Network Security

Project 2:
Anomaly Detector in 5G Core Network

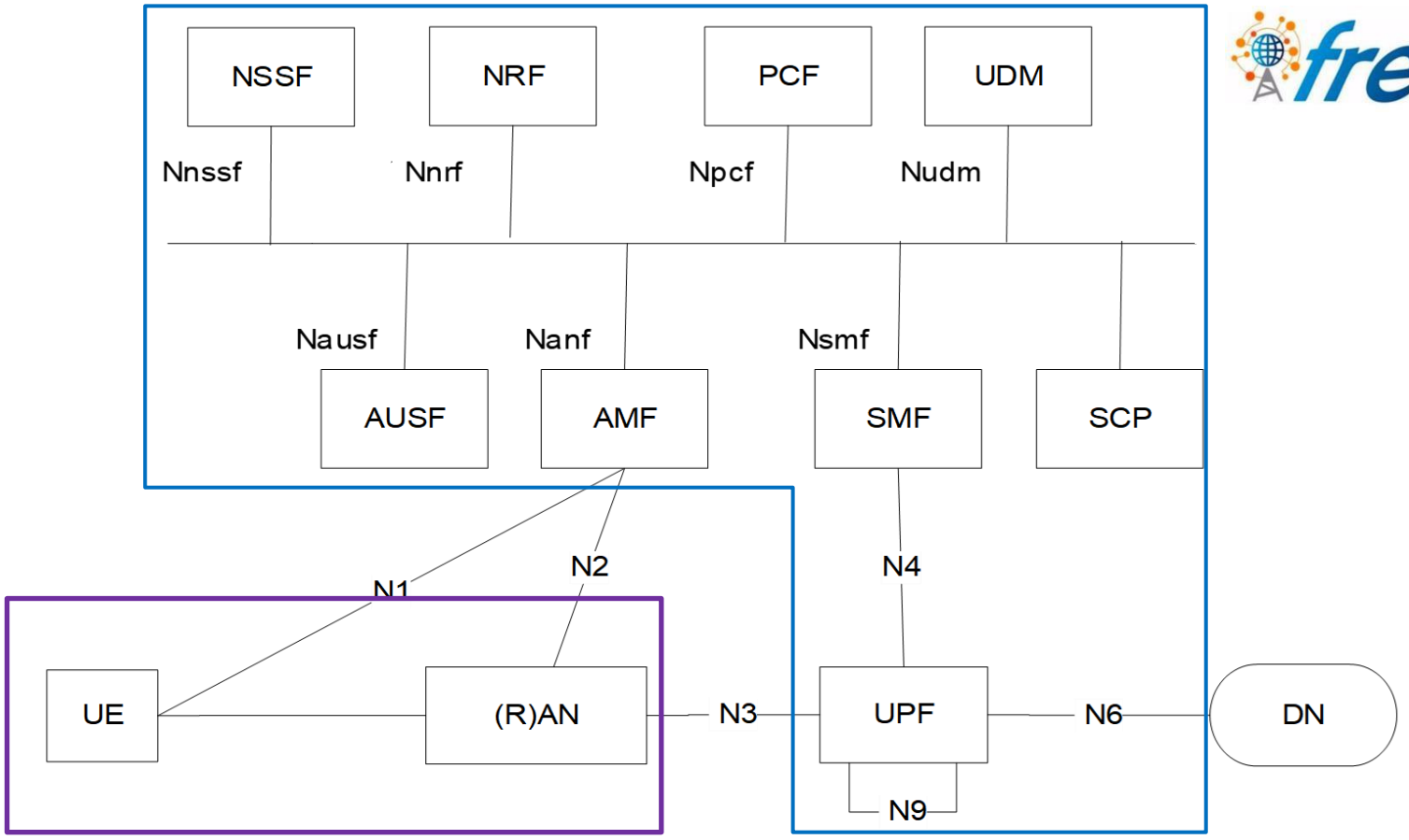
Chi-Yu Li (2024 Fall)
Computer Science Department
National Yang Ming Chiao Tung University

