(Almost) Automatic Testing of Cellular Security

Yongdae Kim

SysSec@KAIST

SysSec Lab.

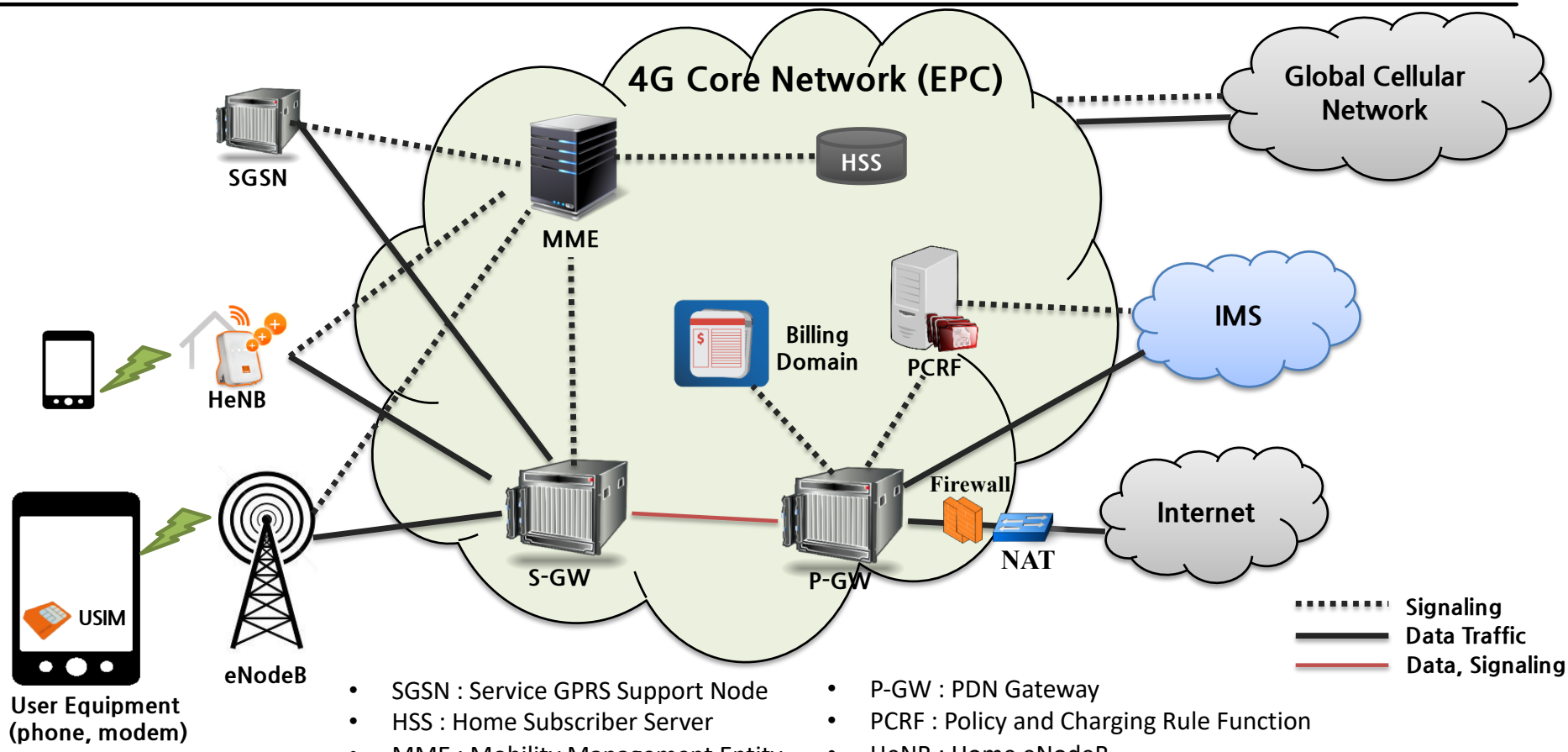
- System Security Lab. @ KAIST, Korea
  - Yongdae Kim
  - Prof @ Electrical Engineering & Information Security

- Research areas: Finding new problems in Emerging Technologies such as Drone, Blockchain, Medical device, Automobiles, Cellular, …
  - Software vulnerability (hacking)
  - Physical system security (sensor, hardware Trojan, …)
  - Wireless communication security (Bluetooth, Zigbee, …)
  - Mobile network security (privacy, abuse, …)

Cellular Security Publications (Selected)

- Location leaks on the GSM Air Interface, NDSS'12
- Gaining Control of Cellular Traffic Accounting by Spurious TCP Retransmission, NDSS' 14
- Breaking and Fixing VoLTE: Exploiting Hidden Data Channels and Mis-implementations, CCS'15
- When Cellular Networks Met IPv6: Security Problems of Middleboxes in IPv6 Cellular Networks, EuroS&P'17
- GUTI Reallocation Demystified: Cellular Location Tracking with Changing Temporary Identifier, NDSS'18
- Peeking over the Cellular Walled Gardens - A Method for Closed Network Diagnosis - , TMC 2018
- Touching the Untouchables: Dynamic Security Analysis of the LTE Control Plane, S&P'19
- Hidden Figures: Comparative Latency Analysis of Cellular Networks with Fine-grained State Machine Models, HotMobile'19
- Hiding in Plain Signal: Physical Signal Overshadowing Attack on LTE, Usenix Sec'19
- BaseSpec: Comparative Analysis of Baseband Software and Cellular Specifications for L3 Protocols, NDSS'21
- Watching the Watchers: Practical Video Identification Attack in LTE Networks, Usenix Sec'22
- DoLTEst: In-depth Downlink Negative Testing Framework for LTE Devices, Usenix Sec'22

