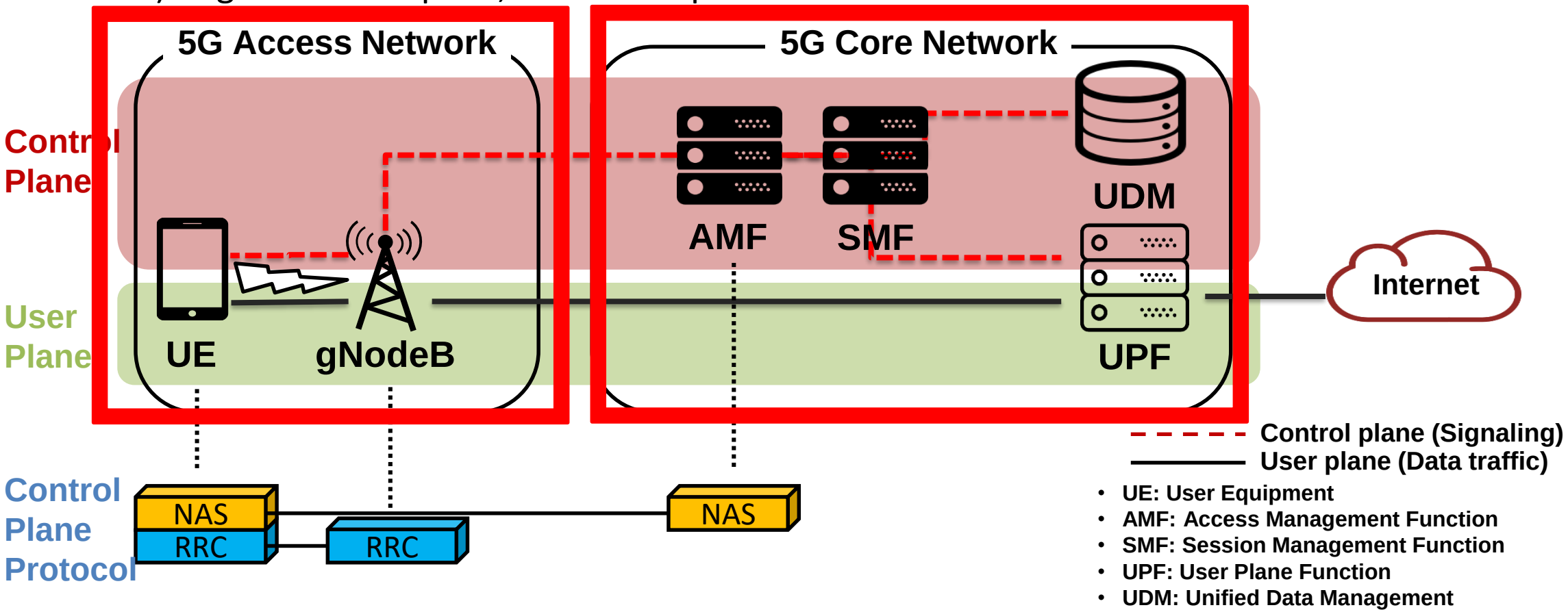
Stateful Black Box Testing for 5G Standalone Network

Yeongbin Hwang

Committee Members
Prof. Yongdae Kim – chair
Prof. Minsuk Kang
Prof. Insu Yun

5G SA Network Architecture

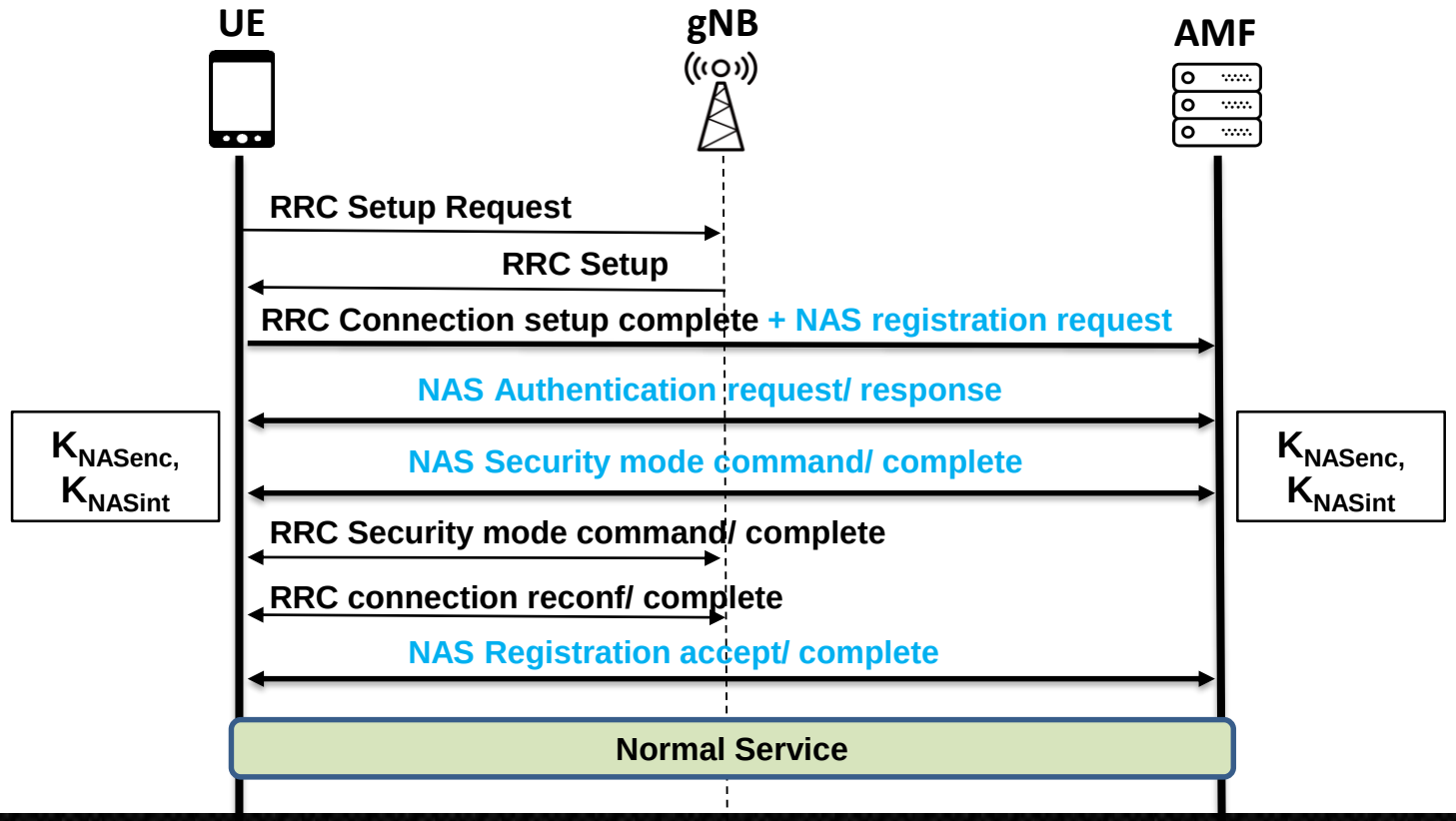
- Control plane procedures **are always preceded** by the user plane procedures
  - ex) Registration request, Service Request



RRC: Radio Resource Control NAS: Non Access Stratum

Control Plane Procedures

- Registration Procedure
  - The first step for a UE to use a cellular services
  - There are many message exchanges.



Fundamental Problems in Cellular Network

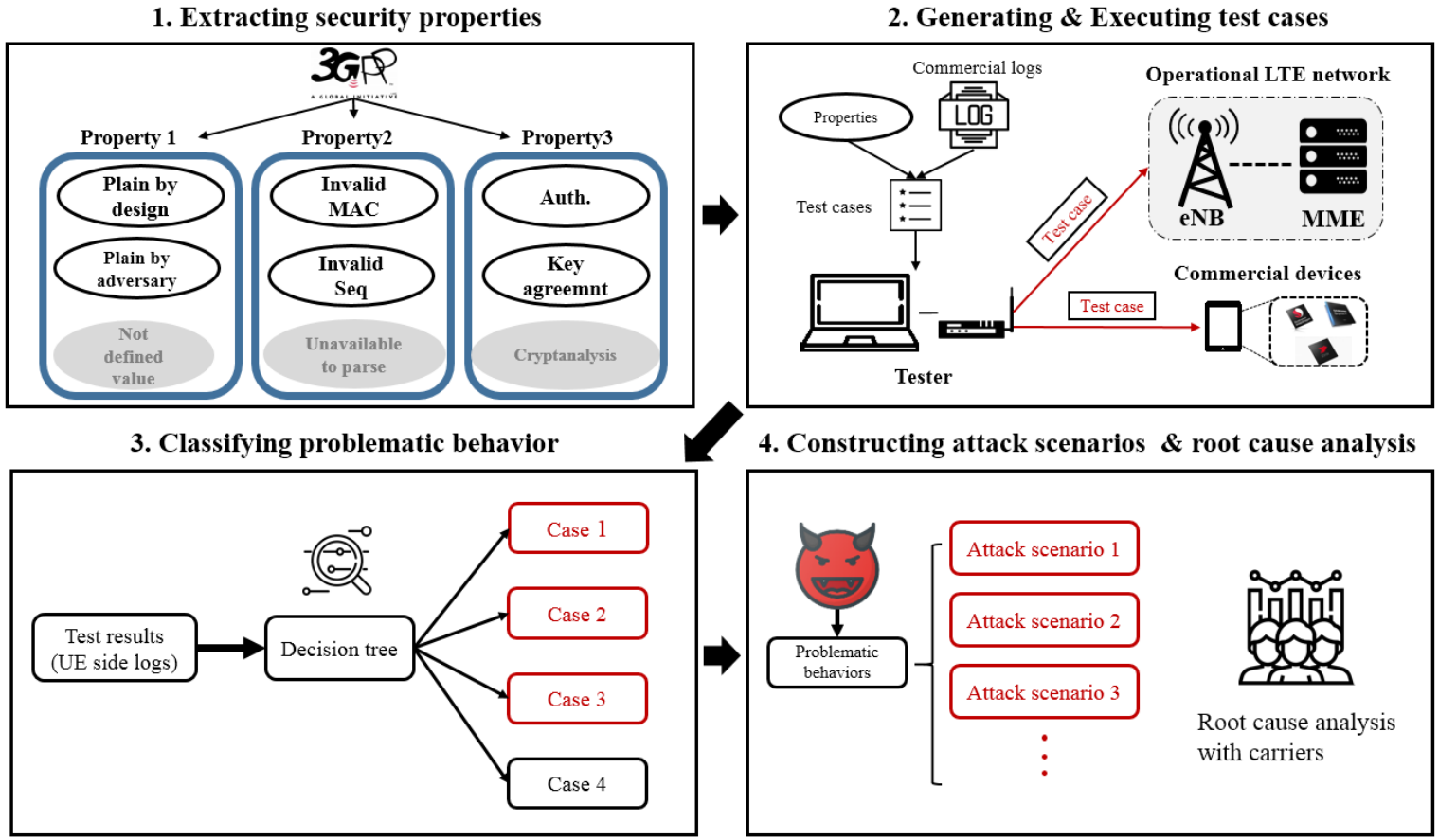
- **Description of standard (3GPP) has amibiguities**
  - The 3GPP specifications are based on natural language
  - Standard leave implementation (exact behavior) details to the vendors
  - There are conformance test specs
    - But, only focused on normal situation (no adversary model)

