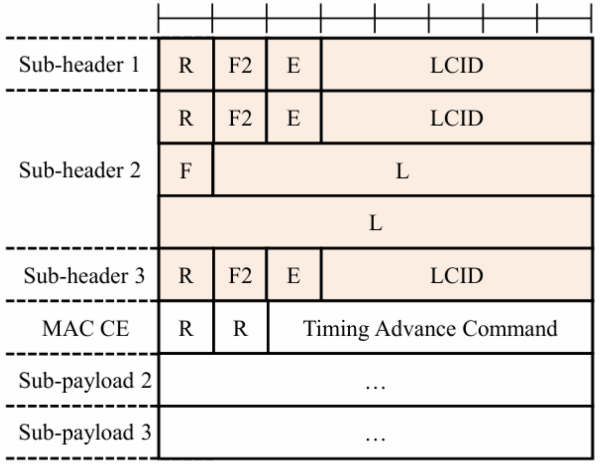
Design & Implementation

- Design:
  - (1) Specification analysis
  - (2) Over-the-air testing
  - (3) Post analysis
- Implementation
  - Built on top of srsENB
  - C/C++, ~11.5K lines of code

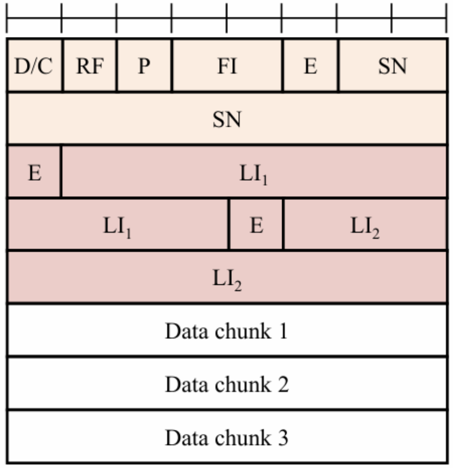


Identify Packet Structures in Lower Layers

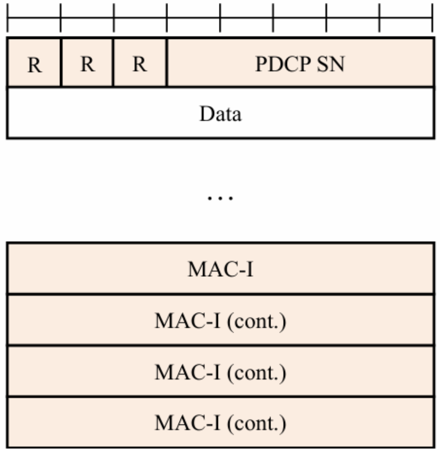
- ❖ MAC Layer: tested 19 structures
  - – Packet structure depends on the UE state
  - – Many sub-header formats
  - – MAC Control Elements (CEs)
- ❖ RLC Layer: tested 18 structures
  - – Packet structure depends on RLC Modes
    - ▪ TM/UM/AM
  - – Configurable sizes for LI and SN fields
  - – Dedicated Control PDUs for RLC ACK
- ❖ PDCP Layer: tested 17 structures
  - – Packet structure differs for signaling/user data
  - – Control PDUs for transmission status
- ❖ PHY Layer: tested 11 DCI structures
  - – DCI structures depend on channel bandwidth, transmission modes, and baseband states