4G LTE Cellular Network Overview



- SGSN : Service GPRS Support Node
- HSS : Home Subscriber Server
- MME : Mobility Management Entity
- S-GW : Serving Gateway
- P-GW : PDN Gateway
- PCRF : Policy and Charging Rule Function
- HeNB : Home eNodeB
- EPC : Evolved Packet Core

Why do we need cellular security testing?

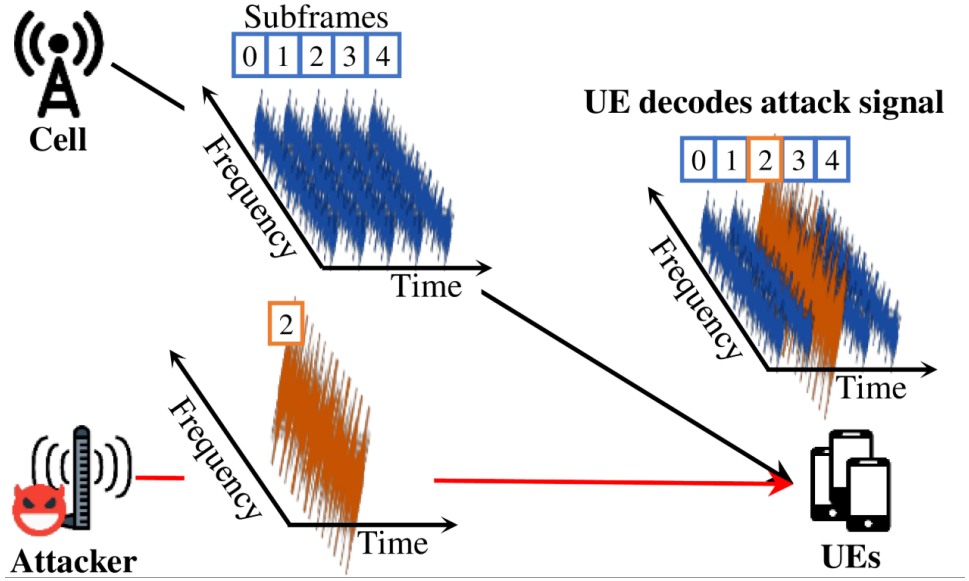
- New Generation (Technology) every 10 year
  - New Standards, Implementation, and Deployment ➔ New vulnerabilities
- Many unpatched design vulnerabilities ➔ SS7, Broadcast channel, …
- Cellular networks are different for each manufacturer and operator
  - Therefore, vulnerabilities are different
- Complicated and huge standards ➔ Hard to implement correctly
  - Leave many implementation details for vendors ➔ Bugs
- Almost no security testing ➔ Only conformance testing
- Walled Garden
  - Carriers (smartphone vendors) don't talk to each other.
  - Carriers don't admit vulnerabilities. ➔ illegal in many countries

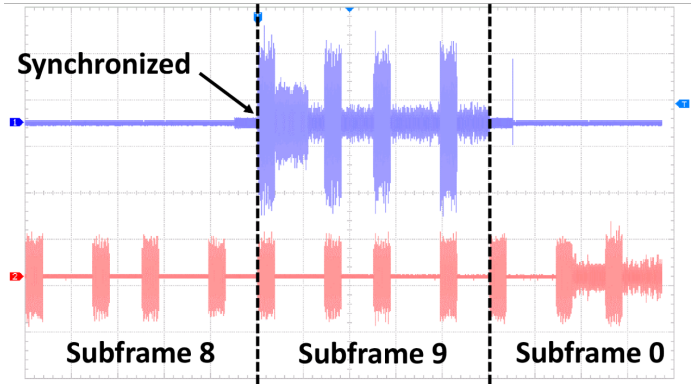
Insecure Standard

Fake CMAS broadcast attack

Signal Overshadowing: SigOver Attack

- Signal injection attack exploits broadcast messages in LTE
  - Broadcast messages in LTE have never been integrity protected!
- Transmit time- and frequency-synchronized signal





