|  |  |
| --- | --- |
| 3GPP TR 33.839 V17.1.0 (2022-03) | |
| Technical Report | |
| 3rd Generation Partnership Project;  Technical Specification Group Services and System Aspects;  Study on security aspects of enhancement of support for edge computing in the 5G Core (5GC)  (Release 17) | |
|  | |
| *5G-logo_175px* | 3GPP-logo_web |
|  | |
| The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP. The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented. This Specification is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification. Specifications and Reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices. | |

|  |
| --- |
|  |
| ***3GPP***  Postal address  3GPP support office address  650 Route des Lucioles - Sophia Antipolis  Valbonne - FRANCE  Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16  Internet  http://www.3gpp.org |
| ***Copyright Notification***  No part may be reproduced except as authorized by written permission. The copyright and the foregoing restriction extend to reproduction in all media.  © 2021, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).  All rights reserved.  UMTS™ is a Trade Mark of ETSI registered for the benefit of its members  3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  GSM® and the GSM logo are registered and owned by the GSM Association |

Contents

Foreword 7

1 Scope 9

2 References 9

3 Definitions of terms, symbols and abbreviations 10

3.1 Terms 10

3.2 Symbols 11

3.3 Abbreviations 11

4 Overview of Edge Computing (EC) 11

5 Key issues 12

5.0 General 12

5.1 Key issue #1: Authentication and Authorization between EEC and EES 12

5.1.1 Key Issue Details 12

5.1.2 Security Threats 12

5.1.3 Potential Security Requirements 13

5.2 Key issue #2: Authentication and Authorization between EEC and ECS 13

5.2.1 Key Issue Details 13

5.2.2 Security Threats 13

5.2.3 Potential Security Requirements 14

5.3 Key issue #3: Authentication and Authorization between EES and ECS 14

5.3.1 Key Issue Details 14

5.3.2 Security Threats 14

5.3.3 Potential Security Requirements 14

5.4 Key Issue #4 Edge Data Network Authentication and Authorization 15

5.4.1 Key issue detail 15

5.4.2 Security threats 15

5.4.3 Potential security requirements 15

5.5 Key Issue #5 Edge Data Network User Identifier and Credential Protection 15

5.5.1 Key issue detail 15

5.5.2 Security threats 15

5.5.3 Potential security requirements 15

5.6 Key issue #6: Transport security for the EDGE-1-9 interfaces 16

5.6.1 Key issue details 16

5.6.2 Threats 16

5.6.3 Potential security requirements 17

5.7 Key Issue #7: Security of Network Information Provisioning to Local Applications with low latency procedure 17

5.7.1 Key issue details 17

5.7.2 Security threats 18

5.7.3 Potential Security requirements 18

5.8 Key Issue #8: authentication and authorization in EES capability exposure 18

5.8.1 Key issue details 18

5.8.2 Security threats 18

5.8.3 Potential security requirements 18

5.9 Key Issue #9: Security of EAS discovery procedure 19

5.9.1 Key issue details 19

5.9.2 Security threats 19

5.9.3 Potential Security requirements 19

5.10 Key issue #10: Authorization during Edge Data Network change 19

5.10.1 Key issue details 19

5.10.2 Security threats 20

5.10.3 Potential security requirements 20

6 Proposed solutions 20

6.0 Mapping of Solutions to Key Issues 20

6.1 Solution #1: DNS request protection 21

6.1.1 Introduction 21

6.1.2 Solution details 21

6.1.3 Solution Evaluation 21

6.2 Solution #2: Authentication between EEC and ECS based on primary authentication 22

6.2.1 Introduction 22

6.2.2 Solution details 22

6.2.2.1 Procedure 22

6.2.2.2 Derivation of Kedge and Kedge ID 23

6.2.2.3 Generation of MACEEC 23

6.2.3 Solution Evaluation 23

6.3 Solution #3: Authentication/Authorization framework for Edge Enabler Client and Servers 24

6.3.1 Introduction 24

6.3.2 Solution details 24

6.3.3 Solution evaluation 26

6.4 Solution #4: Authentication/Authorization framework for Edge Enabler Client and Servers 26

6.4.1 Introduction 26

6.4.2 Solution details 27

6.4.3 Solution evaluation 28

6.5 Solution #5: Authentication and Authorization between the Edge Enabler Client and the Edge Enabler Server 29

6.5.1 Introduction 29

6.5.2 Solution details 29

6.5.3 Solution evaluation 30

6.6 Solution #6: Authentication and Authorization between the Edge Enabler Client and the Edge Enabler Server 30

6.6.1 Introduction 30

6.6.2 Solution details 31

6.6.3 Solution evaluation 32

6.7 Solution #7: Authentication and Authorization with the Edge Data Network 32

6.7.1 Solution overview 32

6.7.3 Solution evaluation 35

6.8 Solution #8: Authentication between EEC and EES 36

6.8.1 Solution overview 36

6.8.2 Solution details 36

6.8.3 Solution evaluation 37

6.9 Solution #9: Authentication and authorization between EEC and ECS based on AKMA 37

6.9.1 Introduction 37

6.9.2 Solution details 38

6.9.3 Solution Evaluation 39

6.10 Solution #10: Authentication and Authorization between the Edge Enabler Client and the Edge Configuration Server 39

6.10.1 Introduction 39

6.10.2 Solution details 40

6.10.3 Solution Evaluation 40

6.11 Solution #11: Authentication between EEC and ECS 40

6.11.1 Solution overview 40

6.11.2 Solution details 41

6.11.3 Solution evaluation 42

6.12 Solution #12: Onboarding and authentication/authorization framework for Edge Enabler Server and Edge Configuration Server 42

6.12.1 Introduction 42

6.12.2 Solution details 43

6.12.3 Solution evaluation 44

6.13 Solution #13: Transport security for EDGE-1-9 interfaces 44

6.13.1 Introduction 44

6.13.2 Solution details 44

13.2.0 General 44

6.13.2.1 Type A 44

6.13.2.2 Type B 44

6.13.2.3 Type C 45

6.13.3 Solution Evaluation 45

6.14 Solution #14: Protection of Network Information Provisioning to Local AF directly 45

6.14.1 Solution overview 45

6.14.2 Solution details 46

6.14.3 Solution evaluation 46

6.15 Solution #15: Network capability re-exposure via Edge Enabler Server 47

6.15.1 Introduction 47

6.15.2 Solution details 47

6.15.3 Solution Evaluation 47

6.16 Solution #16: EEC authentication and authorization framework with ECS and EES 48

6.16.1 Introduction 48

6.16.2 Solution details 48

6.16.3 Solution Evaluation 49

6.17 Solution #17: EEC/EES/ECS authentication and transport protection with TLS 50

6.17.1 Solution overview 50

6.17.2 Solution details 50

6.17.2.1 Authentication and transport protection for the EDGE-1, EDGE-3, EDGE-4, EDGE-6 and EDGE-9 interfaces 50

6.17.2.2 Authentication of the GPSI in EEC-EES/ECS communication 51

6.17.3 Solution evaluation 52

6.18 Solution #18: Authentication and Authorization Framework for EDGE-4 interfaces using Primary authentication and proxy interface 53

6.18.1 Introduction 53

6.18.2 Solution details 54

6.18.3 Solution Evaluation 55

6.19 Solution #19: Authentication/authorization between UE and Edge Data Network based on the secondary authentication 55