- **Mobile network operators & vendors are different**
  - Different carriers with different device vendors suffer from different vulnerabilities

Many approaches to finding implementation vulnerabilities

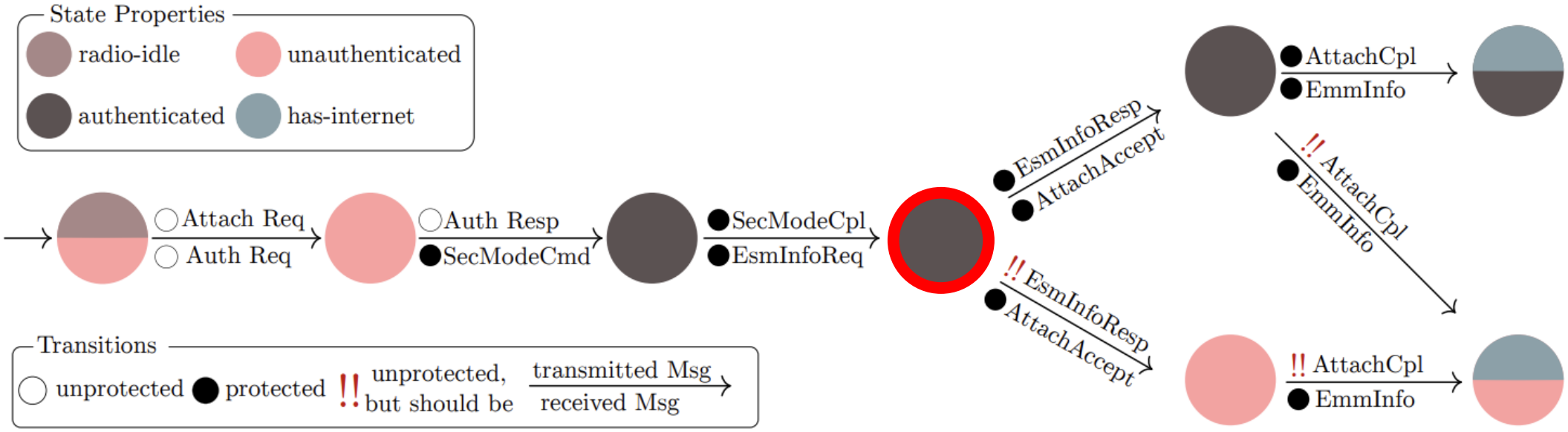
Previous Works

- Touching the Untouchables: Dynamic Security Analysis of the LTE Control Plane



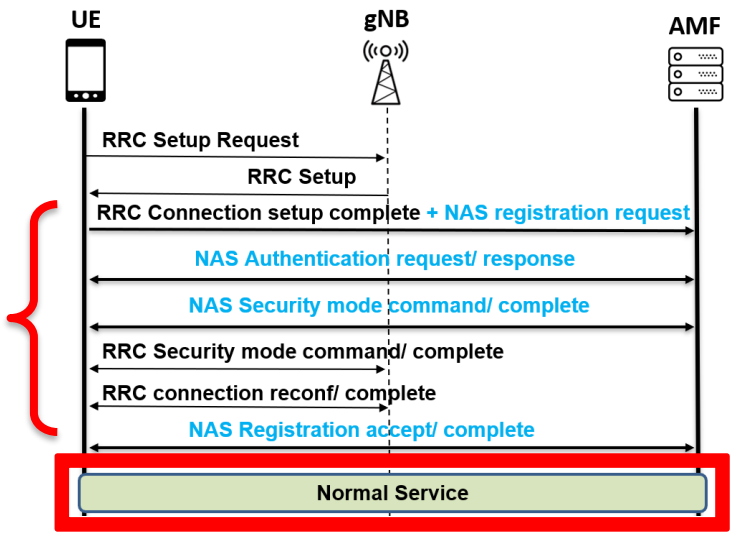
Previous Works

- On the Challenges of Automata Reconstruction in LTE Networks, Wisec 2021
  - They tested **invalid messages** in various states and detected abnormal behavior of the network.



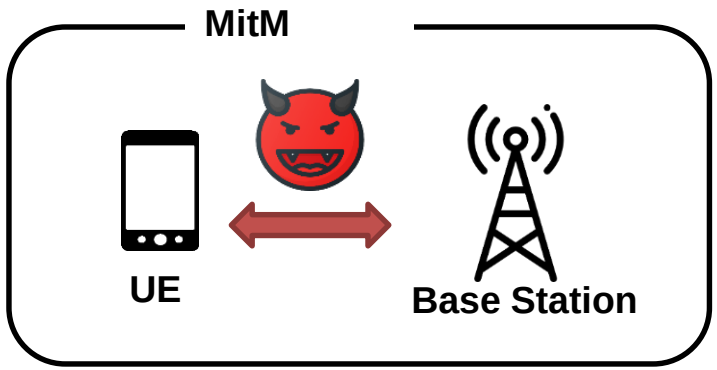
Limitations of Previous Works

- Only considered the **initial**, **REGISTERED** state
  - LTEFuzz
  - Can’t find vulnerabilities in different states



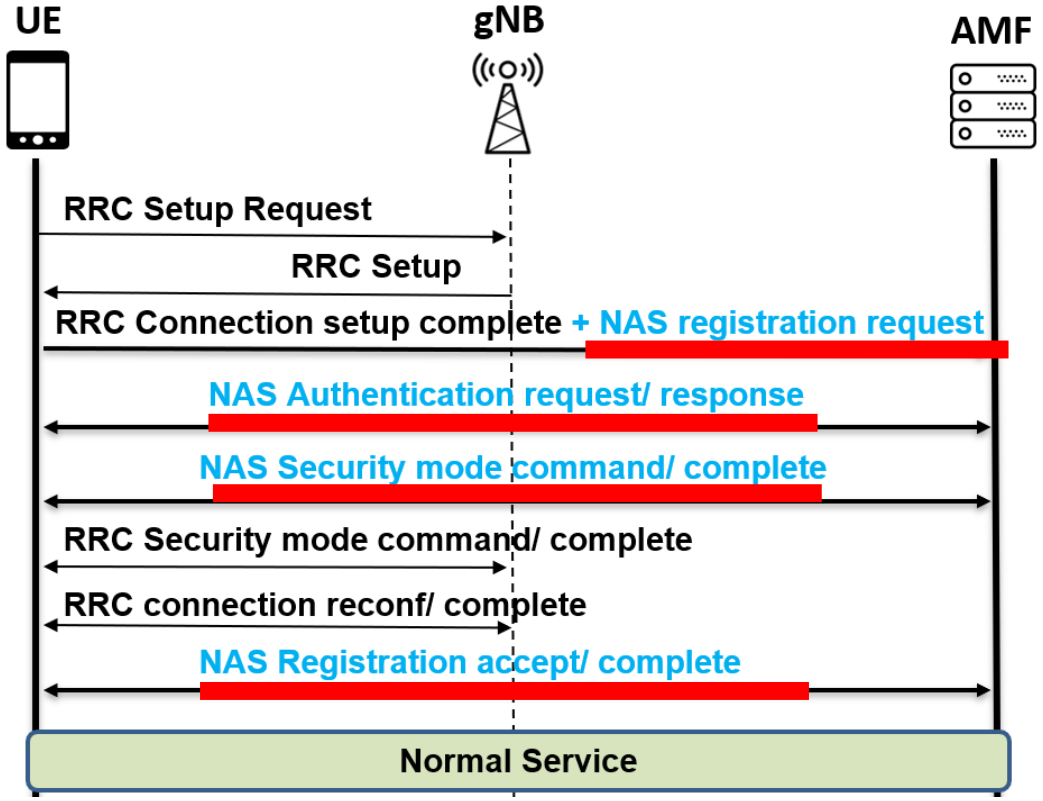
- Only tested the **limited** attack scenarios
  - On the Challenges of Automata Reconstruction
  - Can’t find vulnerabilities in different scenarios

    ex)
    - Invalid sequence number -> Impersonation
    - Unauthenticated message -> DoS



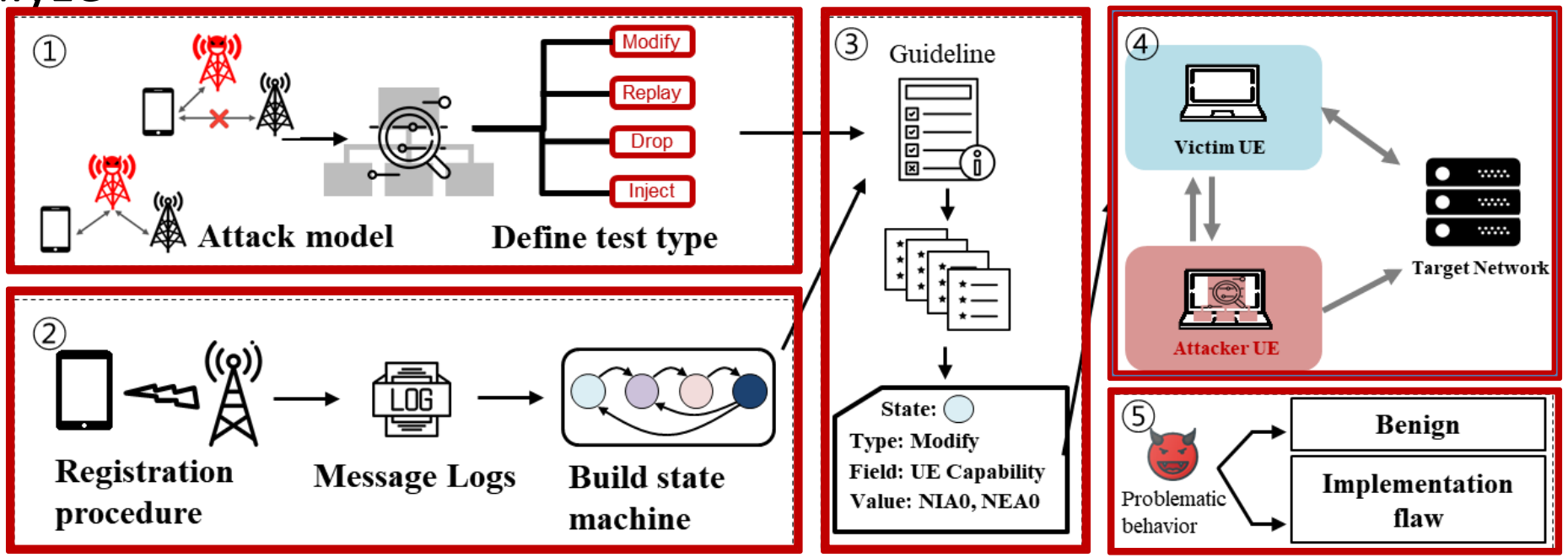
Goal of Our Work

- Conduct the **first uplink stateful testing** that considers the various attack scenarios in **5G SA Network**
  - Attack scenarios
    - **Modify, Replay, Inject, Drop**
  - Implement a testing framework