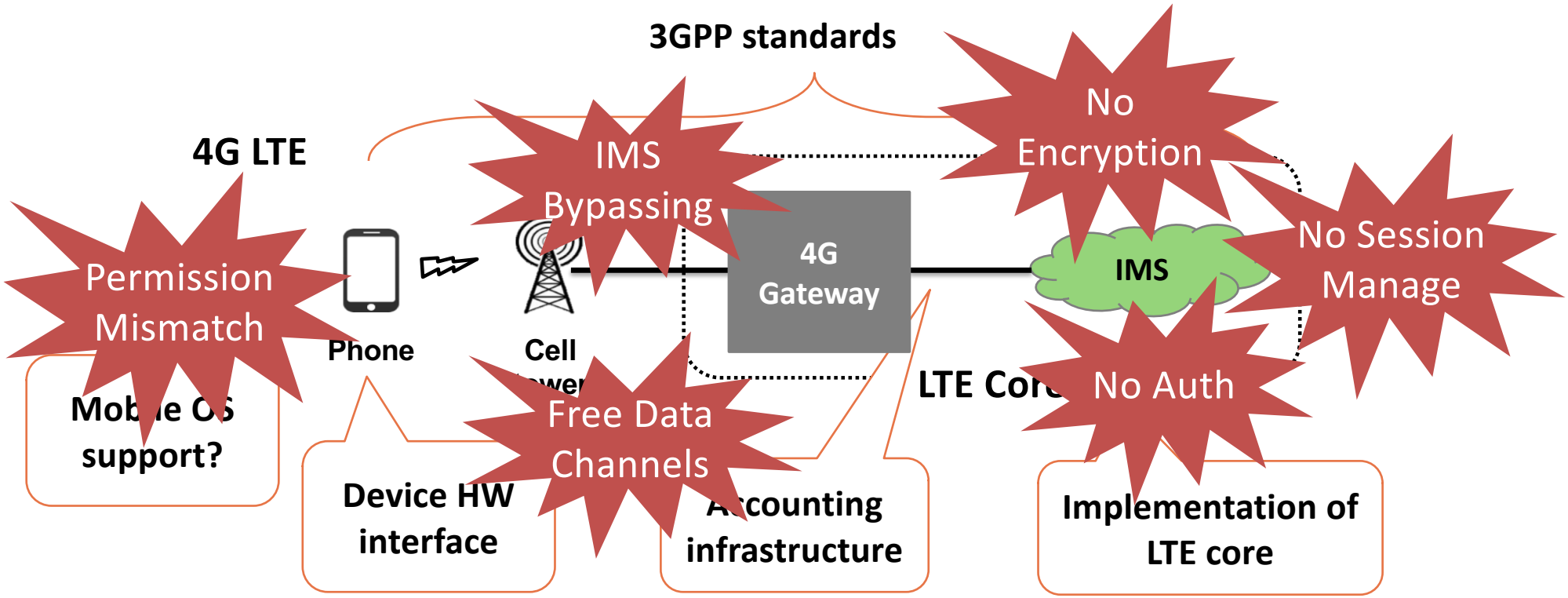
Demonstration of Signal Injection attack

DATA RESTRICTIONS

Security of New Systems

VoLTE makes cellular network more complex

- Let's check potential attack vectors newly introduced in VoLTE



Free Data Channels	Free Channel	US-1	US-2	KR-1	KR-2	KR-3
Using VoLTE Protocol	SIP Tunneling	✓	✓	✓	✓	✓
	Media Tunneling	✓	✓	✓	✓	✓
Direct Communication	Phone to Phone	✓	✗	✓	✗	✗
	Phone to Internet	✗	✓	✓	✗	✗

Weak Point	Vulnerability	US-1	US-2	KR-1	KR-2	KR-3	Possible Attack
IMS	No SIP Encryption						Message manipulation
	No Voice Data Encryption						Wiretapping
	No Authentication						Caller Spoofing
	No Session Management						Denial of Service on Core Network
4G-GW	IMS Bypassing						Caller Spoofing
Phone	Permission Mismatch	Vulnerable for all Android					Denial of Service on Call, Overbilling

 : Vulnerable  : Secure

Discussion

www.kb.cert.org

CERT | Software Engineering

Vulnerability No

Advisory

DATABA

Vulne

Voice

Original R

CWE-732

CWE-284

CWE-287

CWE-384

Elevation Of Privilege Vulnerability in Telephony

A vulnerability in the Telephony component that can enable a local malicious application to pass unauthorized data to the restricted network interfaces, potentially impacting data charges. It could also prevent the device from receiving calls as well as allowing an attacker to control

Acknowledgements

We would like to thank these researchers for their contributions:

- Abhishek Arya, Oliver Chang and Martin Barbella, Google Chrome Security Team: CVE-2015-6608
- Daniel Micay (daniel.micay@copperhead.co) at Copperhead Security: CVE-2015-6609
- Dongkwan Kim of System Security Lab, KAIST (dkay@kaist.ac.kr): CVE-2015-6614
- Hongil Kim of System Security Lab, KAIST (hongilk@kaist.ac.kr): CVE-2015-6614
- Jack Tang of Trend Micro (@jacktang310): CVE-2015-6611
- Peter Pi of Trend Micro: CVE-2015-6611
- Natalie Silvanovich of Google Project Zero: CVE-2015-6608
- Qidan He (@flanker_hqd) and Wen Xu (@antlr7) from KeenTeam (@K33nTeam, http://k33nteam.org/): CVE-2015-6612
- Seven Shen of Trend Micro: CVE-2015-6610