6.19.1 Introduction 55

6.19.2 Solution details 55

6.19.3 Solution Evaluation 56

6.20 Solution #20: Authentication and authorization in EES capability exposure based on CAPIF 56

6.20.1 Introduction 56

6.20.2 Solution details 56

6.20.3 Solution Evaluation 57

6.21 Solution #21: security for the interface between the SMF and LDNSR 57

6.21.1 Solution overview 57

6.21.2 Solution details 57

6.21.3 Solution evaluation 58

6.22 Solution #22: Authorization during Edge Data Network change 58

6.22.1 Introduction 58

6.22.2 Solution details 58

6.22.3 Solution Evaluation 60

6.23 Solution #23: Authentication and Authorization between EEC and ECS/EES 60

6.23.1 Solution overview 60

6.23.2 Solution details 60

6.23.3 Solution evaluation 60

6.24 Solution #24: Using TLS with AKMA to protect edge interfaces 61

6.24.1 Solution overview 61

6.24.2 Solution details 61

6.24.2.1 General 61

6.24.2.2 Shared key-based UE authentication with certificate-based AF authentication 61

6.24.2.2.1 General 61

6.24.1.2.2 Procedures 61

6.24.2.3 Shared key-based mutual authentication between UE and AF 62

6.24.2.3.1 General 62

6.24.2.3.2 Procedures 62

6.24.3 Solution evaluation 62

6.25 Solution #25: Practical authorization during Edge Data Network change 62

6.25.1 Introduction 62

6.25.2 Solution details 63

6.25.3 Solution Evaluation 64

6.26 Solution #26: GBA-based solution for EEC authentication and authorization framework with ECS and EES 64

6.26.1 Introduction 64

6.26.2 Solution details 64

6.26.3 Solution Evaluation 66

6.27 Solution #27: Using TLS with Edge Security Service to protect edge interfaces 67

6.27.1 Solution overview 67

6.27.2 Solution details 68

6.27.3 Solution evaluation 70

6.28 Solution #28: Authentication between EEC and ECS based on AKMA 70

6.28.1 Introduction 70

6.28.2 Solution details 70

6.28.2.1 Procedure 70

6.28.2.2 Derivation of Kedge and Kedge ID 71

6.28.2.3 Generation of MACEEC 72

6.28.3 Solution Evaluation 72

6.29 Solution #29: Using TLS with GBA to protect edge interfaces 72

6.29.1 Solution overview 72

6.29.2 Solution details 72

6.29.2.1 General 72

6.29.2.2 Shared key-based UE authentication with certificate-based AF authentication 72

6.29.2.2.1 General 72

6.29.2.3 Shared key-based mutual authentication between UE and AF 72

6.29.2.3.1 General 72

6.29.3 Solution evaluation 73

6.30 Solution #30: An AKMA-based solution for authentication and interface protection between EEC and EES/ECS 73

6.30.1 Solution overview 73

6.30.2 Solution details 73

6.30.3 Solution evaluation 74

6.31 Solution #31: Enhancing TLS with GBA for usage with Edge 75

6.31.1 Solution overview 75

6.31.2 Solution details 75

6.31.3 Solution evaluation 75

7 Conclusions 75

7.1 Conclusions for Key Issue #1 75

7.2 Conclusions for Key Issue #2 76

7.3 Conclusions for Key Issue #3 76

7.4 Conclusions for Key Issue #4 76

7.5 Conclusions for Key Issue #5 76

7.6 Conclusions for Key Issue #6 76

7.7 Conclusions for Key Issue #7 77

7.8 Conclusions for Key Issue #8 77

7.9 Conclusions for Key Issue #9 77

7.10 Conclusions for Key Issue #10 77

Annex A: Change history 78

# Foreword

This Technical Report has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

In the present document, modal verbs have the following meanings:

**shall** indicates a mandatory requirement to do something

**shall not** indicates an interdiction (prohibition) to do something

The constructions "shall" and "shall not" are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions "must" and "must not" are not used as substitutes for "shall" and "shall not". Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

**should** indicates a recommendation to do something

**should not** indicates a recommendation not to do something

**may** indicates permission to do something

**need not** indicates permission not to do something

The construction "may not" is ambiguous and is not used in normative elements. The unambiguous constructions "might not" or "shall not" are used instead, depending upon the meaning intended.

**can** indicates that something is possible

**cannot** indicates that something is impossible

The constructions "can" and "cannot" are not substitutes for "may" and "need not".

**will** indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**will not** indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**might** indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

**might not** indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

**is** (or any other verb in the indicative mood) indicates a statement of fact

**is not** (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions "is" and "is not" do not indicate requirements.

# 1 Scope

The present document studies the security enhancements on the support for Edge Computing in the 5G Core network define in TR 23.748 [3], and application architecture for enabling Edge Applications defined in TR 23.758 [4] and TS 23.558 [2].

Potential security requirements are provided and possible security enhancements to 5GS and edge application architecture are proposed that meet these security requirements.

NOTE: The user consent for exposure of information to Edge Applications is addressed in TR 33.867 [20].

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 23.558: "Architecture for enabling Edge Applications".

[3] 3GPP TR 23.748: "Study on enhancement of support for Edge Computing in the 5G Core network (5GC)".

[4] 3GPP TR 23.758: "Study on application architecture for enabling Edge Applications".

[5] 3GPP TS 23.502: "Procedure for the 5G System; Stage 2".

[6] 3GPP TS 33.535: "Authentication and Key Management for Applications (AKMA) based on 3GPP credentials in the 5G System (5GS)".

[7] 3GPP TS 33.501: "Security architecture and procedures for 5G System".

[8] 3GPP TS 33.220: "Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA)".

[9] 3GPP TS 23.222: "Functional architecture and information flows to support Common API Framework for 3GPP Northbound APIs; Stage 2".

[10] Void.

[11] 3GPP TS 33.187: "Security aspects of Machine-Type Communications (MTC) and other mobile data applications communications enhancements".

[12] 3GPP TS 33.210: "3G security; Network Domain Security (NDS); IP network layer security".

[13] 3GPP TS 33.310: "Network Domain Security (NDS); Authentication Framework (AF)".

[14] 3GPP TS 23.501: "System Architecture for the 5G System".

[15] 3GPP TS 23.003: "Numbering, addressing and identification".

[16] 3GPP TS 33.122: "Security aspects of Common API Framework (CAPIF) for 3GPP northbound APIs".

[17] Void.

[18] IETF RFC 4279 "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)".

[19] IETF RFC 8446: "The Transport Layer Security (TLS) Protocol Version 1.3".

[20] 3GPP TR 33.867: "Study on user consent for 3GPP services".

[21] IETF RFC 7858: "Specification for DNS over Transport Layer Security (TLS)".

[22] IETF RFC 8310: "Usage Profiles for DNS over TLS and DNS over DTLS".

[23] 3GPP TS 33.434: "Security aspects of Service Enabler Architecture Layer (SEAL) for verticals".

[24] IETF RFC 7616: "HTTP Digest Access Authentication".

[25] IETF RFC 5246: "The Transport Layer Security (TLS) Protocol Version 1.2".

[26] 3GPP TS 33.222: "Generic Authentication Architecture (GAA); Access to network application functions using HypertextTransfer Protocol over Transport Layer Security (HTTPS)".

[27] IETF RFC 2616 (1999): "Hypertext Transfer Protocol (HTTP) – HTTP/1.1".

[28] GSMA IoT.04: "Common Implementation Guide to Using the SIM as "Root of Trust" to Secure IoT Applications".

[29] IETF RFC 7515: "JSON Web Signature (JWS)".

[30] IETF RFC 7519: "JSON Web Token (JWT)".

[31] IETF RFC 7662: "OAuth 2.0 Token Introspection".

[32] OpenID Connect 1.0: "OpenID Connect Core 1.0 incorporating errata set 1", <http://openid.net/specs/openid-connect-core-1_0.html>.

[33] NIST SP 800-63-3. Digital Identity Guidelines. Available at: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63-3.pdf>.

[34] NIST SP 800-63c. Federation and Assertions. Available at: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63c.pdf>.

[35] 3GPP TS 23.548: "5G System Enhancements for Edge Computing; Stage 2; (Release 17)".

[36] IETF RFC 5246: "The Transport Layer Security (TLS) Protocol Version 1.2"..

[37] IETF RFC 8446: "The Transport Layer Security (TLS) Protocol Version 1.3".

# 3 Definitions of terms, symbols and abbreviations

## 3.1 Terms

For the purposes of the present document, the terms given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1].

## 3.2 Symbols

Void

## 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

AC Application Client

EAS Edge Application Server

ECS Edge Configuration Server

EEC Edge Enabler Client

EES Edge Enabler Server

FQDN Fully Qualified Domain Name

LADN Local Area Data Network

# 4 Overview of Edge Computing (EC)

The present document studies the security enhancements on the support for Edge Computing in the 5G Core network defined in TS 23.548 [3], and application architecture for enabling Edge Applications defined in TS 23.558 [2].

For the EC supported in 5GC, please refer to TS 23.548 [3].

For the application architecture for enabling Edge Applications, a new architecture is defined in TS 23.558 [2] as shown in the figure,



Figure 4-1: Architecture for enabling edge applications - reference points representation

Within this architecture, the Edge Enabler Client (EEC) deployed in the UE retrieves the Edge configuration information from the Edge Configuration Server (ECS), The Edge configuration information includes the information for the EEC to connect to the EES (e.g. EDN service area); and the information for establishing a connection with EESs (such as URI). Based on the Edge configuration information, EEC could also acquire the required EAS information from the discovered EES. ECS enables other authorized EES to access their services. Meanwhile, EES enables other authorized EAS to access their services. The ECS, EES and EAS acts as AFs for consuming network services directly from the 3GPP 5G Core Network entities over the service based architecture specified in 3GPP TS 23.501 [14].

For more details on enabling Edge Applications, please refer to TS 23.558 [2].

