3GPP TSG-SA3 Meeting #123 draft\_S3-252945-r2

Goteborg, Sweden, 25 – 29 August 2025

**Source: Qualcomm Incorporated**

**Title: Privacy protection of device ID in individual inventory**

**Document for: Approval**

**Agenda item: 4.1.1**

**Spec: 3GPP TS 33.369**

**Version: 0.2.0**

**Work Item: Ambient\_IoT\_Sec**

**Comments**

This contribution proposes an updated individual inventory procedure (i.e., inventory with AIoT device identifier) that protects AIoT device identifier privacy.

Note that the proposed procedure is identical to the authentication procedure described in S3-252806 as the authentication procedure is based on the Inventory procedure and already considered the identity privacy in sending the Inventory request and constructing the RESAIOT.

Also note that the proposed ID privacy mechanism is based on the option A in the living document (S3-252326) with updates to address the editor’s notes.

The proposed updates address the following Editor’s Notes in the option A of the living document:

Editor’s Note: whether AIoTF or ADM computes T-ID is FFS.

Editor’s Note: whether AIoTF or ADM computes T-ID’ is FFS.

Editor’s Note: in case AIoTF computes T-ID, a key KAIoTF derived from KAIoT in ADM is used. How AIOTF retrieves the KAIoTF is FFS.

Editor’s Note: in case the T-ID is computed by the ADM, whether the cryptographic key is the long-term key KAIoT or a key derived from KAIoT , and the impact of interaction between AIOTF and ADM and the analysis of load of ADM is FFS.

Editor’s Note: which input key (e.g., KAIoT or KAIoTF) to be used is FFS

The above Editor’s Notes are deleted as the updated procedure proposes to have AIOTF hold the KAIoTF and generate a temporary ID (i.e., T-ID) and RESAIOT that corresponds to T-ID’ in the Editor’s Note. The proposal is intended to reduce the signaling overhead between AIOTF and UDM. We note that if AIOTF does not have a valid KAIoT, the bootstrapping procedure is performed as proposed in S3-252807.

Editor’s Note: whether and how to address attacks of an attacker broadcasting T-ID and Nonce triggering all AIoT Devices to constantly compute T’-D (e.g., energy depletion in the AIoT devices) is FFS.

If an attacker constantly triggers Paging message with random T-ID and Nonce, the nearby AIoT devices would compute a T-ID and check if it is matched with the received T-ID for each Paging message. However, energy consumption of T-ID computation is not an issue because AIoT devices are assumed to harvest enough energy from RF signals for backscattering transmission. As long as the AIoT device can receive and decode Paging messages, the AIoT device should be able to perform T-ID computation. Therefore, it is proposed to remove the above Editor’s Note.

Editor’s Note: how to address the attack that manipulates the RANDAIOT\_N in the Paging message is FFS.

If an attacker manipulates the RANDAIOT\_N in the Paging message, the computed T-ID would not be matched with the received T-ID. Consequently, the AIoT device stops further processing of paging message as described in step 2. Therefore, it is impossible for an attacker to receive a valid Paging response from AIoT devices if the attacker manipulated the RANDAIOT\_N. In addition, the Paging message manipulated by the attacker does not affect subsequent Inventory procedure as AIoT devices do not maintain the received RANDAIOT\_N. Therefore, it is proposed to remove the above Editor’s Note.

Lastly, we defined the function FA for temporary ID derivation and RESAIoT calculation. Defining the cryptographic algorithm to realize FA (e.g., HMAC-SHA-256) is not the purpose of this contribution and can be discussed separately.

**Proposed Changes**

\* \* \* \* First change \* \* \* \*

## 5.4 Protection of AIoT device identifier privacy

### 5.4.1 General

This clause describes the mechanism to protect AIoT device identifier privacy during the inventory procedure. The mechanism is based on the use of a Temporary ID (i.e., T-ID). The T-ID is generated based on the key (i.e., KAIoT) shared between AIoT device and ADM. When privacy protection is not used during the inventory procedure, the AIoT device includes its AIoT device permanent identifier as a device identification information in the procedure specified in clause 5.2.2.

### 5.4.2 AIoT device identifier protection for inventory with AIoT device identifier

For the protection of AIoT device permanent identifier during the inventory procedure with AIoT device identifier described in clause 5.2.2, the following changes shall apply:

- In step 1, AIOTF shall retrieve a T-ID in addition to the RANDAIOT\_n from ADM. The ADM shall generate the T-ID using KAIoT, AIoT device permanent identifier and the RANDAIOT\_n as specified in Annex X.Z.

- In step 2, 3 and 4, the T-ID shall be used as a device identification information.

- In step 4, the AIoT device generates the T-ID in the same way as the ADM did in step 1. The AIoT device determines it needs to reply to the NG-RAN if the generated T-ID matches with the received T-ID.

- In step 5 and 6, a device identification information is not included in the D2R message and Inventory Report message.

- In step 7, the AIoT device permanent identifier is used as a device identification information.

NOTE: The AIOTF identifies the AIoT device by checking the received RESAIoT parameter. Therefore, the device identification information is not needed in the D2R message and Inventory Report message.

NOTE: Every AIoT devices that receive an Inventory Request with T-ID need to perform a T-ID matching by generating a T-ID based on the KAIoT and check if the generated T-ID is matched with the received T-ID. It is assumed that the AIoT device that receive the Inventory Request has enough energy to perform this T-ID matching in addition to the Inventory procedure specified in clause 5.2.2.

\* \* \* \* Second change \* \* \* \*

# X.Z T-ID generation

When generating a temporary ID (i.e., T-ID) from KAIOT, the following parameters shall be used to form the input S to the KDF:

- FC = 0xCC,

- P0 = AIoT device permanent identifier,

- L0 = length of AIoT device permanent identifier,

- P1 = RANDAIOT\_n,

- L1 = length of RANDAIOT\_n

The input key KEY shall be KAIOT.

\* \* \* \* End of change \* \* \* \*