PennState

CSE 597: Security of Emerging Technologies
Module: Automata Learning

Kai Tu and Syed Md Mukit Rashid
Prof. Syed Rafiul Hussain

Systems and Network Security (SyNSec) Lab

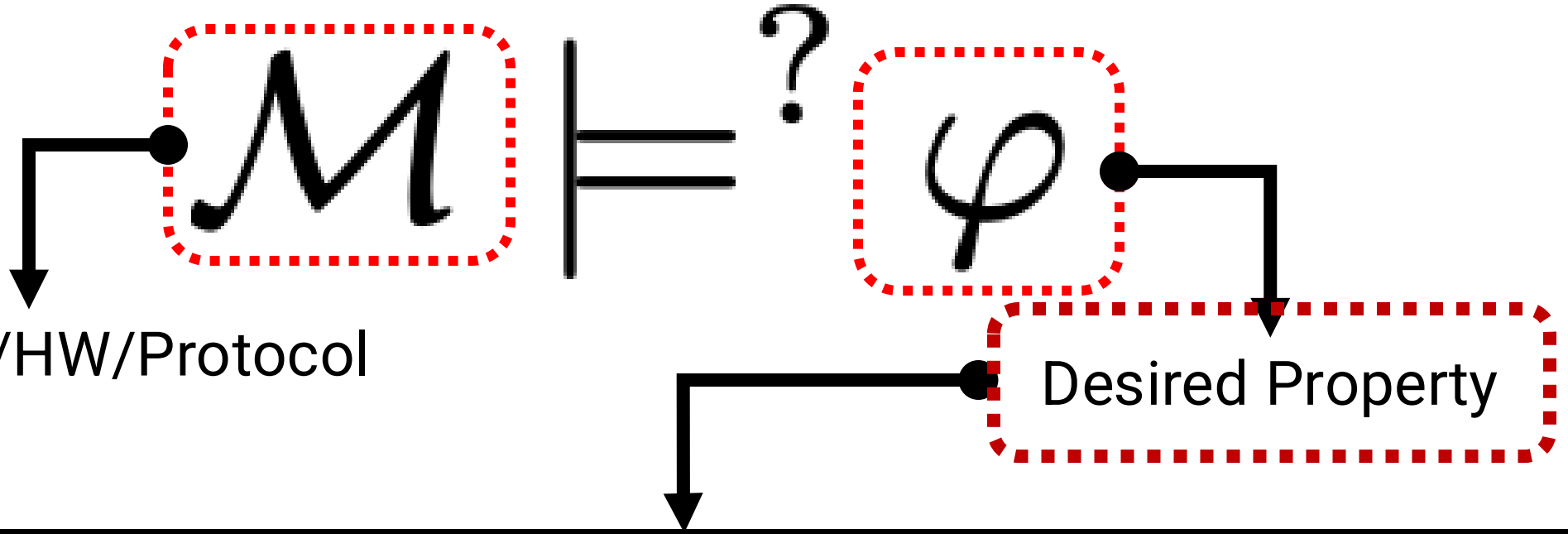
Department of Computer Science and Engineering

Pennsylvania State University

Agenda for Today

- Introduce Model Checking
- Introduce Passive and Active Automata Learning
- Introduce L* algorithm for Automata Learning
- Problem with L* algorithm
- Improving Automata Learning (StateSynth)
- Detecting Vulnerabilities Using Learned Model (DevScan & DevLyzer)
- Attack Demos
- Details of L* algorithm

Model Checking



System correctness Properties (e.g., Turbine speed < 20 rpm)

Security Properties (e.g., Integrity)

Privacy Properties (e.g., Information flow)

Safety Properties (e.g., Reactor temp < 300F)

Model Checking in Practice

Natural Language
System Specification

Manual process

M

Mathematical
Model

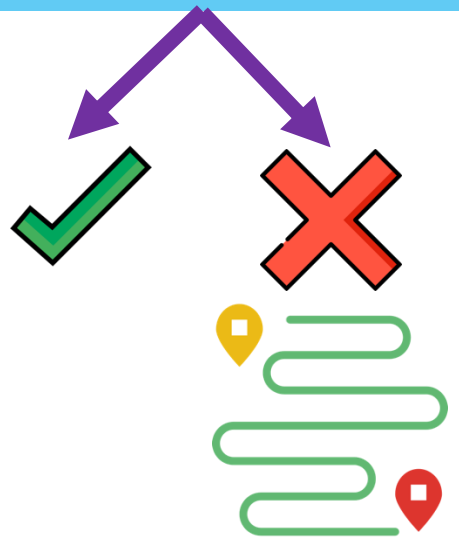
Natural Language
Guarantees

Manual process

ρ

Mathematical
Logic formula

Model Checker



Motivation of Model Learning

What if the system/protocol/HW you want to model check is a black-box system?

Model Checking with Model Learning

System to Model Check

Model Learning

M

Mathematical Model

Model Checker

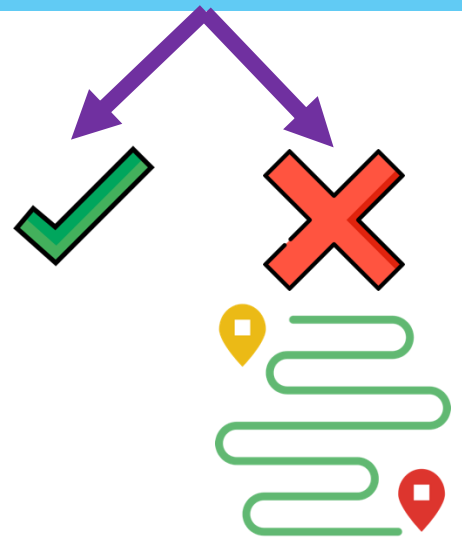
Our Focus

Natural Language Guarantees

Manual process

ρ

Mathematical Logic formula



Automata Learning

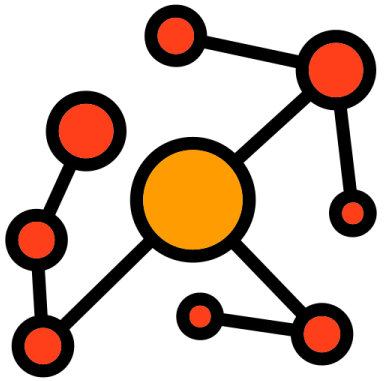
Passive Learning

Active Learning

Passive Automata Learning

System Trace

Passive Learner



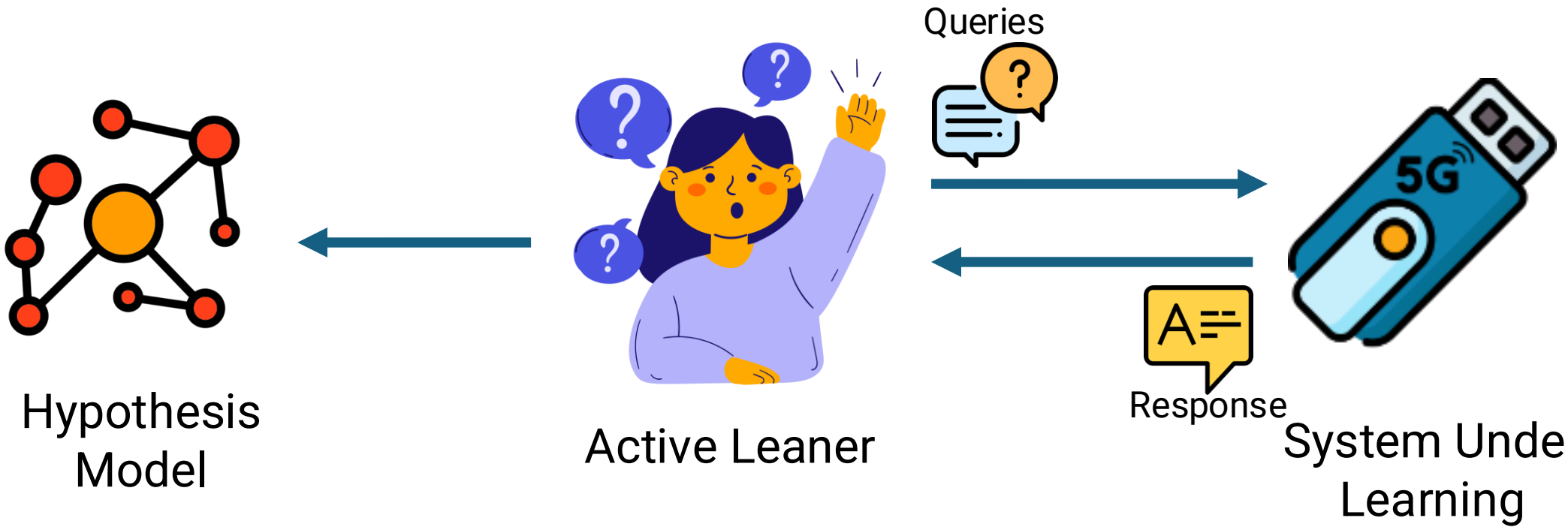
Hypothesis Model

Pros and Cons

- **Non-intrusive methodology** 👍
  Learns exclusively from existing system traces (logs) without requiring active interaction with the target system.

- **Incomplete behavior coverage** 👎
  Model accuracy heavily relies on log quality, limited observed events in logs, potentially leads to missing system behaviors/ states.

Active Automata Learning



Pros and Cons

- **Targeted Exploration** 👍
  Actively queries the system to test hypotheses, enabling strategic discovery of edge cases, rare states.

- **Scalability Limits** 👎
  State-explosion problem: Learning large systems becomes hard due to exponential growth in query volume.

L* Algorithm

Learning Regular Sets from Queries and Counterexamples*

DANA ANGLUIN

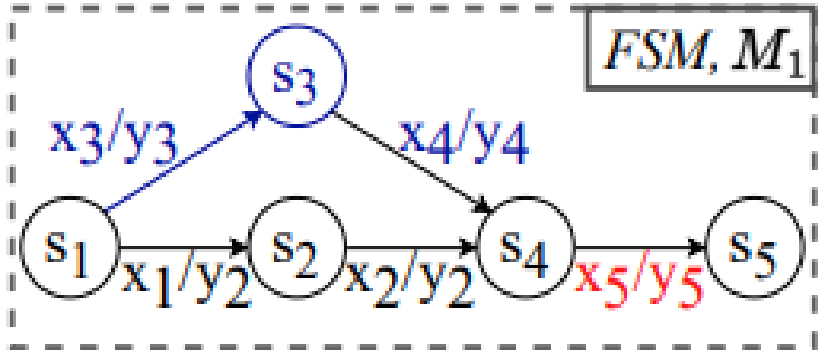
Department of Computer Science, Yale University,
P.O. Box 2158, Yale Station, New Haven, Connecticut 06520

Polynomial time algorithm for correctly learning an unknown regular language (polynomial number of states in the minimized DFA of L and the length of the largest counterexample)

Preconditions for Automata Learning

List of Input Sy
(Alphabe

er Learning (SUL)
inistic behavior



Query Responses Executed
From Initial State

Minimally Adequate Teacher
(MAT) Responding to Queries

Minimally Adequate Teacher (MAT)

Membership queries

$x \in^? \mathcal{L}$

Yes/No

Equivalence queries

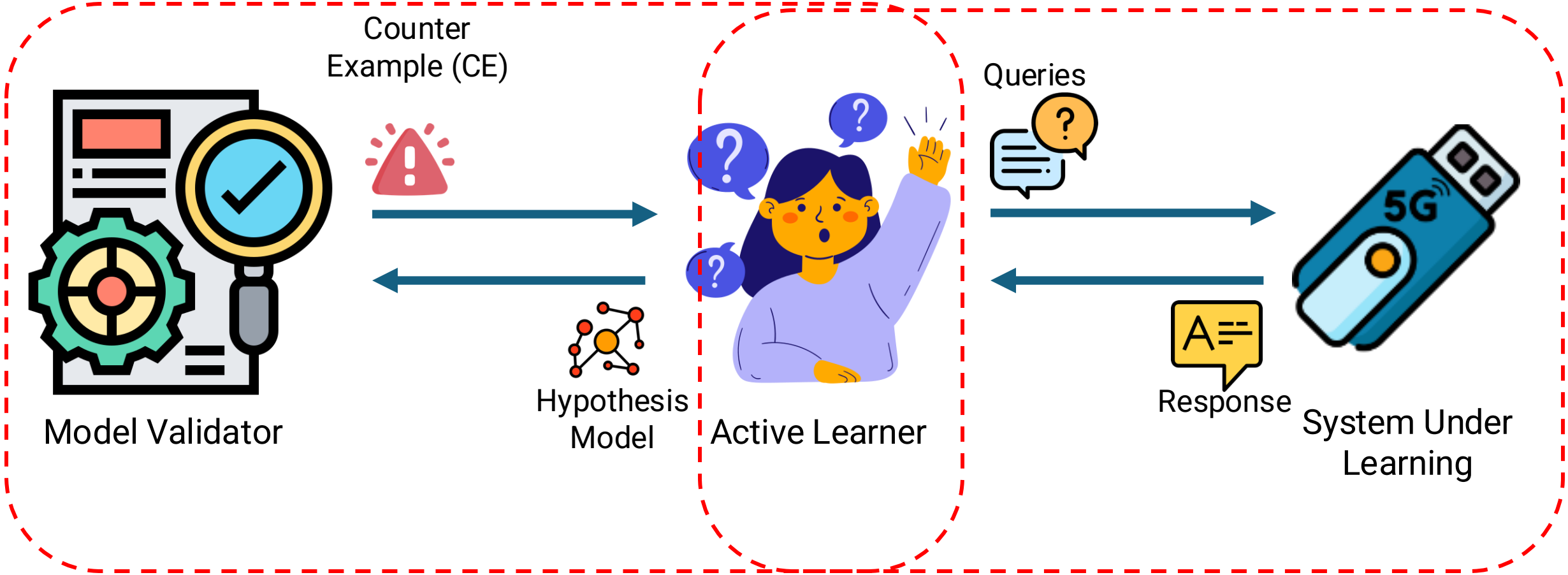
$\mathcal{L}_H \approx^? \mathcal{L}$

<Yes, -> or

<No, σ>

Minimally Adequate Teacher (MAT)