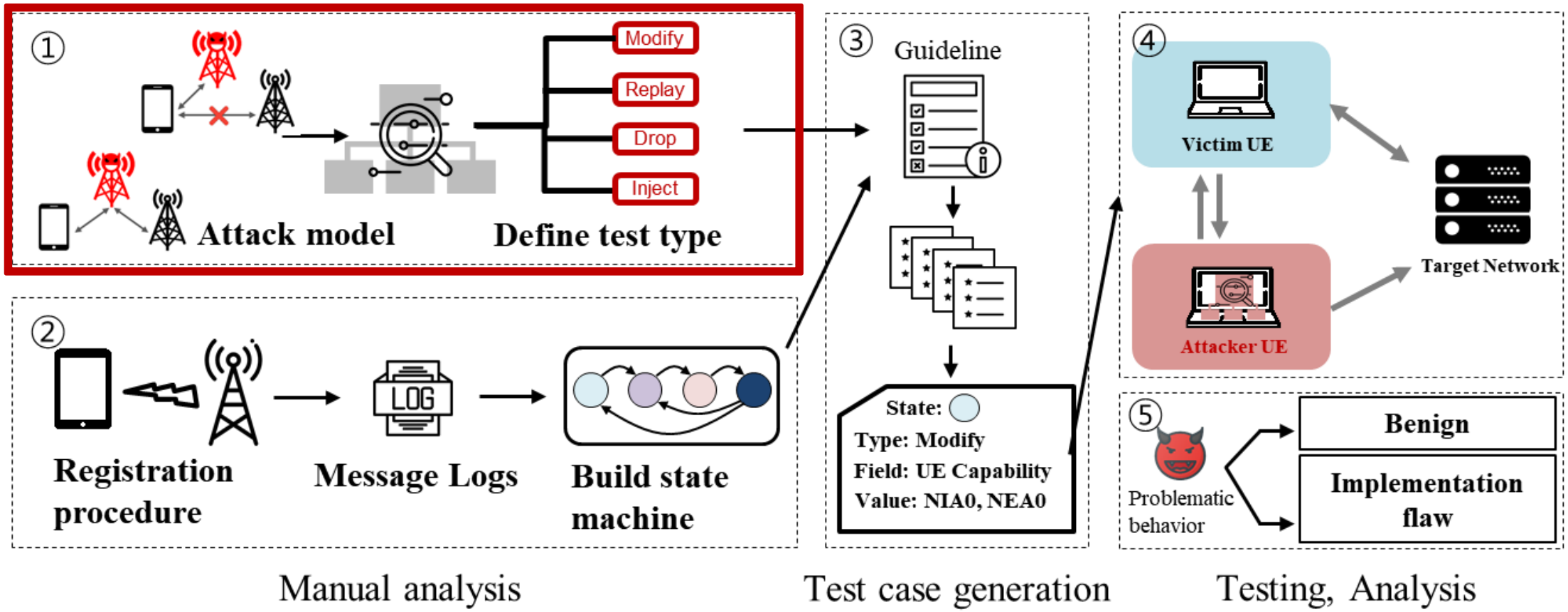


Methodology

1. Define test type and guideline
2. Extract state machine of UE in network
3. Generate test cases
4. Test and Analyze

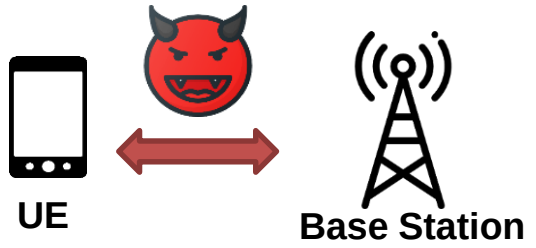


Define Test Type and Guideline

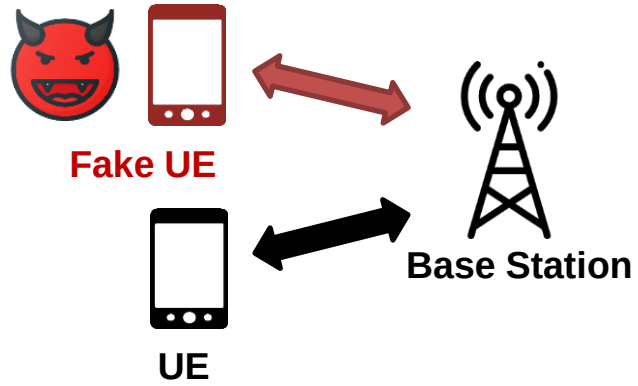


Attack Model

- **MitM** (Man-in-the-Middle)
  - Attacker can eavesdrop, modify, relay, and drop messages between the victim UE and the base station



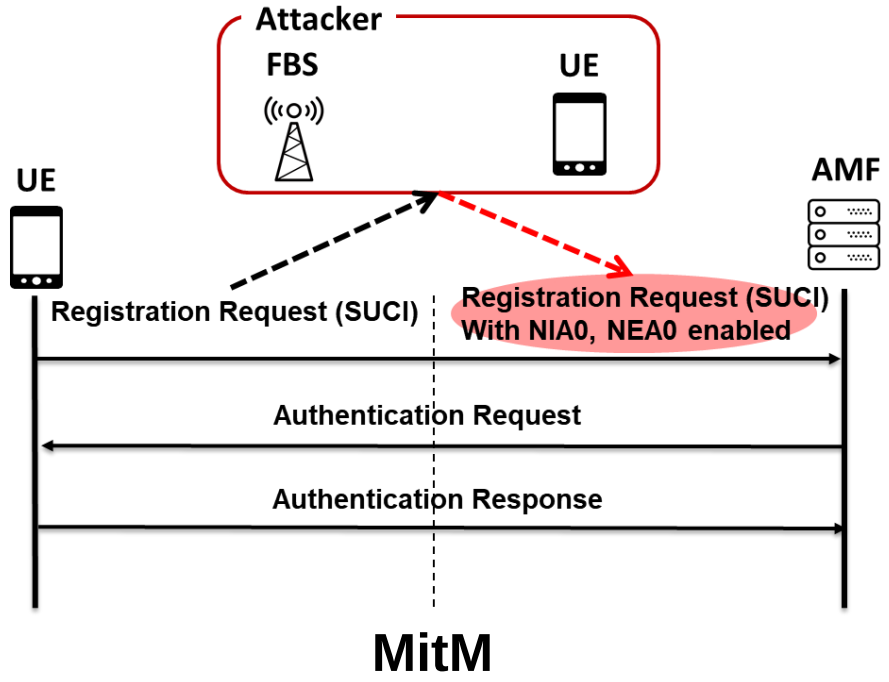
- **Fake UE**
  - Form of man-on-the-side in cellular network.
  - If an attacker knows the identity of the victim UE, he can impersonate the victim UE

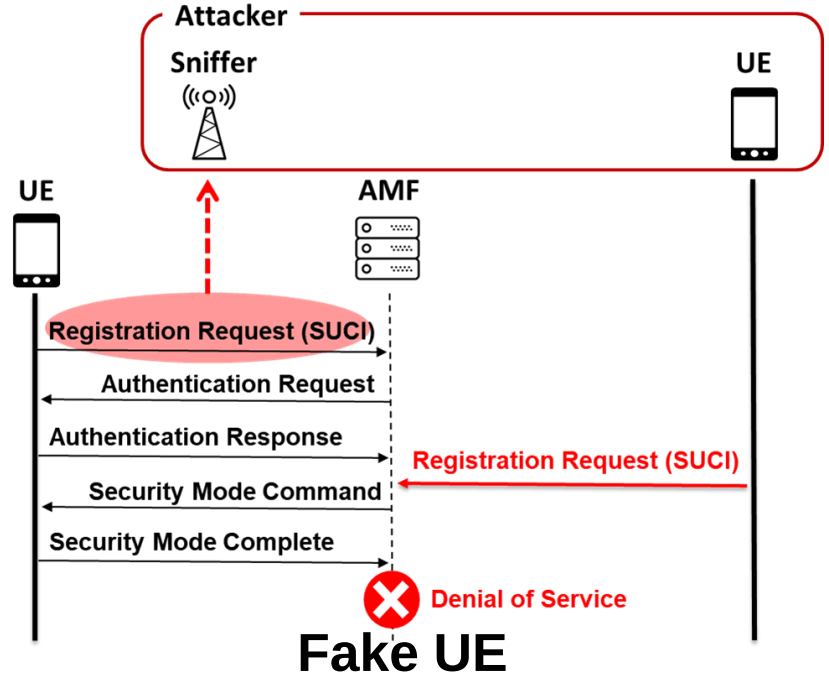


- FBS
- SigOver

Definition of Test Type

- **MitM**
  - MODIFY, DROP, REPLAY, INJECT
- **Fake UE**
  - REPLAY, INJECT





Guidelines for Target Messages

- Target protocol: NAS protocol
- Uplink message

Attack model	Type	Target Messages	Implications
MitM	**MODIFY**	**Messages with capability field**	**Eavesdropping, Impersonate**
		Messages with integrity protection	**Impersonate**
	REPLAY	**Messages with sequence number**	**DoS, Impersonate**
	DROP	Messages that can induce the timer	DoS
	INJECT	Messages with identity field	DoS
Fake UE	**INJECT**	**Messages with identity field**	**DoS**
	REPLAY	**Messages with sequence number**	**DoS, Impersonate**