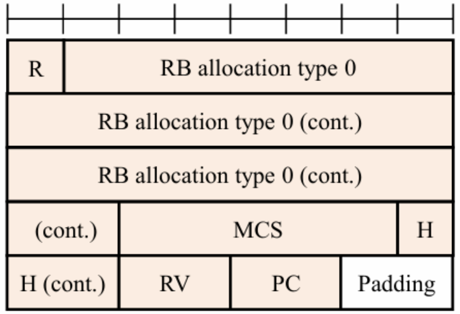
MAC DL-SCH Packet Structure



RLC UM Data Packet Structure



PDCP Data Packet Structure



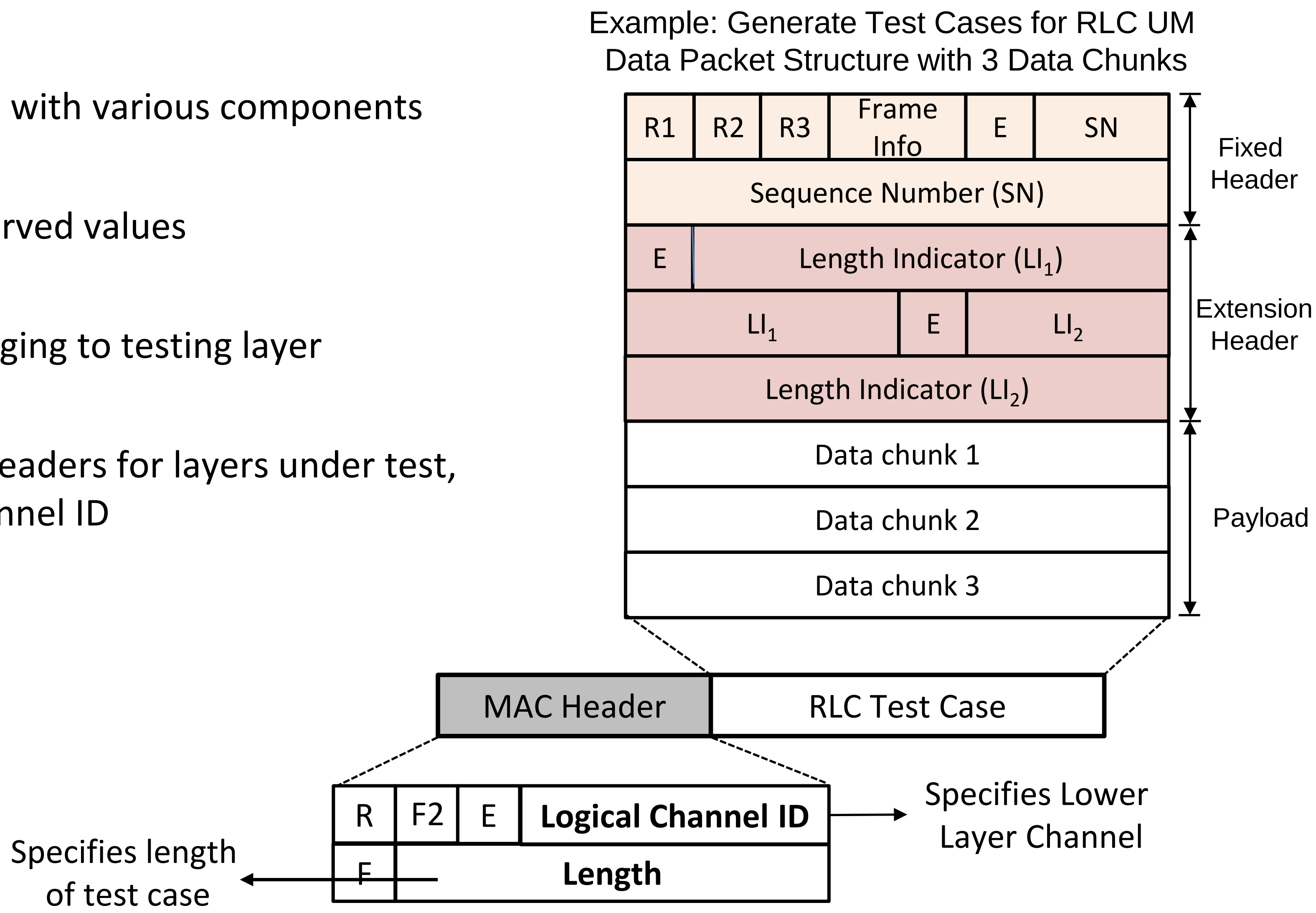
DCI Structure

DCI: Downlink Control Information

SYSSEC KAIST

Test Case Generation

- (1) Initial packet generation
  - Generate legitimate packets with various components
- (2) Header mutation
  - Focus on boundary and reserved values
- (3) Payload mutation
  - Only mutate payloads belonging to testing layer
- (4) Logical channel mapping
  - Generate appropriate sub-headers for layers under test, with the correct Logical Channel ID