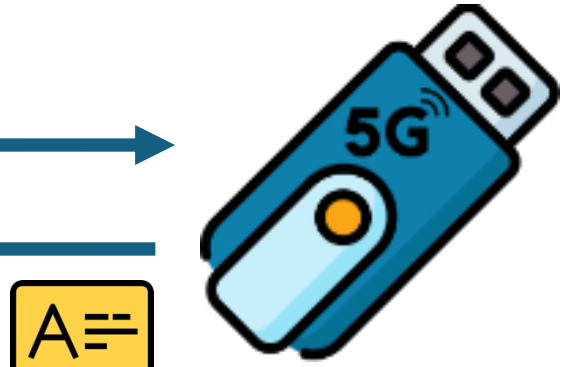
Skeleton of the L* Algorithm



Membership Queries

Active Learner

Queries



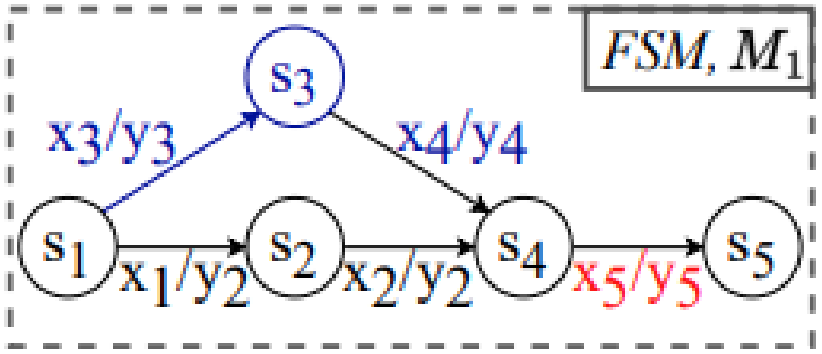
Response

System Under Learning

Hypothesis Model

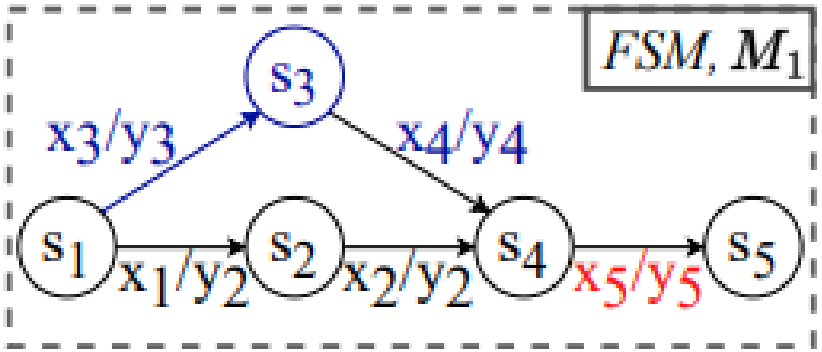
Closed: I know a definite destination state for every input at every state

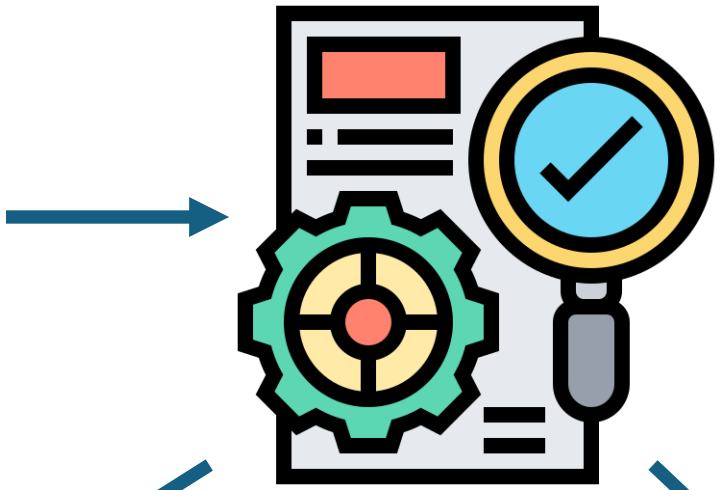
Consistent: However I reach a particular given state, the subsequent behavior is the same



Equivalence Queries

Observational Equivalence





Queries

Response

System Under Learning

Model Validator

Counter Example (CE)

Observational Equivalence

Skeleton of the L* Algorithm

Input: Alphabet ***A*** and MAT ***M***

Output: DFA of ***L***

```
L*(A, M){
        Hypothesis DFA D_H = {}
        While(true)
        {
                Construct hypothesis DFA DH through a series of membership queries
                If D_H is equivalent to L, then return D_H
                else use the obtained counterexample to update D_H
        }
}
```

A Fundamental Problem

To learn an automata that is observationally equivalent to the SUL,
A LOT of queries need to be generated

Device	M	E	Time (min)	# of states	# of transitions
Motorola Nexus 6	3129	21300	37620	21	556
HTC One E9+	8060	42432	77757	35	1172
Samsung Galaxy S6	3097	10612	21111	20	529
HTC Desire 10 Lifestyle	3129	21300	37676	21	560
Huawei Nexus 6P	3129	21300	37450	21	568
Samsung Galaxy S8+	2908	20961	36762	21	554
Google Pixel 3 XL	3110	20501	36345	21	548
Huawei Y5 Prime	8100	44432	80899	35	114
Honor 8X	4623	16813	33011	28	725
Huawei P8lite	6228	7863	21700	34	1054
Xiaomi Mi A1	3105	21045	37191	21	570
Apple iPhone XS	2340	22450	75361	17	448
4G LTE USB Modem	2905	18332	39953	21	562
Fibocom L860-GL	2322	20470	35099	16	430

Table 2: M = Membership and E = Equivalence queries.

Alphabet Size

Time Per Query

Size of State Machine

[1] Hussain, Syed Rafiul, et al. "Noncompliance as deviant behavior: An automated black-box noncompliance checker for 4g lte cellular devices." *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 2021.

PennState

Logic Gone Astray: A Security Analysis Framework for the Control Plane Protocols of 5G Basebands

Kai Tu, Abdullah Al Ishtiaq, Syed Md Mukit Rashid, Yilu Dong, Weixuan Wang, Tianwei Wu, Syed Rafiul Hussain

Systems and Network Security (SyNSec) Lab

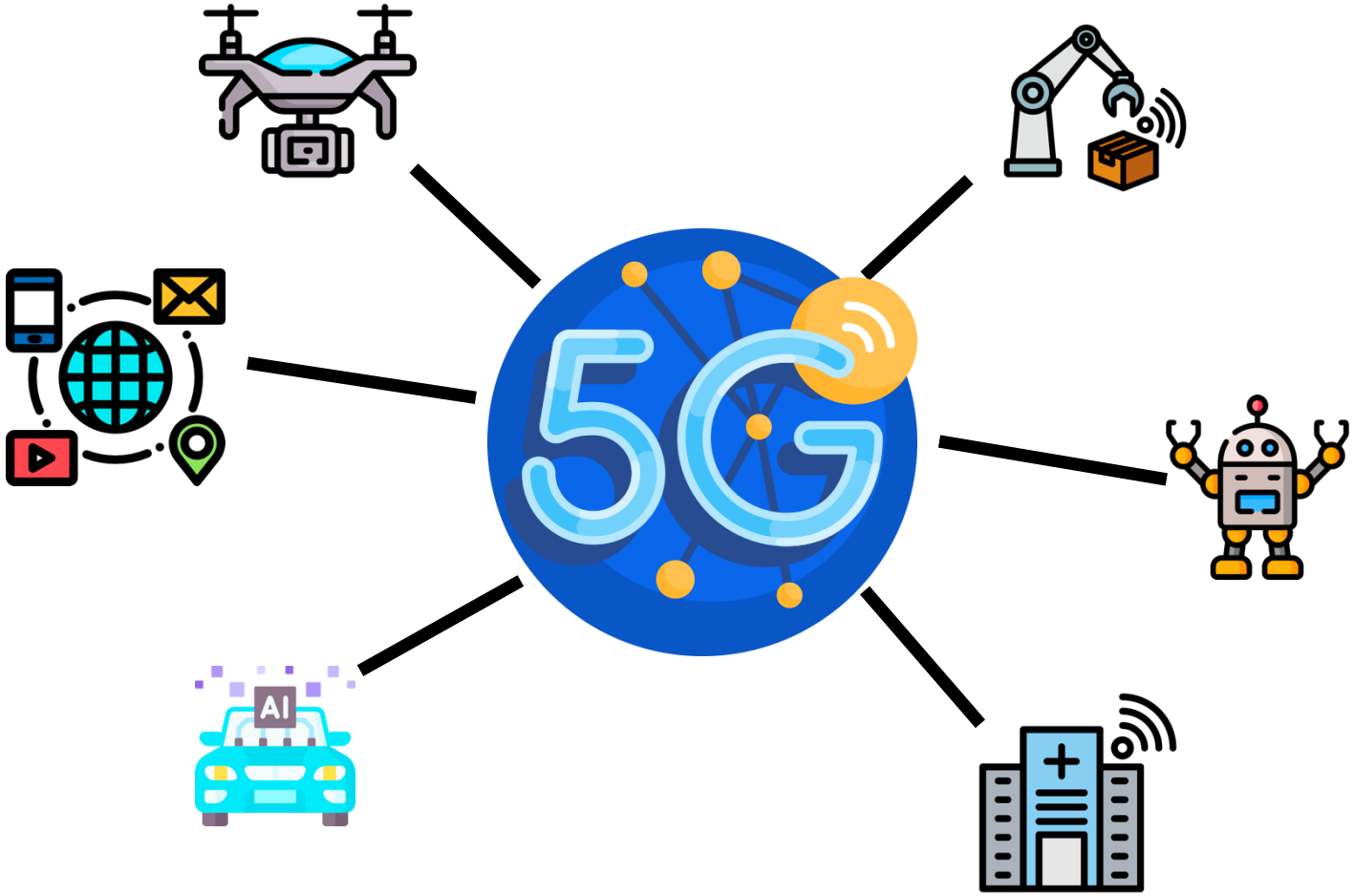
Department of Computer Science and Engineering

