**3GPP TSG-SA3 Meeting #103-e *S3-211712r1***

**e-meeting, 17- 28 May 2021**

**Source: Apple**

**Title: New solution on authentication between EEC and ECS based on AKMA**

**Document for: Approval**

**Agenda Item: 5.8**

1 Decision/action requested

***It is proposed to add a new solution in MEC TR 33.839.***

2 References

[1] 3GPP TS 23.558: "Architecture for enabling Edge Applications (EA)"

3 Rationale

This pCR proposes a new solution in TR 33.839 to address the authentication between EEC and ECS based on AKMA.

4 Detailed proposal

**\*\*\*\*START OF CHANGES \*\*\***

6.X Solution #X: Authentication between EEC and ECS based on AKMA

6.2.1 Introduction

This solution addressed key issue#2 Authentication and Authorization between EEC and ECS.

This solution proposes the authentication between EEC (Edge Enabler Client) and ECS (Edge Configuration Server) based on AKMA. To be more specific, it is proposed to use the KAKMA derived from the AKMA procedure as the trust root to perform the authentication between EEC and ECS.

It is assumed in this solution that ECS is located outside of the MNO’s network.

6.2.2 Solution details

6.2.2.1 Procedure

Timeline

Description automatically generated

Figure-6.2.2.1-1. Authentication between the EEC and ECS based on AKMA

The authentication procedure details are as following:

Step 0: UE performs primary authentication with the network. Then KAUSF is shared between UE and AUSF in Home network.

Step 1.1: UE generates KAKMA and A-KID following AKMA procedure in TS 33.535 and stores them securely.

Step 1.2: AAnF generates KAKMA and A-KID following AKMA procedure in TS 33.535 and stores them securely.

Step 2: Every EEC in this UE fetches the KAKMA and generates Kedge from KAKMA and EEC ID.

NOTE：In this way, there will be one KAKMA and multiple Kedge in every UE.

Step 3: Every EEC computes MACEEC using the Kedge and EEC ID.

Step 4: UE sends Application Registration request (EEC ID, MACEEC, A-KID) to ECS.

Step 5: ECS sends Authentication verification (EEC ID, MACEEC, A-KID) to AAnF for verification.

Step 6: AAnF retrieves KAKMA using A-KID and calculates Kedge using KAKMA and EEC ID, then verify MACEEC using the (Kedge and EEC ID).

Step 7: If AAnF verification success, then AAnF sends Authentication verification response(success) back to ECS, otherwise, AAnF sends Authentication verification response(fail) to ECS.

Step 8: Based on the verification results, ECS decides whether to accept or reject the authentication request, and sends Authentication Request accept/rejection to EEC in the UE.

Editor’s Note: It is FFS how EEC ID is authenticated.

Editor’s Note: It is FFS whether ECS could perform the authentication instead of AAnF.

6.2.2.2 Derivation of Kedge and Kedge ID

Kedge is generated using KDF defined in Annex B.2.0 of TS 33.220 [8]. When deriving a Kedge from KAKMA, the following parameters should be used to form the input S to the KDF:

- FC = xxxx(to be allocated by 3GPP)

- P0 = <SUPI>,

- L0 = length of <SUPI>.

The input key KEY should be KAKMA.

6.2.2.3 Generation of MACEEC

When deriving MACEEC in the UE and AAnF, the following parameters should be used to form the input S to the SHA-256 hashing algorithm:

- P0 = Kedge,

- P1 = EEC ID,

The input S should be equal to the concatenation P0||P1 of the P0 and P1.

The MACEEC is identified with the 32 least significant bits of the output of the SHA-256 function.

### 6.X.3 Solution Evaluation

This solution requires AAnF to perform the verification of the MACEEC.

This solution applies to the case when there are multiple EECs in one UE.

Editor’s Note: Further evaluation is FFS.

**\*\*\*\*END OF CHANGES \*\*\***