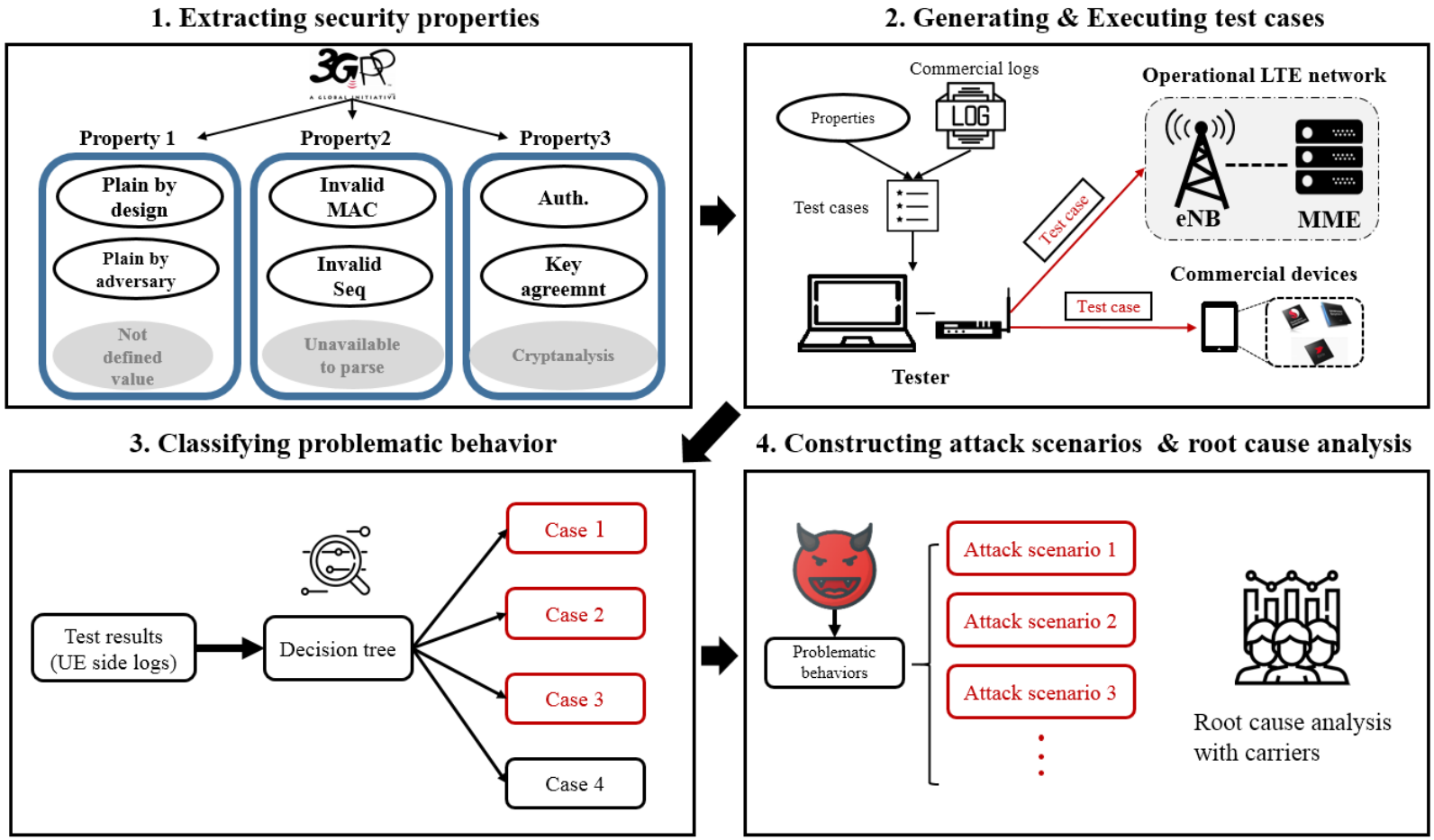
Fuzzing LTE Core and Baseband

Fundamental Problems in cellular network

- Description of standard (3GPP) is ambiguous
  - The 3GPP specifications are based on natural language
  - Standard leave implementation (exact behavior) details to the vendors
  - There are conformance test specs…
    - But, no security testing specs

- Mobile network operators & vendors rarely communicate with each other
  - Different carriers with different device vendors suffer from different vulnerabilities

LTEFuzz


1. Extracting security properties
Property 1
Property2
Property3
Plain by design
Plain by adversary
Not defined value
Invalid MAC
Invalid Seq
Unavailable to parse
Auth.
Key agreemnt
Cryptanalysis
2. Generating & Executing test cases
Commercial logs
Operational LTE network
Properties
LOG
Test cases
Test case
eNB
MME
Commercial devices
Test case
Tester
3. Classifying problematic behavior
Test results (UE side logs)
Decision tree
Case 1
Case 2
Case 3
Case 4
4. Constructing attack scenarios & root cause analysis
Problematic behaviors
Attack scenario 1
Attack scenario 2
Attack scenario 3
Root cause analysis with carriers