Pennsylvania State University

PennState

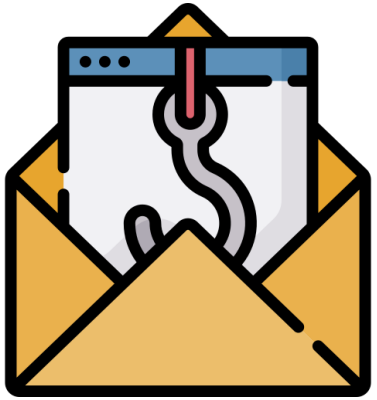
SyNSec

5G Cellular Networks

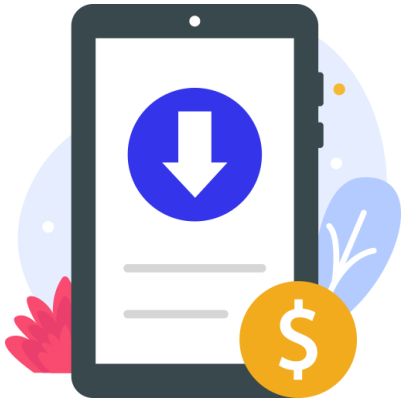


Impacts of Security Policies Violations

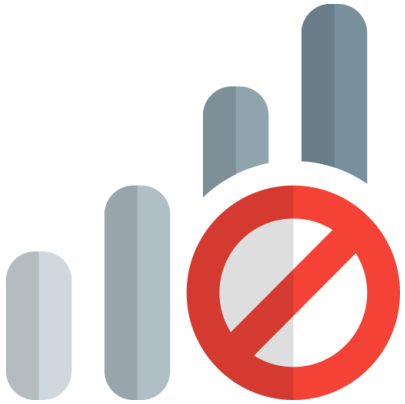
Information Leak



Phishing



Downgrade



Denial-of-Service

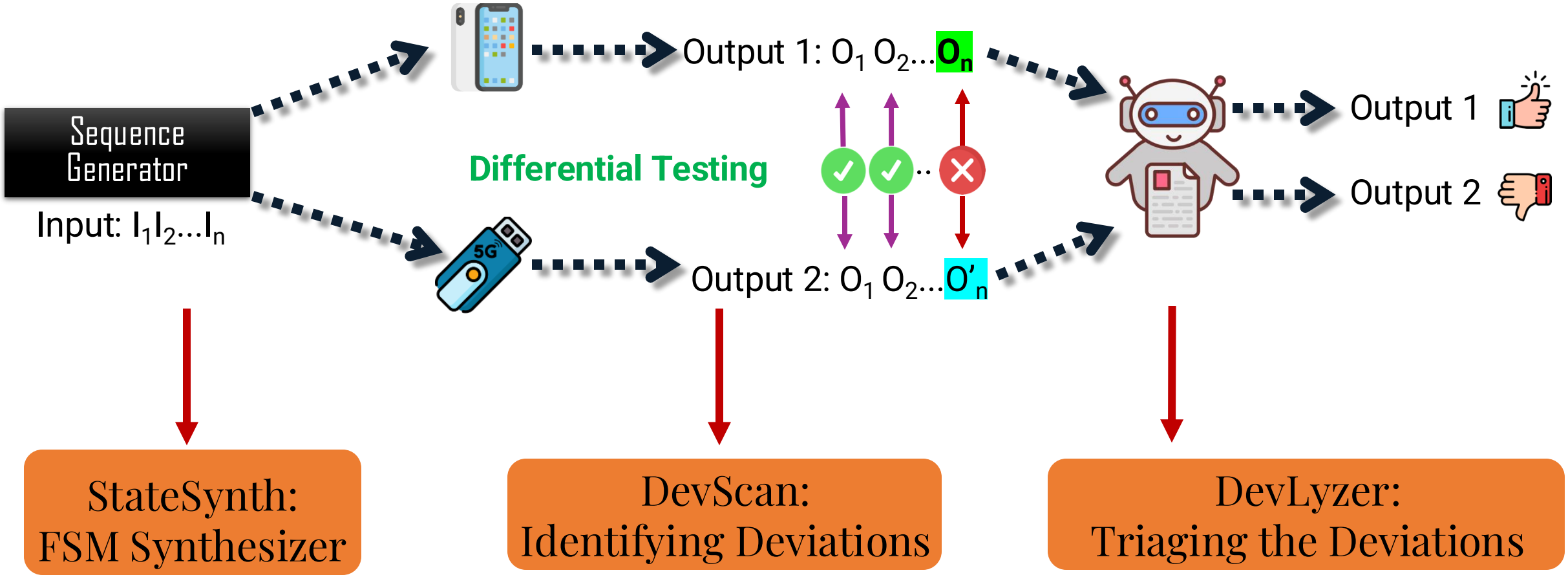
Our Goal

Is it possible to develop an automated framework to efficiently identify security policy violations in 5G UE implementations?

No Comprehensive List of Security Policies

3GPP does not provide such a list which contains a complete set of security policies

High-level Philosophy of Our Approach



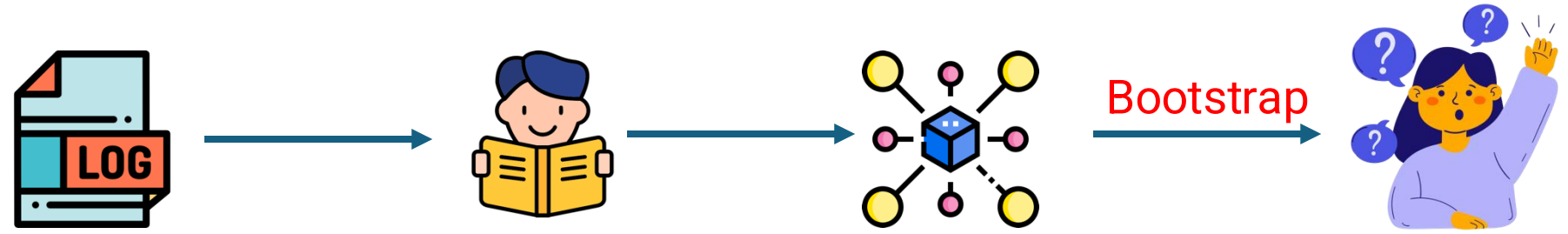
Challenges of Active Automata Learning

Large number of equivalence checking queries are generated in the model validation stage and most of them are not CE.

Active learner, initially does not have any idea about the 5G protocol interactions. It will generate many meaningless queries.

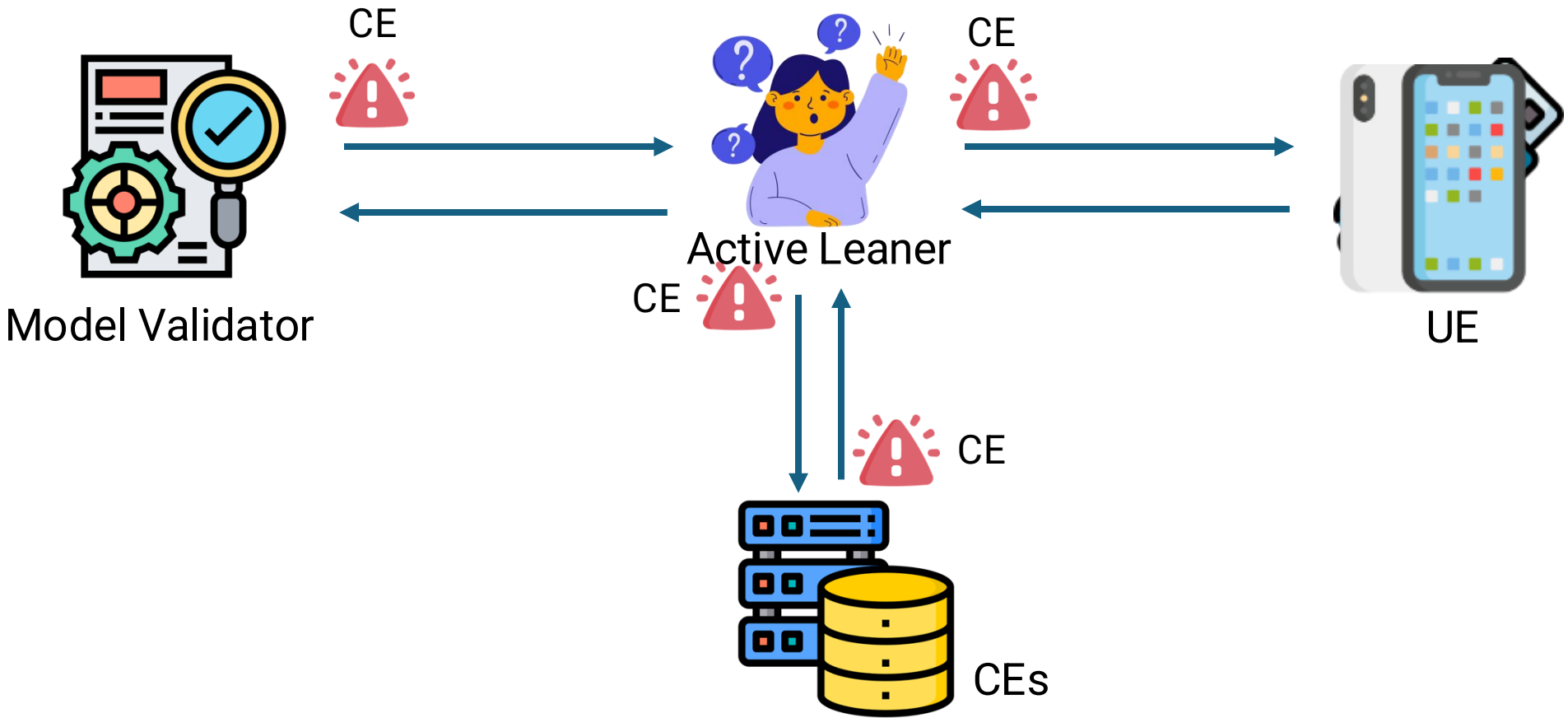
Hybrid Automata Learning

- Synthesize an initial FSM to provide guidance at the beginning of the active learning!