Guidelines for Target Messages

- Target protocol: NAS protocol
- Uplink message

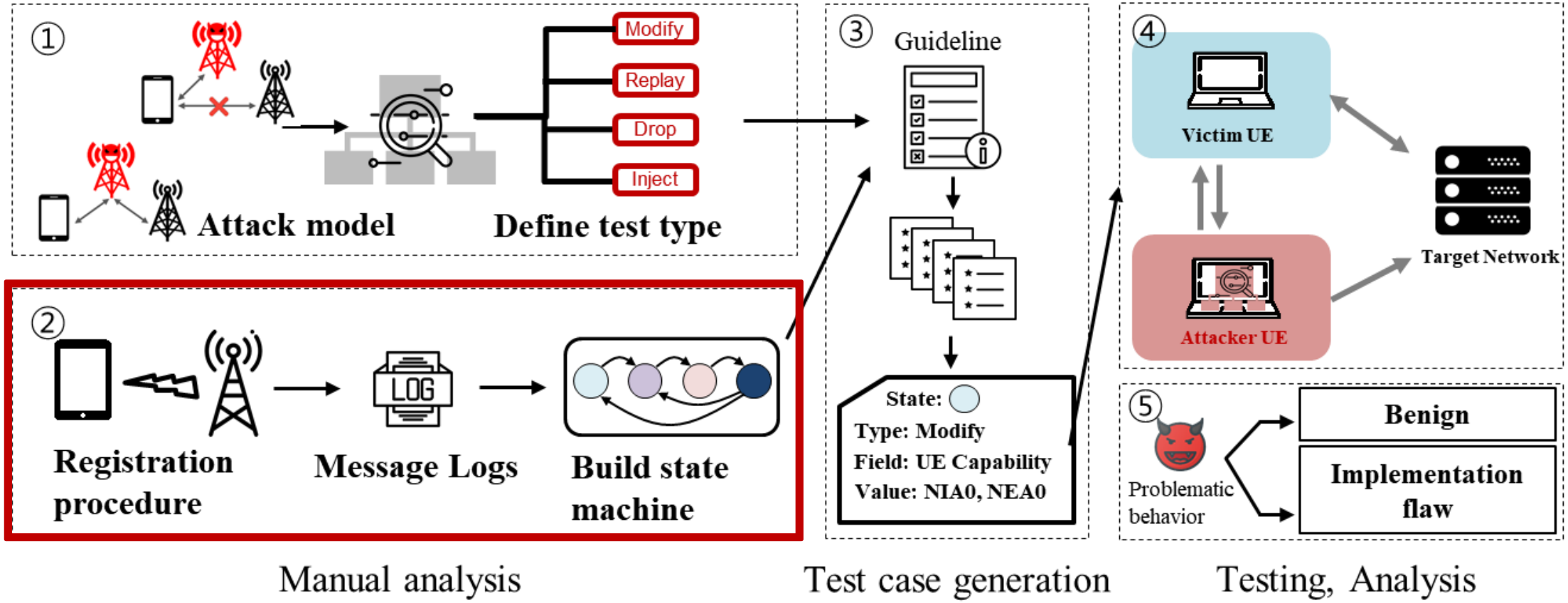
Attack model	Type	Target Messages	Implications
MitM	**MODIFY**	**Messages with capability field**	**Eavesdropping, Impersonate**
		Messages with integrity protection	**Impersonate**
	REPLAY	**Messages with sequence number**	**DoS, Impersonate**
	~~DROP~~	~~Messages that can induce the timer~~	~~DoS~~
	~~INJECT~~	~~Messages with identity field~~	~~DoS~~
Fake UE	**INJECT**	**Messages with identity field**	**DoS**
	REPLAY	**Messages with sequence number**	**DoS, Impersonate**

Guidelines for Target Messages

- Target protocol: NAS protocol
- Uplink message

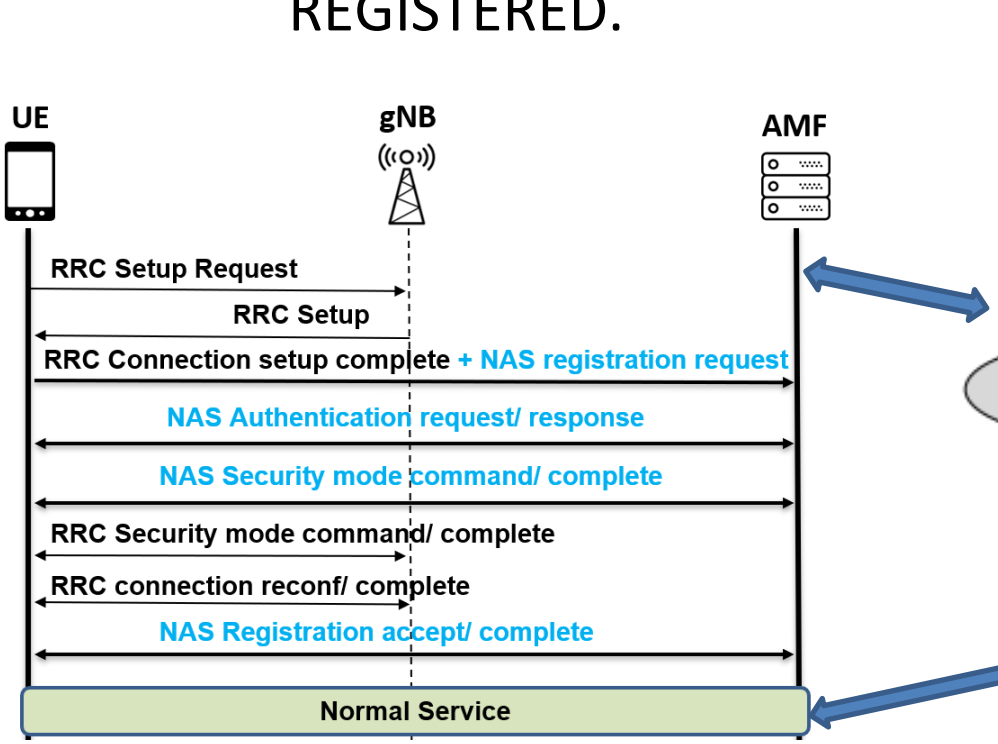
Attack model	Type	Target Messages	Examples
MitM	**MODIFY**	**Messages with capability field**	**Registration request**
		Messages with integrity protection	**Regi req, SMComplete, Regi Comp**
	REPLAY	**Messages with sequence number**	**Regi req, SMComplete, Deregi req**
	~~DROP~~	~~Messages that can induce the timer~~	
	~~INJECT~~	~~Messages with identity field~~	
Fake UE	**INJECT**	**Messages with identity field**	**Regi req, Deregi req, Service req**
	REPLAY	**Messages with sequence number**	**Regi req, SMComplete, Deregi req**