Specification issues

Vendor issues

Test messages	Direction	Property 1-1	Property 2-1 (I)	Property 2-2 (R)	Property 3	Affected component
NAS						
Attach request (IMSI/GUTI)	UL	-	DoS	DoS	DoS	-	Core network (MME)
Detach request (UE originating detach)	UL	-	DoS [1]	DoS	DoS	-	Core network (MME)
Service request	UL	-	-	B	Spoofing	-	Core network (MME)
Tracking area update request	UL	-	DoS	DoS	FLU and DoS	-	Core network (MME)
Uplink NAS transport	UL	-	SMS phishing and DoS	SMS phishing and DoS	SMS replay	-	Core network (MME)
PDN connectivity request	UL	B	B	DoS	DoS	-	Core network (MME)
PDN disconnect request	UL	-	B	DoS	selective DoS	-	Core network (MME)
Attach reject	DL	DoS [2]	DoS [3]	-	-	-	Baseband
Authentication reject	DL	DoS [4]	-	-	-	-	Baseband
Detach request (UE terminated detach)	DL	-	DoS [4]	-	-	-	Baseband
EMM information	DL	-	Spoofing [5]	-		-	Baseband
GUTI reallocation command	DL	-	B	B	ID Spoofing	-	Baseband
Identity request	DL	Info. leak [6]	B	B	Info. leak	-	Baseband
Security mode command	DL	-	B	B	Location tracking [4]	-	Baseband
Service reject	DL	-	DoS [3]	-	-	-	Baseband
Tracking area update reject	DL	-	DoS [3]	-	-	-	Baseband
RRC						
RRCConnectionRequest	UL	DoS and con. spoofing	-	-	-	-	Core network (eNB)
RRCConnectionSetupComplete	UL	Con. spoofing	-	-	-	-	Core network (eNB)
MasterInformationBlock	DL	Spoofing	-	-	-	-	Baseband
Paging	DL	DoS [4] and Spoofing	-	-	-	-	Baseband
RRCConnectionReconfiguration	DL	-	MitM	DoS	B	-	Baseband
RRCConnectionReestablishment	DL	-	Con. spoofing	-	-	-	Baseband
RRCConnectionReestablishmentReject	DL		DoS			-	Baseband
RRCConnectionReject	DL	DoS	-	-	-	-	Baseband
RRCConnectionRelease	DL	DoS [2]	-	-	-	-	Baseband
RRCConnectionSetup	DL	Con. spoofing	-	-	-	-	Baseband
SecurityModeCommand	DL	-	B	B	B	MitM	Baseband
SystemInformationBlockType1	DL	Spoofing [4]	-	-	-	-	Baseband
SystemInformationBlockType 10/11	DL	Spoofing [4]	-	-	-	-	Baseband
SystemInformationBlockType12	DL	Spoofing [4]	-	-	-	-	Baseband
UECapabilityEnquiry	DL	Info. leak	-	Info. leak	Info. leak	-	Baseband

Attacks exploiting MME

- Result of dynamic testing against different MME types
  - Carrier 1: MME1, MME2, Carrier2: MME3 (MME1 & MME3: the same vendor)

Exploited NAS Messages	Implications		
	MME_1	MME_2	MME_3
Attach Request	DoS (**P**, **I**, **R**)	×	DoS (**P**, **I**, **R**)
TAU Request	DoS (**P**, **I**, **R**)	×	DoS (**I**), False location update (**R**)
Uplink NAS Transport	DoS (**P**, **I**), SMS phishing (**R**)	SMS phishing (**P**, **I**, **R**)	-
PDN Connectivity Request	DoS (**I**)	×	DoS, DosS (**R**)
PDN Disconnect Request	DoS (**I**), DosS (**R**)	×	DosS (**R**)
Detach Request	DoS (**P**, **R**)	DoS (**P**, **I**, **R**)	DoS (**P**, **I**, **R**)

DosS: Denial of selective Service, **P:** Plain, **I:** Invalid MAC, **R:** Replay

Negative Testing of Core and Basebands

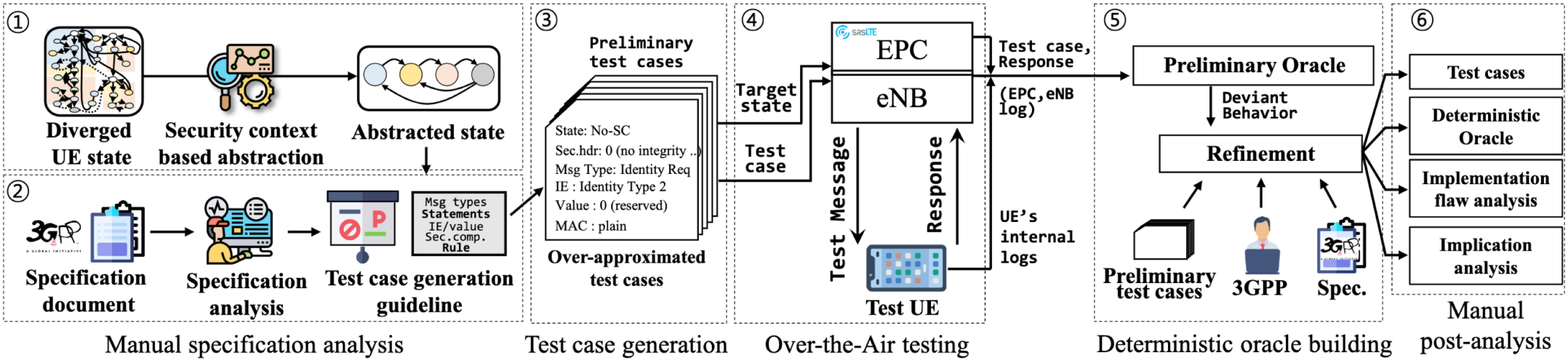
Negative Testing

- Conformance testing ➔ check if valid messages are correctly handled

- Negative testing?
  - check if invalid or prohibited messages are appropriately handled
  - Among 993 test scenarios in conformance spec, only 14 cases are negative.