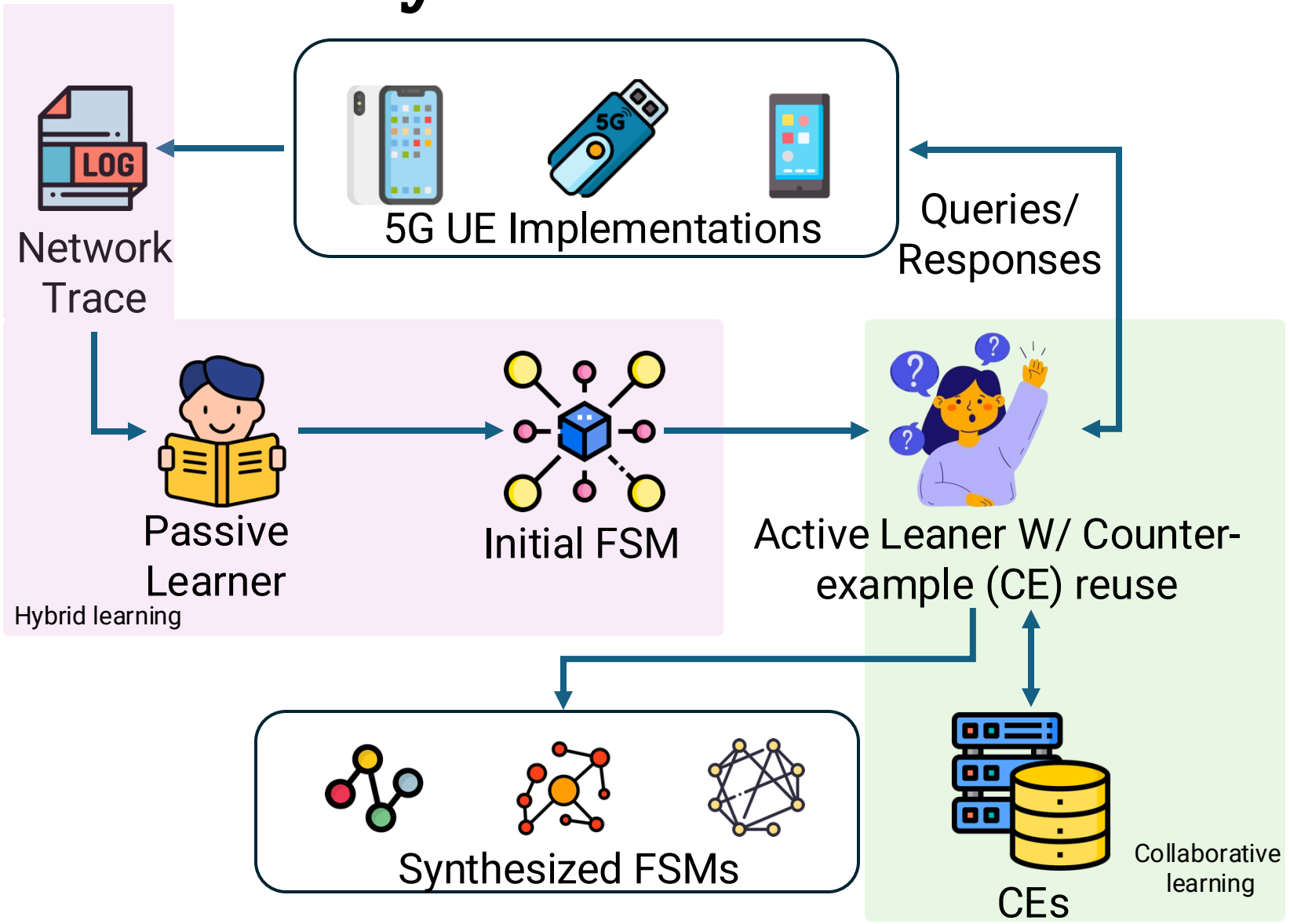


Collaborative Automata Learning

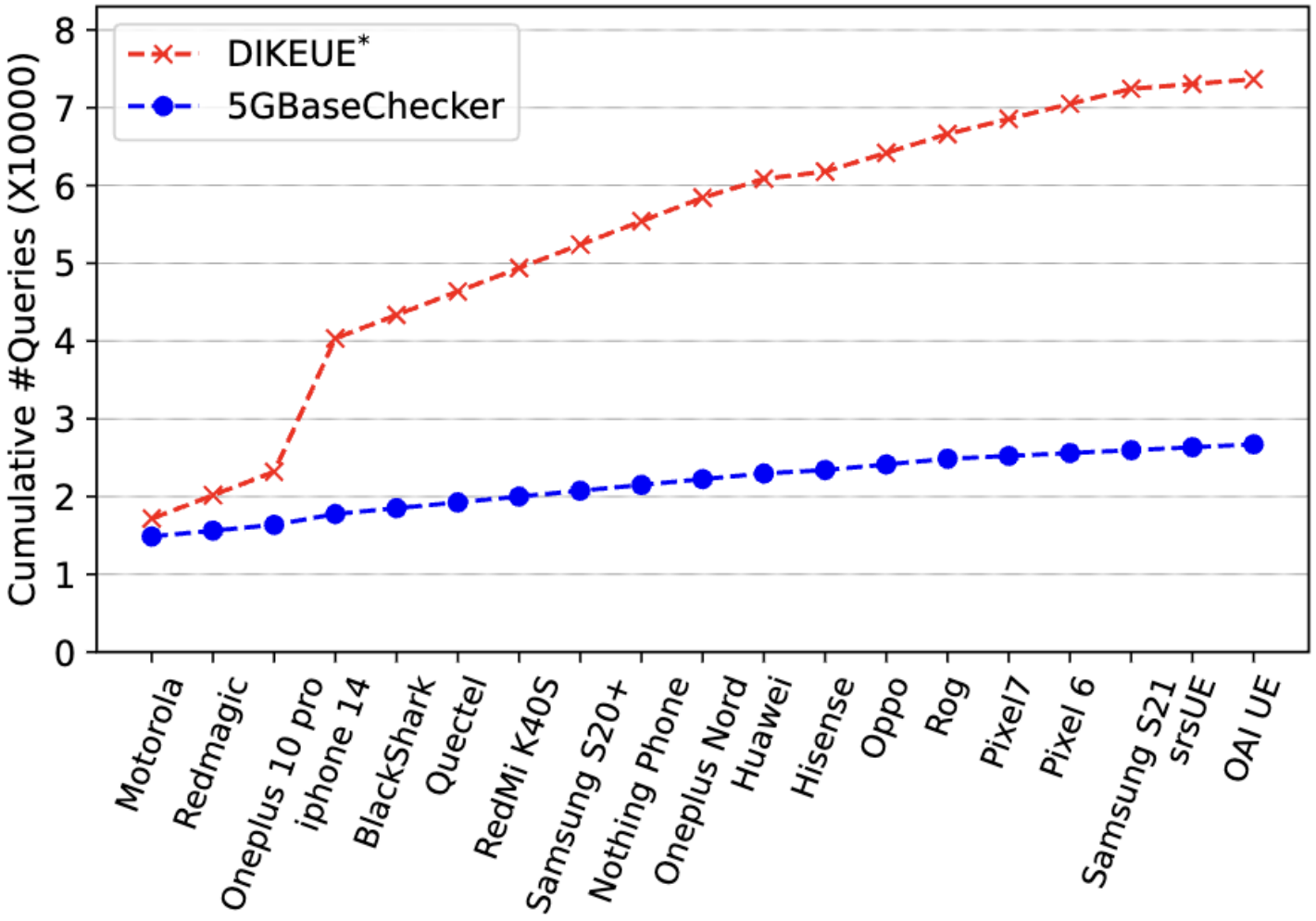
- Since all basebands implement the same protocol, and CEs found during FSM construction of one device are likely to be applicable to other basebands as well.



StateSynth: Workflow



Evaluation of StateSynth

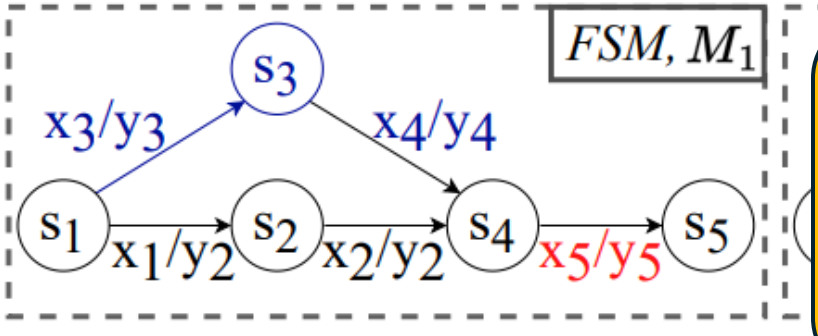


Security Analysis Using Learned Model

$\mathcal{M} \models^{?} \varphi$

Model

Desired Property



Protocols usually do not provide a list that contains a complete set of security policies

Significant manual work to manually analyze all deviations

DevScan

FSM 1

FSM 2

Deviation scanner using symbolic model checker

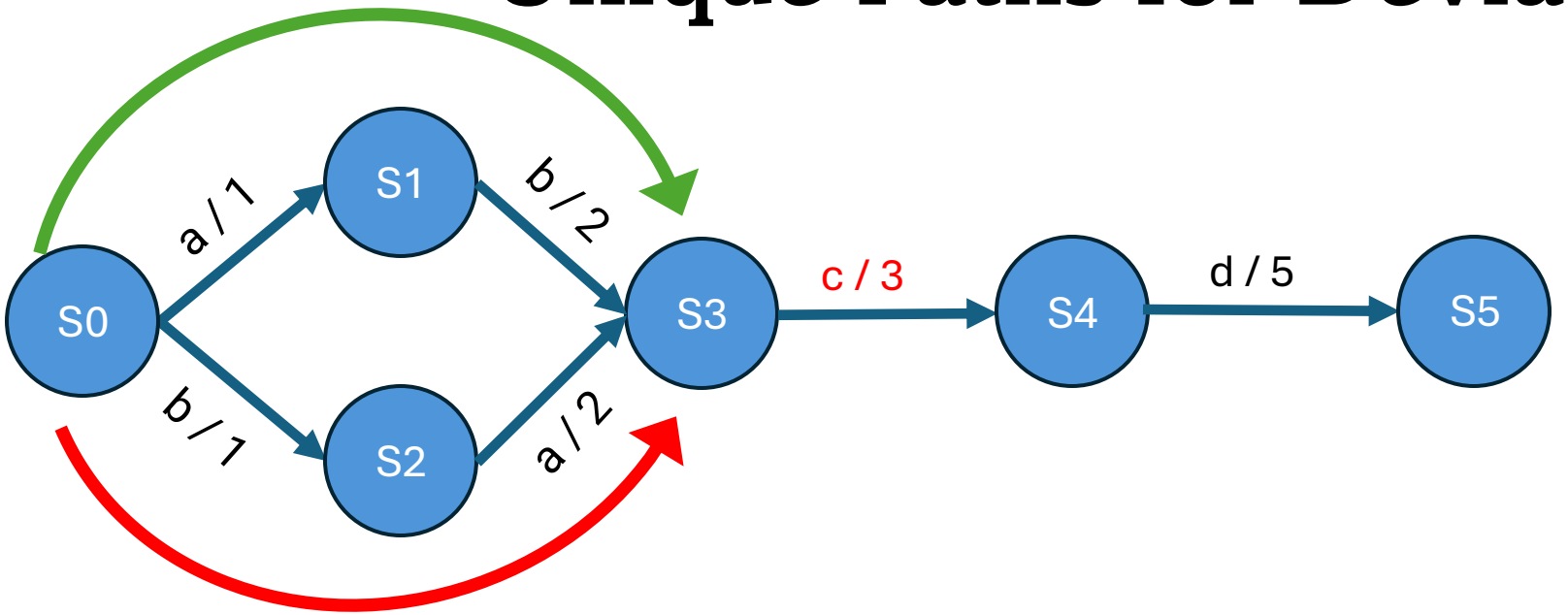
$I_1\ I_2\ I_n\ /\ O_1\ O_2\ O_n$
$I_1\ I_2\ I_n\ /\ O_1\ O_2\ O'_n$

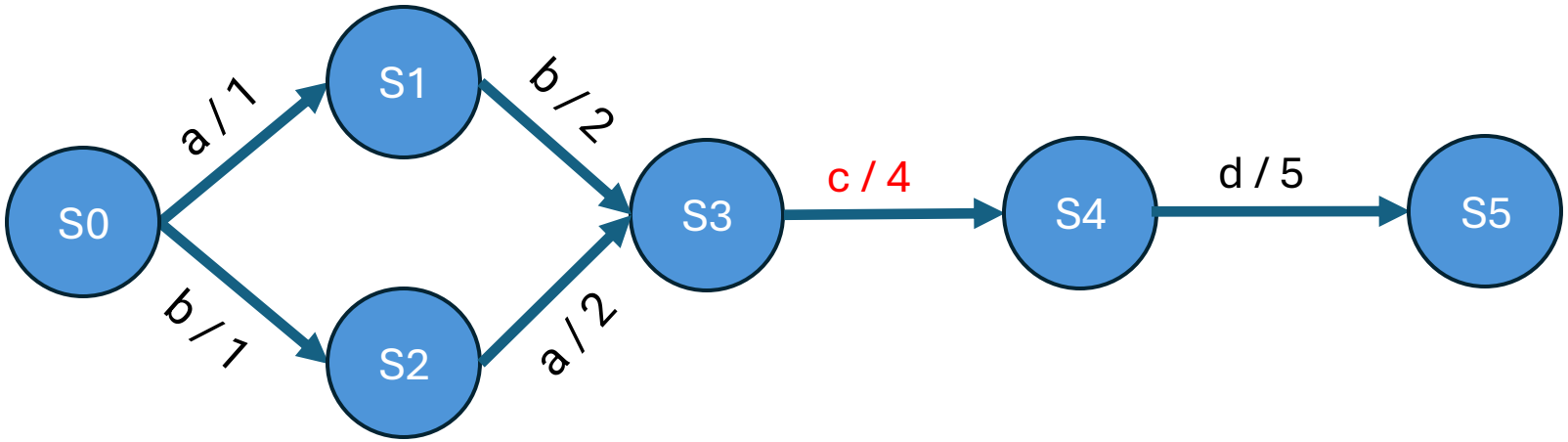
deviations

⚠ Limitation of previous work[1]: prematurely stop their exploration for different variations of a deviation.

[1] Hussain, Syed Rafiul, et al. "Noncompliance as deviant behavior: An automated black-box noncompliance checker for 4g lte cellular devices." *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 2021.

Unique Paths for Deviation



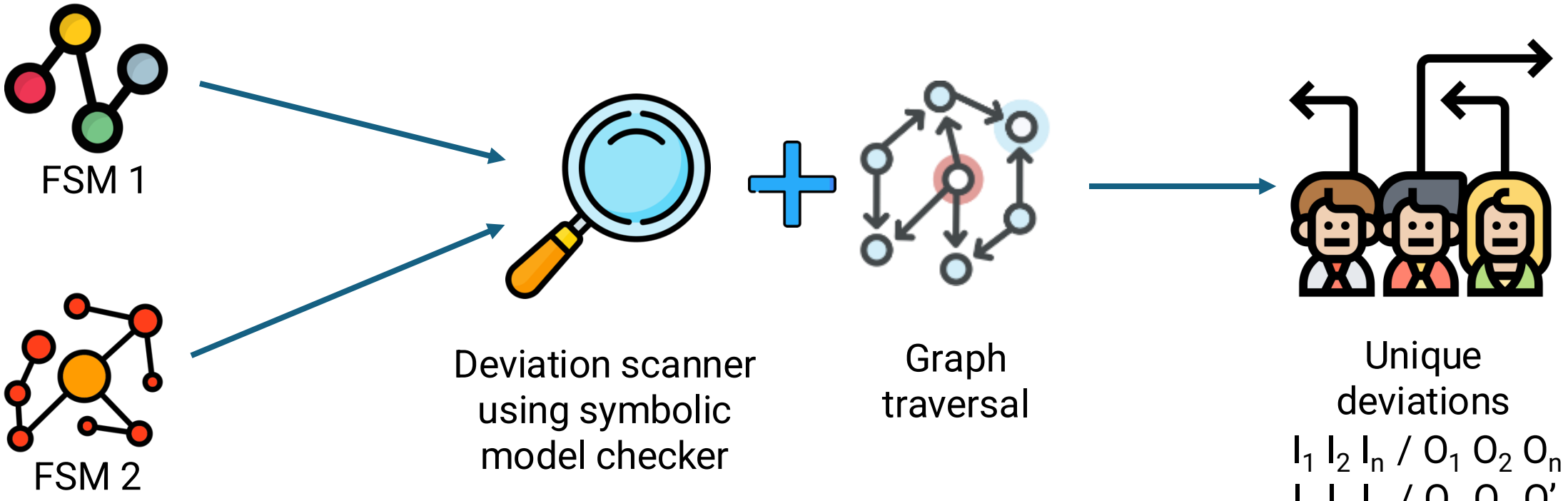


Unique paths for deviation:

