Mobile Network Security

Anomaly Detector in 5G Core Network

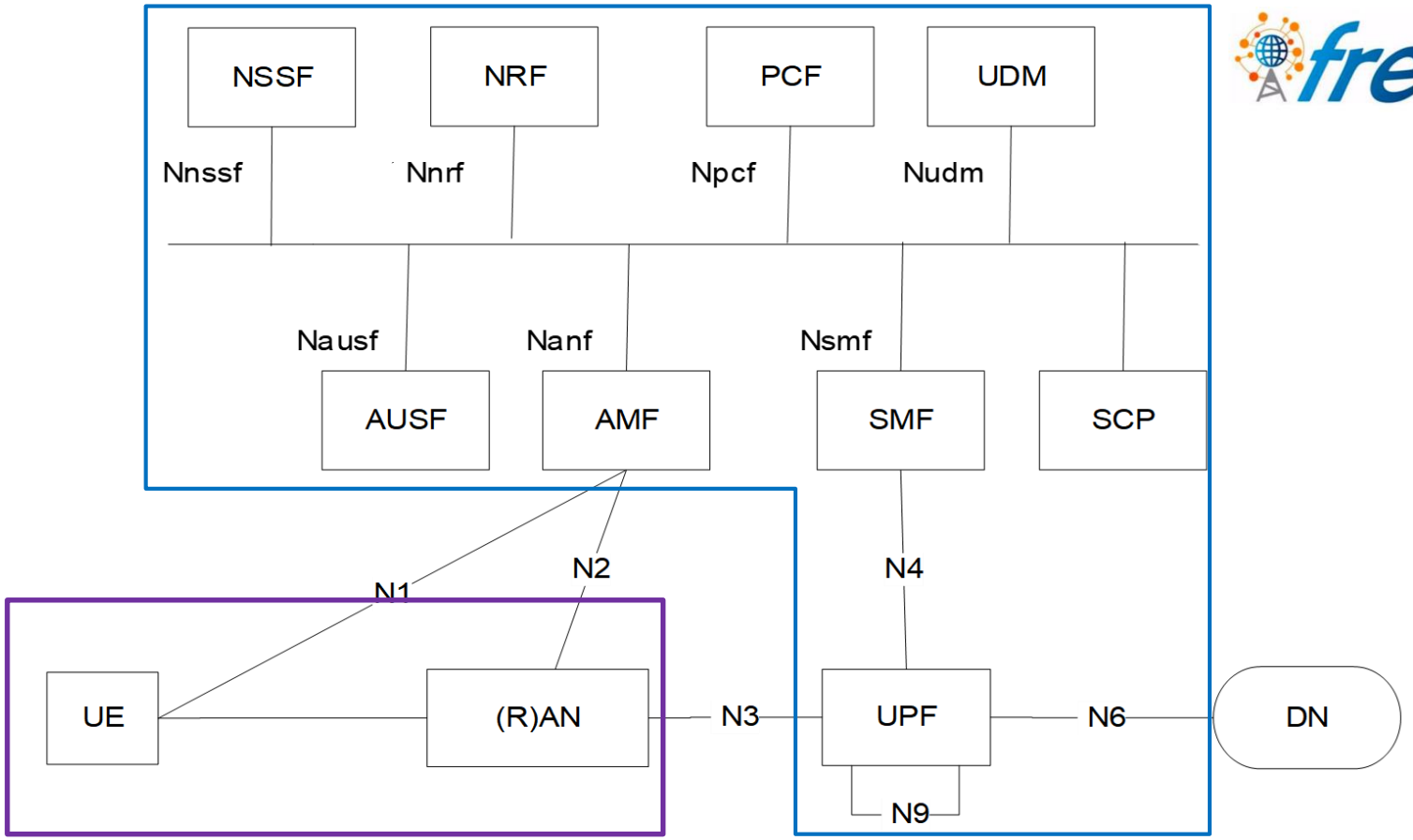
Chi-Yu Li (2022 Spring)
Computer Science Department
National Yang Ming Chiao Tung University

