**S3-181394**
**Source:** Ericsson
**Title:** SBA: Example for SEPP Protection Policies
**Document for:** Information
**Agenda Item:** 7.2.13.1

# SEPP Protection Policies

This document has been presented at the SA3 conference call 2018-03-22 and is submitted to SA3#91 for information.

## Example and Motivation

# Reminder: Problem formulation

- Context: Application layer solution for protection on N32 interface

- Issue: a SEPP of a given release will not know which IEs to protect if the message belongs to a service that was introduced or modified later.
  - If this problem is not solved, it will put a severe restriction for SBA, as the idea behind SBA is that it should be easy to introduce new NFs and new services. If it is necessary to update the SEPP of both PLMNs for any new service that involves roaming, adding new services would be significantly harder.

# Reminder: SEPP Protection Policies

- Idea: For a newly introduced service and for every message included in this service, it is expressed by a policy which elements should be encrypted and/or integrity protected.
- Received positive feedback during SA3#90bis, and was included in the living document (S3-180888) as Solution #6, by S3-180897.
- However, it seems like clarifications are needed. Therefore, we present an example to show the usefulness of protection policies.

# Example: Setup

- The example shows how protection policies could work for the first interaction between AMF and AUSF for 5G-AKA authentication.

- Three steps:
  - Signaling content
  - Full signaling flows
  - Application of protection policies

# Example: Signaling content

- Actual content of the signaling, for the first interaction between AMF and AUSF for 5G-AKA authentication.

- Highlights in yellow the elements we need to protect
  - SUPI in the request
  - SUPI and AV in the response

vAMF

hAUSF

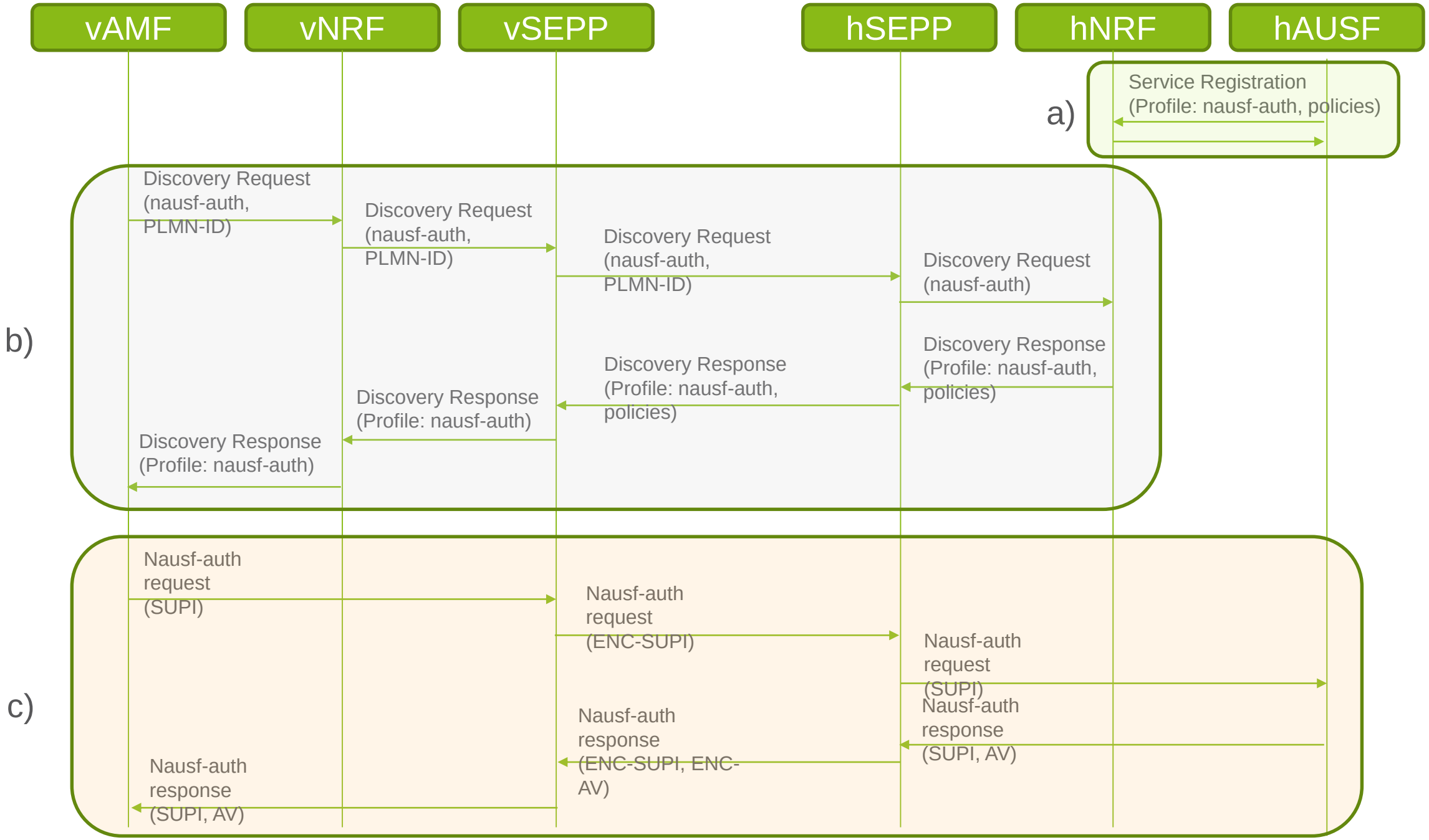
ERICSSON

```
POST {apiRoot}/nausf-auth/v1/ue-authentications

{
  "ueId": "imsi-123456789012345",
  "servingNetworkName": "5G:123456",
  "reSynchronizationInfo": {
    "rand": "01234567890ABCDEF0123456789ABCDEF",
    "auts": "01234567890ABCDEF0123456789AB"
  }
}
```

```
HTTP 200 OK

{
  "ueId": "imsi-123456789012345",
  "authenticationMethod": "5G_AKA",
  "5gAkaData": {
    "linkToConfirm": "...",
    "av5gAka": {
      "rand":     "01234567890ABCDEF0123456789ABCDEF",
      "xResStar": "01234567890ABCDEF0123456789ABCDEF",
      "autn":     "01234567890ABCDEF0123456789ABCDEF",
      "kausf":    "01234567890ABCDEF34...CDE456789ABCDEF"
    }
  },
  "servingNetworkName": "5G:123456"
}
```