Path 1: S0 → S1 → S3

Path 2: S0 → S2 → S3

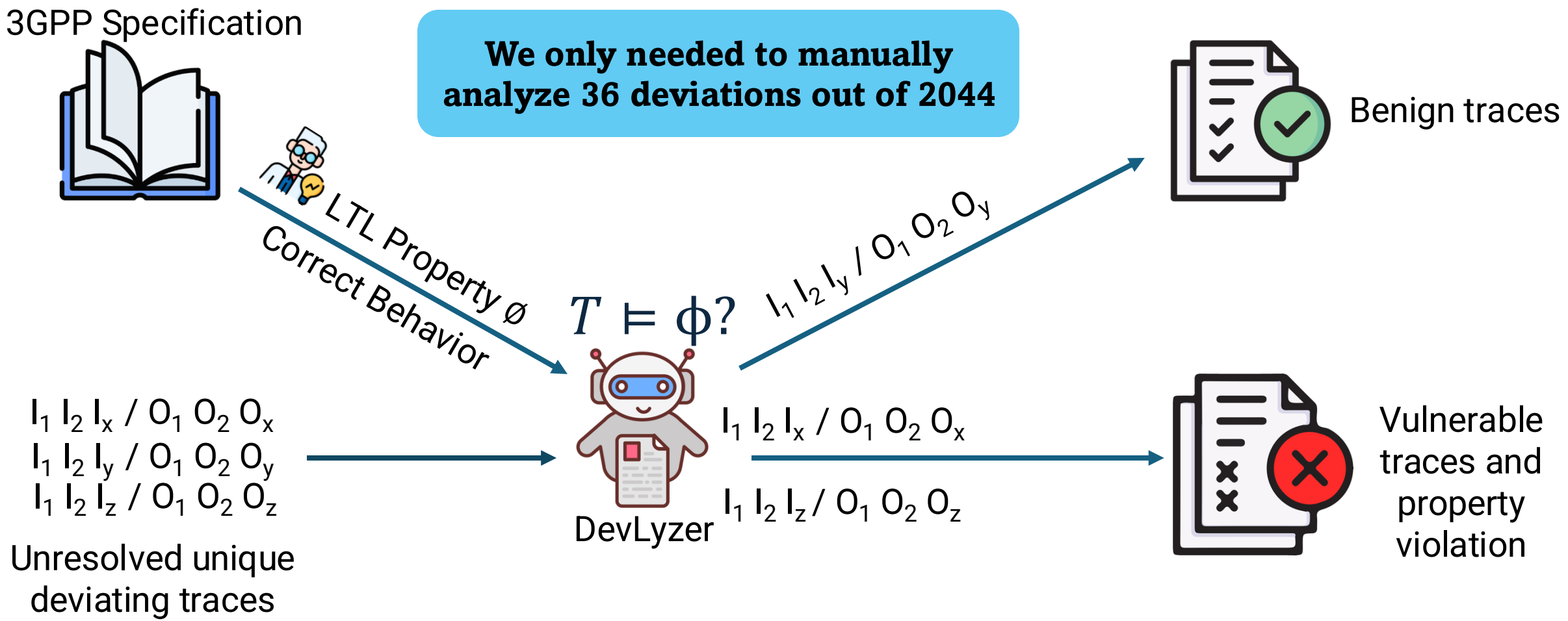
DevScan: Workflow



DevScan: Results

Approach	# Deviations	Avg. time (sec.)
DIKEUE [35]	1325	83
BLEDiff [39]	724	118
`5GBaseChecker`	2044	98

DevLyzer: Workflow



LTL: Linear temporal logic

Evaluation

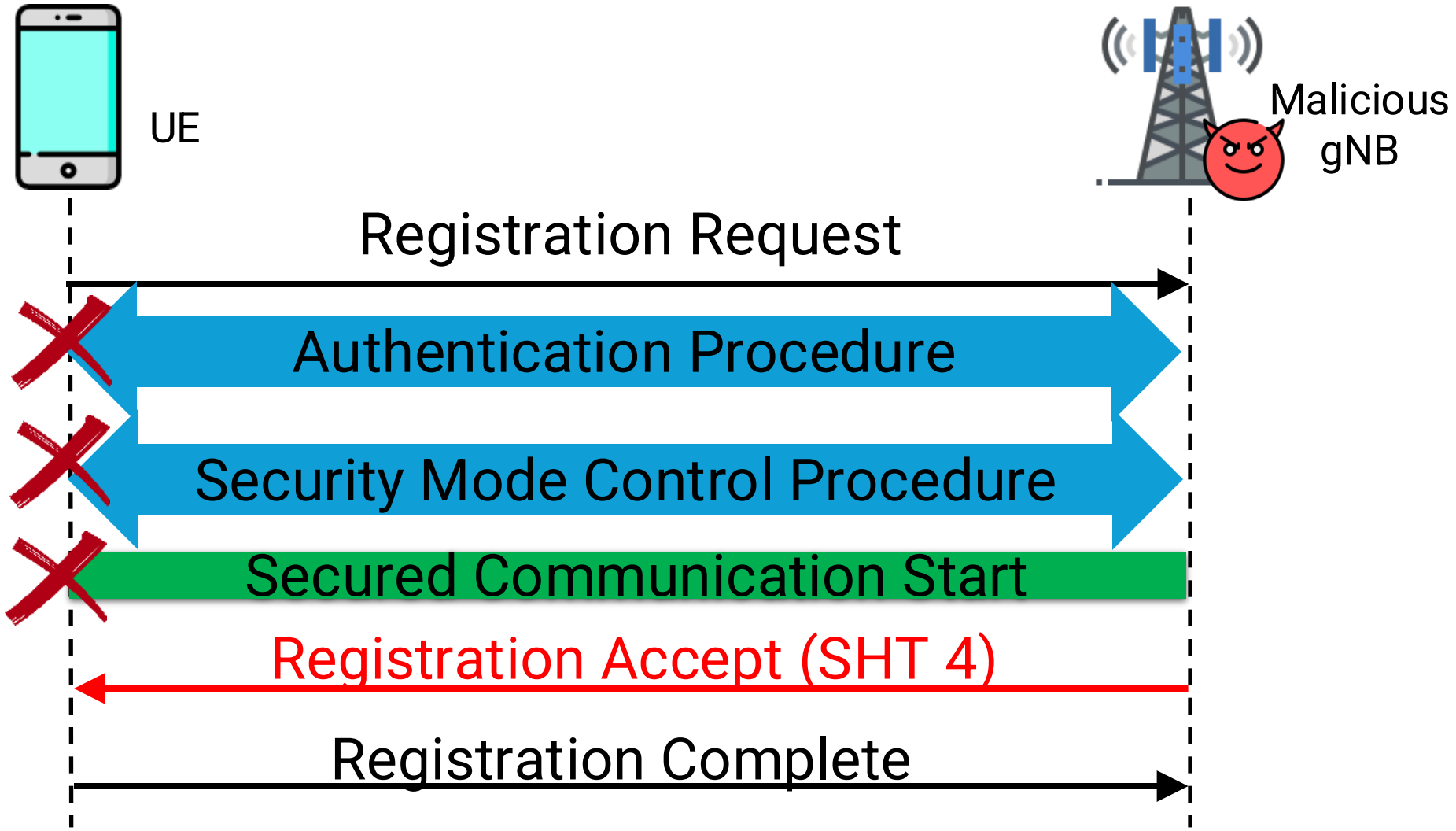
- We tested 17 Commercial Devices from 5 vendors + 2 Open-Source UE Implementations with 5GBaseChecker.

Findings and Impact

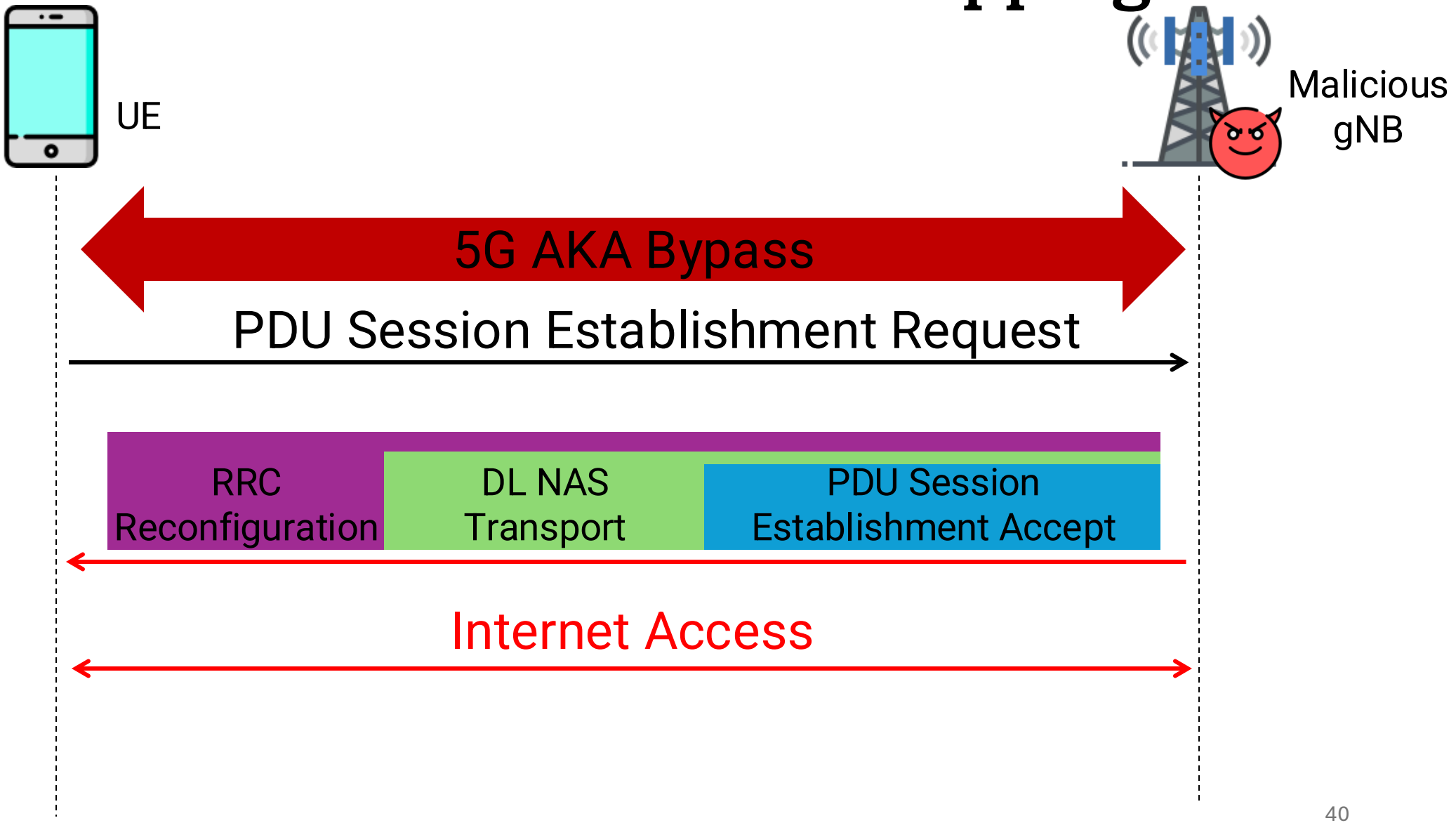
- Uncovered 22 unique issues, 13 could lead to exploitable attacks.
- 12 CVEs assigned and some vendor acknowledgements.
  - CVE-2023-52341, -49928, -50804, -49927, -50803, -52343, -52533, -52534, -52342, -52344; CVE-2024-29152, -28818
- GSMA Mobile Security Research Acknowledgements (CVD-2023-0081)

CVD-2023	0081	Kai Tu, Abdullah Al Ishtiaq, Syed MD Mukit Rashid, Yilu Dong, Weixuan Wang, Tianwei Wu, Syed Rafiul Hussain	Pennsylvania State University

5G AKA Bypass



Internet Traffic Eavesdropping



kai@kai: ~/Desktop/5GBaseChecker_Core

```
NN[internet] IPv4[10.45.0.2] IPv6[] (../src/sm
497)
04/06 20:19:47.410: [upf] INFO: [Added] Number of UPF-Sessions
is now 1 (../src/upf/context.c:178)
04/06 20:19:47.410: [gtp] INFO: gtp_connect()
(../lib/gtp/path.c:60)
04/06 20:19:47.410: [upf] INFO: UE F-SEID[CP:0x1 UP:0x1] APN[in
ternet] PDN-Type[1] IPv4[10.45.0.2] IPv6[] (../src/upf/context.
c:397)
04/06 20:19:47.410: [upf] INFO: UE F-SEID[CP:0x1 UP:0x1] APN[in
ternet] PDN-Type[1] IPv4[10.45.0.2] IPv6[] (../src/upf/context.
c:397)
04/06 20:19:47.410: [gtp] INFO: gtp_connect() [127.0.0.7]:2152
(../lib/gtp/path.c:60)
04/06 20:19:47.411: [amf] WARNING: 0x7f40a981c010 (../src/amf/n
amf-handler.c:83)
04/06 20:19:47.411: [sctp] INFO: sctp_senddata (../lib/sctp/ogs
-sctp.c:73)
04/06 20:19:47.446: [amf] INFO: number of events in queue 1 (..
/src/amf/event.c:106)
04/06 20:19:47.446: [gtp] INFO: gtp_connect() [127.0.0.5]:2152
(../lib/gtp/path.c:60)
04/06 20:19:47.446: [amf] INFO: set e->h.sbi.message (../src/am
f/amf-sm.c:511)
```