# 5 Key issues

## 5.0 General

The following key issues are defined based on the security enhancements on the support for Edge Computing in the 5G Core network define in TR 23.748 [3] and TS 23.548 [35], and application architecture for enabling Edge Applications defined in TR 23.758 [4] and TS 23.558 [2], i.e.:

- Support for Edge Computing in the 5G Core network: Key issues 4, 5, 7, 9, and 10.

- Application architecture for enabling Edge Applications: Key issues 1, 2, 3, 6, and 8.

## 5.1 Key issue #1: Authentication and Authorization between EEC and EES

### 5.1.1 Key Issue Details

As per TR 23.558 [2], EDGE-1 reference point enables interactions between the Edge Enabler Server and the Edge Enabler Client. EDGE-1 reference point supports registration and de-registration of the Edge Enabler Client to the Edge Enabler Server, retrieval and provisioning of Edge Application Server configuration information; and discovery of Edge Application Servers available in the Edge Data Network.

Edge Enabler server provides functionalities to Edge Enabler client over EDGE-1 reference point such as provisioning of configuration information to Edge Enabler Client and support the functionalities of application context transfer.

Edge Enabler Client performs the functionalities like configuration information retrieval from the edge enabler server and discovering of the edge application servers available in Edge Data Network. The Edge Data Network is a local Data Network. Edge Application Server(s) and the Edge Enabler Server are contained within the EDN.

The UE is initially provisioned with the configurations required to connect to the Edge Data Network. Upon initial provisioning, the Edge Enabler Client of the UE registers with the selected Edge Enabler Server(s) from the list of provisioned Edge Enabler Server(s). Edge Enabler Client consumes service offered by the Edge Enabler Server, e.g. discovering Edge Application Servers in an area of interest. The procedure enables the initialization or update of the Edge Enabler Client context information at the Edge Enabler Server. The Edge Enabler Client sends the Edge Enabler Client registration request to the Edge Enabler Server. Edge Application Server discovery enables Edge Enabler Clients to obtain information about available Edge Application Servers of interest. The identification of the Edge Application Servers is based on matching query filters or Application Client Profiles provided in the request.

GPSI can be used as a UE identifier inside and outside of 5G networks, as specified in TS 23.501 [14] and TS 23.003 [15]. As specified in TS 23.558 [2], a new edge enabler layer is defined. In order to identify the UE's Edge Enabler Client, the UE uses Edge Enabler client ID as the client identifier at the edge enabler layer. And the Edge Enabler client ID may be used along with GPSI. Then the EEC uses two different identifiers towards the EES, EEC ID and UE identifier (could be GPSI)). Solutions to this key issue need to clearly state which identifier of the EEC they authenticate.

Note 1: It is not addressed here whether the EEC ID will be unique across different UEs.

Note 2: Whether the binding issue between EEC ID and UE identifier is required is not addressed here.

### 5.1.2 Security Threats

When Registration, Discovery, Deregistration is used without authorization, malicious Edge enabler client receives a list of Services and topology structure within Edge Data Network from Edge Enabler Server discovery response message. The received information can reveal Edge Data Network's topology (e.g. URI, IP address, number of Edge Application Servers, Application Server Functionalities, API type, protocols). Malicious Edge Enabler Client may use this information to launch attacks on Edge Data Network or use this information for competitive reasons.

If GPSI is not authenticated, then an Edge Enabler Client that spoofs a victim UE's GPSI can learn some information about the location of the victim UE's location because the server lists returned to the Edge Enabler Client is constructed considering the UE location learned from the 3GPP network.

### 5.1.3 Potential Security Requirements

Edge Enabler Server shall be able to provide mutual authentication with Edge Enabler Client over EDGE-1 Interface.

Edge Enabler Server shall be able to determine whether Edge Enabling client is authorized to access Edge Enabling Server's services.

Edge Enabler Server shall be able to authenticate the GPSI sent by the Edge Enabler Client if the client sends the GPSI to the server.

## 5.2 Key issue #2: Authentication and Authorization between EEC and ECS

### 5.2.1 Key Issue Details

As per TR 23.558 [2], the EDGE-4 reference point enables interactions between the Edge Configuration Server (ECS) and the Edge Enabler Client. Edge Configuration Server (ECS) (Edge Configuration Server (ECS)) provides supporting functions needed for the Edge Enabler Client to connect with an Edge Enabler Server(EES). EDGE-4 reference point supports provisioning of Edge configuration information (e.g. URI or LADN service information) to the Edge Enabler Client.

Edge Enabler Client performs the functionalities like configuration information retrieval from the edge configuration sever over the EDGE-4 interface.

As per TR 23.558 [2], The Edge Configuration Server(ECS) can be deployed in the MNO domain or can be deployed in 3rd party domain by the service provider in which one Edge Enabling Client may communicate with one or more Edge Configuration Server(ECS)(s) concurrently. If the Edge Configuration Server (ECS) is deployed by MNO, the Edge Configuration Server (ECS) provides one or more Edge Enabling Server configuration information. If the Edge Configuration Server (ECS) is deployed by a non-MNO Edge computing service provider, the Edge Configuration Server(ECS) endpoint address is pre-configured with the Edge Enabling Client. The Edge enabling client that is configured with multiple Edge Configuration Server (ECS) endpoint addresses (es), may perform the service provisioning procedure per the Edge Configuration Server(ECS) of each Edge Configuration Server(ECS) multiple times. UE can contain a single Application Client (AC) or multiple Application Client(AC)s, which are served by a single Edge Configuration Server(ECS). In another scenario, UE has multiple Application Client(AC)s where each Application Client(AC) can be served by an Edge Application Server, which in turn is served by a different Edge Configuration Server(ECS)'s Edge Enabling Server.

GPSI can be used as a UE identifier inside and outside of 5G networks, as specified in TS 23.501 [14] and TS 23.003 [15]. As specified in TS 23.558 [2], a new edge enabler layer is defined. In order to identify the UE's Edge Enabler Client, the UE uses Edge Enabler client ID as the client identifier at the edge enabler layer. And the Edge Enabler client ID may be used along with GPSI. Then the EEC uses two different identifiers towards the EES, EEC ID and UE identifier (could be GPSI)). Solutions to this key issue need to clearly state which identifier of the EEC they authenticate.

Note 1 : It is not addressed here whether the EEC ID will be unique across different UEs.

Note 2: Whether the binding issue between EEC ID and UE identifier is required is not addressed here.

### 5.2.2 Security Threats

If access to Provisioning and configuration information is retrieved without authentication and authorization, malicious Edge enabler client will be able to receive a list of Edge Enabling Server configuration information and topology structure within Edge Data Network from the provisioning response message. The received information can reveal Edge Data Network's topology (e.g. URI, FQDN, IP address, LADN service information, Application Server Functionalities, API type, protocols).

Malicious Edge Enabler Client may use this information to launch attacks on Edge Data Network or use this information for competitive reasons.

If GPSI is not authenticated, then an Edge Enabler Client that spoofs a victim UE's GPSI can learn some information about the location of the victim UE's location because the server lists returned to the application is constructed considering the UE location learned from the 3GPP network.

### 5.2.3 Potential Security Requirements

Edge Configuration Server(ECS) Requirements:

- Edge Configuration Server(ECS) shall be able to provide mutual authentication with Edge Enabler Client over EDGE-4 Interface.

- Edge Configuration Server(ECS) shall be able to determine whether Edge Enabling the client is authorized to access provisioning services offered by Edge Configuration Server(ECS).

- Edge Configuration Server shall be able to authenticate the GPSI sent by the Edge Enabler Client if the client sends the GPSI to the server.

## 5.3 Key issue #3: Authentication and Authorization between EES and ECS

### 5.3.1 Key Issue Details

As per 23.558 [2], the EDGE-6 reference point enables interactions between the Edge Configuration Server (ECS) and the Edge Enabler Server. EDGE-6 supported the registration and registration updates, deregistration, of Edge Enabler Server information to the Edge Enabler Network Configuration Server. The Edge Enabler Server Registration procedure allows an Edge Enabler Server to provide information to an Edge Configuration Server to request the use of its edge configuration capabilities. The Edge Enabler Server registration update procedure allows an Edge Enabler Server to update the Edge Configuration Server if there is a change in the information at the Edge Enabler Server. The Edge Enabler Server uses the Edge Enabler Server deregistration procedure to remove its information from the Edge Configuration Server. As per 23.558 [2], The Edge Configuration Server(ECS) can be deployed in the MNO domain or can be deployed in 3rd party domain by the service provider in which one Edge Enabling Client may communicate with one or more Edge Configuration Server(ECS)(s) concurrently. One Edge Enabling Server may concurrently connect to one or more Edge Configuration Server with a separate EDGE-6 reference point interface. The Edge enabling server that is configured with multiple Edge Configuration Server (ECS) endpoint addresses (es) may perform the service registration, updates, or deregistration procedures per the Edge Configuration Server(ECS) of each Edge Configuration Server(ECS) multiple times. In this context, the Security Context of each of EDGE-6 interfaces needs to be separate from each other as the trust domain may be different.