Goals

- Understand the procedure of 5G AKA authentication

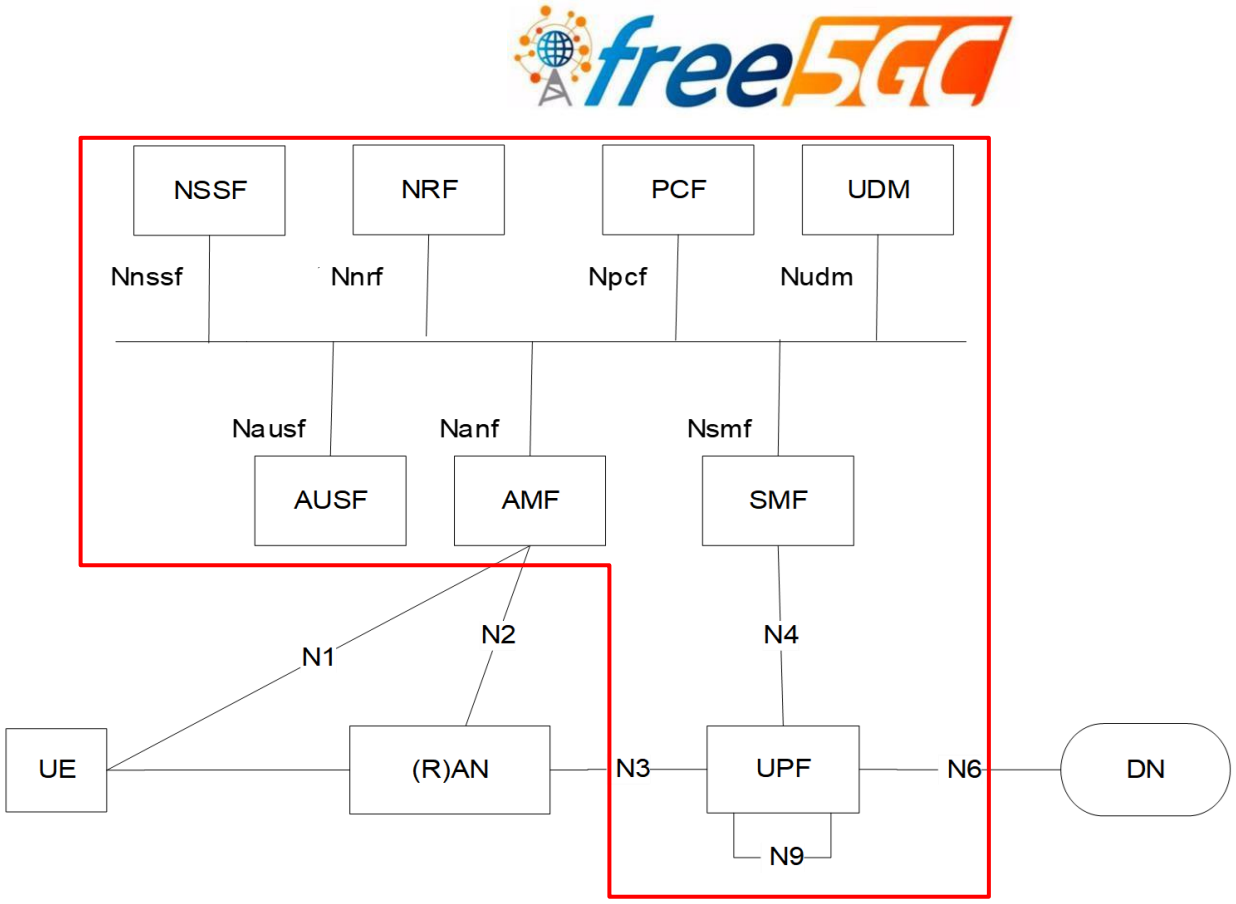
- You will learn
  - 5G AKA authentication
  - 5G SBA operation
  - free5GC with docker compose
  - golang programming
  - reading 3GPP Spec