Attacker Terminal

kai@kai: ~/Desktop/clean/openairinter...

```
 CellGroup
[NR_MAC]   Activating RRC processing timer fo
ms
[NR_MAC]   (949.2) De-activating RRC processi
16
[NR_MAC]   Modified rnti 4a16 with CellGroup
[NR_MAC]   Added new CBRA process for UE RNTI 4a16 with initial
 CellGroup
[NR_RRC]   Receive RRC Reconfiguration Complete message UE 4a16
[PDCP]   ../../../openair2/LAYER2/nr_pdcp/nr_pdcp_oai_api.c:860
:add_drb_am: warning DRB 1 already exist for UE ID/RNTI 18966,
do nothing
[PDCP]   ../../../openair2/LAYER2/nr_pdcp/nr_pdcp_oai_api.c:add
_drb:911: added DRB for UE ID/RNTI 18966
[RLC]   ../../../openair2/LAYER2/nr_rlc/nr_rlc_oai_api.c:761:ad
d_drb_am: DRB 1 already exists for UE with RNTI 4a16, do nothin
g
[RLC]   ../../../openair2/LAYER2/nr_rlc/nr_rlc_oai_api.c:nr_rlc
_add_drb:860: added DRB to UE with RNTI 0x4a16
[NR_RRC]   [gNB 0] Frame  0 : Logical Channel UL-DCCH, Received
 NR_RRCReconfigurationComplete from UE rnti 4a16, reconfiguring
 DRB 1
[NR_RRC]   msg index 0, pdu_sessions index 0, status 2, xid 0):
 nb_of_pdusessions 1,  pdusession_id 5, teid: 1166204179
 [NR_RRC]   NGAP_PDUSESSION_SETUP_RESP: sending the message
[NGAP]   pdusession_setup_resp_p: pdusession ID 5, gnb_addr 127
.0.0.5, SIZE 4
[PDCP]   discard NR PDU rcvd_count=6, entity->rx_deliv 10,sdu_i
n_list 0
```

Attacker Terminal

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Trace Capture

ip.addr == 127.0.0.5 && dns || ngap || gtp

No.	Protocol	Info
961	NGAP	NGSetupRequest
963	NGAP	NGSetupResponse
3169	NGAP/N…	InitialUEMessage, Registration request, Registration request
3249	NGAP/N…	DownlinkNASTransport, Identity request
3255	NGAP/N…	SACK (Ack=1, Arwnd=106496) , UplinkNASTransport, Identity response
3340	NGAP/N…	DownlinkNASTransport, Registration accept
3351	NGAP/N…	SACK (Ack=2, Arwnd=106496) , UplinkNASTransport, Registration complete
3465	NGAP/N…	UplinkNASTransport, UL NAS transport, PDU session establishment request
3602	NGAP/N…	SACK (Ack=3, Arwnd=106496) , PDUSessionResourceSetupRequest, DL NAS transport, PDU s
3608	NGAP	SACK (Ack=4, Arwnd=106496) , PDUSessionResourceSetupResponse

Attack Message

Frame (118 bytes) Bitstring tvb (4 bytes) Unaligned OCTET STRING

Authentication Bypassed!!!