  - Challenges
    - How do we enumerate violating cases?
    - UE/Network state dependence
    - Spec is difficult to understand ➔ Oracle?
    - Baseband/UE implementation diversity

DoLTEst


①
Diverged UE state
Security context based abstraction
Abstracted state
②
Specification document
Specification analysis
Test case generation guideline
Msg types
Statements
IE/value
Sec.comp.
Rule
③
Preliminary test cases
State: No-SC
Sec.hdr: 0 (no integrity ..)
Msg Type: Identity Req
IE : Identity Type 2
Value : 0 (reserved)
MAC : plain
Over-approximated test cases
④
EPC
eNB
Target state
Test case
Test Message
Response
Test UE
Test case, Response
(EPC,eNB log)
UE's internal logs
⑤
Preliminary Oracle
Deviant Behavior
Refinement
Preliminary test cases
3GPP
Spec.
⑥
Test cases
Deterministic Oracle
Implementation flaw analysis
Implication analysis
Manual post-analysis
Manual specification analysis
Test case generation
Over-the-Air testing
Deterministic oracle building

1	iPhone 6	Apple	Qualcomm	MDM9625	7.21.00 / 7.80.04
2	iPhone 8	Apple	Intel	XMM 7480	4.02.01
3	iPhone XS	Apple	Intel	XMM 7560	1.03.08
4	iPhone 12 Pro	Apple	Qualcomm	Snapdragon X55	1.62.11
5	Y9	Huawei	HiSilicon	Kirin 659	21C60B269S003C000
6	P10 Lite	Huawei	HiSilicon	Kirin 658	21C60B268S000C000
7	P10	Huawei	HiSilicon	Kirin 960	21C30B323S003C000
8	Mate 10 Pro	Huawei	HiSilicon	Kirin 970	21C10B551S000C000
9	P20 pro	Huawei	HiSilicon	Kirin 970	21C20B369S007C000
10	Mate 20 pro	Huawei	HiSilicon	Kirin 980	21C10B687S000C000
11	X401	LG	Mediatek	MT6750	MOLY.LR11.W1552.MD.TC01.LVSF.SP.V1.P22
12	X6	LG	Mediatek	Helio P22 MT6762	MOLY.LR12A.R3.TC01.PIE.SP.V1.P10.T12
13	K50	LG	Mediatek	Helio P22 MT6762	MOLY.LR12A.R3.TC01.PIE.SP.V1.P26
14	G6	LG	Qualcomm	MSM8996 Snapdragon 821	MPSS.TH.2.0.1.c3.1-00024-M8996FAAAANAZM-1.142344.1.143233.1
15	V35 ThinQ	LG	Qualcomm	SDM845 Snapdragon 845	MPSS.AT.4.0.c2.9-00057-SDM845_GEN_PACK-1
16	G7 ThinQ	LG	Qualcomm	SDM845 Snapdragon 845	MPSS.AT.4.0.c2.9-00088-SDM845_GEN_PACK-1.299473
17	G8 ThinQ	LG	Qualcomm	SM8150 Snapdragon 855	MPSS.HE.1.0.c4-00104-SM8150_GEN_PACK-1
18	V50	LG	Qualcomm	SM8150 Snapdragon 855	MPSS.HE.1.5.c4-00270.1-SM8150_GENFUSION_PACK-1.215515.14
19	Oppo find X	OPPO	Qualcomm	SDM845 Snapgragon 845	Q_V1_P14,Q_V1_P14
20	Galaxy S4	Samsung	Qualcomm	MSM8974 Snapdragon 800	E330KKKUCNG5
21	Galaxy S5	Samsung	Qualcomm	MSM8974AC Snapdragon 801	G900VVRU1ANI2
22	Galaxy S5 A	Samsung	Qualcomm	APQ8084 Snapdragon 805	G906LKLU1CPK2
23	Galaxy Note5	Samsung	Samsung	Exynos 7 (7420)	N920SKSU2DQH2
24	Galaxy S6	Samsung	Samsung	Exynos 7 (7420)	G920SKSU3EQC9
25	Galaxy Note FE	Samsung	Samsung	Exynos 8 (8890)	N935JJJU4CTJ1
26	Galaxy Note8	Samsung	Samsung	Exynos 9 (8895)	N950NKOU4CRH2
27	Galaxy S8	Samsung	Qualcomm	MSM8998 Snapdragon 835	G950U1UES5CSB2
28	Galaxy Note9	Samsung	Samsung	Exynos 9 (9810)	N960NKOU3DSLA
29	Galaxy S10	Samsung	Samsung	Exynos 9 (9820)	G977NKOU2BTA2 / G977NKOU4DK1
30	Galaxy S10	Samsung	Qualcomm	SM8150 Snapdragon 855	G977UVRS3YSJK
31	Galaxy A31	Samsung	Mediatek	Helio P65 MT6768	A315NKOU1BUA1
32	Galaxy S20	Samsung	Qualcomm	SM8250 Snapdragon 865	G981NKSU1CTKD
33	Galaxy A71	Samsung	Samsung	Exynos 9 (980)	A716SKSU1ATF4 / A716SKSU3BTL2
34	Galaxy Note20	Samsung	Qualcomm	SM8250 Snapdragon 865	N986NKSU1CUC9
35	Redmi 5	Xiaomi	Qualcomm	SDM450 Snapdragon 450	MPSS.TA.2.3.c1-00522-8953_GEN_PACK-1_V042
36	Redmi note 4x	Xiaomi	Qualcomm	MSM8953 Snapdragon 625	953_GEN_PACK-1.122638.1.123338.1
37	Mi Max 3	Xiaomi	Qualcomm	SDM636 Snapdragon 636	AT32-00672-0812_2359_46aa9a7
38	Mi 5S	Xiaomi	Qualcomm	MSM8996 Snapdragon 821	TH20c1.9-0612_1733_9fe7ce8
39	Mi Mix 2	Xiaomi	Qualcomm	MSM8998 Snapdragon 835	AT20-0608_2116_6c4a86b
40	Black Shark	Xiaomi	Qualcomm	SDM845 Snapdragon 845	00888-SDM845_GEN_PACK-1.163713.1
41	POCOphone F1	Xiaomi	Qualcomm	SDM845 Snapdragon 845	AT4.0.c2.6-144-1008_1436_e3055ba
42	ZTE Blade V8 Pro	ZTE	Qualcomm	MSM8953 Snapdragon 625	-8953_GEN_PACK-1.79091.1.79899.1
43	ZTE Axon 7	ZTE	Qualcomm	MSM8996 Snapdragon 820	TH.2.0.c1.9-00104-M8996FAAAANAZM