# Example: Full signaling flows

- Signaling flow for 3 processes
  a) Registration of AUSF in NRF in hPLMN
  b) Service discovery from AMF in vPLMN
  c) Service invocation from AMF in vPLMN
    - If vAMF has not triggered a discovery request b), instead the vSEPP could initiate the discovery request towards hNRF, for retrieving the policies, during service invocation c).


vAMF
vNRF
vSEPP
hSEPP
hNRF
hAUSF
a)
Service Registration
(Profile: nausf-auth, policies)
b)
Discovery Request
(nausf-auth,
PLMN-ID)
Discovery Request
(nausf-auth,
PLMN-ID)
Discovery Request
(nausf-auth,
PLMN-ID)
Discovery Request
(nausf-auth)
Discovery Response
(Profile: nausf-auth,
policies)
Discovery Response
(Profile: nausf-auth,
policies)
Discovery Response
(Profile: nausf-auth)
Discovery Response
(Profile: nausf-auth)
c)
Nausf-auth
request
(SUPI)
Nausf-auth
request
(ENC-SUPI)
Nausf-auth
request
(SUPI)
Nausf-auth
response
(SUPI, AV)
Nausf-auth
response
(ENC-SUPI, ENC-
AV)
Nausf-auth
response
(SUPI, AV)

# Example: Protection policies

- Example of how the protection policies could look like inside the NF Profile of the AUSF
  - "protPolicies" element is an array of objects, where each object is a policy.
  - Each policy contains
    - the resource to be applied to,
    - the method,
    - the direction (requests or responses),
    - the actual JSON element where the policy should be applied,
    - and the action (only encryption, so far).
  - The policies can be formatted in a very different way. This is just an example, to show it is feasible. The formatting of the policies is inspired by the JSON Patch format, but this is simply anecdotical.
  - The NF profile quite realistic, with same element names used currently in NRF specification.

AUSF NF Profile


```
{
  "nfInstanceID": "24dac1c0-7cf6-4713-99ac-74e5581ee398",
  "nfType": "AUSF",
  "fqdn": "ausf.5gc.mnc123.mcc456.3gppnetworks.org",
  "nfServiceList": [
    {
      "serviceInstanceId": "1",
      "serviceName": "nausf-auth",
      "version": "v1",
      "schema": "http",
      "ipEndPoint": {
        "transport": "TCP",
        "port": 80
      },
      "protPolicies": [
        { "r": "/ue-authentications", "m": "POST", "d": "req", "e": "/ueId", "a": "enc" },
        { "r": "/ue-authentications", "m": "POST", "d": "resp", "e": "/ueId", "a": "enc" },
        { "r": "/ue-authentications", "m": "POST", "d": "resp", "e": "/5gAkaData/av5gAka", "a": "enc" }
      ]
    }
  ]
}
```


r: resource
m: method
d: direction (request/response)
e: element (JSON Pointer)
a: action (encrypt, ...)

# Motivation for protection policies

- Not necessary to update the SEPP.
- Instead, the service producer provides the protection policy, which is distributed dynamically during registration, discovery and service invocation.
- Protection policies are slim and can easily be sent in signaling, as opposed to large OpenAPI files.

ERICSSON