SM-G991B

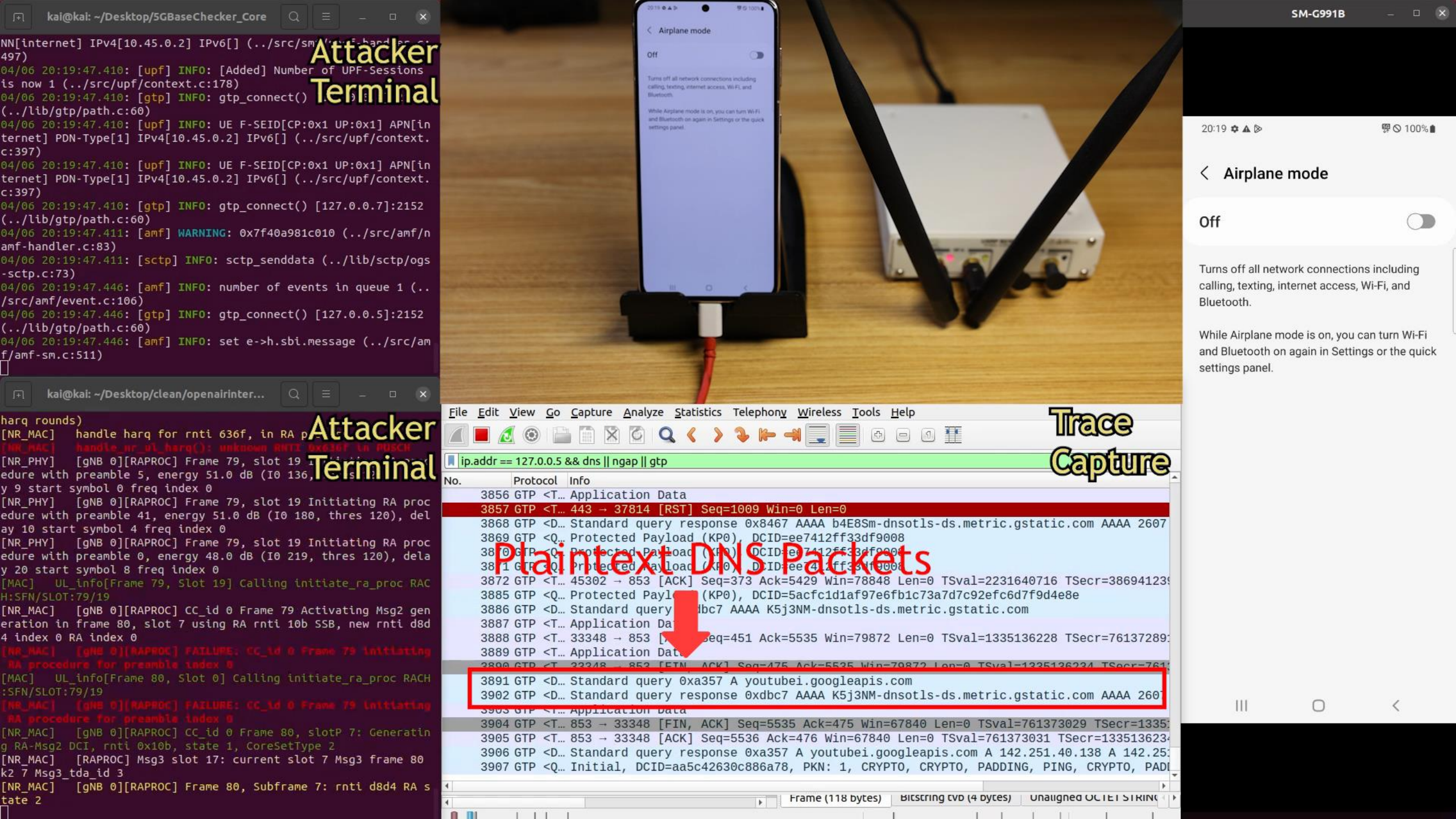
20:19 100%

Airplane mode

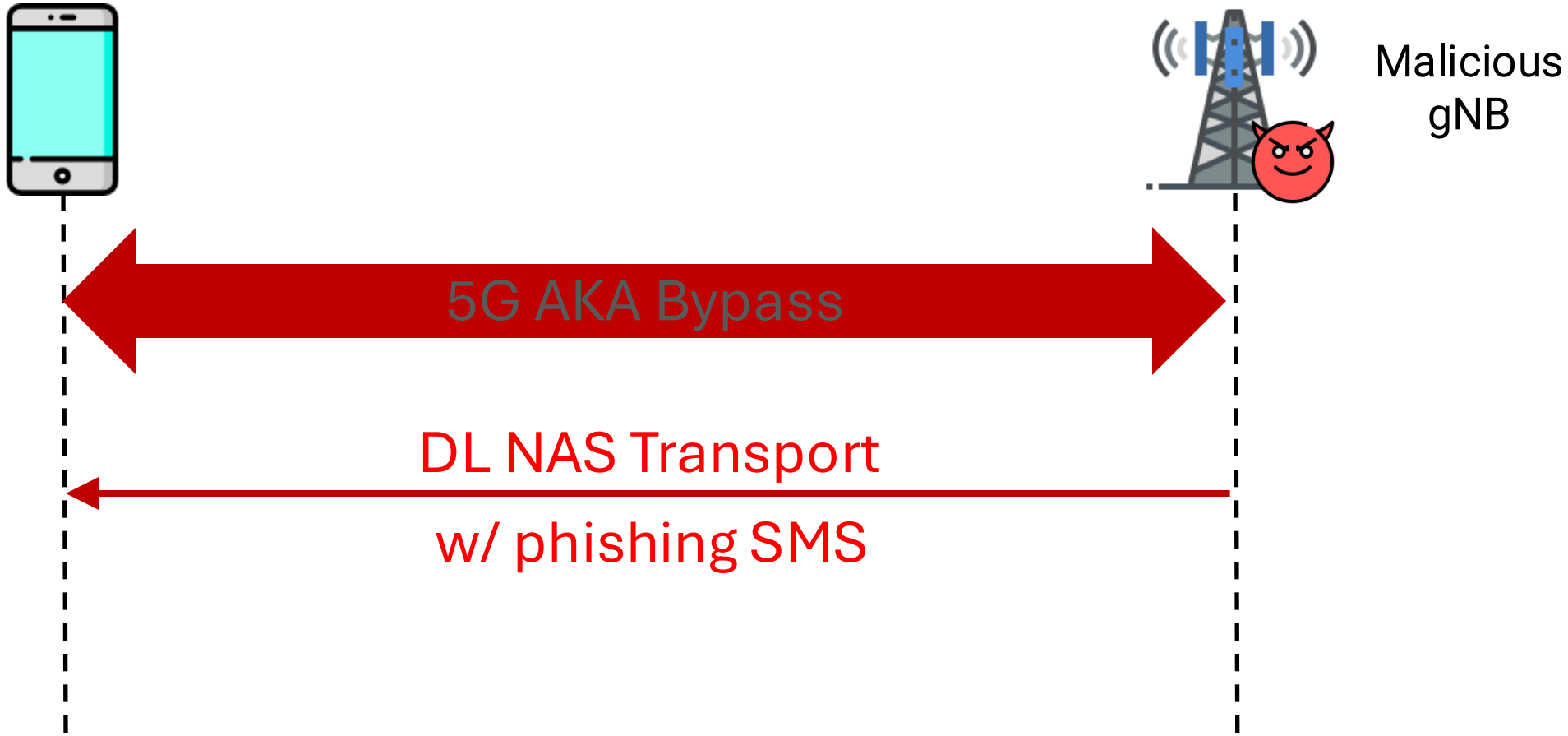
Off

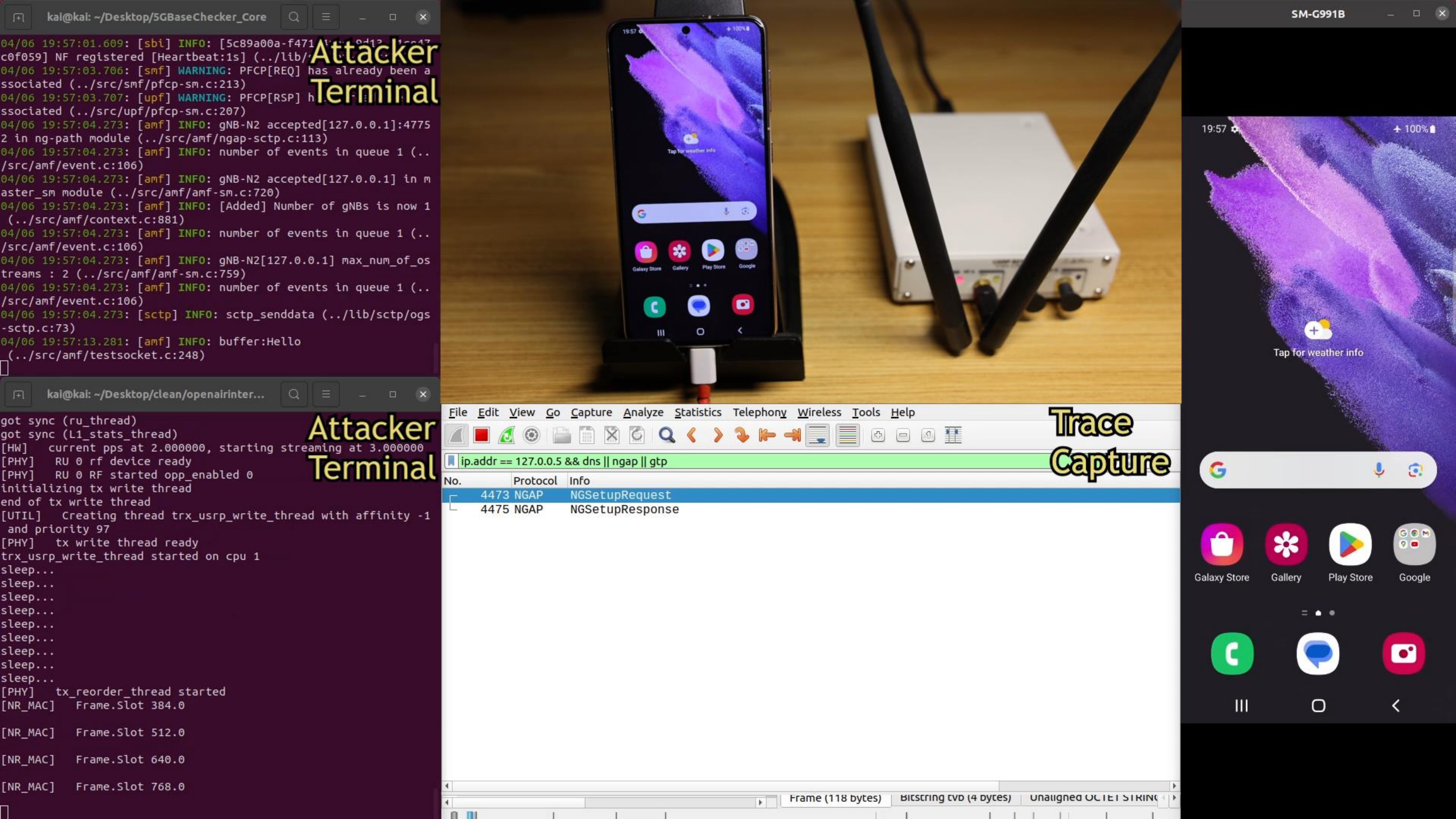
Turns off all network connections including calling, texting, internet access, Wi-Fi, and Bluetooth.

While Airplane mode is on, you can turn Wi-Fi and Bluetooth on again in Settings or the quick settings panel.


kai@kai: ~/Desktop/5GBaseChecker_Core
NN[internet] IPv4[10.45.0.2] IPv6[] (../src/sm
497)
04/06 20:19:47.410: [upf] INFO: [Added] Number of UPF-Sessions
is now 1 (../src/upf/context.c:178)
04/06 20:19:47.410: [gtp] INFO: gtp_connect()
(../lib/gtp/path.c:60)
04/06 20:19:47.410: [upf] INFO: UE F-SEID[CP:0x1 UP:0x1] APN[in
ternet] PDN-Type[1] IPv4[10.45.0.2] IPv6[] (../src/upf/context.
c:397)
04/06 20:19:47.410: [upf] INFO: UE F-SEID[CP:0x1 UP:0x1] APN[in
ternet] PDN-Type[1] IPv4[10.45.0.2] IPv6[] (../src/upf/context.
c:397)
04/06 20:19:47.410: [gtp] INFO: gtp_connect() [127.0.0.7]:2152
(../lib/gtp/path.c:60)
04/06 20:19:47.411: [amf] WARNING: 0x7f40a981c010 (../src/amf/n
amf-handler.c:83)
04/06 20:19:47.411: [sctp] INFO: sctp_senddata (../lib/sctp/ogs
-sctp.c:73)
04/06 20:19:47.446: [amf] INFO: number of events in queue 1 (..
/src/amf/event.c:106)
04/06 20:19:47.446: [gtp] INFO: gtp_connect() [127.0.0.5]:2152
(../lib/gtp/path.c:60)
04/06 20:19:47.446: [amf] INFO: set e->h.sbi.message (../src/am
f/amf-sm.c:511)
Attacker Terminal
kai@kai: ~/Desktop/clean/openairinter...
harq rounds)
[NR_MAC] handle harq for rnti 636f, in RA p
[NR_MAC] handle_nr_ul_harq(): unknown RNTI 0x636f in PUSCH
[NR_PHY] [gNB 0][RAPROC] Frame 79, slot 19
edure with preamble 5, energy 51.0 dB (I0 136,
y 9 start symbol 0 freq index 0
[NR_PHY] [gNB 0][RAPROC] Frame 79, slot 19 Initiating RA proc
edure with preamble 41, energy 51.0 dB (I0 180, thres 120), del
ay 10 start symbol 4 freq index 0
[NR_PHY] [gNB 0][RAPROC] Frame 79, slot 19 Initiating RA proc
edure with preamble 0, energy 48.0 dB (I0 219, thres 120), dela
y 20 start symbol 8 freq index 0
[MAC] UL_info[Frame 79, Slot 19] Calling initiate_ra_proc RAC
H:SFN/SLOT:79/19
[NR_MAC] [gNB 0][RAPROC] CC_id 0 Frame 79 Activating Msg2 gen
eration in frame 80, slot 7 using RA rnti 10b SSB, new rnti d8d
4 index 0 RA index 0
[NR_MAC] [gNB 0][RAPROC] FAILURE: CC_id 0 Frame 79 initiating
RA procedure for preamble index 0
[MAC] UL_info[Frame 80, Slot 0] Calling initiate_ra_proc RACH
:SFN/SLOT:79/19
[NR_MAC] [gNB 0][RAPROC] FAILURE: CC_id 0 Frame 79 initiating
RA procedure for preamble index 0
[NR_MAC] [gNB 0][RAPROC] CC_id 0 Frame 80, slotP 7: Generatin
g RA-Msg2 DCI, rnti 0x10b, state 1, CoreSetType 2
[NR_MAC] [RAPROC] Msg3 slot 17: current slot 7 Msg3 frame 80
k2 7 Msg3_tda_id 3
[NR_MAC] [gNB 0][RAPROC] Frame 80, Subframe 7: rnti d8d4 RA s
tate 2
Attacker Terminal
Airplane mode
Off
Turns off all network connections including calling, texting, internet access, Wi-Fi, and Bluetooth.
While Airplane mode is on, you can turn Wi-Fi and Bluetooth on again in Settings or the quick settings panel.
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.addr == 127.0.0.5 && dns || ngap || gtp
Trace Capture
No. Protocol Info
3856 GTP <T… Application Data
3857 GTP <T… 443 → 37814 [RST] Seq=1009 Win=0 Len=0
3868 GTP <D… Standard query response 0x8467 AAAA b4E8Sm-dnsotls-ds.metric.gstatic.com AAAA 2607
3869 GTP <Q… Protected Payload (KP0), DCID=ee7412ff33df9008
3872 GTP <T… 45302 → 853 [ACK] Seq=373 Ack=5429 Win=78848 Len=0 TSval=2231640716 TSecr=38694123
3885 GTP <Q… Protected Payl (KP0), DCID=5acfc1d1af97e6fb1c73a7d7c92efc6d7f9d4e8e
3886 GTP <D… Standard query dbc7 AAAA K5j3NM-dnsotls-ds.metric.gstatic.com
3887 GTP <T… Application Da
3888 GTP <T… 33348 → 853 [eq=451 Ack=5535 Win=79872 Len=0 TSval=1335136228 TSecr=76137289
3889 GTP <T… Application Dat
3891 GTP <D… Standard query 0xa357 A youtubei.googleapis.com
3902 GTP <D… Standard query response 0xdbc7 AAAA K5j3NM-dnsotls-ds.metric.gstatic.com AAAA 260
3904 GTP <T… 853 → 33348 [FIN, ACK] Seq=5535 Ack=475 Win=67840 Len=0 TSval=761373029 TSecr=1335
3905 GTP <T… 853 → 33348 [ACK] Seq=5536 Ack=476 Win=67840 Len=0 TSval=761373031 TSecr=133513623
3906 GTP <D… Standard query response 0xa357 A youtubei.googleapis.com A 142.251.40.138 A 142.25
3907 GTP <Q… Initial, DCID=aa5c42630c886a78, PKN: 1, CRYPTO, CRYPTO, PADDING, PING, CRYPTO, PADI
Plaintext DNS Packets
Frame (118 bytes)
SM-G991B
20:19
100%
Airplane mode
Off
Turns off all network connections including calling, texting, internet access, Wi-Fi, and Bluetooth.
While Airplane mode is on, you can turn Wi-Fi and Bluetooth on again in Settings or the quick settings panel.

Phishing SMS Injection





kai@kai: ~/Desktop/5GBaseChecker_Core
Attacker Terminal
04/06 19:57:03.706: [smf] WARNING: PFCP[REQ] has already been a
ssociated (../src/smf/pfcp-sm.c:213)
04/06 19:57:03.707: [upf] WARNING: PFCP[RSP]
ssociated (../src/upf/pfcp-sm.c:207)
04/06 19:57:04.273: [amf] INFO: gNB-N2 accepted[127.0.0.1]:4775
2 in ng-path module (../src/amf/ngap-sctp.c:113)
04/06 19:57:04.273: [amf] INFO: number of events in queue 1 (..
/src/amf/event.c:106)
04/06 19:57:04.273: [amf] INFO: gNB-N2 accepted[127.0.0.1] in m
aster_sm module (../src/amf/amf-sm.c:720)
04/06 19:57:04.273: [amf] INFO: [Added] Number of gNBs is now 1
(../src/amf/context.c:881)
04/06 19:57:04.273: [amf] INFO: number of events in queue 1 (..
/src/amf/event.c:106)
04/06 19:57:04.273: [amf] INFO: gNB-N2[127.0.0.1] max_num_of_os
treams : 2 (../src/amf/amf-sm.c:759)
04/06 19:57:04.273: [amf] INFO: number of events in queue 1 (..
/src/amf/event.c:106)
04/06 19:57:04.273: [sctp] INFO: sctp_senddata (../lib/sctp/ogs
-sctp.c:73)
04/06 19:57:13.281: [amf] INFO: buffer:Hello
(../src/amf/testsocket.c:248)
kai@kai: ~/Desktop/clean/openairinter...
Attacker Terminal
got sync (ru_thread)
got sync (L1_stats_thread)
[HW] current pps at 2.000000, starting streaming at 3.000000
[PHY] RU 0 rf device ready
[PHY] RU 0 RF started opp_enabled 0
initializing tx write thread
end of tx write thread
[UTIL] Creating thread trx_usrp_write_thread with affinity -1
and priority 97
[PHY] tx write thread ready
trx_usrp_write_thread started on cpu 1
sleep...
sleep...
sleep...
sleep...
sleep...
sleep...
sleep...
sleep...
sleep...
[PHY] tx_reorder_thread started
[NR_MAC] Frame.Slot 384.0
[NR_MAC] Frame.Slot 512.0
[NR_MAC] Frame.Slot 640.0
[NR_MAC] Frame.Slot 768.0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Trace Capture
ip.addr == 127.0.0.5 && dns || ngap || gtp
No. Protocol Info
4473 NGAP NGSetupRequest
4475 NGAP NGSetupResponse
Frame (118 bytes)
SM-G991B
19:57
100%
Tap for weather info
Galaxy Store
Gallery
Play Store
Google

DFA and NFA

- Finite acceptor: $\mathcal{M} = <Q, \Sigma, \delta, q_0, \mathcal{F}>$
  - A set of finite states- Q
  - An alphabet- Σ
  - Transition function- $\delta : Q \times \Sigma \rightarrow Q$
  - Initial state- $q_0 \in Q$
  - Accepting states- $\mathcal{F} \subseteq Q$
- The transition function δ decides whether $\mathcal{M}$ is a DFA or NFA

Insight of the algorithm

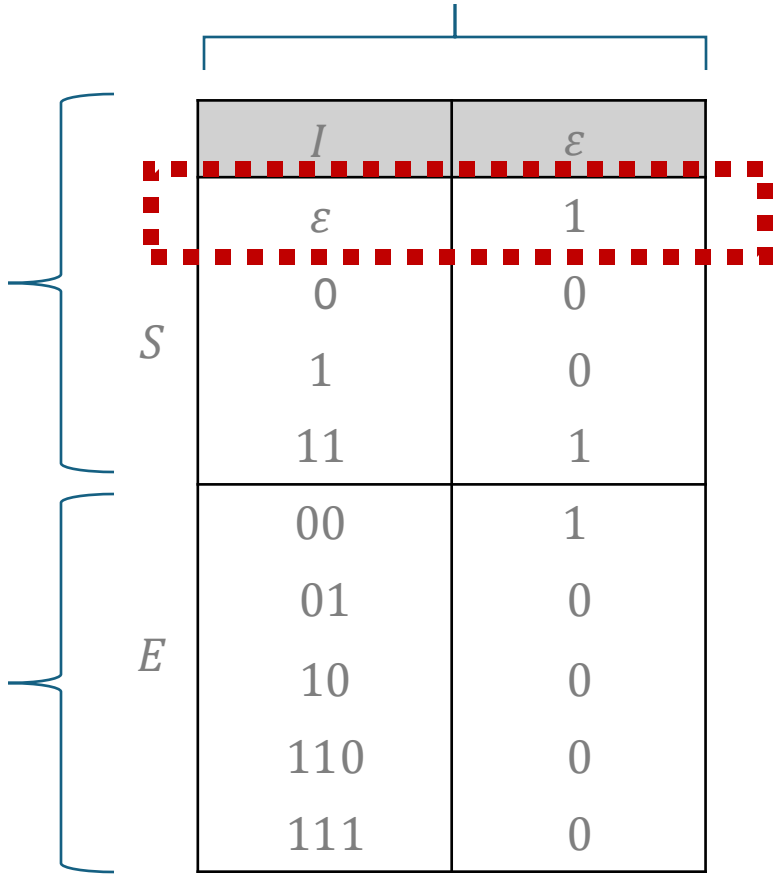
- Discriminating sequence:
  - The concept is used to distinguish two non-equivalent states
  - Two states are equivalent if for all strings they end up in the same accepting or non-accepting state
  - A discriminating sequence is a string suffix that will distinguish between non-equivalent states (one will end up in an accepting state and the other in a non-accepting state)

Representation of Hypothesis Automata

Discriminating sequence

Potential states

Potential transitions



	I	ε
	ε	1
S	0	0
	1	0
	11	1
E	00	1
	01	0
	10	0
	110	0
	111	0

- $A \in \{0, 1\}$
- $I = \{a_1, a_2, \ldots, an\} \mid a \in A^*$
- $O = \{0, 1\}$
- $S =$ Prefix-closed set of I
- $E = S.A =$ Suffix-closed set of I

When do we pose equivalence queries?