Goals

- Understand the procedure of 5G AKA authentication

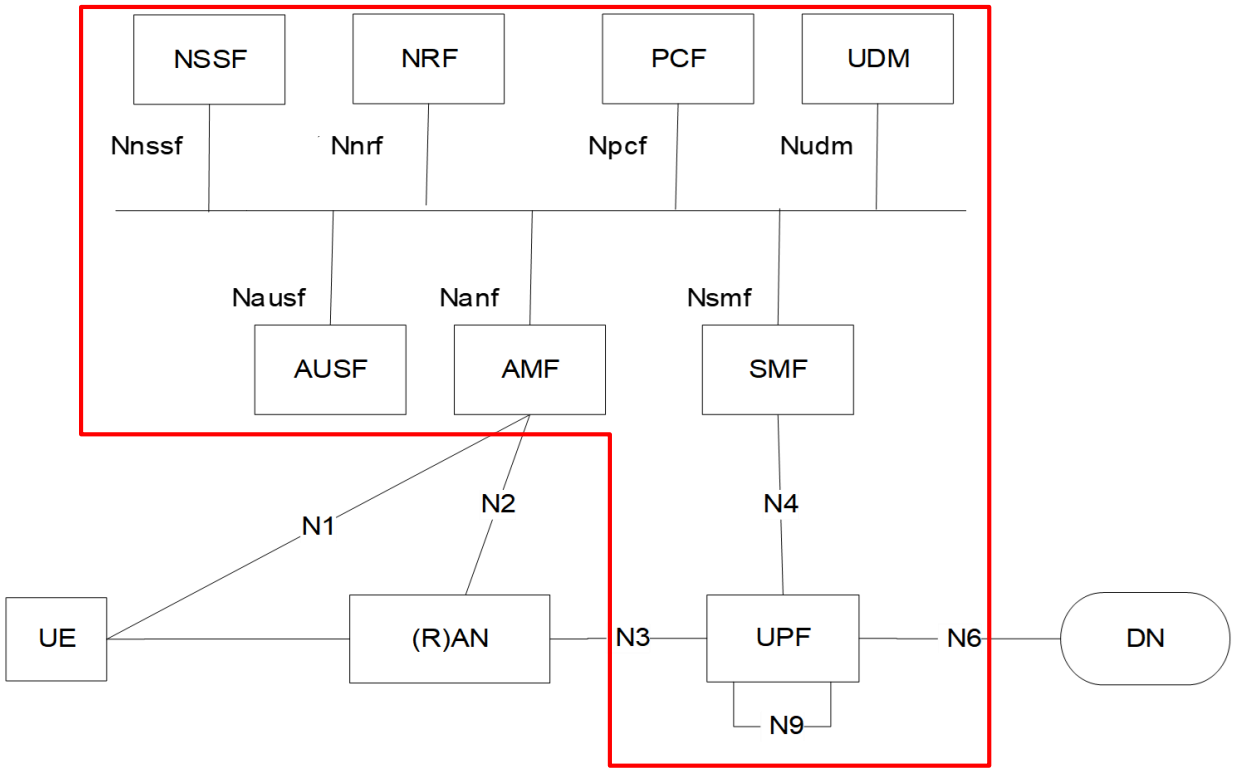
- You will learn
  - 5G AKA authentication
  - 5G SBA operation
  - free5GC
  - golang programming
  - reading 3GPP Spec

5G Testbed


NSSF
NRF
PCF
UDM
Nnssf
Nnrf
Npcf
Nudm
Nausf
Nanf
Nsmf
AUSF
AMF
SMF
SCP
N1
N2
N4
UE
(R)AN
N3
UPF
N6
DN
N9

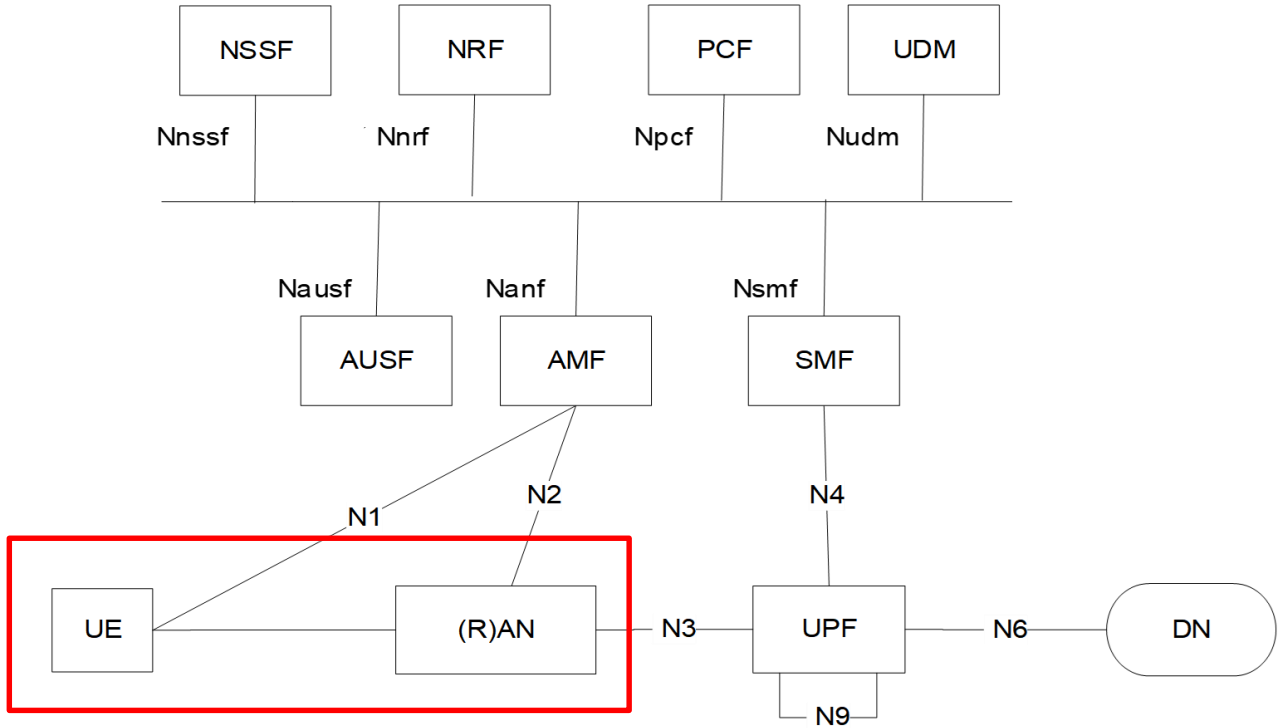
free5GC

- Open source 5G core network
  - Based on Release 15
  - https://github.com/free5gc/free5gc
  - https://www.free5gc.org/
- In this project, we use a modified version of free5GC