### 5.3.2 Security Threats

Without authentication or authorization, the Malicious Edge Enabling server may be able to register with the Edge configuration server, further exposing its services to UE's Edge, enabling clients and applications running on UE.

Registration updates without any confidentiality or integrity may be able to help a Man In the middle actor impersonating the Edge configuration server to the Edge Enabling server exposing and possibly altering the registration updates with a falsified Edge Enabling Server profile to Edge configuration server. Also, this attack leads to exposing the topology details, server information within the PLMN domain. Malicious actors can use this exposed information for the benefit of PLMN's or Edge Computing Service provider's competitors.

### 5.3.3 Potential Security Requirements

The Edge Configuration Server and the Edge Enabling Server shall perform mutual authentication, to register and update the server profile information.

The Edge Configuration Server shall be able to authorize the Edge Enabling Server to register and update the server profile information.

## 5.4 Key Issue #4 Edge Data Network Authentication and Authorization

### 5.4.1 Key issue detail

The concept of edge computing is analogous to that of an (external) data network in the sense that the UE's edge client and the edge application server needs to be authenticated and authorized before UE can access the edge data network. In the case of an edge data network, the data network itself is much closer to the UE than a traditional data network. UE authentication and authorization are normal part of UE network access. For UEs accessing the edge data network, the authentication to the edge data network is in addition to the primary authentication for 3GPP network access. However, depending on the relationship between the edge data network operator and the 3GPP PLMN, the authentication to the edge data network may be implicit.

### 5.4.2 Security threats

Authentication and authorization are fundamental necessities in establishing security and providing access to the UEs by the network. Without it, there is no security and unauthenticated and unauthorized UEs may be able to enjoy the services provided by an edge data network that the UEs have not subscribed to.

### 5.4.3 Potential security requirements

UEs and Edge Data Network shall be mutually authenticated. When the Edge Data Network is outside of the 3GPP domain, non-3GPP credentials may be used.

UE's access to Edge Data Network shall be authorized.

Existing security mechanisms shall be re-used as much as possible (e.g. secondary authentication or slice-specific authentication).

## 5.5 Key Issue #5 Edge Data Network User Identifier and Credential Protection

### 5.5.1 Key issue detail

For each UE, there may be multiple sets of user identifiers and credentials that are used between UE and different edge data networks that are different from the long term identifiers and credentials (i.e. 5G AKA credentials) used for primary authentication. These user identifiers and credentials used in edge data network authentication are stored in the UE and the edge data networks. The identifiers and credentials need to be identified and protected in the UE, in the network, and in transition, even in the case where the edge data network is operated by a third party.

### 5.5.2 Security threats

If user identifiers and credentials are not protected, a number of well-documented attacks can result in the loss of privacy, user data, and other sensitive information for the users.

### 5.5.3 Potential security requirements

Edge data network application user identifiers and credentials shall be protected in storage and transit.

NOTE: How edge data network application user identifiers and credentials are provisioned in the UE is out of the scope of the present document.

## 5.6 Key issue #6: Transport security for the EDGE-1-9 interfaces

### 5.6.1 Key issue details

TS 23.558 [2], clause 6.2 describes a new architecture for enabling edge applications, i.e.



Figure 5.6.1-1

New interfaces (i.e. EDGE-1-9) were introduced in the architecture for enabling Edge Applications. This key issues studies the related transport security, i.e. confidentiality, integrity, and replay-protection.

- Type A (Between UE and Edge servers):

- EDGE-1: between EEC and EES

- EDGE-4: between EEC and ECS

- EDGE-5: between EEC and Application Client(s)

NOTE: Details of the EDGE-5 is out of scope of this release of the present document, according to TS 23.558 [2]

- Type B (Between 3GPP core and Edge servers):

- EDGE-2: between 3GPP Core network and EES

- EDGE-7: between 3GPP Core network and EAS

- EDGE-8: between 3GPP Core network and ECS

- Type C (Between Edge servers):

- EDGE-3: between EAS and EES

- EDGE-6: between EES and ECS

- EDGE-9: between EES(s)

### 5.6.2 Threats

Without confidentiality, integrity, and replay protection, an attacker may eavesdrop or manipulate or replay the communication or initiate the MITM attacks on the interface.

### 5.6.3 Potential security requirements

Confidentiality protection, integrity protection, and replay-protection shall be supported on the EDGE-1-4, and EDGE 6-9 interfaces.

## 5.7 Key Issue #7: Security of Network Information Provisioning to Local Applications with low latency procedure

### 5.7.1 Key issue details

In the solutions for network information provisioning to local application procedure in TR 23.748 [3], the following two ways are proposed to perform network information exposure to a local application:

UPF exposes the network information (i.e. QoS monitoring) to local AF via Local NEF.

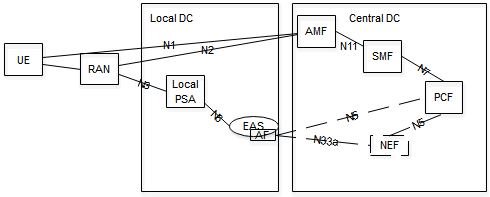


For this case, the following two alternatives proposed:

- The EAS/AF subscribes the network information notification according to the blue dashed line path, and the local PSA provisions the networking information to EAS/AF via local NEF (i.e. according to the blue solid line path).

- The EAS/AF subscribes the network information notification according to the red dashed line path, in this case, the local NEF retrieves the UPF information before subscribing to the event from UPF for AF which is not showed in the figure. When the request event happens, the local PSA provisions the networking information also to EAS/AF via local NEF (i.e. according to the blue solid line path).

UPF exposes the network information to local AF directly.



- For this case, the following two alternatives proposed:

- The EAS/AF subscribes the network information notification according to the blue dashed line path, and the local PSA provisions the networking information to EAS/AF directly (i.e. according to the blue solid line path).

- The EAS/AF subscribes the network information notification with UPF directly (i.e. according to the red dashed line path).

A new interface between UPF and local NEF/local AF/EAS is introduced, which has a new security issue.

NOTE: Local PSA UPF can expose the QoS monitoring results to local AF via N6. How to deliver the information on N6 is out of scope.

### 5.7.2 Security threats

Without authentication and protection, an attacker may eavesdrop or manipulate or replay the communication on the new interface.

### 5.7.3 Potential Security requirements

For the case that UPF exposes the network information to local AF via Local NEF.

- Mutual authentication mechanism between UPF and local NEF shall be supported.

- Confidentiality protection, integrity protection, and replay protection shall be supported on the new interface between UPF and local NEF.

For the case that UPF exposes the network information to local AF directly:

- The UPF enables the secure provisioning of information in the 3GPP network by authenticated and authorized Application Functions.

Confidentiality protection, integrity protection, and replay protection shall be supported on the interface between UPF and Application Functions.

## 5.8 Key Issue #8: authentication and authorization in EES capability exposure

### 5.8.1 Key issue details

TS 23.558, clause 8.6 [2] describes service capability APIs exposed by the Edge Enabler Server to the Edge Application Server(s). The service capability APIs exposed include EES capabilities and re-exposed 3GPP Core Network capabilities. To support EES capability exposure, the following open issues need to be studied:

- Whether and how to support the Edge Application Server to access the EES capability exposure function directly, e.g. how CAPIF, as specified in 3GPP TS 23.222 [9] can be utilized, and whether there is a need to enhance functionalities of CAPIF?

- How the Edge Enabling Server re-exposes service API(s) to the Edge Application Server, where the service API(s) are relying on the SCEF/NEF northbound API(s)?

### 5.8.2 Security threats

If the access to EES capability APIs is not authenticated and authorized, attackers would potentially be able to perform the following types of attacks:

- Requesting service from the EES that unauthorized parties are not allowed to consume, e.g. in order to gain user's privacy information

- Flooding the EES with resource-demanding operations may lead to a Denial of Service situation

### 5.8.3 Potential security requirements

EES capability exposure to EAS shall be authenticated and authorized.

## 5.9 Key Issue #9: Security of EAS discovery procedure

### 5.9.1 Key issue details

In the solutions for the EAS discovery procedure in TR 23.748 [3], the following DNS based solution is proposed. The solution requires a new Functionality, an enhanced DNS Forwarder here referred to as "LDNSR". LDNSR supports Edge AS Discovery using DNS using knowledge of the 5GC connectivity of the UE.