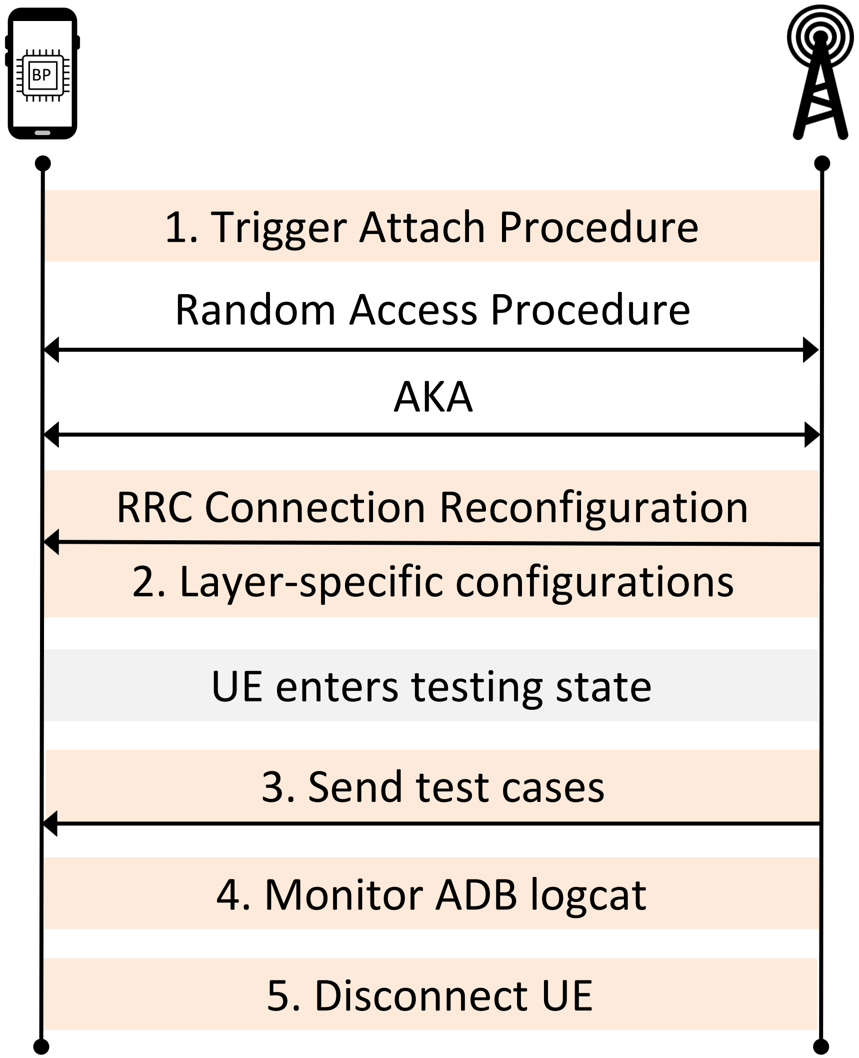
SYSSEC KAIST

Oracle for Detecting Bugs

- ❖ Leverage debug messages from ADB logcat
- ❖ Separate thread for ADB monitoring
- ❖ Magic strings:
  - “RADIO_OFF_OR_UNAVAILABLE”
  - “Modem Reset”
  - “Everybody panic!”