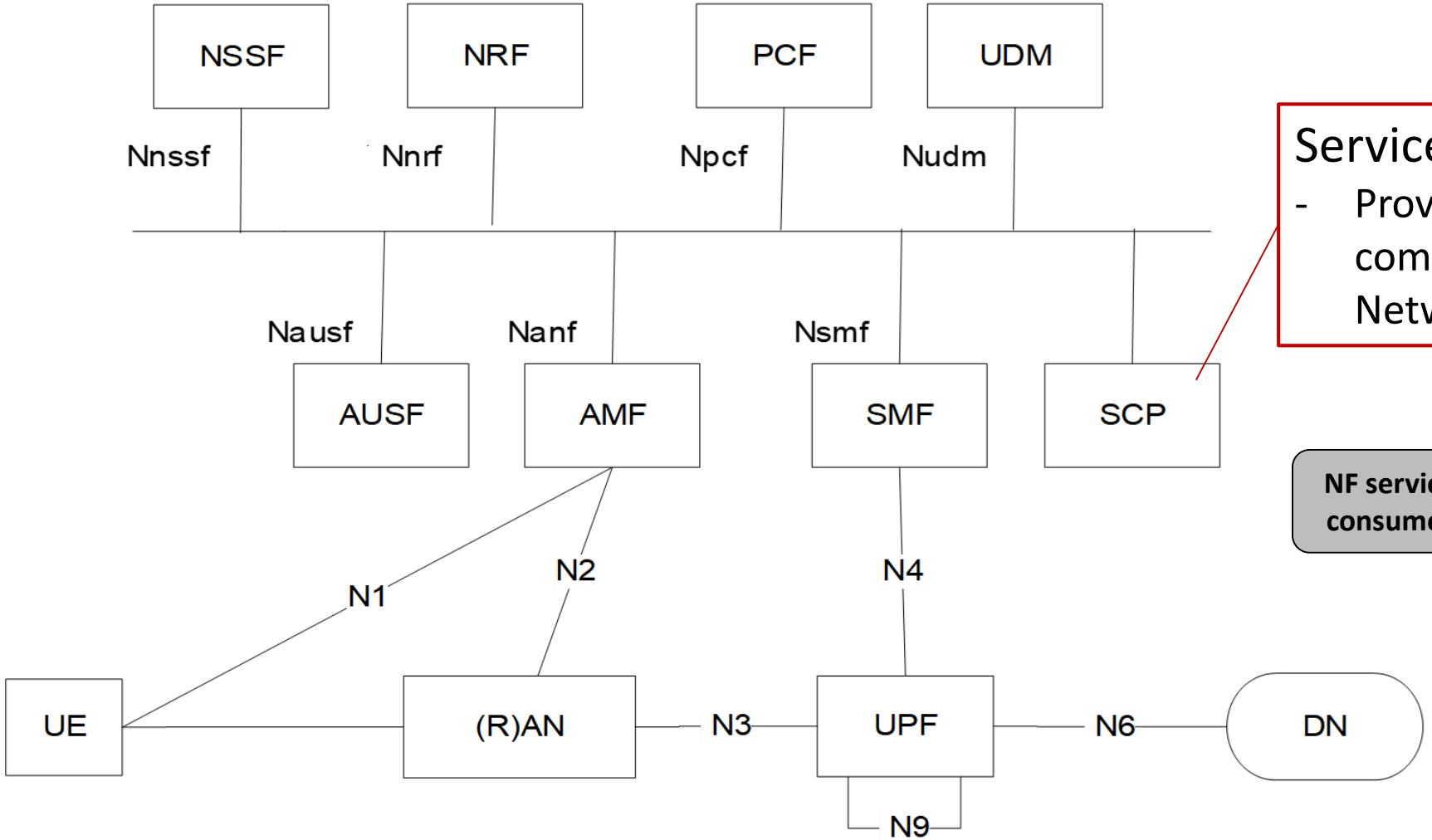


UERANSIM

- Open source 5G UE and RAN (gNodeB)
  - https://github.com/aligungr/UERANSIM



5G System Architecture with SCP

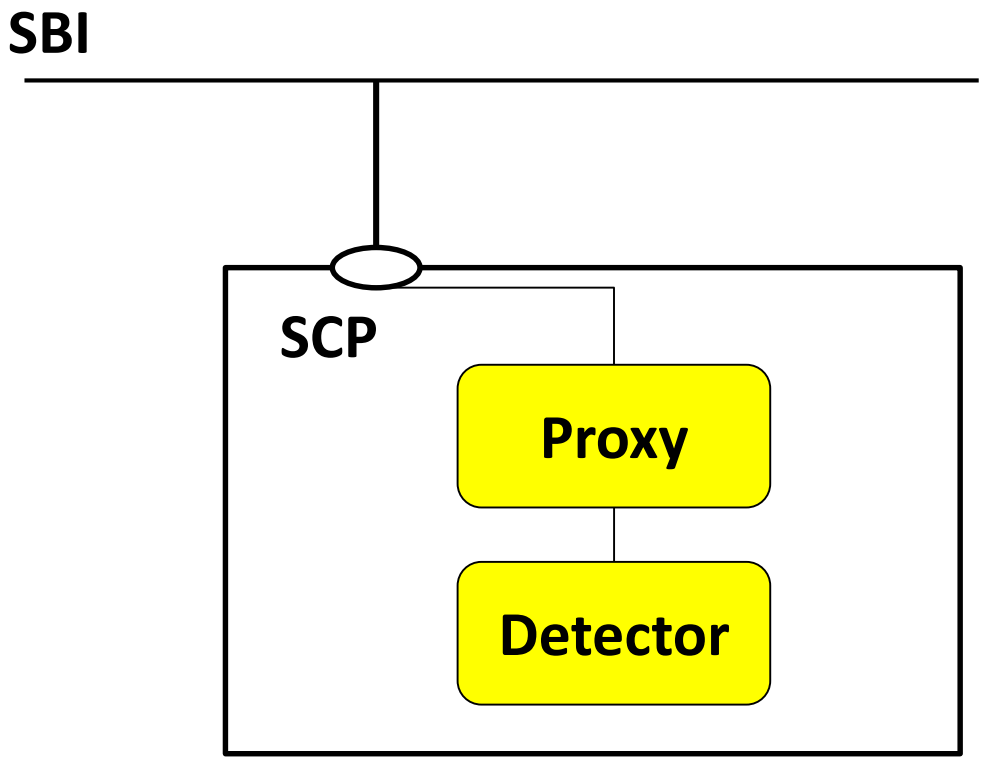


Service Communication Proxy (SCP)

- Providing indirect and secure communication proxy between Network Functions (NFs) in Release 16