Figure 5.9.1-1: Options for the EAS discovery using LDNSR for PDU session breakout

New function LDNSR is introduced for EAS discovery, and the interaction between SMF and LDNSR is also introduced. The SMF may provide knowledge of the 5GC connectivity of the UE to LDNSR, the information about the knowledge of the 5GC connectivity of the UE is sensitive material which should be security protected.

In the above solution, a DNS request is sent to query the Edge Server's address. If the DNS destination address is modified by the attacker, DNS request will be sent to the compromised DNS server, then the wrong Edge Server address may be allocated. This attack may make UE connected to a far Edge Server and ruin the advantage of the MEC, even worse, the compromised DNS server may lead UE to connect to a compromised Edge Server.

### 5.9.2 Security threats

Without protection, an attacker may eavesdrop or manipulate or replay the communication on the new interface.

Without protection about the DNS message, an attacker may manipulate the DNS message which may cause the UE is not able to find a suitable EAS.

### 5.9.3 Potential Security requirements

The interaction message between the SMF and LDNSR shall be confidentiality, integrity, and replay protected. Secure discovery of EDGE Services should be supported.

## 5.10 Key issue #10: Authorization during Edge Data Network change

### 5.10.1 Key issue details

TR 23.748 [3] clause 5.2 describes a key issue #2 "Edge relocation", which raises an issue on "How to handle the change of the serving EAS (without UE mobility) to support seamless change, e.g. preventing or reducing packet loss". Currently, several solutions (such as solution #22-40) in TR 23.748 [3] were proposed to address this key issue.

Here, Edge Data network connectivity will be modified during the edge relocation, which is different from changing the PDU session anchor only. When a new PDU session is created, secondary authentication will be triggered (distributed anchor with SSC mode 1/2/3 and multiple PDU sessions). This will not happen when additional PSA-UPFs are added to an existing PDU session. Authorization needs to be investigated in relation to session breakout and ULCL as well as IPv6MH.

Specifically, it needs to be studied how the authorization provided by the secondary authentication is addressed during Edge Data network change with requirements on service continuity as studied in Key issue #2 of TR 23.748 [3].

This key issue is to study the authentication and authorization requirement between the UE and the target Edge Data network during the Edge Data network change.

### 5.10.2 Security threats

If authentication between the UE and the target EDN is not performed for the sake of seamless change, the threats of key issue #4 apply in this key issue.

Without authorization, an unauthorized UE may be able to consume the services provided by the target Edge Data network.

### 5.10.3 Potential security requirements

Authentication of the UE and the target Edge Data Network shall be supported during Edge Data network relocation with seamless change.

Authorization of UE for EAS service access during Edge Data network relocation with seamless change shall be supported.

# 6 Proposed solutions

## 6.0 Mapping of Solutions to Key Issues

Table 6.0-1: Mapping of Solutions to Key Issues

| Solutions | Key Issues | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Solution #1: DNS request protection |  |  |  |  |  |  |  |  | x |  |
| Solution #2: Authentication between EEC and ECS based on primary authentication |  | x |  |  |  |  |  |  |  |  |
| Solution #3: Authentication/Authorization framework for Edge Enabler Client and Servers | x | x |  |  |  |  |  |  |  |  |
| Solution #4: Authentication/Authorization framework for Edge Enabler Client and Servers | x | x |  | x |  | x |  |  |  |  |
| Solution #5: Authentication and Authorization between the Edge Enabler Client and the Edge Enabler Server | x |  |  |  |  |  |  |  |  |  |
| Solution #6: Authentication and Authorization between the Edge Enabler Client and the Edge Enabler Server | x |  |  |  |  |  |  |  |  |  |
| Solution #7: Authentication and Authorization with the Edge Data Network | x | x |  |  |  | x |  |  |  |  |
| Solution #8: Authentication between EEC and EES | x |  |  |  |  |  |  |  |  |  |
| Solution #9: Authentication and authorization between EEC and ECS based on AKMA |  | x |  |  |  |  |  |  |  |  |
| Solution #10: Authentication and Authorization between the Edge Enabler Client and the Edge Configuration Server |  | x |  |  |  |  |  |  |  |  |
| Solution #11: Authentication between EEC and ECS |  | x |  |  |  |  |  |  |  |  |
| Solution #12: Onboarding and authentication/authorization framework for Edge Enabler Server and Edge Configuration Server |  |  | x |  |  |  |  |  |  |  |
| Solution #13: Transport security for EDGE-1-9 interfaces |  |  |  |  |  | x |  |  |  |  |
| Solution #14: Protection of Network Information Provisioning to Local AF directly |  |  |  |  |  |  | x |  |  |  |
| Solution #15: Network capability re-exposure via Edge Enabler Server |  |  |  |  |  |  |  | x |  |  |
| Solution #16: EEC authentication and authorization framework with ECS and EES | x | x |  |  |  |  |  |  |  |  |
| Solution #17: EEC/EES/ECS authentication and transport protection with TLS and HTTP Digest with AKMA PSK | x | x | x |  |  | x |  |  |  |  |
| Solution #18: Authentication and Authorization Framework for EDGE-4 interfaces using Primary authentication and proxy interface |  | x |  |  |  |  |  |  |  |  |
| Solution #19: Authentication/authorization between UE and Edge Data Network based on the secondary authentication |  |  |  | x |  |  |  |  |  |  |
| Solution #20: Authentication and authorization in EES capability exposure based on CAPIF |  |  |  |  |  |  |  | x |  |  |
| Solution #21: security for the interface between the SMF and LDNSR |  |  |  |  |  |  |  |  | x |  |
| Solution #22: EC: New solution on authorization during Edge Data Network change |  |  |  |  |  |  |  |  |  | x |
| Solution #23: Authentication and Authorization between EEC and ECS/EES | x | x |  |  |  |  |  |  |  |  |
| Solution #24: Using TLS with AKMA to protect edge interfaces |  |  |  |  |  | x |  |  |  |  |
| Solution #25: Practical authorization during Edge Data Network change |  |  |  |  |  |  |  |  |  | x |
| Solution #26: GBA-based solution for EEC authentication and authorization framework with ECS and EES | x | x |  |  |  |  |  |  |  |  |
| Solution #27: Using TLS with Edge Security Service to protect edge interfaces | x | x |  |  |  | x |  |  |  |  |
| Solution #28: Authentication between EEC and ECS based on AKMA |  | x |  |  |  |  |  |  |  |  |
| Solution #29: Using TLS with GBA to protect edge interfaces | x | x |  |  |  | x |  |  |  |  |
| Solution #30: An AKMA-based solution for authentication and interface protection between EEC and EES/ECS | x | x |  |  |  | x |  |  |  |  |
| Solution #31: Enhancing TLS with GBA for usage with Edge | x | x |  |  |  |  |  |  |  |  |

## 6.1 Solution #1: DNS request protection

### 6.1.1 Introduction

The key issue #9 is proposed to protect the DNS request modification attack. In an edge computing environment, DNS request is needed to query the Edge Server's address. If the DNS destination address is modified by the attacker, then the wrong Edge Server address may be allocated. This attack may make UE connected to a far Edge server and ruin the advantage of the MEC, even worse, the false DNS server may lead UE to connect to a compromised Edge Server.

TS 33.501 [7] has an informative annex P.2 describing security aspects on DNS for 5G, and it is proposed to reuse the enhanced DNS in the MEC system.

### 6.1.2 Solution details

DNS server should support DNS over (D)TLS, as specified in RFC 7858 [21] and RFC 8310 [22]. The DNS server(s) that are deployed within the 3GPP network can enforce the use of DNS over (D)TLS. The UE can be pre-configured with the DNS server security information (out-of-band configurations specified in the IETF RFCs similar to, credentials to authenticate the DNS server, supported security mechanisms, port number, etc.), or the core network can configure the DNS server security information to the UE. When DNS over (D)TLS is used, a TLS cipher suite that supports integrity protection needs to be negotiated.

### 6.1.3 Solution Evaluation

This solution reuses the security recommendations from TS 33.501 and requires UE and DNS server to support DNS over (D)TLS, while introducing no extra impact to other network entities.