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

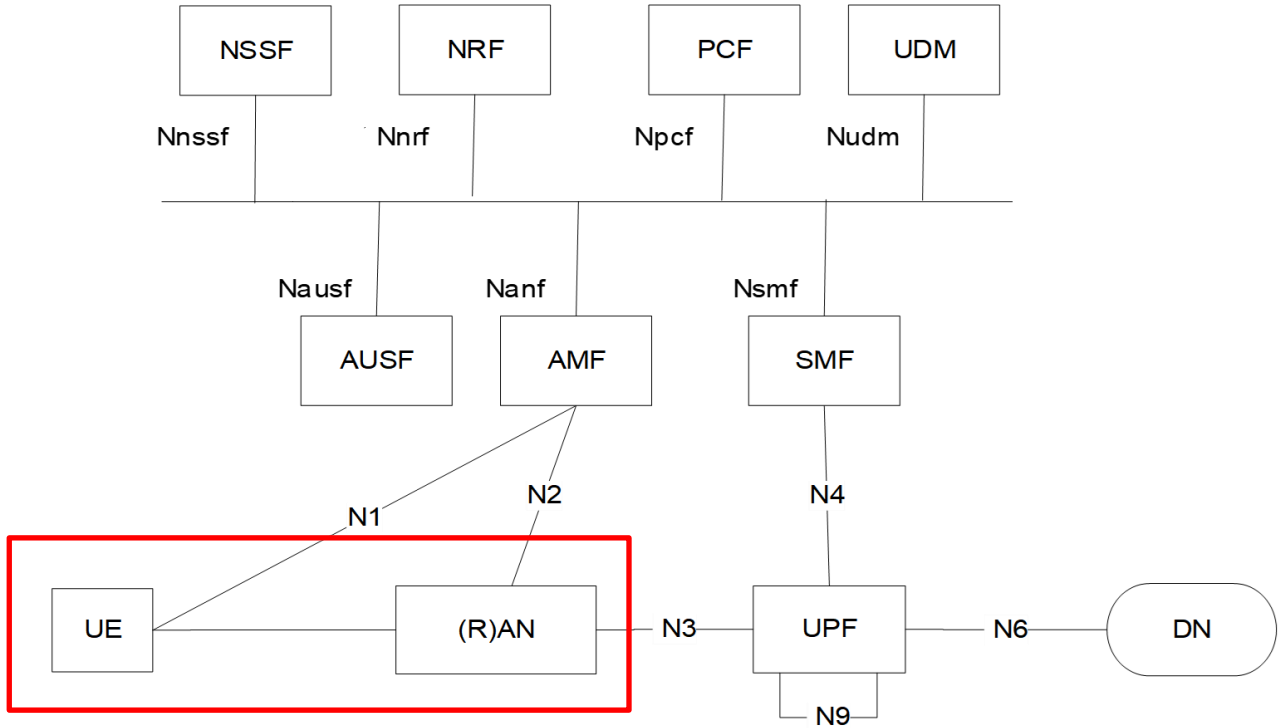
free5GC

- 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

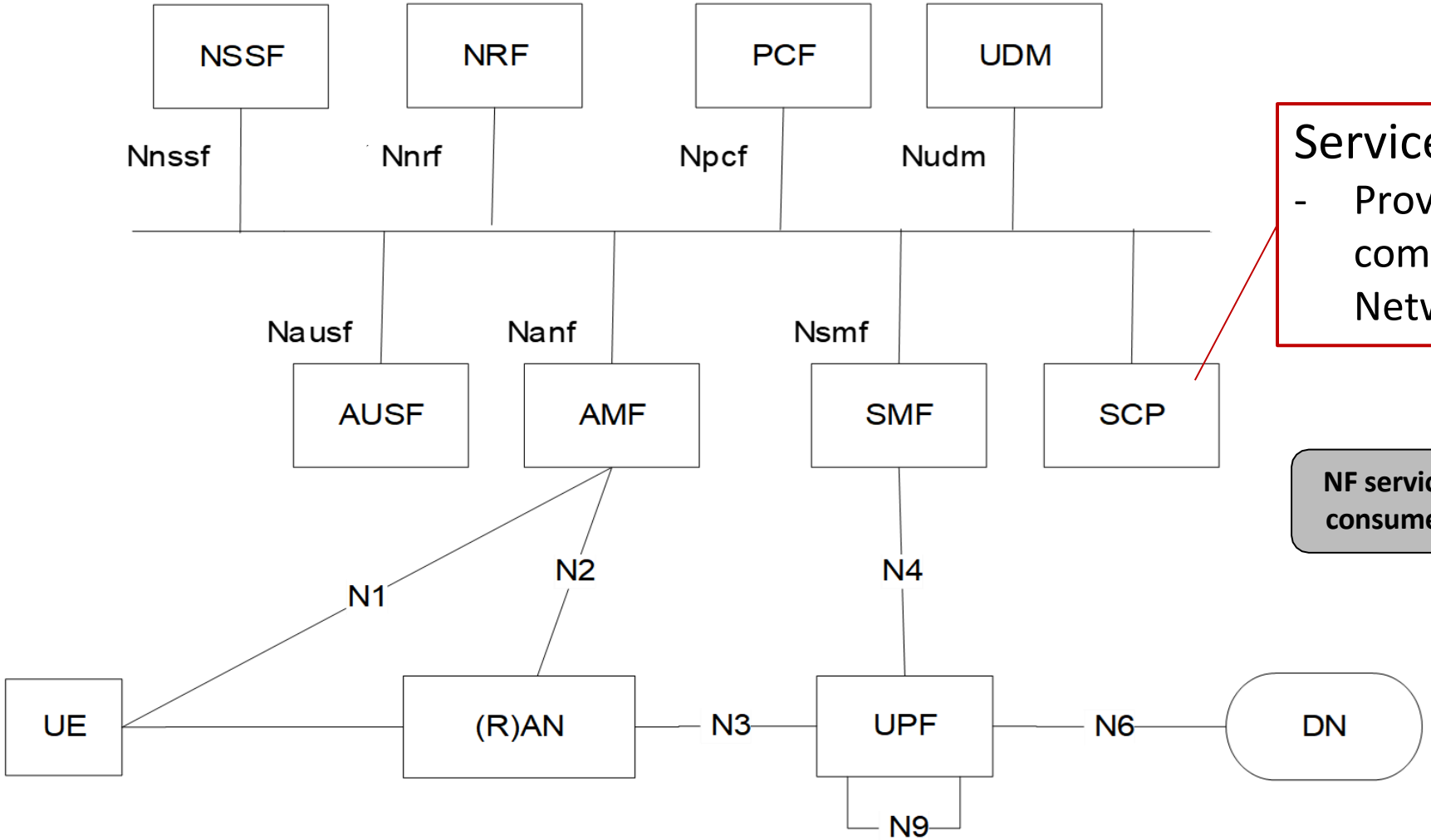