SCP Architecture

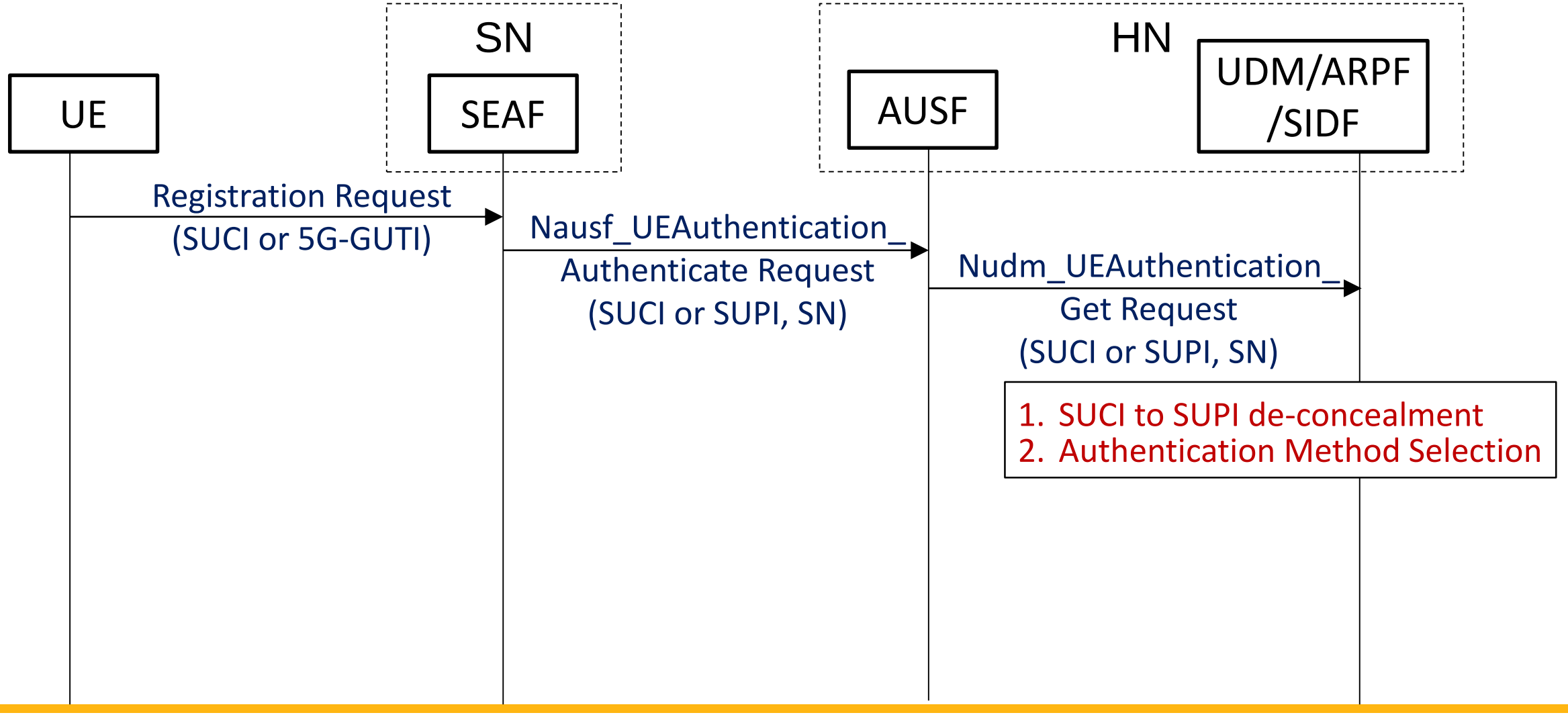
- We aim to develop an anomaly detector at SCP
  - SCP can monitor and filter all the forwarded messages
- Proxy
  - Forwarding SBI message to detector
  - Forwarding SBI message to target NF
- Detector
  - Detecting abnormal message
  - Recovering abnormal content