## 6.2 Solution #2: Authentication between EEC and ECS based on primary authentication

### 6.2.1 Introduction

This solution is addressing 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). To be more specific, it is proposed to use the Kausf derived from the primary authentication 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

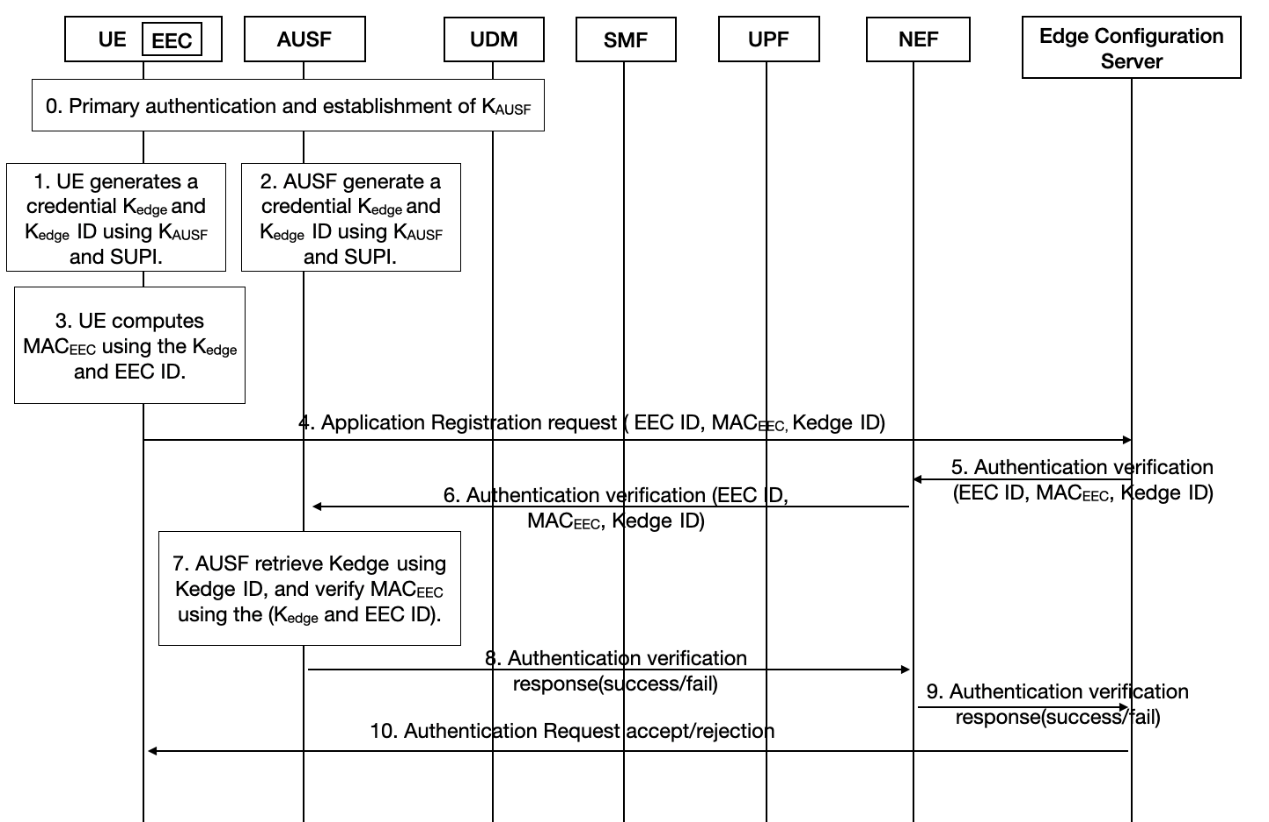


Figure-6.2.2.1-1: Authentication between the EEC and ECS based on primary authentication

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. UE performs PDU session establishment procedure as defined in TS 23.502.

Step 1: UE generates a credential Kedge and Kedge ID using KAUSF and SUPI, and stored securely. The method to derive generate Kedge and Kedge ID is in 6.2.2.2.

Step 2: AUSF generates a credential Kedge and Kedge ID using KAUSF and SUPI, and stored securely.

Step 3: UE computes MACEEC using the Kedge and EEC ID (defined in TS 23.558 [2]). The method to generate MACEEC is in 6.2.2.3.

Step 4: UE sends Application Registration request (EEC ID, MACEEC, Kedge ID) to ECS. Whether this message is send using NAS or user plane is based on SA2's decision.

Step 5: ECS sends Authentication verification (EEC ID, MACEEC, Kedge ID) to NEF for verification.

Step 6: NEF discovers the AUSF based on Kedge ID, and sends Authentication verification (EEC ID, MACEEC, Kedge ID) to AUSF for MACEEC verification.

Note 1: How to discover the AUSF is not addressed here.

Step 7: AUSF retrieves Kedge using Kedge ID, and verify MACEEC using the (Kedge and EEC ID).

Step 8: If verification in AUSF succeed, then AUSF sends Authentication verification response(success) back to NEF, otherwise, AUSF sends Authentication verification response(fail) to NEF.

Step 9: NEF sends Authentication verification response(success/fail) from AUSF to ECS.

Step 10: 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.

Note 2: How the AUSF can be aware of each specific Kedge per UE is not addressed here.

#### 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 KAUSF, 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 KAUSF.

Kedge ID is generated by AUSF and UE, and uniquely identify only one Kedge.

Kedge ID should contain the "routing indicator|| service identifier part || uniqueness part".

- The routing indicator part is used by ECS to find the correct NEF,

- The service identifier part is used to represent different service, and it could be the string of the service name or the EAS ID defined in TS 23.558.

The uniqueness part is used to make sure there is no collision among different Kedge IDs for different EECs.

Note: It is not addressed here how to generate the uniqueness part.

#### 6.2.2.3 Generation of MACEEC

When deriving MACEEC in the UE and AUSF, 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.

### 6.2.3 Solution Evaluation

Not addressed here.

## 6.3 Solution #3: Authentication/Authorization framework for Edge Enabler Client and Servers

### 6.3.1 Introduction

This solution addresses the security requirement for the Authentication and Authorization of EEC in key issue #1 and key issue #2, Key issue #6(for EDGE-1, EDGE-4 interfaces).

The Edge Configuration Server (ECS) act as the token server for issuance and validation of access tokens to the UE and also to the EES and optionally EAS. Access tokens are issued to EEC for the Edge Computing service, after verification of the UE authenticity using AKMA service. AKMA service is used as to use the network access credentials for the UE's authentication. Access token is used for authorization of the UE to access/obtain the Edge Computing service.

### 6.3.2 Solution details

6.3.2.1 Authentication and Authorization procedure between EEC and ECS/EES



Figure 6.3.2-1: Authentication/Authorization framework for Edge Enabler Client and Servers

Step 1: The UE performs the procedures as defined in TS 23.502 [5] to get the 5GC network access.

Step 1A: At the end of the network access authentication procedure (Primary authentication and key agreement TS 33.501, clause 6.1)), the UE and the AUSF are in possession of the key KAUSF.

Step 2A-2C: The UE and the AUSF derives the AKMA key as specified in TS 33.535 [6]. The AUSF provides the AKMA key to the AAnF as specified in TS 33.535 [6].

Step 2D-2J: The UE initiates the Initial provisioning procedure with the ECS and includes AKMA Key ID. ECS is Application Function (AF) for the AAnF as specified in TS 33.535 [6]. The ECS contacts the AAnF (using AKMA key ID) to obtain the corresponding key KECS (KAF) of the UE, if it does not hold a valid KECS of the UE or the AKMA Key ID provided by the UE is different from the previous AKMA Key ID. The AAnF provides the derived key (KAF) to the ECS for the Edge Computing service. The KECS is the AKMA Application Key (KAF) and derived as specified in TS 33.535 [6] by both the UE and the ECS.

The key KECS is used by the ECS to derive the key KECS-PSK. The KECS-PSK is derived as defined in clause 6.3.2.1 of the present document, which is used as the PSK to establish TLS between the EEC and the ECS. Once the KECS-PSK is derived, the ECS includes the CounterECS used to derive the KECS-PSK, to the UE in the initial provisioning response message. On receiving the initial provisioning response message, the UE derives KECS-PSK, as derived by the AUSF using the received CounterECS value.

Step 2K: EEC establish the TLS session with the ECS, to secure the communication. TLS is used to provide integrity protection, replay protection and confidentiality protection for EDGE-4 interface. Mutual authentication is performed between the EEC and the ECS using TLS, based on pre-shared keys (KECS-PSK) following RFC 4279 [18] for TLS 1.2 and RFC 8446 [19] for TLS 1.3.

Step 2L-2M: Once TLS session is created successfully, the EEC initiates the service provisioning procedure with the ECS (as specified in clause 8.3 in TS 23.558 [2]) over the established TLS. If the UE is authorised to access the EES, then the ECS generates (as detailed in clause 6.3.2.2 of the present document) and provide the access token and ID token to the UE over the established TLS session. Additionally, the ECS may provide EES root CA certificate to the EEC, which is used to validate the EES's certificate.

Step 3: The UE performs EEC registration (as specified in clause 8.4.2 in TS 23.558 [2]) and discovery (as specified in clause 8.5 in TS 23.558 [2]) with the EES.

Step 3A: Before sending the access token and ID token to the EES, the UE and the EES establish a secure TLS connection using EES server certificate. Edge Configuration Server may provide EES root CA certificate during the initial provisioning procedure (Step 2M) to the EEC to validate the EES's certificate. TLS provides integrity protection, replay protection, and confidentiality protection over the EDGE-1 interface. It is required to protect and to provide the access token to an authentic EES.