UERANSIM

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



5G System Architecture with SCP

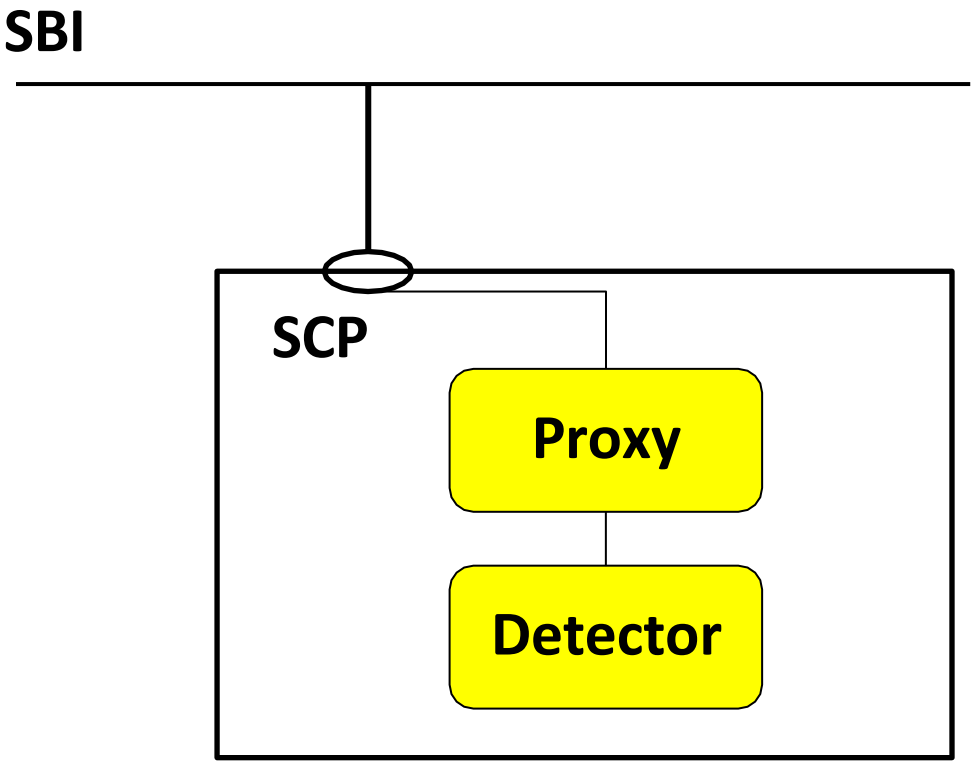


Service Communication Proxy (SCP)