Main Features at SCP Detector

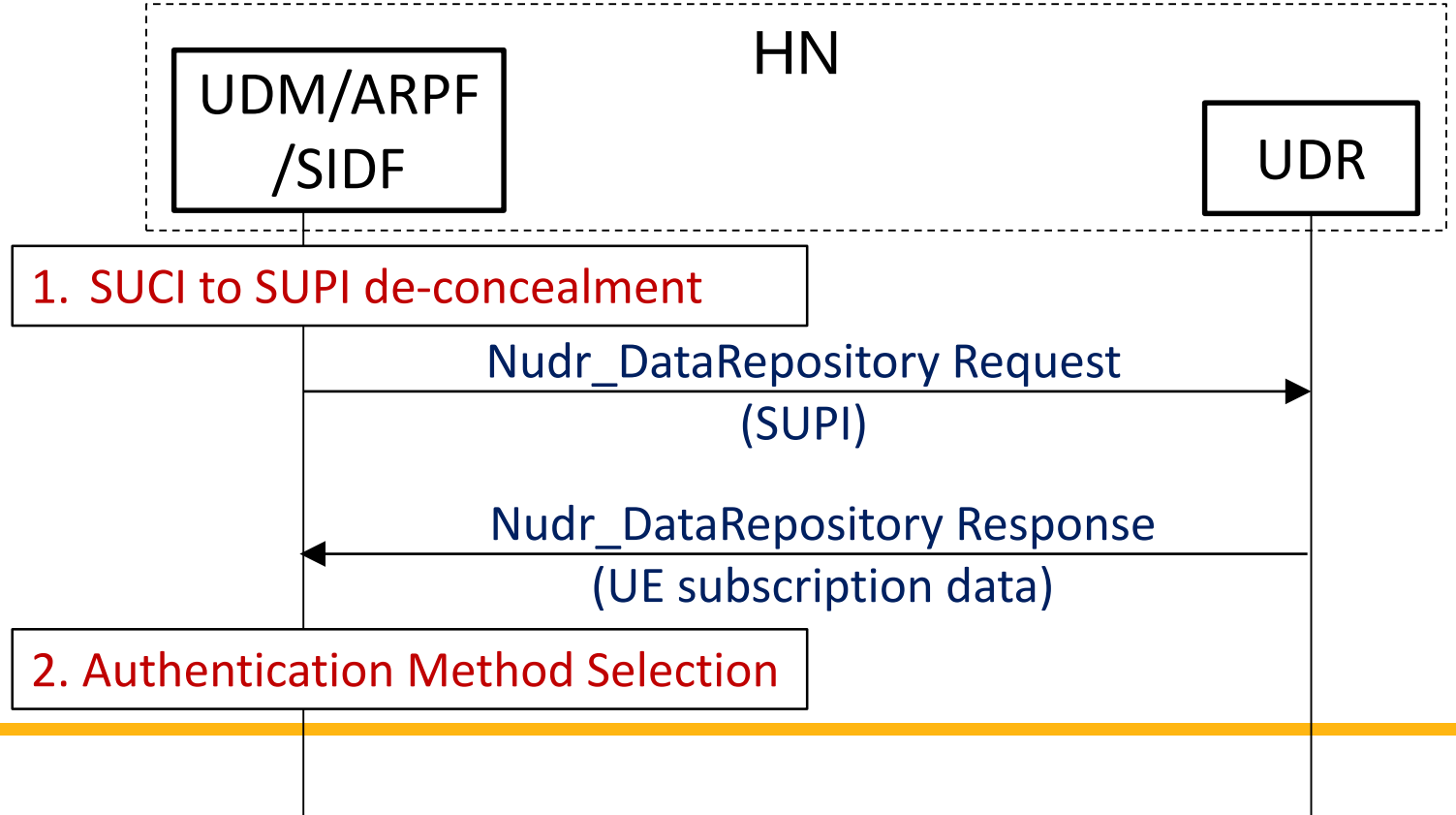
- Handling 5G AKA authentication procedure messages
  - Only authentication messages are sent to SCP
- Verifying correctness of messages
  - Including all the Information Elements (IE) in authentication messages
- Recovering problematic messages
  - Only NF binaries are given in this project