Step 3C-3E: The UE initiates EEC registration procedure with the EES, including the access token and ID token obtained from the ECS in Step 2J. The access token and ID token included in registration request provides authentication and the authorization check for the EEC registration request by verifying of the access token and ID token issued by the ECS to the UE. The EES obtains the token validation service from the ECS.

Step 3F-3I: When the UE initiates EAS discovery procedure with the EES by including the same access token and ID token obtained from the ECS in Step 2M, if it is valid. Again the EES obtains the access token validation service from the ECS. The EES also optionally requests and obtains the token(s) from the ECS for the UE to grant access to the EAS(s). Then in response to the request, the EES optionally includes the EAS access grant token(s), with relevant information like validity time, to the UE.

If the obtained tokens from the ECS (in Step 2M) is not valid (due to time limitation), then the EEC requests ECS for a new access token as shown in figure 6.3.2-1. The token request message includes the necessary parameters to identify the EEC security context and parameters for authenticity verification. After verification of the authenticity, the ECS provides new tokens to the EEC, in response to the request.

NOTE : The authentication and authorization between AC and EAS is out of scope of the present document. For completeness, steps 4A-4F only detail a possible procedure to be used in application layer.

Steps 4A-4F: The UE obtains service from EAS, by producing the access token and ID token obtained from the EES, over the secure TLS connection. The UE also obtains security policy and the relevant tokens from the EES in Step 3I. Before sending the access token and ID token to the EAS, the UE and the EAS establish a secure channel using EAS server certificate. It is required to protect and to provide the access token and ID token to an authentic EAS. The EAS obtains the token validation service from the ECS via EES. After successful validation of the tokens, the UE obtains the Edge Computing service from the EAS.

6.3.2.2 KECS-PSK derivation

KECS-PSK generation is performed using the key derivation function (KDF) specified in Annex B.2.0 of TS 33.220 [8]. When deriving a KECS-PSK from KECS, the following parameters, KECS and CounterECS are used to form the input S to the KDF.

To generate the KECS-PSK, the ECS use a counter, called a CounterECS. The EEC and the ECS associates a 16-bit counter, CounterECS, with the key KECS. The ECS initializes the CounterECS to 0x00 0x01 when the KECS is derived. The EEC and the ECS maintains the CounterECS for lifetime of the KECS. The ECS sets the CounterECS to 0x00 0x02 after the first derived KECS-PSK, and monotonically increment it for each additional derived KECS-PSK.

The CounterECS is incremented by the ECS for every new computation of the KECS-PSK. The CounterECS is used as freshness input into KECS-PSK derivations, to mitigate the replay attack. The ECS sends the value of the CounterECS (used to generate the KECS-PSK) to the EEC. The EES accepts CounterECS value that is greater than stored CounterECS value. The ECS suspends the Initial provisioning procedure, if the CounterECS associated with the KECS is about to wrap around. When a fresh KECS is generated, the CounterECS at the ECS is reset to 0x00 0x01 and the ECS resumes the Initial provisioning procedure.

6.3.2.3 Access token generation

The access token is opaque to EEC and is consumed by the EES. The access token shall be encoded as a JSON Web Token as defined in IETF RFC 7519 [30]. The access token shall include the JSON web digital signature profile as defined in IETF RFC 7515 [29]. The access tokens shall convey the standards-based claims as defined in IETF RFC 7662 [31].

NOTE: Additional claims to be conveyed by access token is up to the EDGE service requirement.

6.3.2.4 ID token

The ID Token can be a JSON Web Token (JWT) and contain the standard claims as defined in by the OpenID Connect 1.0 specification [32] and are required for EDGE service implementation.

NOTE: Additional claims to be conveyed by ID token is up to the EDGE service requirement.

### 6.3.3 Solution evaluation

This solution reuses AKMA service as defined in TS 33.535 to use the network access credentials for the UE's authentication for Edge service. ID token and Access token are used for authentication and authorization of the UE respectively to obtain the Edge Computing service. These methods (OAuth2.0 and OpenID Connect 1.0) are already being utilized in Mission Critical Services (see TS 33.180), SEAL based procedures (see TS 33.434) and Service-Based procedures in TS 33.501 (OAuth 2.0).

## 6.4 Solution #4: Authentication/Authorization framework for Edge Enabler Client and Servers

### 6.4.1 Introduction

This solution addresses the security requirement for the Authentication and Authorization of EEC in key issue #1 and key issue #2, Key issue #6(for EDGE-1, EDGE-4 interfaces). The solution should work for all the scenarios described in 23.558 [2]. e.g. MNO Owned ECSP and non-MNO owned ECSP. Another scenario where the solution should be beneficial where UE already has a business relationship (e.g. subscribed to services) with ECSP and MNO has a business relationship ECSP then UE should use existing authentication/authorization methodologies to connect to ECSP to avail services.

Note: Secondary Authentication is performed in this solution.

### 6.4.2 Solution details



Figure 6.4.2-1: Secondary Authentication Based Authentication/Authorization framework for Edge Enabler Client and Servers

The procedure includes the following steps:

Step 0: UE pre-configuration: If the ECS deployed by MNO is contracted with one or more ECSP(s), the ECS provides EES configuration information of MNO owned, and ECSP owned EESs via MNO ECS as described in clause 8.3.3.2 in 23.558 [2]. If a non-MNO ECSP deploys the ECS, the ECS endpoint address may be configured with the EEC. An EEC that is aware of multiple ECSP's ECS endpoint addresses may perform the service provisioning procedure per ECS ECSP multiple times. As part of provisioning EEC may have installed ECS's TLS certificates.

Step 1: Primary Authentication: In this step, UE performs primary authentication with the network.

Step 2a, 2b: PDU session: As a result of UE initiating the service provisioning procedure with the ECS (as specified in clause 8.3 in TS 23.558 [2]), UE establishes a PDU session. This PDU Session may be established either to a well-known or pre-configured S-NSSAI or DNN, or the 5GC derives the S-NSSAI by using the registration for UE to network in step 1. Based on this information, the AMF selects an SMF, which in turn selects a PSA that provides a data connection to the Edge Cloud Service Provider's (Edge Data Network's) AAA Server. SMF continues secondary authentication as per clause 11.1.2 in 33.501[7]. ECS may act as DN-AAA Server.

Step 3a, 3b: After successful UE-requested PDU Session Establishment authentication/authorization by an EDN-AAA server, the device discovers and connects to a ECS server address (that was preconfigured in the UE in step 0 or is derived from the application identifier and/or Service Provider Identifier provided by the user in step 1) for provisioning EEC with ECS. The UE performs EEC registration (as specified in clause 8.4.2 in TS 23.558 [2]) and Discovery (as specified in clause 8.5 in TS 23.558 [2]) with the EES.

EEC establish the TLS session with the ECS, to secure the communication. TLS is used to provide integrity protection, replay protection and confidentiality protection for EDGE-4 interface. Certificate-based Mutual authentication is performed between the EEC and the ECS using TLS, following RFC 5246 [25] for TLS 1.2 and RFC 8446 [19] for TLS 1.3 After successfully establishing the secure session over EDGE-4 as in step 2, the Edge Enabling Client should send an Initial Provisioning request with Access Token Request message to the Edge Configuration Server as per the OAuth 2.0 specification. The Edge Configuration Server should verify the Access Token Request message per OAuth 2.0 specification. If the Edge Configuration Server successfully verifies the Access Token Request message, the Edge Configuration Server should generate an access token specific to the Edge Enabling Client and return it in an Initial Provisioning Response (Access Token Response) message.

Step 4.a: On EDGE-1, the Edge Enabling Client authenticates to the Edge Enabling Server by establishing a TLS session with the Edge Enabling Server based on the Server (Edge Enabling Server) side certificate authentication or certificate-based mutual authentication) as indicated by Edge Configuration Server. Edge Configuration Server may provide Edge Enabling Client's root CA certificate during the registration response (as specified in clause 8.4 in TS 23.558[2]) to the Edge Enabling Server to validate the Edge Enabling Client's certificate. TLS provides integrity protection, replay protection, and confidentiality protection over the EDGE-1 interface. It is required to protect and to provide the access token to an authentic EES.

Step 4.b: The UE initiates the EEC registration procedure with the EES, including the access token obtained from the ECS in Step 3.b. The authorization check for the EEC registration request is performed by verifying the access token issued by the ECS to the UE. The EES obtains the access token validation service from the ECS. In another option, the access token validation service by the ECS could be replaced by an authorization service by the ECS that does not require a token to be issued by the ECS to the UE but details are not in scope of this solution.