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

SCP Architecture

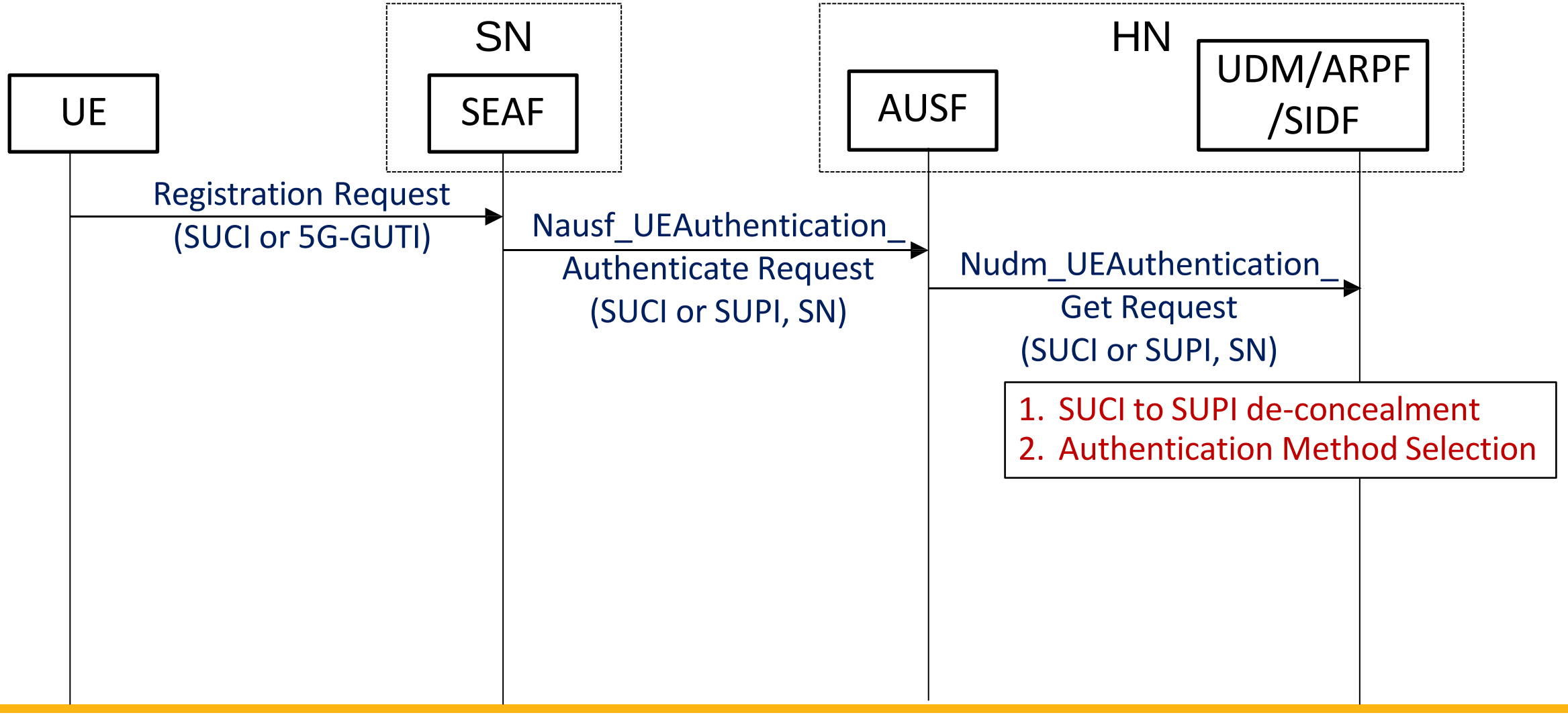
- We aim to develop an anomaly detector at SCP
  - SCP can monitor and filter all the forwarded messages
- Proxy
  - Forwarding SBI message to detector
  - Forwarding SBI message to target NF
- Detector
  - Detecting abnormal message
  - Recovering abnormal content