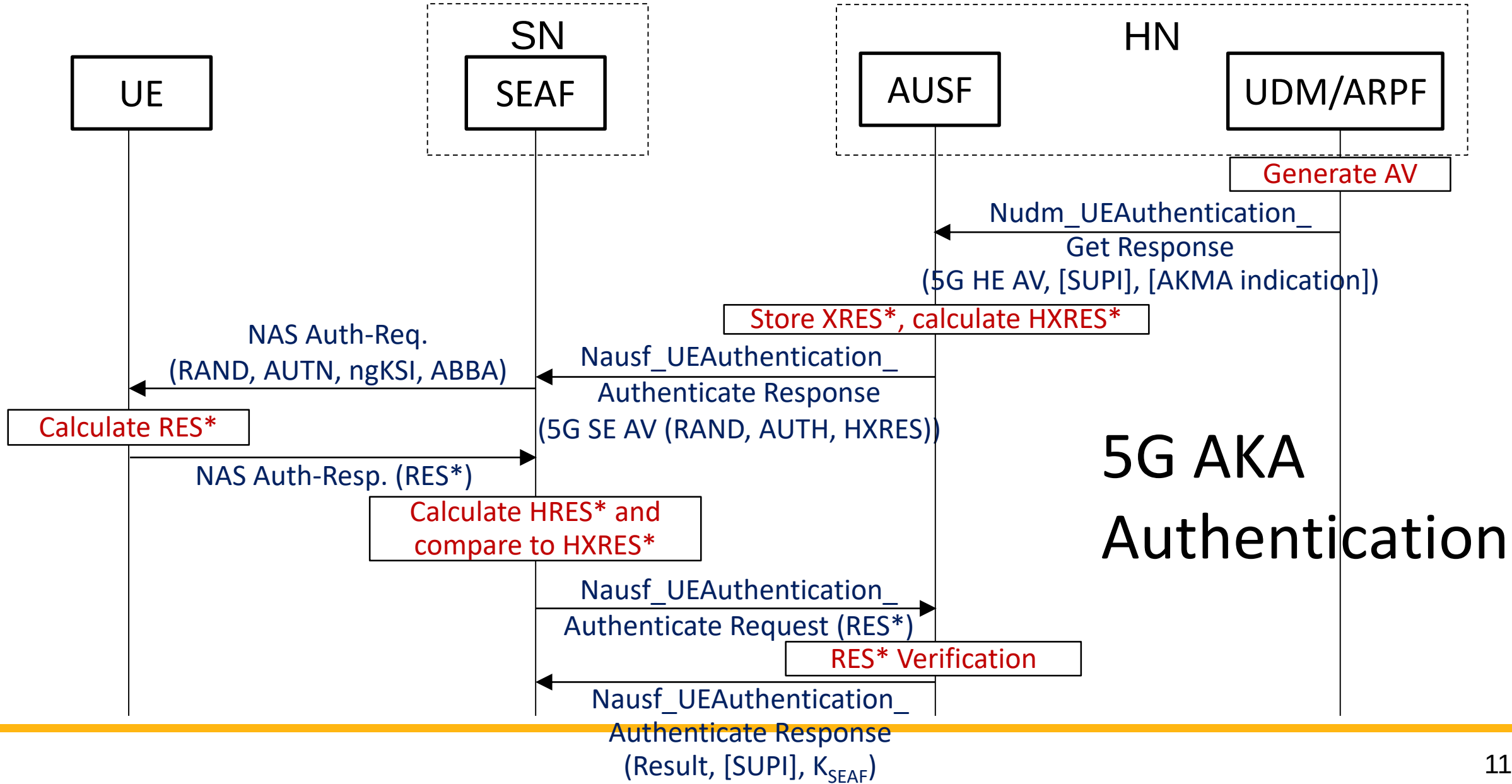
5G AKA Authentication: Initiation



5G AKA Authentication (cont.)

- Getting UE authentication subscription data from UDR
  - UDM doesn't have UE subscription data in memory





5G AKA Authentication

Tasks

- Task I: Authentication messages forwarding (50%)
  - Forwarding authentication messages to correct NFs
- Task II: Detecting abnormal messages (30%)
  - Abnormal messages include
    - missing mandatory IE
    - incorrect IE value
    - mismatch conditional IE
- Task III: Recovering abnormal messages (20%)
  - Using given functions to obtain correct IE values