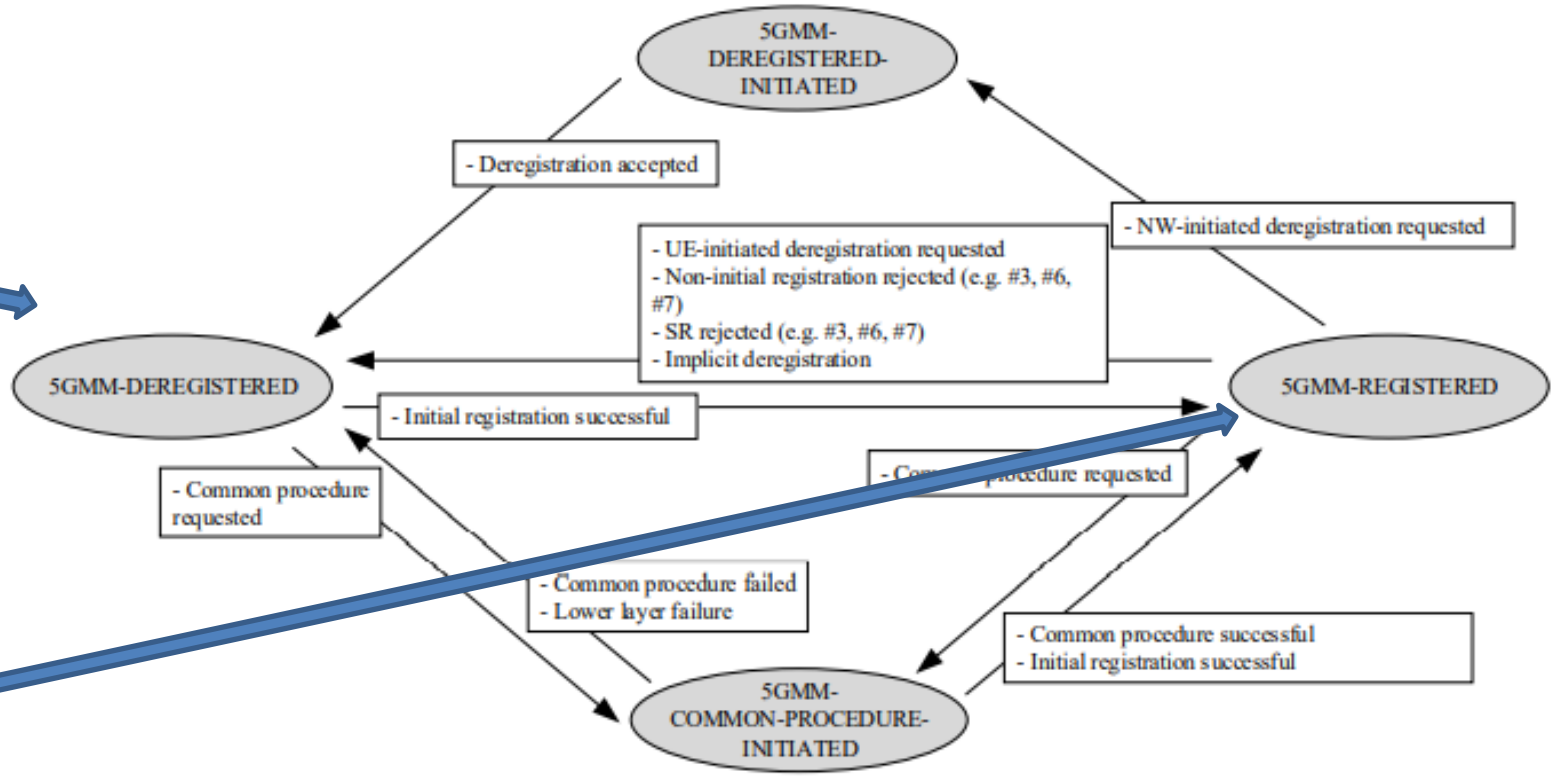
Extract State Machine of UE in Network



State Machine of UE in Network

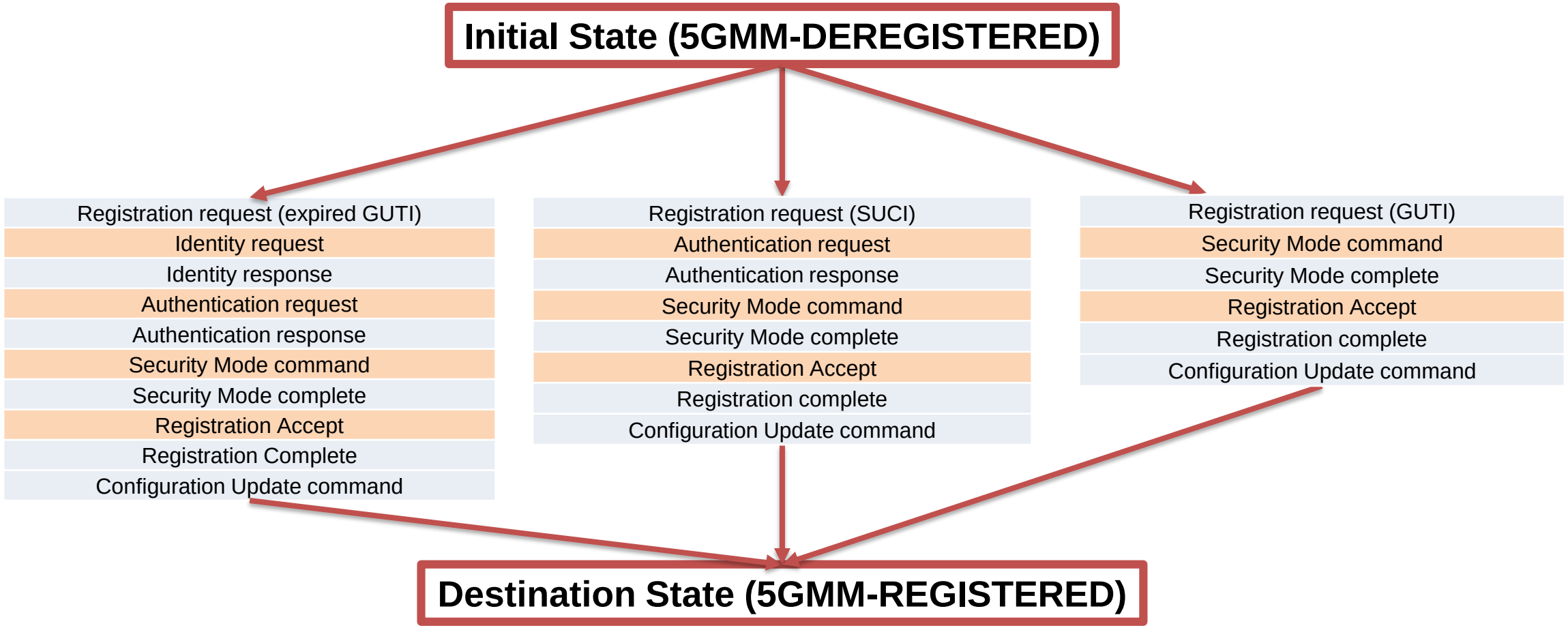
- Defined in 3GPP specification
- Abstracted state machine
  - Many message exchanges between 5GMM-DEREGISTERED and 5GMM-REGISTERED.