Main Features at SCP Detector

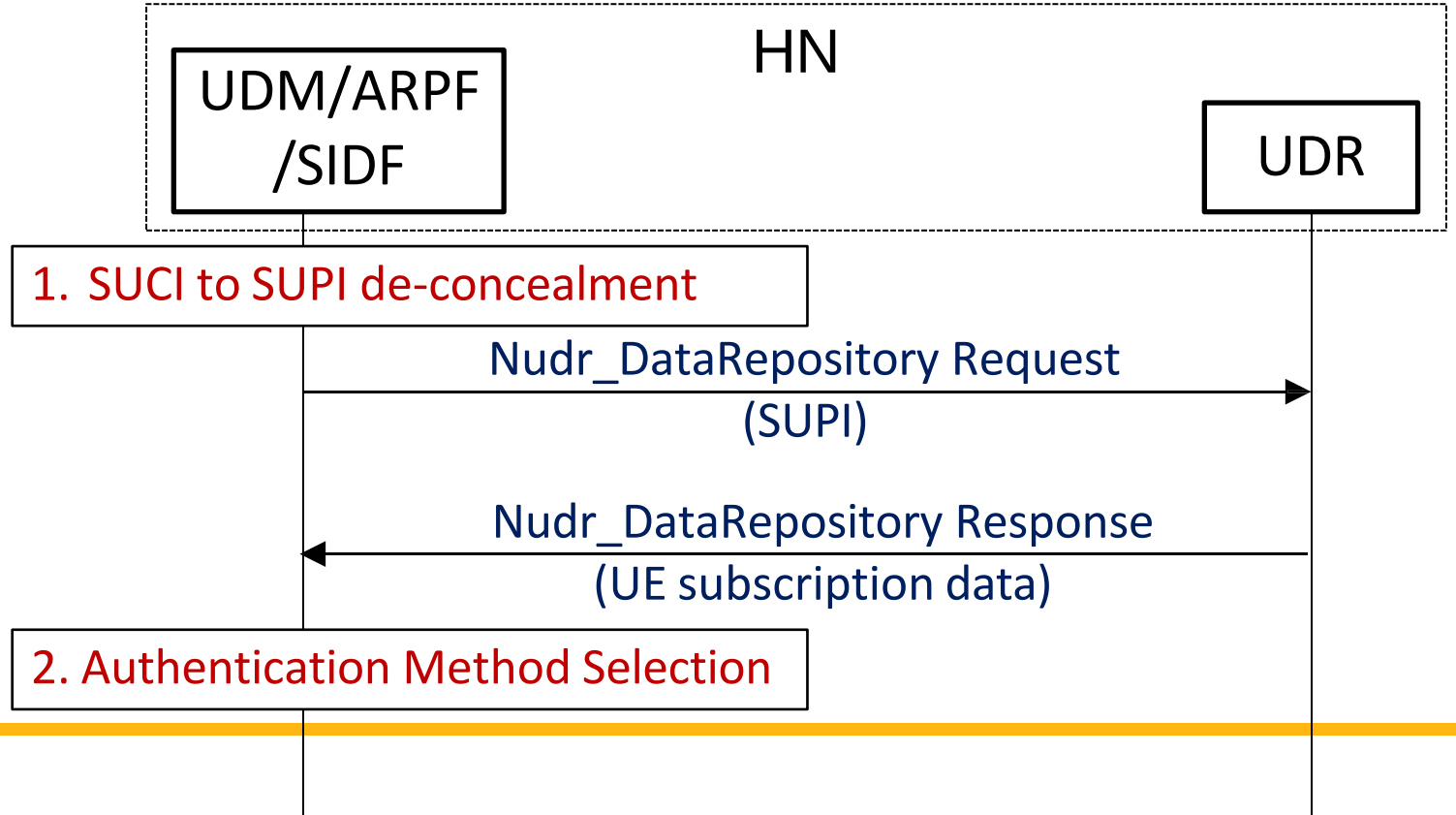
- Handling 5G AKA authentication procedure messages
  - Only authentication messages are sent to SCP
- Verifying the correctness of messages
  - Including all the Information Elements (IEs) in authentication messages
- Recovering problematic messages
  - Only NF images are given in this project