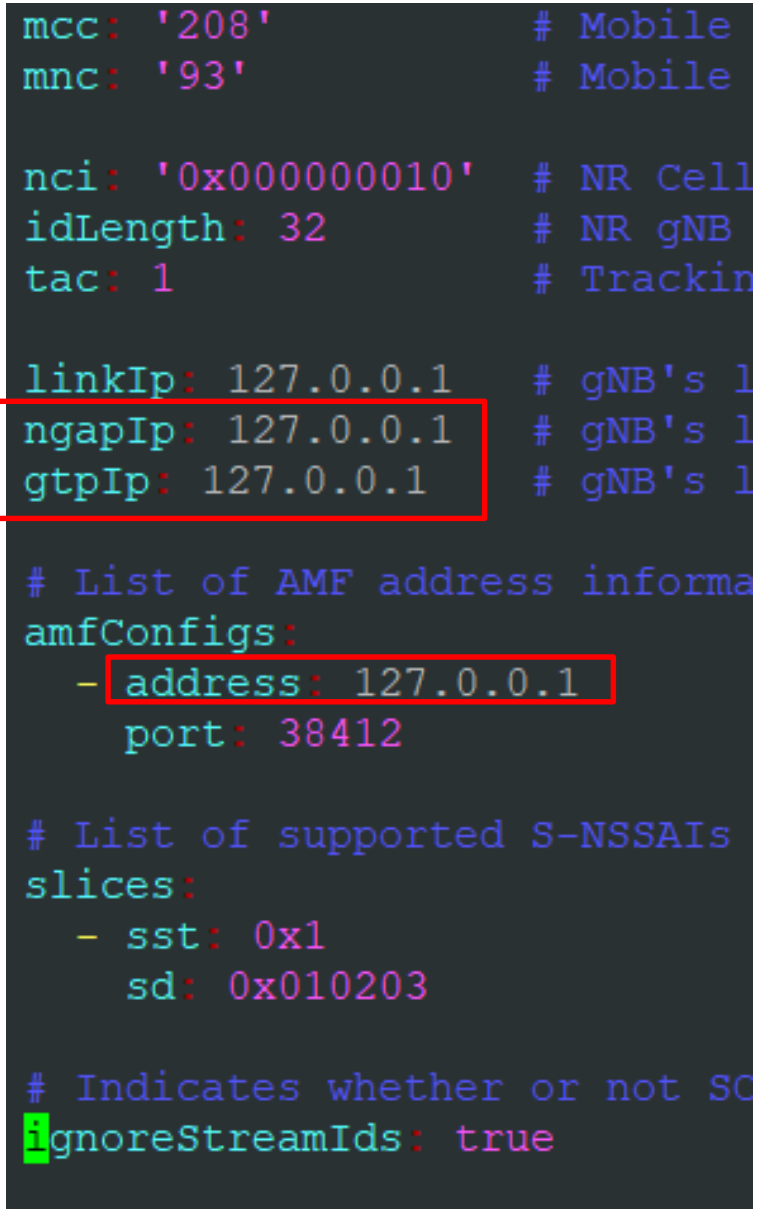
Environment Setup

- Download a given VM image from Link
  - Building 2 machines: UE/RAN and 5G Core Network
  - VirtualBox is recommended
  - Login account: mns2022/mns2022

- 5GC VM network config
  - Interface1: NAT
  - Interface2: Host-Only
- UE VM network config
  - Interface1: Host-Only

Environment Setup (cont.)

- Edit UERANSIM config file in UE VM
  - File path: ~/UERANSIM/config/free5gc-gnb.yaml
- Attributes need be modified
  - Set “ngapIp”, “gtpIp” to IP of UE VM
    - Should be the IP of Host-only Interface
    - e.g., 192.168.56.102
  - Set “amfConfigs”→ “address” to IP of 5GC VM
    - Should be the IP of Host-only Interface
    - e.g., 192.168.56.101



Observation of Normal 5GC Operation

- Start normal 5GC without SCP
  - command: ~/project1/run.sh
- Connect UE to 5GC
  - command @UE VM
    - First, start gnb: ~/UERANSIM/build/nr-gnb -c ~/UERANSIM/config/free5gc-gnb.yaml
    - Then, ue: ~/UERANSIM/build/nr-ue -c ~/UERANSIM/config/free5gc-ue.yaml
  - Authentication will succeed and UE can connect Internet
    - Testing with ‘ping -I uesimtun0 8.8.8.8’
- Observing normal operation
  - Normal authentication procedure packets
    - Captured at loopback interface using Wireshark
  - 5GC logs printed on screen

Observation of Abnormal 5GC Behavior

- Start buggy 5GC without SCP
  - command: ~/project1/run.sh --buggy
  - Abnormal IEs in authentication procedure messages
- Connect UE to 5GC
  - command @UE VM
    - First, start gnb: ~/UERANSIM/build/nr-gnb -c ~/UERANSIM/config/free5gc-gnb.yaml
    - Then, ue: ~/UERANSIM/build/nr-ue -c ~/UERANSIM/config/free5gc-ue.yaml
  - Authentication will fail, so UE can’t connect Internet
- Observing abnormal operation
  - Abnormal authentication procedure packets
    - Captured at loopback interface using Wireshark
  - 5GC logs printed on screen

Observation of Normal 5GC Operation with SCP

- Start normal 5GC with SCP
  - command: ~/project1/run.sh --with-scp
- Connect UE to 5GC
  - command @UE VM
    - First, start gnb: ~/UERANSIM/build/nr-gnb -c ~/UERANSIM/config/free5gc-gnb.yaml
    - Then, ue: ~/UERANSIM/build/nr-ue -c ~/UERANSIM/config/free5gc-ue.yaml
  - Authentication will fail, so UE can’t connect Internet
- Observing abnormal operation
  - SCP logs printed on screen
  - 5GC logs printed on screen

SCP Detector Development

- Service messages need to be handled
  - {apiRoot}/nausf-auth/v1/ue-authentications
  - {apiRoot}/nudm-ueau/v1/{supiOrSuci}/securityinformation/generate-auth-data
  - {apiRoot}/nudr-dr/subscription-data/{ueid}/authentiocation-data/authentication-subscription
  - {apiRoot}/nausf-auth/v1/ue-authentications/{authCtxId}/5g-aka-confirmation
- Assume the following messages and IEs are correct
  - Messages from two NFs, AMF and UDR
  - IE rand from UDM
  - IE ausfInstanceId, authResult, _links from AUSF

SCP Detector Development (cont.)

