**3GPP TSG-SA3 Meeting #123 draft\_S3-252942-r7**

**Goteborg, Sweden, 25 – 29 August 2025 merge of S3-252863, S3-252726, S3-252655, S3-252848, S3-252558, S3-252611, S3-252827, S3-252806, S3-252837, S3-252852, S3-252853, S3-252865, S3-252847, S3-252754…**

**Source: Xiaomi, Nokia, Huawei, Lenovo, OPPO, Vivo, ZTE, Ericsson, Qualcomm, CATT, Apple, Samsung**

**Title: Addressing ENs in AIoT authentication procedure**

**Document for: Approval**

**Agenda item: 4.1.1**

**Spec: 3GPP TS 33.369**

**Version: 0.2.0**

**Work Item: AmbientIoT-SEC**

**Comments**

It is proposed to address the following ENs in the AIoT authentication procedure.

Editor’s Note: Whether ADM or AIOTF generates RANDAIOT\_n is FFS.

To reduce the interaction between the AIOTF and ADM, it is proposed that AIOTF generate RANDAIOT\_n for authentication. Otherwise, one additional round of messages is necessary for transmitting the RANDAIOT\_n, i.e. one message from the AIOTF to ADM for requesting the generation of RANDAIOT\_n, and one message from the ADM to AIOTF for providing the RANDAIOT\_n.

Editor’s Note: Whether replay attack is possible is FFS.

Editor’s Note: Whether RANDAIOT\_d is required for inventory procedure is FFS.

Since the Inventory request is not protected, the replay attack can always be launched. However, the device authentication is determined by verifying the RES, which is generated at least by using RANDAIOT\_n included in the Inventory request. If the verification of RES fails, the replay attack can be detected by the network side. Therefore, the attacker cannot manipulate the AIoT system by launching a replay attack during the inventory procedure.

During the authentication procedure, each peer entity should provide a random number to ensure freshness. Therefore, it is proposed to include RANDAIOT\_d in the inventory response and use it as one of the input parameters for generating the RES. With RANDAIOT\_d and RANDAIOT\_n, the security of authentication can be further enhanced.

Editor’s Note: How RESAIOT is derived and whether it is derived from KAIoT or intermediate key is FFS.

Editor’s note: The impact of interaction between AIOTF and ADM is FFS. If the authentication is expected to be run more often than normal UE, (e.g., during each inventory procedure), the analysis of load of ADM is FFS.

Considering the storage limitation of AIoT device, it is proposed that the KAIOTF is refreshed during each inventory + command procedure. In this case, the AIoT device do not need to store the security context, which can meet the requirement from RAN. The KAIOTF can be generated by the ADM from KAIOT and provided to the AIOTF to protect the command message. Regarding the load of ADM, it can be left to the implementation of the private network.

Considering the security functionality of AIOTF and ADM, and the primary authentication procedure in SNPN (clause I.2.2.2.2 of TS 33.501), it is proposed that ADM calculate XRES by using the KAIoT and determine the authentication result by verifying RES. Therefore, the security functionality of AIOT system will be:

* The ADM is responsible for the authentication and credential management.
* The AIOTF is responsible for the communication protection.

Editor’s note: How to perform the mutual authentication for command procedure will be specified.

To save the calculation resources and transmission resources of AIoT device, the implicit authentication method can be used for mutual authentication. By verifying the security of command request message, the AIoT device can implicitly determine the authentication result to the network.

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

### 5.2.2 Authentication procedure

The authentication procedure is aligned with inventory procedure and command procedure in 6.2.2 and 6.2.3 of TS 23.369[2].



Figure 5.2.1-1: Authentication procedure

0. Step 1-6 of clause 6.2.2 Procedure for Inventory or clause 6.2.3 Procedure for command in TS 23.369 [2] is performed.

1. ADM shall generate RANDAIOT\_n. AIOTF shall retrieve RANDAIOT\_n from ADM.

2. AIOTF shall include RANDAIOT\_n in the inventory request message to NG-RAN, in addition to other device identification information.

Editor’s Note: The inclusion of RANDAIOT\_n in Paging Request and the size of RANDAIOT\_n needs RAN confirmation.

3. NG-RAN shall include RANDAIOT\_n in the paging request message to the AIoT device in addition to other device identification information.

NOTE x: An active attack may send a new paging request to the device while there is an ongoing procedure in device. The device will abort the ongoing procedure and respond to the new paging. The security measure to such denial-of-service attack is not specified in present document.

NOTE y: While a legitimate network is performing an inventory operation, an attacker may cause amplification of resource exhaustion at the legitimate network side by sending AIoT paging messages for all devices or to a large group of devices, which causes large number of devices sending D2R messages to the legitimate network that the legitimate network does not expect to receive. The security measure to such amplification of resource exhaustion attack is not specified in present document.

4. Upon receiving the paging request message, if the device determines it needs to respond based on the device identification information, AIoT device shall generate RANDAIOT\_d using the pseudo-random bit generation. AIoT device shall calculate RESAIOT using KAIoT\_root, AIoT device permanent identifier, RANDAIOT\_d, and RANDAIOT\_n (see Annex X) for network authenticating AIoT Device.

 Editor’s Note: the randomness of RANDAIOT\_d is FFS.

5. AIoT device sends D2R message to the NG-RAN, including RESAIOT and RANDAIOT\_d from device.

Editor’s note: Device identification information is FFS.

6. NG-RAN sends Inventory report message to AIOTF, including the RESAIOT and RANDAIOT\_d.

7. AIOTF sends device identification information, RANDAIOT\_n and RANDAIOT\_d to ADM.

NOTE: the authentication is expected to be run more often than normal UE, (e.g., during each inventory procedure), which has load impact to ADM.

8. ADM shall calculate XRESAIOT using the same method as in AIoT device (see Annex X).

9. ADM sends XRESAIOT to AIOTF.

10. AIOTF verifies RESAIOT. If the verification is successful, f ADM sends KAIoTF to AIOTF.

The steps 12-14 in clause 6.2.2 for inventory procedure or the step 8-11of clause 6.2.3 for command procedure in TS 23.369 [2] continues.

For the command procedure, the AIoT device implicitly authenticates the network via the verification of MAC which is derived using the KCommand\_int as specified in clause 5.2.3 of present document.

\* \* \* Second Change \* \* \* \*

Annex X (normative):
Key derivation functions

# X.1 KDF interface and input parameter construction

## X.1.1 General

All key derivations (including input parameter encoding) for 5GC shall be performed using the key derivation function (KDF) specified in Annex B.2.0 of TS 33.220 [yy].

This clause specifies how to construct the input string, S, and the input key, KEY, for each distinct use of the KDF. Note that "KEY" is denoted "Key" in TS 33.220 [yy].

## X.1.2 FC value allocations

The FC number space used is controlled by TS 33.220 [yy], FC value allocated for the present document is 0xAA-0xZZ.

# X.Y RESAIOT and XRESAIOT derivation function

When deriving a RESAIOT and XRESAIOT from KAIOT\_root, the following parameters shall be used to form the input S to the KDF:

- FC = 0xZZ,

- P0 = RANDAIOT\_n,

- L0 = length of RANDAIOT\_n (i.e. 0x00 0x10),

- P1 = RANDAIOT\_d.

- L1 = length of RANDAIOT\_d (i.e. 0x00 0x10),

- P2 = AIoT device permanent identifier,

- L2 = length of AIoT device permanent identifier,

The input key KEY shall be KAIOT\_root.

\* \* \* End of Changes \* \* \* \*