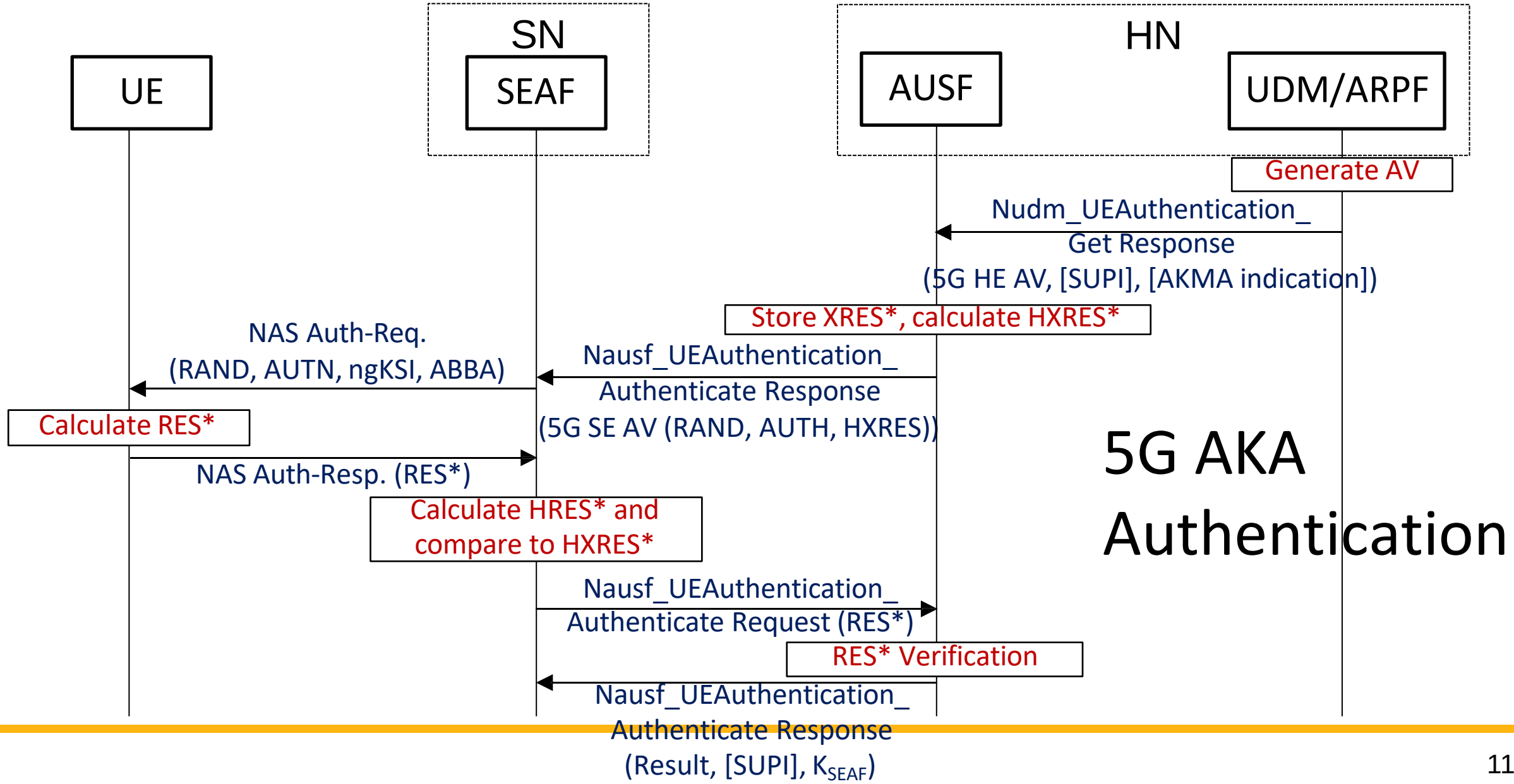
5G AKA Authentication: Initiation



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 the supplement from E3
  - Follow the step in the README.md to do the environment setup
  - Once the prerequisites are fulfilled and a new UE is successfully added in the Webconsole, you can enter the bash of UERANSIM's container to make it connect to the free5GC to observe the normal 5GC operation
  - If the setup is successful, the authentication will pass, and UE can establish the PDU session to connect to the internet

```
[debug] PDU Session Establishment Accept received
[info] PDU Session establishment is successful PSI[1]
[info] Connection setup for PDU session[1] is successful, TUN interface[uesimtun0, 10.60.0.1] is up.
[debug] PDU Session Establishment Accept received
[info] PDU Session establishment is successful PSI[2]
[info] Connection setup for PDU session[2] is successful, TUN interface[uesimtun1, 10.61.0.1] is up.
```

SCP Detector Development

- Four service messages need to be handled
  - {apiRoot}/nausf-auth/v1/ue-authentications
  - {apiRoot}/nudm-ueau/v1/{supiOrSuci}/security-information/generate-auth-data
  - {apiRoot}/nudr-dr/subscription-data/{ueId}/authentication-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
  - Rand from UDM
  - ausfInstanceId from AUSF

SCP Detector Development (cont.)