- Related files
  - handler.go, derivation.go, and context.go
- ~/project1/scp/detector/handler.go
  - Checking and recovering messages
  - Set target uri to forward message requests
  - Complete TODO parts in the files

```
detector/
├── context.go
├── derivation.go
└── handler.go
```

```go
func HandleUeAuthPostRequest(request *http_wrapper.Request) *http_wrapper.Response {
    logger.DetectorLog.Infof("HandleUeAuthPostRequest")
    updateAuthenticationInfo := request.Body.(models.AuthenticationInfo)

    // NOTE: The request from AMF is guaranteed to be correct

    // TODO: Send request to target NF by setting correct uri
    targetNfUri := ""

    response, respHeader, problemDetails, err := consumer.SendUeAuthPostRequest(targetNfUri, &updateAuthenticationInfo)

    // TODO: Check IEs in response body is currect
```

SCP Detector Development (cont.)

- ~/project1/scp/detector/derivation.go
  - SUCI decryption function
  - Key derivation functions
  - See 33.501 Annex A for function input information

- ~/project1/scp/detector/context.go
  - Providing a global variable to store message information

```
// Define every thing you want in this struct,
// so that you can use them in different message handler
type AuthProcedureInfo struct {
    AuthSubsData       models.AuthenticationSubscription
}
```

How to Test Your SCP Detector?

- Compiling SCP Detector
  - Write a makefile
  - Be sure your SCP binary is put in ~/project1/scp/bin and named as scp
- Starting abnormal 5GC and SCP
  - command: ~/project1/run.sh --with-scp --buggy
- Connect UE to free5GC
  - command @UE VM
    - First, start gnb: ~/UERANSIM/build/nr-gnb -c ~/UERANSIM/config/free5gc-gnb.yaml
    - Then, ue: ~/UERANSIM/build/nr-ue -c ~/UERANSIM/config/free5gc-ue.yaml

How to Test Your SCP Detector? (cont.)

- Check Internet reachability of UE
  - command: `ping -I uesimtun0 8.8.8.8`
- Check SCP detector output
  - Shall report found problems

Output Rules of SCP Detector

- Must use logger function with ‘error’ level
  - logger.DetectorLog.Errorln()
  - logger.DetectorLog.Errorf()
- Format: “<Fully-Qualified-Type-Name>: <Error message>”
  - <Fully-Qualified-Type-Name>: From top message IE type to member IE type
    - Connect each type name with ‘.’
    - Case insensitive
  - 3 Types of error messages
    - consts defined in handler.go is helpful
    - “mandatory type is absent"
    - “missing conditions"
    - “unexpected value is received"

Output Rules of SCP Detector (cont.)

- Sample Output

```
[ERRO][SCP][Detector] AuthenticationInfoRequest.ServingNetworkName: Mandatory type is absent

[ERRO][SCP][Detector] UeAuthenticationCtx.Av5gAka.HxresStar: Unexpected value is received

[ERRO][SCP][Detector] ConfirmationDataResponse.Kseaf: Miss condition
```

Needed 5G Specification

- 3GPP TS 33.501 (Security architecture and procedures for 5G System): Sections 6.1, Annex A
  - Message flows of UE authentication
  - Jobs of NFs in UE authentication
  - Annex A is for key derivation functions
- 3GPP TS 29.503 (Unified Data Management Services): Sections 5.4 and 6.3
  - UDM service used in UE authentication
  - Definition of UDM service message structure
- 3GPP TS 29.509 (Authentication Server Services): Sections 5.2 and 6.1
  - Definition of AUSF service message structure

Other 5G Specification

- 3GPP TS 29.505(Usage of the Unified Data Repository services for Subscription Data): Section 5.2.2, 5.4
  - For the information of response from UDR in authentication procedure
- 3GPP TS 29.571(Common Data Types for SBI)
  - Common data type definition used in SBI
- 3GPP TS 29.501(Principles and Guidelines for Services Definition): Section 5.2
  - SBI API definition
  - Helpful to understand tables in specification

NFs IP Configuration

NFs	IP Configuration
AMF	127.0.0.18:8000
AUSF	127.0.0.9:8000
UDM	127.0.0.3:8000
UDR	127.0.0.4:8000
SCP (Detector)	127.0.0.113:8000

Project Submission

- Due date: 4/6 11:55pm

- Submission rules
  - Zip the whole directory scp/ and name it using your student ID
  - Upload the zip file to E3
  - A sample of the zip file: 01212112.zip

- No team up

Questions?