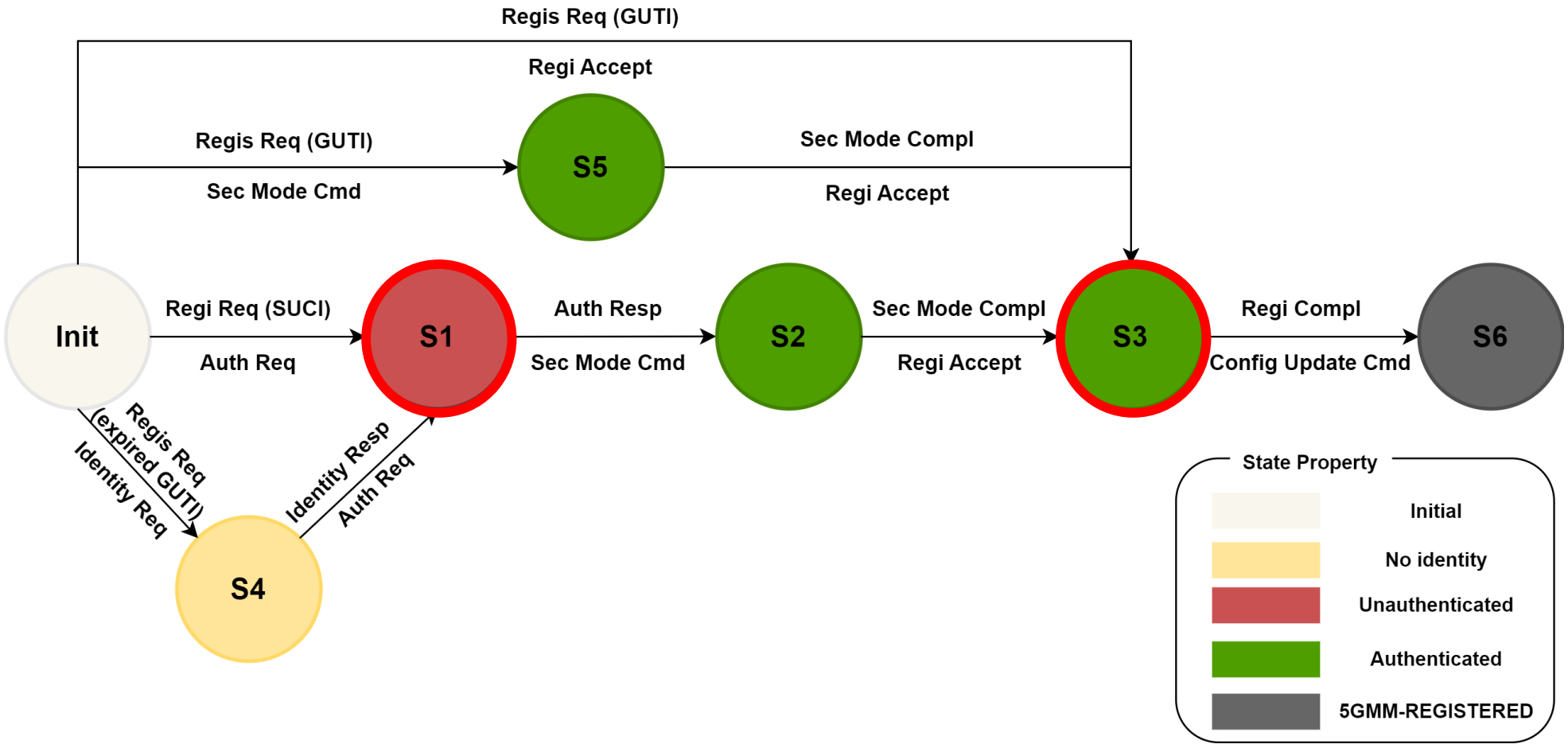
Message Exchanges during Registration

- Transition: Send message/ Receive message

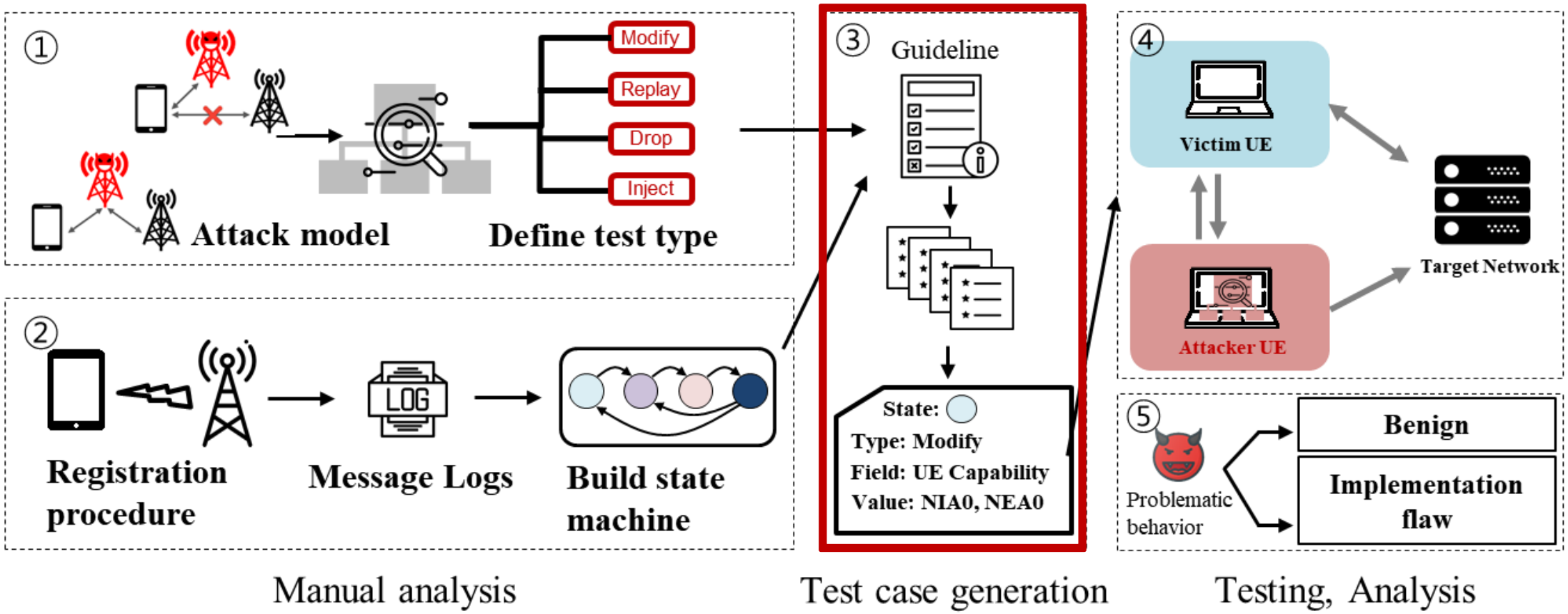


Build the State Machine

- Transition: Send message/ Receive message

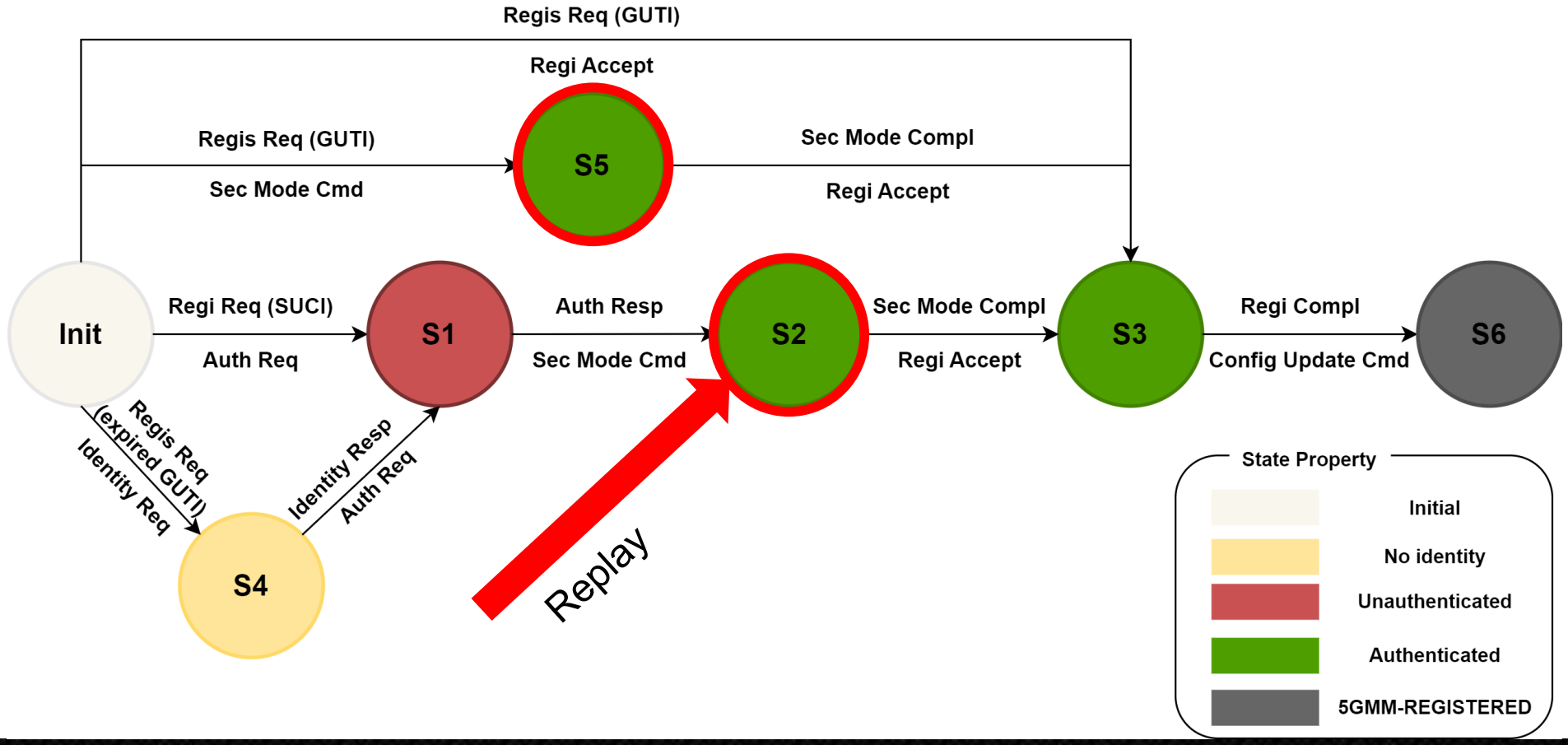


Generate Test Cases



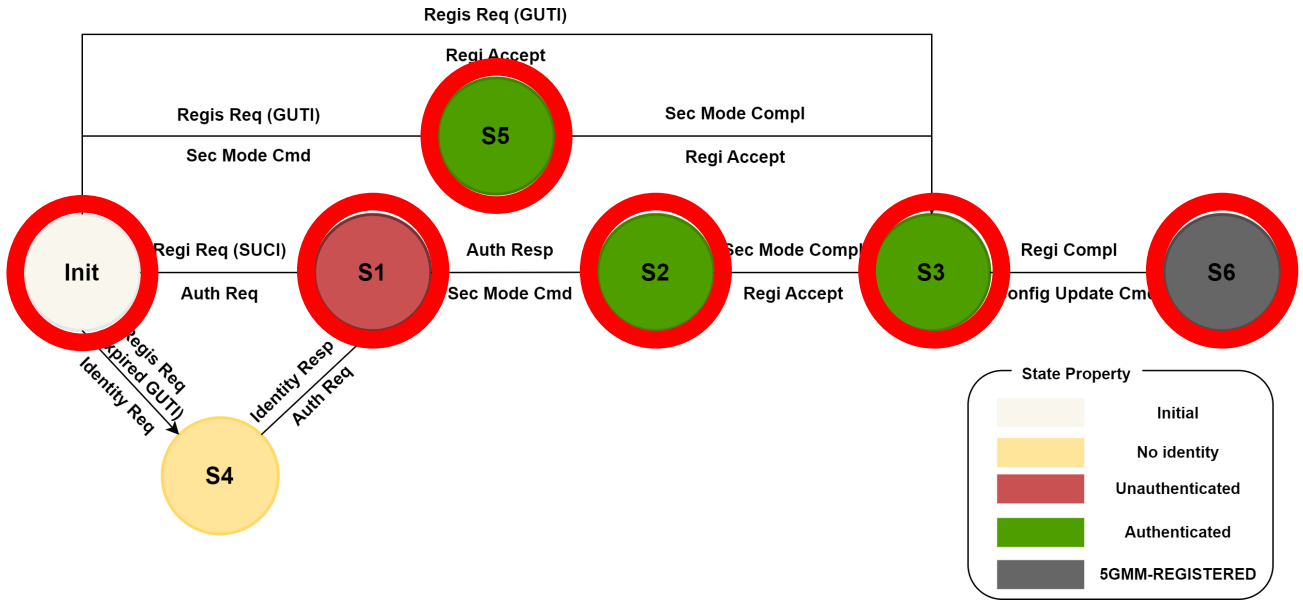
Select the Valid States

- Replay: check sequence number
  - Can test before obtaining the new security context



Select the Valid States

- MODIFY (capability)
  - States that can send unprotected message that contain capability
  - Ex) Init
- REPLAY
  - States that do not have a new security context
  - Ex) S1, S3, S6
- MODIFY (protected message)
  - States that contain security context
  - Ex) Init, S2, S3, S5
- INJECT
  - State that have valid identity
  - Ex) S1, S2, S3, S5, S6

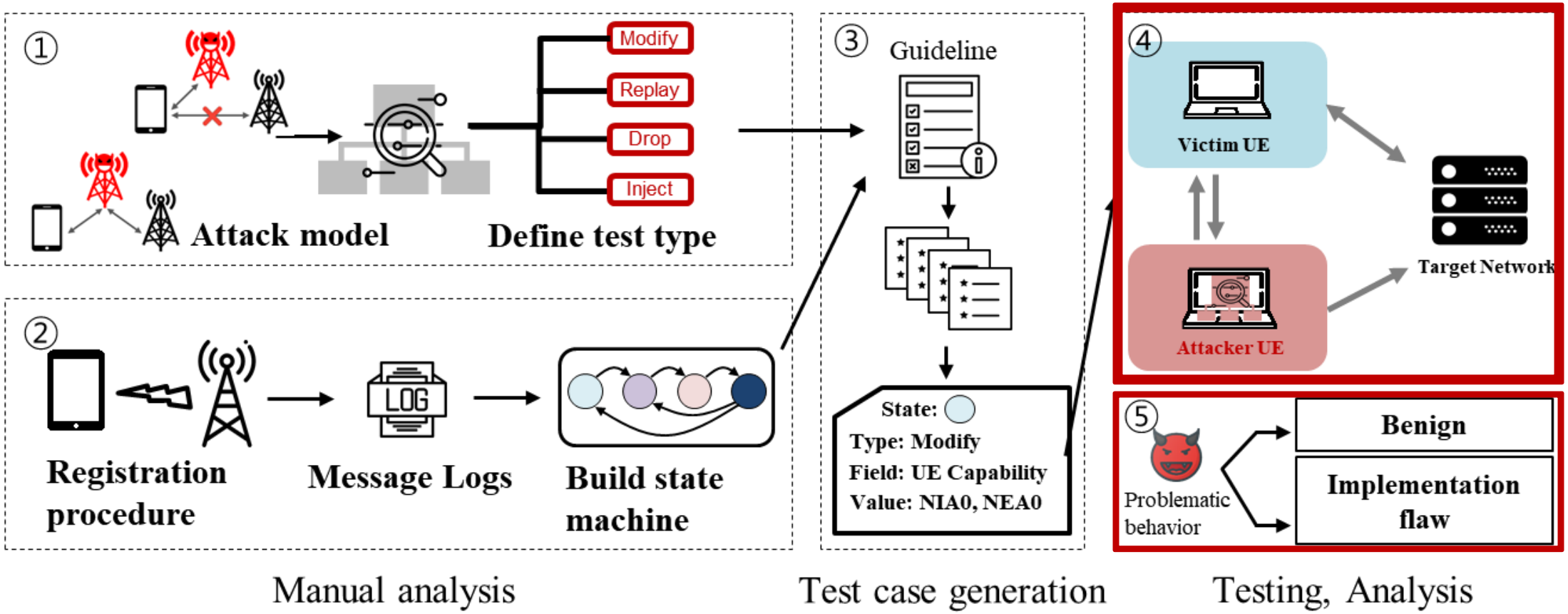


Generate Test Cases

- MODIFY (capability)
  - States that can send unprotected message that contain capability
  - Ex) Init
- REPLAY
  - States that do not have a new security context
  - Ex) S1, S3, S6
- MODIFY (protected message)
  - States that contain security context
  - Ex) Init, S2, S3, S5
- INJECT
  - State that have valid identity
  - Ex) S1, S2, S3, S5, S6