- When the observation table is closed and consistent

- ***Closed***: An observation table is closed if only if
  - $\forall \mathrm{e} = \mathrm{s}'.\mathrm{a} \in \mathrm{E}\ \exists s \in S \mid \mathrm{row}(\mathrm{s}) = \mathrm{row}(e),\ a \in A$

- ***Consistent:*** An observation table is consistent if only if
  - $\forall s_1, s_2 : S \mid \mathrm{row}(s_1) = \mathrm{row}(s_2) \rightarrow \mathrm{row}(s_1 \cdot a) = \mathrm{row}(s_2 \cdot a)$

The L* Algorithm - Explained

- $A \in \{0, 1\}$
- $I = \{a_1, a_2, \ldots, an\} \mid a \in A^*$
- $O = \{0, 1\}$
- $S =$ Prefix-closed set of I
- $E =$ Suffix-closed set of I
- Develop an observation table that is *closed* and *consistent*.

- We will explain the algorithm through an example:
  - Learn the observation table/DFA of a DFA that only accepts strings with even $0s$ and $1s$.

	I	ε
S	ε	1
E	0	0
	1	0

Observation table – Closed and consistent

The table is initialized as follows:

	I	ε
S	ε	1
E	0	0
	1	0

Closed? No!
$\text{row}(\varepsilon) \neq \text{row}(0)$ is in the table

Consistent? Yes!
Only one row in S

Observation table – Closed and consistent

- An observation table is closed if:
  - $\forall e \in E\ \exists s \in S \mid \mathrm{row}(s) = \mathrm{row}(e), a \in A$
- An observation table is consistent if:
  - $\forall s_1, s_2 \mid \mathrm{row}(s_1) = \mathrm{row}(s_2) \rightarrow \mathrm{row}(s_1 \cdot a) = \mathrm{row}(s_2 \cdot a)$
- Objective 1: Make the table closed.

	I	ε
S	ε	1
E	0	0
	1	0

Observation table – Closed and consistent

- An observation table is closed if:
  - $\forall e \in E \; \exists s \in S \mid \text{row}(s) = \text{row}(e), a \in A$
- An observation table is consistent if:
  - $\forall s_1, s_2 \mid \text{row}(s_1) = \text{row}(s_2) \rightarrow \text{row}(s_1 \cdot a) = \text{row}(s_2 \cdot a)$
- Objective 1: Make the table closed.

	I	ε
S	ε	1
	0	0
E	1	0
	00	1
	01	0

Closed? Yes!
For all rows in E, there exists a row in S such that the rows are equal.

Consistent? Yes!
S only has two distinct rows, therefore the if condition doesn't apply.

Observation table – Closed and consistent

- An observation table is closed if:
  - $\forall e \in E\ \exists s \in S \mid \mathrm{row}(s) = \mathrm{row}(e), a \in A$
- An observation table is consistent if:
  - $\forall s_1, s_2 \mid \mathrm{row}(s_1) = \mathrm{row}(s_2) \rightarrow \mathrm{row}(s_1 \cdot a) = \mathrm{row}(s_2 \cdot a)$
- Objective 1: Make the table closed.
- The learner can now formulate a hypothesis and query the system to see if the inferred FSA is equivalent to the SUL FSA.

	I	ε
S	ε	1
	0	0
E	1	0
	00	1
	01	0

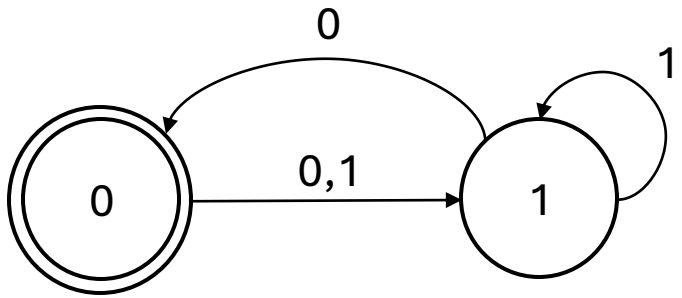
Closed? Yes!
For all rows in E, there exists a row in S such that the rows are equal.

Consistent? Yes!
S only has two distinct rows, therefore the if condition doesn't apply.

Observation table to DFA

- We can construct a DFA using the unique rows in the table.
- Each row in S represents a state in the DFA.
- The first row with the first column of 1 is the accepting state.
- For this table, the DFA has 2 states.

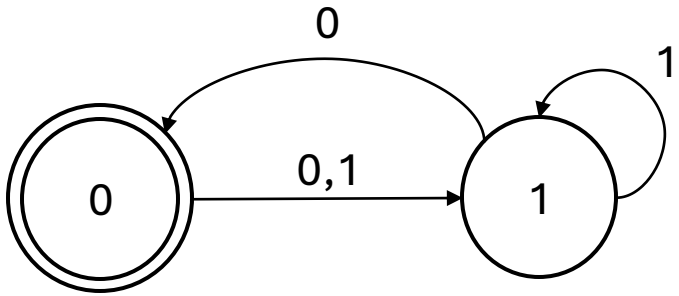
	I	ε
S	ε	1
	0	0
E	1	0
	00	1
	01	0



Equivalence query

- We now ask the SUL, through an equivalence oracle, is this construction an accurate representation of the target system?
  - The equivalence oracle responds with either a *yes* or a *no.*
  - If the oracle responds with *yes,* learning is terminated.
  - If the oracle responds with *no,* the oracle provides a counter-example. An input for which the DFA behaves differently from the target system.
- The L* algorithm, therefore, uses this *counter-example* to update the observation table.

	I	ε
S	ε	1
	0	0
E	1	0
	00	1
	01	0



Counter-example

- We incorporate the counter-example whilst keeping the table closed and consistent.
- Assume the oracle gives the counter-example 11, our current DFA will reject this string.
- Add 1 and 11 to S and their $\{0,1\}$ appended suffixes to E.

	I	ε
S	ε	1
	0	0
	1	0
	11	1
E	00	1
	01	0
	10	0
	110	0
	111	0

Closed? Yes!

Consistent? No.
$\text{row}(0) = \text{row}(1)$ but $\text{row}(00) \neq \text{row}(10)$

Consistency

- Objective 1: Make the table closed.
- Objective 2: Make the table consistent.
  - Add column of the $a \in A$, that causes the inconsistency, in this case, 0.

	I	ε
S	ε	1
	0	0
	1	0
	11	1
E	00	1
	01	0
	10	0
	110	0
	111	0

Closed? Yes!

Consistent? No.
$\text{row}(0) = \text{row}(1)$ but $\text{row}(00) \neq \text{row}(10)$

Consistency

- Objective 1: Make the table closed.
- Objective 2: Make the table consistent.
  - Add column of the $a \in A$, that causes the inconsistency, in this case, 0.

	I	ε	0
S	ε	1	0
	0	0	1
	1	0	0
	11	1	0
E	00	1	0
	01	0	0
	10	0	0
	110	0	0
	111	0	0

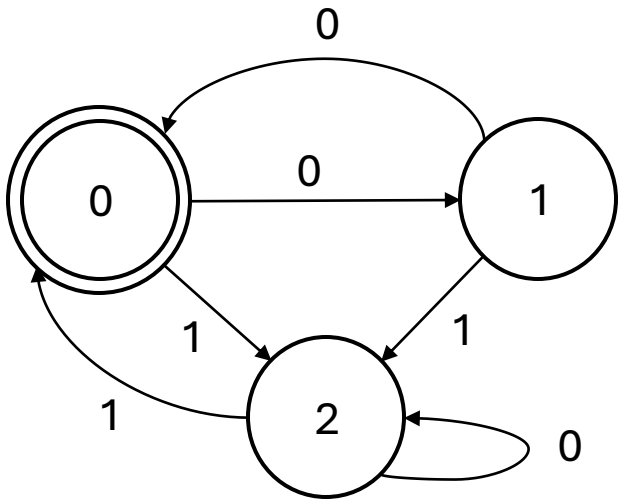
Closed? Yes.

Consistent? Yes.

Hypothesis refinement

- The learner again queries the EQ oracle for testing the equivalence of hypothesis $\mathcal{H}$, namely if:
  $\mathcal{H} \approx \mathcal{M}$ where $\mathcal{H}$ is the hypothesized FSA and $\mathcal{M}$ is the system FSA
- Since the table has 3 unique rows, the DFA has 3 states.

	I	ε	0
S	ε	1	0
	0	0	1
	1	0	0
	11	1	0
E	00	1	0
	01	0	0
	10	0	0
	110	0	1
	111	0	0



Since the DFA does not accept only even **0s and 1s,** the EQ oracle returns a *counter-example*. Assume it returns 011. Our DFA will accept this string.

Hypothesis refinement

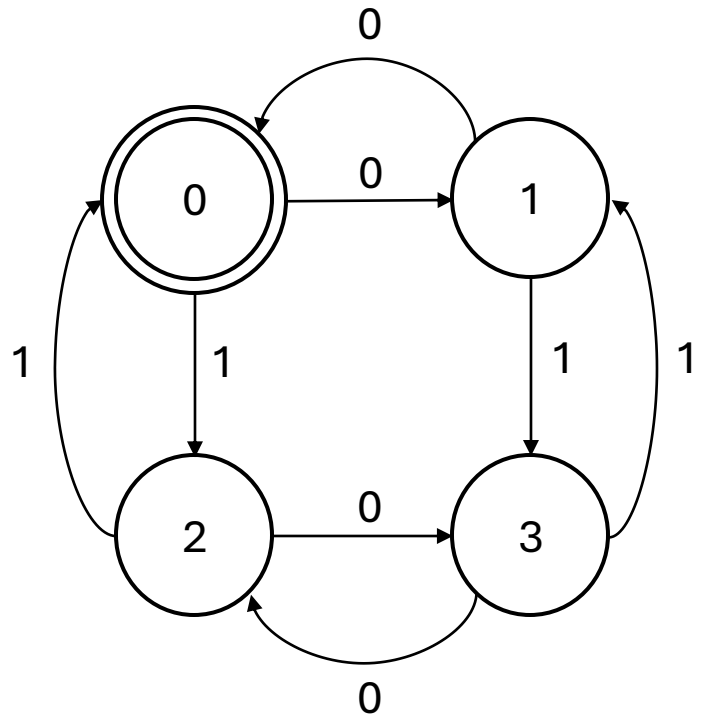
- The algorithm then repeats the process of making the table closed and consistent, eventually adding the 1 column.
- Our observation table after another closing and consistency iteration becomes:

	I	ε	0	1
S	ε	1	0	0
	0	0	1	0
	1	0	0	1
	11	1	0	0
	011	0	1	0
E	00	1	0	0
	01	0	0	0
	10	0	0	0
	110	0	1	0
	111	0	0	1
	0110	1	0	0
	0111	0	1	0

Hypothesis refinement

- This table has 4 unique rows, therefore, the resulting DFA has 4 states.

	I	ε	0	1
S	ε	1	0	0
	0	0	1	0
	1	0	0	1
	01	0	0	0
	11	1	0	0
	011	0	1	0
E	00	1	0	0
	10	0	0	0
	010	0	0	1
	110	0	1	0
	111	0	0	1
	0110	1	0	0
	0111	0	1	0



The EQ oracle returns *True* and the algorithm terminates.