- TODO files
  - ausf_service.go, udm_service.go, udr_service.go (internet/sbi/consumer)
  - ausfueauth.go, udmueauth.go, udrauthsubdata.go (internet/sbi/processor)
- internet/sbi/consumer
  - Call the NF's API with OAuth access token to do the authentication messages forwarding

```go
func (s *nausfService) SendUeAuthPostRequest(uri string,
    authInfo *models.AuthenticationInfo) (*models.UeAuthenticationCtx, *models.ProblemDetails, error) {

    client := s.getUEAuthenticationClient(uri)
    if client == nil {
        return nil, nil, openapi.ReportError("ausf not found")
    }

    // TODO: OAuth AUSF Ue Auth Post
    var ueAuthenticationCtx models.UeAuthenticationCtx
    return &ueAuthenticationCtx, nil, nil
```

SCP Detector Development (cont.)

- internet/sbi/processor
  - Set the target URI for calling the NF's API
  - Detect and recover abnormal messages
  - Utilize the util.go to do the derivation of information elements (IEs) in authentication messages

```go
// NOTE: The response from AMF is guaranteed to be correct
func (p *Processor) PostUeAutentications(
    authInfo models.AuthenticationInfo,
) *HandlerResponse {
    logger.ProxyLog.Debugln("[AMF->AUSF] Forward AMF UE Authentication Request")

    // TODO: Send request to target NF by setting correct uri
    var targetNfUri string

    // TODO: Verify that the Information Elements (IEs) in the response body are correct
    //       Recover and handle errors if the IEs are incorrect
    response, problemDetails, err := p.Consumer().SendUeAuthPostRequest(targetNfUri, &authInfo)
```

How to Test Your SCP Detector?

- Compile and build the SCP Detector
  - Be sure your scp source file is put in base/free5gc/NFs
  - Working directory: supplement
  - Build: make scp && docker compose -f docker-compose-scp.yaml build
- Start free5GC with SCP
  - Normal case: ./run.sh --with-scp
  - Abnormal case: ./run.sh --buggy --with-scp
- Connect UE to free5GC
  - Command @UE container
    - First, enter ue bash: docker exec -it ueransim bash
    - Then, run ue: ./nr-ue -c config/uecfg.yaml

How to Test Your SCP Detector? (cont.)

- Check internet reachability of UE for task I and task III
  - ping -I uesimtun0 8.8.8.8
- Check SCP detector output for task II
  - Shall report found problems on the screen

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
  - <Error message>: 3 Types of error messages defined in util.go
    - "Mandatory type is absent"
    - "Miss condition"
    - "Unexpected value is received"

Output Rules of SCP Detector (cont.)

- Some sample outputs

```
[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 6.3
  - UDM service used in UE authentication
  - Definition of UDM service message structure
- 3GPP TS 29.509 (Authentication Server Services): Sections 6.1
  - Definition of AUSF service message structure

Other 5G Specification

- 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

Project Submission

- Due date: 12/11 11:55pm
- Makeup submission (75 points at most): TBA (After the final)
- Submission Rules
  - Put your source code files into a directory and name it using your student ID(s)
    - If your team has two members, please concatenate your IDs separated by “-”
  - Just zip the whole scp source file and upload the zip file to New E3 (only upload the scp source file)
  - A sample of the zip file: 01212112-02121221.zip
    - scp
  - If the scp source file are not in the working directory after unzip, 10 points will be deducted

Online Project Demo

- Demo data: 12/13
- TA will prepare your zip file and run your programs for the demo on behalf of you
  - TA will run your program in the same given supplement on E3
- You will
  - be asked some questions
  - be responsible to show and explain the outcome to the TA

Questions?