**3GPP TSG-SA3 Meeting #103-e *S3-211636***

**e-meeting, 17 – 28 May 2021**

**Source: Huawei, HiSilicon**

**Title: Update to the solution 3.3**

**Document for: Approval**

**Agenda Item: 5.5**

# 1 Decision/action requested

***This pCR proposes update to the solution 3.3 in TR 33.846.***

# 2 Rationale

For legal UEs and illegal UEs, AMF will always send Auth request message as the response of the uplink NAS message. When UEs receive Auth request message, there are three possible scenarios:

1. Authentication response
   1. Legal UE: If AUTN in Auth request message is verified successfully, the UE returns RES\* to the SEAF in a NAS message Authentication Response. If the verification of RES\* fails in SEAF, SEAF responses with authentication failure.
   2. Illegal UE: If UE returns fake RES\* to the SEAF in a NAS message Authentication Response, SEAF responses with authentication failure.
   3. Observation: authentication failure is both used for legal and illegal UEs. No additional information is acquired for the attacker.
2. MAC failure
   1. Legal UE: If the verification of MAC fails, the UE returns MAC failure, SEAF may initiate the identification procedure.
   2. Illegal UE: If UE returns fake MAC failure to the SEAF, SEAF may initiate the identification procedure.
   3. Observation: identification procedure may be triggered both for legal and illegal UEs. No additional information is acquired for the attacker.
3. Synch failure
   1. Legal UE: If the verification of SQN fails, the UE returns AUTS in a NAS message Authentication Response. If the verification of AUTS fails in network side, SEAF responses with authentication failure.
   2. Illegal UE: If UE returns fake AUTS to the SEAF in a NAS message Authentication Response, SEAF responses with authentication failure.
   3. Observation: authentication failure is both used for legal and illegal UEs. No additional information is acquired for the attacker.

Based on the analysis, the attacker is hard to guess the SUPI if generating fake response to SEAF.

# 3 Detailed proposal

SA3 is kindly requested to agree to the below pCR to TR 33.846.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* First Change \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### 6.3.3 Solution #3.3: Mitigation of SUPI guessing attack

#### 6.3.3.1 Introduction

This solution addresses Key issue #3.2 to mitigate the SUPI guessing attack. The basic idea is to use the same response message for valid and invalid subscriber.

#### 6.3.3.2 Solution details

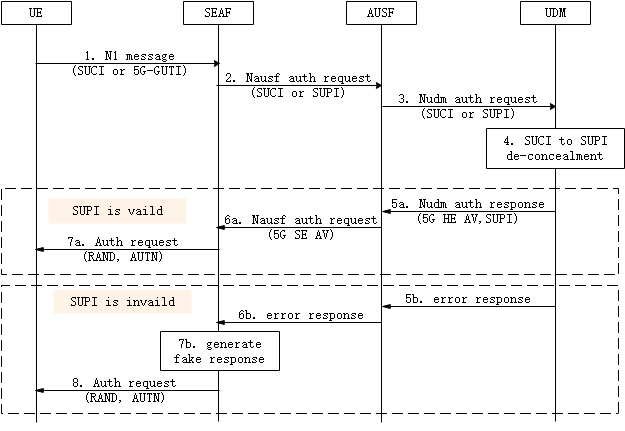


Figure 6.3.3.2-1: Procedure for mitigating SUPI guessing attack

Step 1-4: The initiation of the primary authentication is defined in TS 33.501 [2] clause 6.1.2.

Step 5a-7a: If the SUPI is valid, the authentication procedure is performed as defined in TS 33.501 [2] clause 6.1.3.

Step 5b: If the SUPI is invalid, UDM sends error response to AUSF with failure cause.

Step 6b: AUSF send error response to SEAF with failure cause. Step 7b: When the error response indicating invalid SUPI is received, SEAF generates Auth request message including the RAND and AUTN. RAND and AUTN are randomly generated.

Step 8: SEAF sends the generated Auth request message to UE. Afterwards, if UE with an invalid SUPI returns an auth response with RES\* or sych failure, SEAF responses with authentication failure. If the attacker returns a MAC failure, SEAF may initiate the identification procedure as define in TS 24.501 [6].

With the proposed method, the attacker cannot ascertain that the guessed SUPI is valid or not with the Authentication Request message sent by the network.

#### 6.3.3.3 Evaluation

This solution addresses Key issue #3.2 to mitigate the SUPI guessing attack.

The solution only requires changes on the SEAF. There is no change to any entities if the SUPI is valid. For illegitimate UEs with invalid SUPI, SEAF will respond to the UE with the appropriate message as described in the solution. SEAF could mark the UE as illegal and reject the potential response from UE.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of pCR\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*