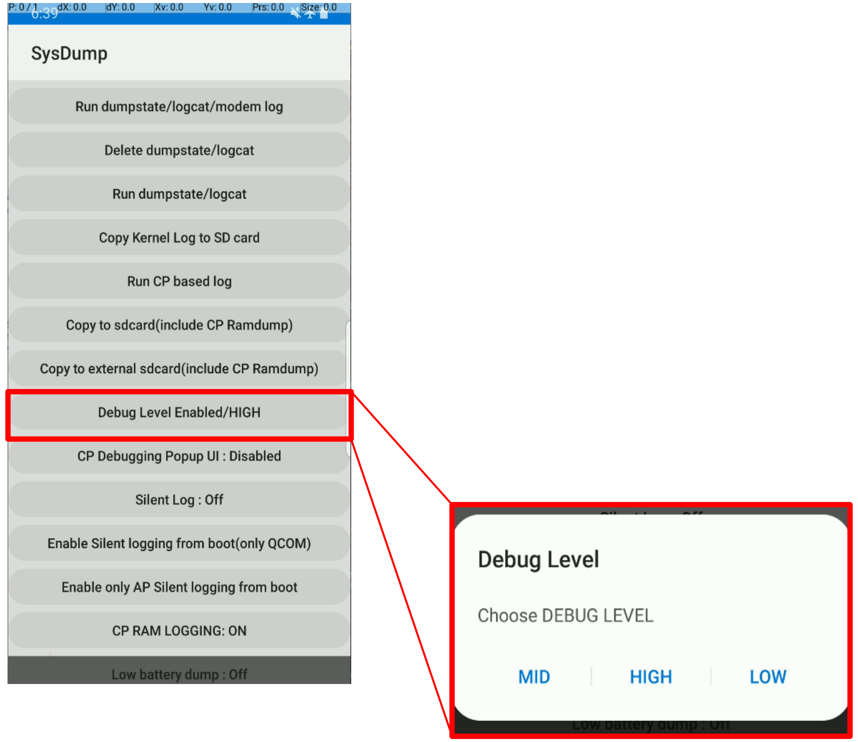
```
10-11 18:27:07.071  1885  2306 E RILD    : LookupProfile: Failed to get profile list
10-11 18:27:07.074  1885  2568 E RILD    : All Service is closed, Modem Reset!!
10-11 18:27:07.076  1885  2306 E RILD    : ApplyQoS: [sqos] mLinkFlow for cid(2) is set(1)
10-11 18:27:07.076  1885  2306 E RILD    : [sqos]Qos entry with flow ID 0x0 is not found
```

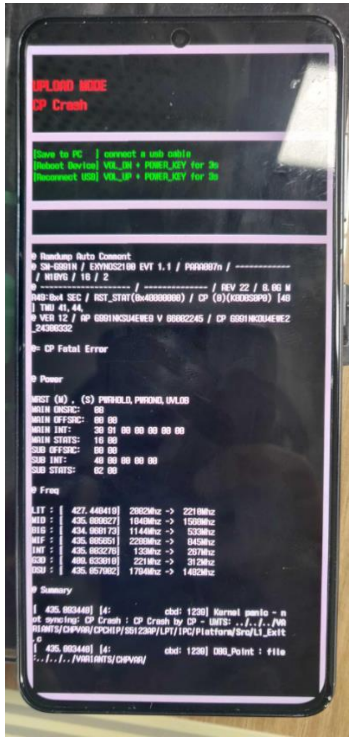
OTA Testing Procedure



Post Analysis

- Crashes detected by ADB-based oracle might have false positives
- Use vendor’s debug mode (*#9900#) to verify bug candidates