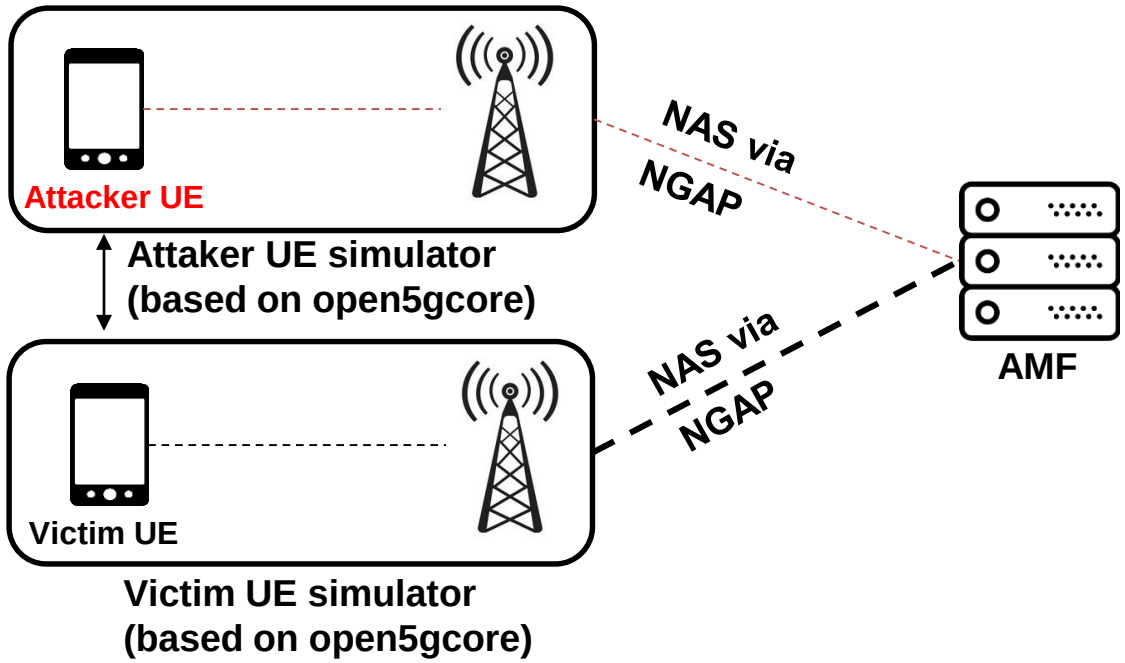
state	Test type			
	Inject	Modify		Replay
		Capability	Message	
Init	-	O	O	-
S1	O	-	-	O
S2	O	-	O	-
S3	O	-	O	O
S4	-	-	-	-
S5	O	-	O	-
S6	O	-	-	O

Test and Analyze



Experiment Setup & Implementation

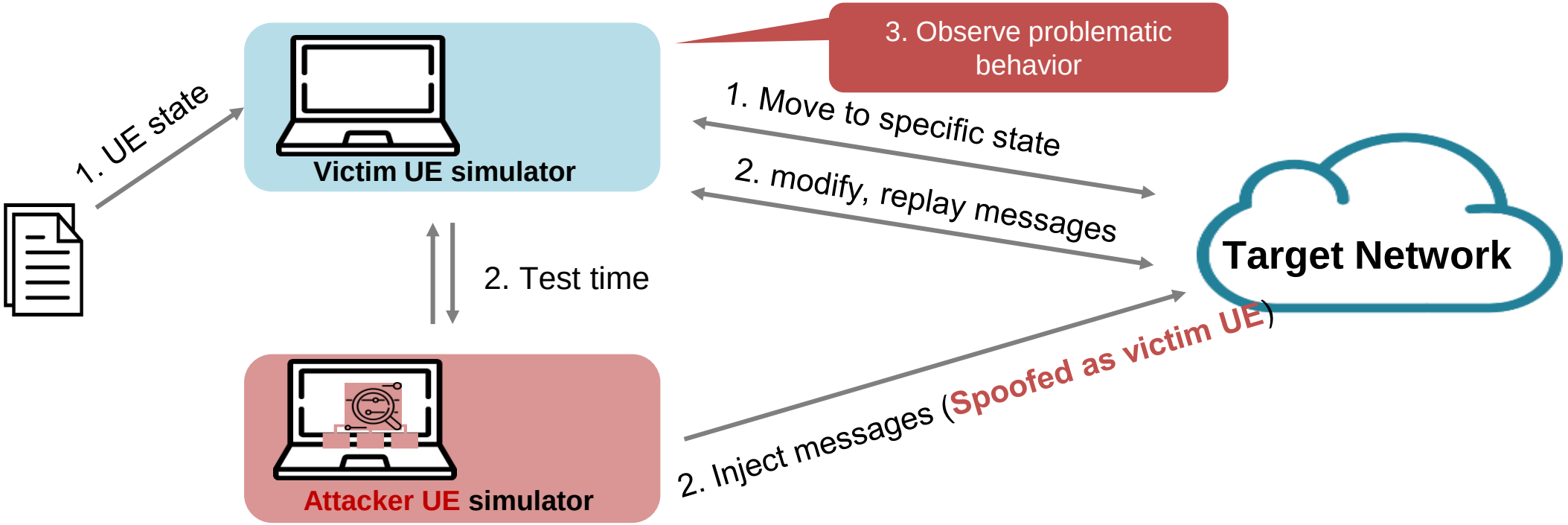
- Tester
  - UE-gNB simulator based on Open5GCore
  - Support 3 type scenarios.
    - Modify, Replay, Inject
  - Automate a stateful testing



- Target Core Network (AMF)
  - Open source project: Open5GS
  - Commerical equipment: Amarisoft
  - Operator's vendor: $Vendor_1$, $Vendor_2$

Testing

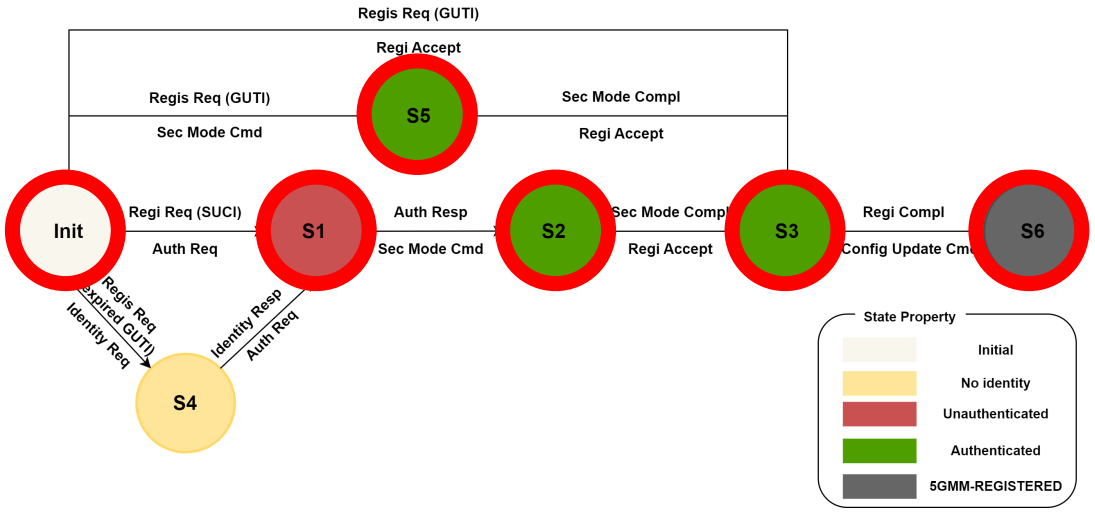
1. Move the victim UE to targeted state
2. Execute test scenario (Inject, modify, replay)
3. Observe problematic behavior and logging
4. message, state mutation (1~3 repeat)



Result – Open Source

- We tested a total of **1155** messages in Open5GS
  - Found 12 implementation flaws
- -: not supported, X: Benign

state	Open5GS		
	Inject	Modify	Replay
Init	-	NEA0 allowed, Invalid MAC allowed	-
S1	Victim UE's Connection release	-	Victim UE's Connection release
S2	Victim UE's Connection release	X	-
S3	Victim UE's Connection release	Plain allowed, Invalid MAC allowed	Victim UE's Connection release
S4	-	-	-
S5	Victim UE's Connection release	X	-
S6	Victim UE's Connection release	-	Victim UE's Connection release

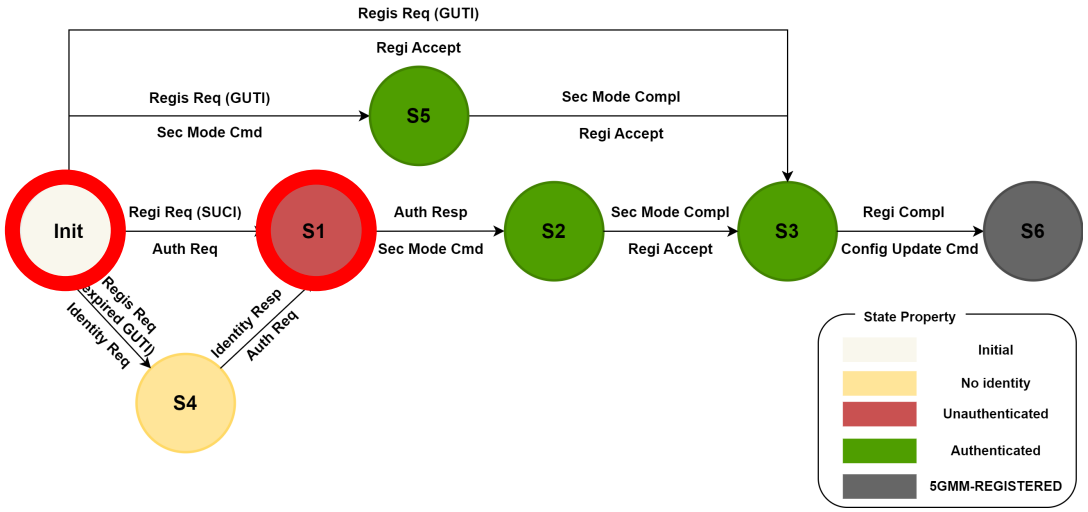


SYSSEC KAIST

Result – Commercial Equipment

- We tested a total of **1155** messages in Amarisoft
  - Found 3 implementation flaws
- -: not supported, X: Benign

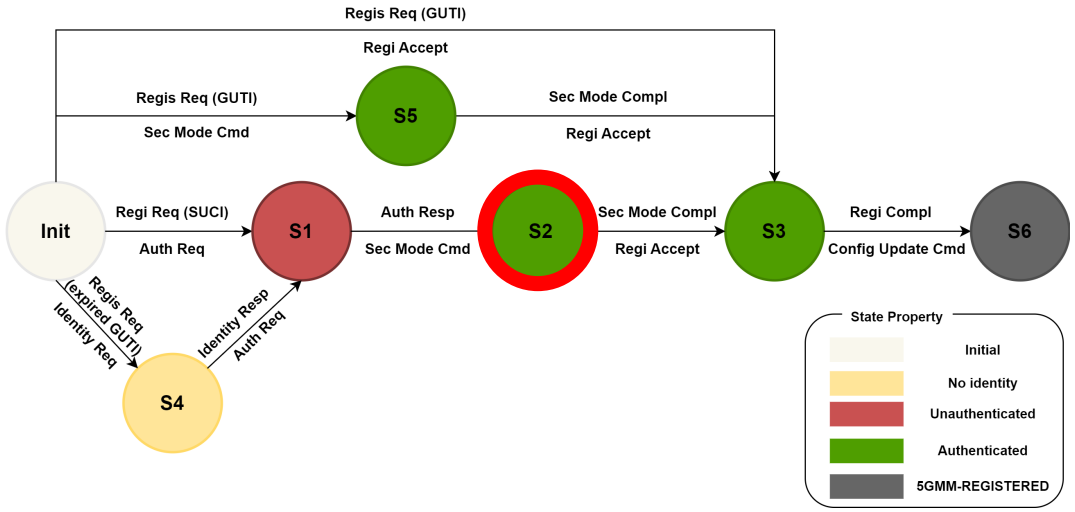
state	Amari		
	Inject	Modify	Replay
Init	-	NIA0, NEA0 allowed	-
S1	Victim UE's Connection release	-	Victim UE's Connection release
S2	X	X	-
S3	X	X	X
S4	-	-	-
S5	X	X	-
S6	X	-	X