Baseband Fingerprinting

Baseband	Device	Message				
		#1	#2	#3	#4	#5
Intel	Apple iPhone XS	·	·	·	A_5	·
Qualcomm	Xiaomi Mi Mix 2	·	A_2	A_4	A_5	A_3
Exynos	Samsung Galaxy S10	A_1	·	A_4	A_5	·
MediaTek	LG K50	·	·	A_4	A_6	·
HiSilicon	Huawei Mate 20 Pro	·	A_3	·	A_5	·

Lessons Learned

- Found 26 misimplementations (22 were unknown previously)
- Almost all BBs have misimplementations.
- Not all BB vendors are responsive.
- Patch cycle of BB seems larger than that of other softwares

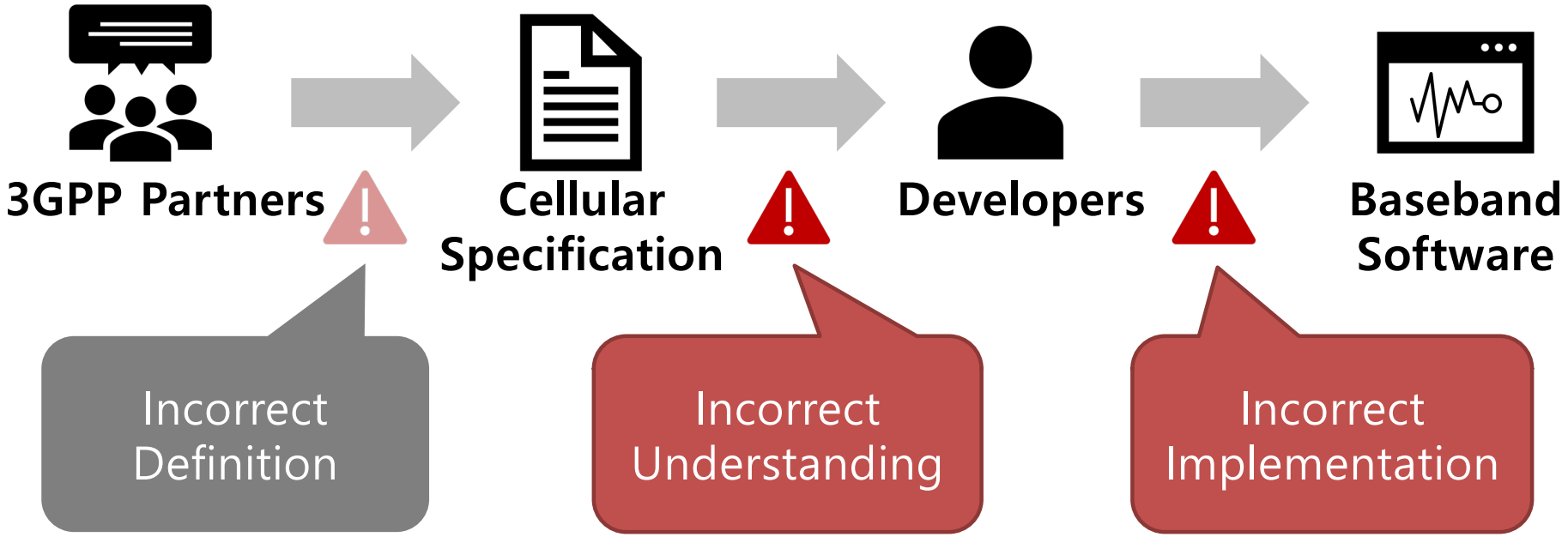
Limitations of Dynamic Testing

- Over-the-air testing is painful.
  - Violating messages often “reboot” BBs.
  - Slow ➔ Testing 1,000 messages takes about 5 hours.
  - Debugging is expensive ➔ expensive DM and SDRs, no memory access
- Hard to find memory bugs
  - No memory access
- Huge manual effort
  - Test case generation, reasoning, reading manual

BaseSpec: Comparative Analysis of Baseband Software and Cellular Specifications

Errors in Protocol Implementation

- Many points of **human errors** in development process



How about directly comparing?