Target Devices

Vendor	No.	Smartphone	Baseband Model
Qualcomm	1	SS* Galaxy Note 20 Ultra	Snapdragon 865+
	2	SS Galaxy S20	Snapdragon 865
	3	SS Galaxy S22 Plus	Snapdragon 8 Gen 1
	4	SS Galaxy S24 Ultra	Snapdragon 8 Gen 3
	5	OnePlus 9 Pro	Snapdragon 888
MediaTek	6	SS Galaxy A31	Helio P65
	7	SS Galaxy A32	Helio G80
	8	Xiaomi K40 Gaming	Dimensity 1200
	9	Xiaomi Redmi Note 9T	Dimensity 800U
Samsung Exynos	10	SS Galaxy S21	Exynos 2100
	11	SS Galaxy S24	Exynos 2400
	12	SS Galaxy S10e	Exynos 9820
Google Tensor	13	Pixel 6a	Google Tensor
	14	Pixel 8 Pro	Google Tensor G3
Huawei Kirin	15	Huawei P30 Pro	Kirin 980

SS: Samsung

Results (LTE)

- ❖ Found 9 previously unknown memory corruptions: 2 in PDCP, 2 in RLC, and 5 in MAC layers.
- ❖ Affect basebands from 4 major vendors: Qualcomm, MediaTek, Samsung, Google

No.	Vendor*	Layer	State	Configuration	Disclosure
B1	Qualcomm	MAC	S2, S3, S4	-	CVE-2025-21477, Patched
B2		MAC	S1	-	CVE-2024-23385, Patched
B3		RLC	S4	UM, 5-bit SN	Verified
B4	MediaTek	MAC	S3	-	CVE-2024-20076, Patched
B5		MAC	S2, S3, S4	-	CVE-2024-20077, Patched
B6		MAC	S2, S3	-	Affects only old firmwares
B7		PDCP	S4	-	CVE-2025-20659, Patched
B8	Tensor, Exynos	RLC	S3, S4	AM, 11-bit LI, 10-bit SN	CVE-2025-26781/26782
B9	Exynos	PDCP	S4	12-bit SN	CVE-2025-26780, Patched

* A list of tested and affected basebands is provided in our paper

Impact of Lower-layer Bugs

- CVE summary: 9 CVEs were assigned
  - Qualcomm: CVE-2025-21477, CVE-2024-23385 – Affecting 90+ baseband chipsets
  - MediaTek: CVE-2025-20659, CVE-2024-20076/77 – Affecting 80+ baseband chipsets
  - Samsung: CVE-2025-26780 – Found in Exynos 2400 and Modem 5400
  - Google: CVE-2025-26781/82 – Found in Google Tensor and Exynos 2400
  - Apple: CVE-2024-27870 – Overlapping with Qualcomm

- Not guaranteed to be patched – supply-chain issue
  - Whether a patch is applied depends on the device vendors (e.g. smartphones, IoT devices, cars, …)

- Bugs remain exploitable even after Authentication and Key Agreement (AKA)
  - Lower layers are **not cryptographically protected** by design

Fuzzing 5G Basebands (in 2 weeks)

- LLFuzz’s approach can be extended to 5G
  - 5G and LTE lower layers share a similar design principle

- Developed a minimum LLFuzz-5G version for testing 5G PDCP layer
  - Took 2 weeks with Augment Code
  - Tested Xiaomi K40 Gaming
  - Found 2 unknown bugs, 1 in PDCP and 1 in RRC layer
  - Will provide details in our open-source release after the patch

- Practical challenges for testing 5G
  - Most UEs in our lab do not connect to open-source gNBs (OAI + srsRAN)
  - UEs do not automatically reconnect after a crash

Conclusion

- LLFuzz:
  - Over-the-Air testing framework for cellular baseband lower layers
  - Channel-driven, stateful, configuration-aware testing
  - Specification-guided test case generation
  - Tested 15 basebands from 5 vendors, uncovered 11 previously unknown memory bugs
  - Source code available at: https://github.com/SysSec-KAIST/LLFuzz

- Lessons Learned:
  - Memory corruptions are prevalent in lower layers
  - Memory corruptions in uplink?
  - No security testing requirement in the 3GPP specification
  - Supply-chain issue needs to be resolved
    - There could be many unpatched devices.

SCAN ME

Thank you!