Result – Operator's Vendor

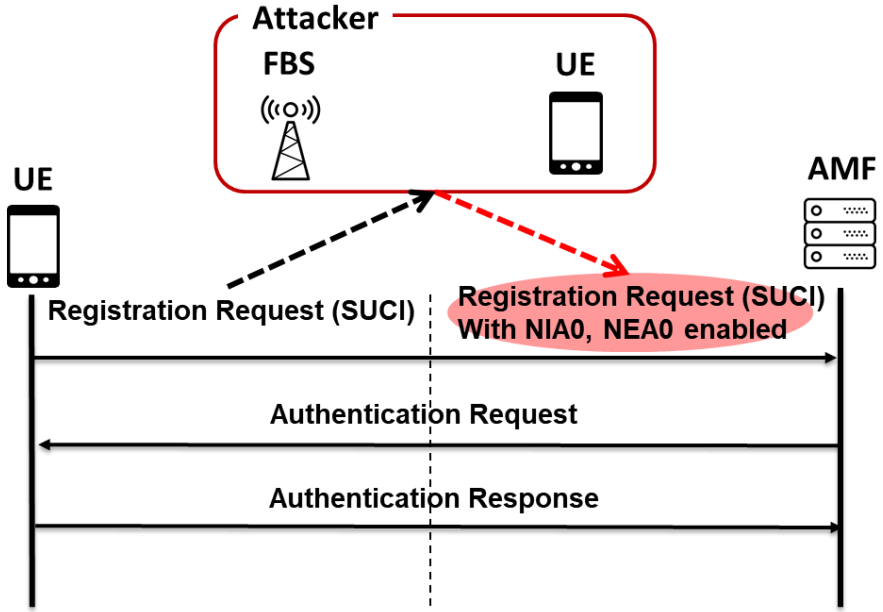
- We tested only **Inject** type in two networks
  - a total of **96** messages in $Vendor_1$
  - a total of **480** messages in $Vendor_2$
- Found 1 implementation flaw

state	$Vendor_1$	$Vendor_2$
	Inject	Inject
Init	-	-
S1	-	X
S2	Victim UE's Connection release	X
S3	-	X
S4	-	-
S5	-	X
S6	-	X



Result – Modify

- Registration request with modified NEA, NIA algorithms.
- Detect invalid behaviors
  - Amarisoft: NIA0, NEA0 allowed
  - Open5GS: NEA0 allowed




```
Message type: Registration request (0x41)
> 5GS registration type
> NAS key set identifier
> 5GS mobile identity
v UE security capability
    Element ID: 0x2e
    Length: 4
    1... .... = 5G-EA0: Supported
    .0.. .... = 128-5G-EA1: Not supported
    ..0. .... = 128-5G-EA2: Not supported
    ...0 .... = 128-5G-EA3: Not supported
    .... 0... = 5G-EA4: Not supported
    .... .0.. = 5G-EA5: Not supported
    .... ..0. = 5G-EA6: Not supported
    .... ...0 = 5G-EA7: Not supported
    1... .... = 5G-IA0: Supported
```

```
Message type: Security mode command (0x5d)
v NAS security algorithms
    0000 .... = Type of ciphering algorithm: 5G-EA0 (null ciphering algorithm) (0)
    .... 0000 = Type of integrity protection algorithm: 5G-IA0 (null integrity protection
  0000 .... = Spare Half Octet: 0
```

Result – Modify

- In 5G, UE send the registration request in the **security mode complete.**
- Valid behaviors
  - If AMF received the NAS message container,
  - AMF shall consider the NAS message as the initial NAS message that triggered the procedure
    - Have to restart the security mode command.

```
InitialUEMessage, Registration request
DownlinkNASTransport, Authentication request
UplinkNASTransport, Authentication response
DownlinkNASTransport, Security mode command
UplinkNASTransport, Security mode complete, Registration request
InitialContextSetupRequest, Registration accept
```

When the AMF receives an integrity protected initial NAS message which includes a NAS message container IE, the AMF shall decipher the value part of the NAS message container IE. If the received initial NAS message is a REGISTRATION REQUEST message or a SERVICE REQUEST message, the AMF shall consider the NAS message that is obtained from the NAS message container IE as the initial NAS message that triggered the procedure.

Result – Replay

- Capture a valid **Deregistration request**
- Replay the message when victim UE receives the Registration accept
- Detect invalid behaviors
  - Open5GS: **release** the victim's connection

```
InitialUEMessage, Registration request
InitialContextSetupRequest, Registration accept
InitialUEMessage, Deregistration request (UE originating)
InitialContextSetupResponse
UplinkNASTransport, Registration complete
UEContextReleaseCommand
UEContextReleaseComplete
DownlinkNASTransport, Deregistration accept (UE originating)
```

Replay

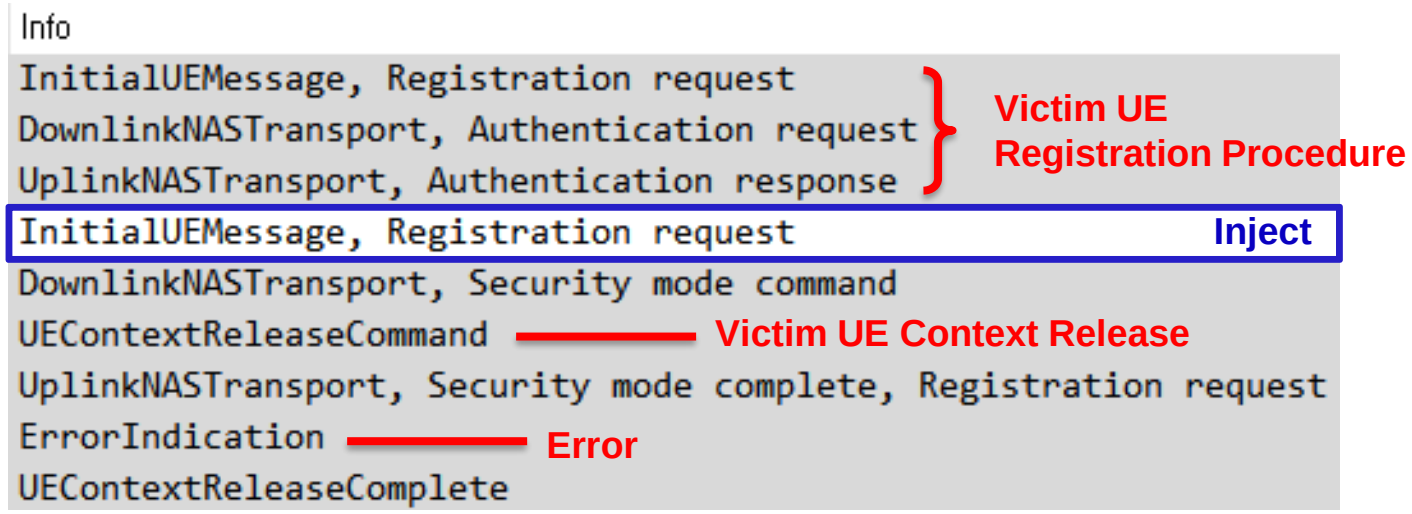
```
Item 0: id-RAN-UE-NGAP-ID
  ProtocolIE-Field
      id: id-RAN-UE-NGAP-ID (85)
      criticality: reject (0)
    value
      RAN-UE-NGAP-ID: 97
```

```
    id: id-UE-NGAP-IDs (114)
    criticality: reject (0)
  value
    UE-NGAP-IDs: uE-NGAP-ID-pair (0)
      uE-NGAP-ID-pair
          aMF-UE-NGAP-ID: 162
          rAN-UE-NGAP-ID: 97
```

Result – Inject

- Inject a message that contain **victim's identity**

- Detect invalid behaviors
  - Vendor$_1$: **release** the victim's connection



DoS Attack - Inject

- Additional scenario
  - Induce T3502 using the attempt counter
  - DoS attacks for default 12 min.

c) T3510 timeout.

The UE shall abort the registration update procedure and the N1 NAS signalling connection, if any, shall be released locally.

If the UE has initiated the registration procedure in order to enable performing the service request procedure for emergency services fallback, the UE shall inform the upper layers of the failure of the emergency services fallback (see 3GP P TS 24.229 [14]). Otherwise, the UE sha

d) REGISTRATION REJECT message, other 5GMM cause va
cases of 5GMM cause values #11, #22, #31, #72, #73, #74,
according to subclause 5.5.1.3.5.

For the cases c, d and e the UE shall proceed as follows:

Timer T3510 shall be stopped if still running.

The registration attempt counter shall be incremented, unless it was already set to 5.

TIMER NUM.	TIMER VALUE	STATE	CAUSE OF START	NORMAL STOP	ON EXPIRY
T3502	Default 12 min. NOTE 1	5GMM-DEREGISTERED 5GMM-REGISTERED	At registration failure and the attempt counter is equal to 5	Transmission of REGISTRATION REQUEST message	Initiation of the registration procedure, if still required

Limitations

- Only NAS protocol
  - Currently, the implementation of the lower layer is not proper for testing the core network, so it is difficult to test the RRC layer.
- Incomplete state machine
  - It is difficult to build complete state machine because we only considered UE-side log.
- Tests were not fully performed in operator's vendor

Conclusion

- In this work, first uplink stateful testing was performed to resolve the previous work's limitation in 5G SA network.

- As a result, we found 16 implementation flaws in the 4 networks.
  - 1 from commercial equipment ($Vendor_1$)
  - 3 from commercial equipment for research (Amarisoft)
  - 12 from open source projects (Open5GS)

- Problems vary depending on implementations.
  - Different networks have different vulnerabilities.
  - Carriers can identify the problem of the equipment without the vendor’s source code.

Thank you