- Software analysis with specification is a common approach
  - Formal verification of software
    - Using manually defined formal specification
  - Protocol specification extraction from binary
    - For malware analysis

➔ Can be applied to **cellular baseband software**?

Challenges

Cellular Specification

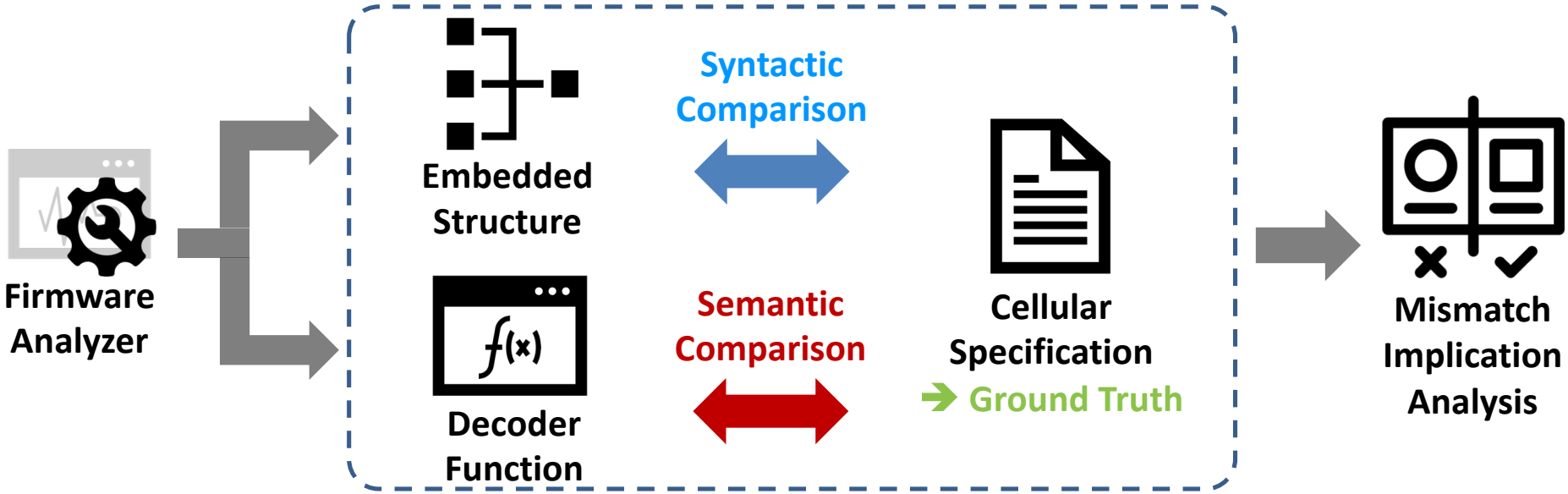
- Large volume of documents
  - Over a hundred documents
  - Each has hundreds of pages
- Mainly written in natural language

Baseband Software

- Obscure embedded system
  - Vendors do not open details
  - Hard to analyze dynamically
- Complex implementation
  - Low-level embedded software
  - Numerous functions (>90K)

Our Approach

- **Comparative analysis** of message structures in baseband and specification
  - Compare embedded message structures with specification
  - Compare logic of decoder function with specification
  - Analyze implication of identified mismatches



Mismatch Results (vendor x)

- Missing Mismatches of mandatory IE & Unknown Mismatches
  - Directly indicate **functional errors** (drop of benign IE / undefined behavior)
- Invalid Mismatches
  - Numerous incorrect length limit / ad-hoc length checkers
  - Can lead to **memory-related bugs**
- Missing optional IEs
  - May not be buggy

9 Error cases
(4 Memory-related including 2 RCEs)

		Missing Mismatch		Unknown Mismatch		Invalid Mismatch	
Models	Total IEs	Mandatory IE	Optional IE	Mandatory IE	Optional IE	Mandatory IE	Optional IE
Model A	1475	5	189	6	58	94	364
Model B	1475	5	192	6	58	94	361
Model C	1475	5	192	6	58	94	361
Model D	1475	5	203	6	58	94	349
Model E	1475	5	203	6	58	94	349

*IE: Information Element (= message field)

Conclusion

- Spec could be written better.
  - Formally verifiable?
  - Sample implementation
  - Negative testing (security testing) should be standardized!
- Or use of NLP to understand 3GPP Spec
  - Seems impossible… Too many inconsistencies and ambiguities…
- Emulation of BB seems promising
  - Still requires quite a bit of manual effort
  - HITL BB? Using Avatar? (debugging interface)
- Many design vulnerabilities should be patched as well.

Questions?

- Yongdae Kim
  - email: yongdaek@kaist.ac.kr
  - Home: http://syssec.kaist.ac.kr/~yongdaek
  - Facebook: https://www.facebook.com/y0ngdaek
  - Twitter: https://twitter.com/yongdaek
  - Google "Yongdae Kim"