Step 5: EEC requests a service (e.g. Discovery) with access token obtained in step 4. The Edge Enabling Server should validate the access token. The Edge Enabling Server verifies the integrity of the access token by verifying the Edge Configuration Server signature. If validation of the access token is successful, the Edge Enabling Server should verify the Edge Enabling Client's Service request against the authorization claims in the access token, ensuring that the Edge Enabling Client has access permission for the requested service.

e.g. When the UE initiates the EAS discovery procedure with the EES by including the same access token obtained from the ECS in Step 3.b if it is valid. Again, the EES obtains the access token validation service from the ECS. The EES also optionally requests and obtains the access token(s) from the ECS for the UE to grant access to the EAS(s). In response to the request, the EES optionally includes the EAS access grant token(s), with relevant information like validity time, to the UE.

If the obtained access token from the ECS (in Step 3.b) is not valid, then the EEC requests ECS for a new access token, as shown in figure 6.3.X-1. The access token request message includes the necessary parameters to identify the EEC security context and parameters for authenticity verification. After verifying the authenticity, the ECS provides a new access token to the EEC in response to the request.

NOTE: The authentication and authorization between AC and EAS is out of scope of the present document. Step 6 is only a possible procedure to be used in application layer for completeness.

Step 6: The UE obtains service from EAS by producing the access token obtained from the EES over the secure TLS connection. The UE also obtains security policy and the relevant access token from the EES in Step 5. Before sending the access token to the EAS, the UE and the EAS establish a secure channel using the EAS server certificate. It is required to protect and to provide the access token to an authentic EAS. The EAS obtains the access token validation service from the ECS via EES. After successful validation of the access token, the UE obtains the Edge Computing service from the EAS.

### 6.4.3 Solution evaluation

The above solution proposes reusing the authentication and authorization between UE and Edge Data network using existing secondary authentication mechanisms as defined in TS 33.501. There is no impact on network entities and existing procedures.

After establishing the authentication and authorization using secondary authentication with Edge AAA server, EDGE-1, Edge-4 interfaces are further protected using TLS. TLS provides integrity protection, replay protection, and confidentiality protection over the EDGE-1 and Edge 4 interfaces.

An access token mechanism provides authorization for Edge-1. The solution can be amended by an authorization service by the ECS instead of an access token mechanism.

Solutions comply with all app-based platforms and the majority of deployed application solutions on the Internet today, which rely on the basic principle where a network server (in the role of Authenticator) authenticates the device (in the role of Supplicant) by communicating with a backend Authentication Server. The key benefit of this Solution compared with AKMA based solutions proposed in the present document is that the additional system impact on enabling AKMA on the ECSP network is avoided. Also, it avoids putting a burden on the ECSP to support AKMA.

## 6.5 Solution #5: Authentication and Authorization between the Edge Enabler Client and the Edge Enabler Server

### 6.5.1 Introduction

The following solution addresses the security requirement for the key issue #1 on Authentication and Authorization between the EEC and the EES.

In clause 8.3.2.3 of TS 23.558[2], before the service provisioning procedure, the Edge Enabler Client should been authorized to communicate with the Edge Configuration Server. From the security perspective, three security requirements are specified for the access of UE to Edge Data Network.

- It needs to ensure that only PLMN authorized UE can access to the Edge Data Network.

- It needs to ensure that only edge computing service authorized UE can access to the Edge Data Network.

- The URI or address information of Edge Enabler Server is the entry information for Edge Data Network when the ECS is within the MNO.

This solution proposes a mechanism to reuse the secondary authentication for the authorization of the PLMN PDU session establishment for the authentication between the EEC and the EES.

Based on the secondary authentication procedure, the client is authenticated by the EES. The EES will allocate the Edge Application Server information to the client. Then the client can use this URI information of the Edge Application Server to consume the edge service.

### 6.5.2 Solution details



Figure 6.5.2-1: Authentication and Authorization between the EEC and the EES

For this solution implement, there is a prerequisite: both the UE and the EES support the secondary authentication.

The procedure assumes that the Edge Configuration Server is deployed by the MNO. In this case, the EES is the authentication server in the Edge Data Network.

1. The UE registers in the operator network and perform the primary authentication procedure. After primary authentication, the UE has the information of Edge Configuration Server.

Note 1: How the ECS is involved in the primary authentication is to be clarified.

2 When the UE triggers the edge service it sends the PDU session establishment request to the AMF to setup the PDU session for the services provided by Edge Data Network. The SMF should trigger EAP Authentication procedure and perform the role of the EAP Authenticator.

3-4. The steps 3, 4, are the same as steps 5a-5b in clause 11.1.2 of TS 33.501[7].

5. The secondary authentication procedure is required to perform if the SMF check the UE has not been authenticated and authorized by the EES. The EES is the authentication server (AAA) of the Edge Data Network.

Note 2: How to make SMF aware that it should communicate with EES for secondary authentication is to be clarified.

6. This step is the same as steps 8-15 in clause 11.1.2 of TS 33.501[7].

7. After the successful completion of the secondary authentication procedure, the EES sends EAP Success message to the SMF including the registration response.

8. The SMF sends a Namf\_Communication\_N1N2MessageTransfer to the AMF with the received information.

9. The AMF forwards NAS SM PDU Session Establishment Response message including EAP Success.

### 6.5.3 Solution evaluation

This solution reuses the secondary authentication procedure to address the security requirement for the key issue #1. This solution provides the authentication between the EEC and the EES in case where the ECS is deployed by the MNO.

In this solution, the authentication of EEC by the EES is based on the authentication of UE with User ID (EAP ID), the EES needs to check the mapping relationship between the UE ID and the EEC ID additionally.

Note 1: The mapping relationship between the UE ID used during the secondary authentication and the EEC ID is not addressed here.

Note 2: It is not addressed here how the EEC is authenticated with the relationship between EEC ID and User ID.

The SMF checks whether the UE should be authenticated by the EES or not.

The EES plays the role of EDN-AAA server and support secondary authentication.

## 6.6 Solution #6: Authentication and Authorization between the Edge Enabler Client and the Edge Enabler Server

### 6.6.1 Introduction

The following solution addresses the security requirement for the key issue #1 on Authentication and Authorization between the EEC and the EES.

In clause 8.3.2.3 of TS 23.558 [2], before the service provisioning procedure, the Edge Enabler Client should be authorized to communicate with the Edge Configuration Server. From the security perspective, three security requirements are specified for the access of UE to Edge Data Network:

- It needs to ensure that only PLMN authorized UE can access to the Edge Data Network.

- It needs to ensure that only edge computing service authorized UE can access to the Edge Data Network.

- The URI or address information of Edge Configuration Server is the entry information for Edge Data Network when the ECS is deployed by the ECSP.

When the Edge Configuration server is deployed by the ECSP, the ECS information is preconfigured in the UE. This solution proposes a mechanism how the Edge Enabler Client is authenticated and authorized by the Edge Enabler Server when the Edge Configuration server is deployed by the ECSP. The secondary authentication for the authorization of the PLMN PDU session establishment will be reused as service authentication. The SMF should perform the role of the EAP Authenticator and communicates with the ECS (AAA).

Based on the secondary authentication procedure, the client is authenticated. The ECS will allocate the Edge Enabler Server information to the EEC. Then the client can use this URI information of the Edge Enabler Server to communicate with the Edge Date Network. It takes advantage of the secondary authentication between EEC and ECS to realize the authentication between the EEC and the EES. The authorization between the EEC and the EES is performed via the ECS.

### 6.6.2 Solution details



Figure 6.6.2-1: Authentication and Authorization between the EEC and the EES

For this solution implement, there is a prerequisite: both the UE and the ECS support the secondary authentication.

The procedure assumes that the Edge Data Network is deployed by the ECSP. Both the ECS and the EES stores the mapping between the EEC ID and GPSI for each EEC. The ECS will store the allowed EES list and the subscription expiration time.

Note 1: It is not addressed here whether the ECS and the EES can obtain a mapping between the EEC ID and GPSI for each EEC.

1. The UE registers in the operator network and perform the primary authentication procedure.

2. When the UE trigger the edge service it sends the PDU session establishment request to the AMF to setup the PDU session for the services provided by Edge Data Network.

3-4. The steps 3, 4 are the same as steps 5a-5b in clause 11.1.2 of TS 33.501[7]. The secondary authentication procedure is required to perform if the SMF check the UE has not been authenticated and authorized by the ECS. The ECS is the authentication server of the Edge Data Network.

Note 2: How to make SMF aware that it should communicate with ECS for secondary authentication is to be clarified.

6. This step is the same as steps 8-15 in clause 11.1.2 of TS 33.501[7].

7. After the successful completion of the authentication procedure, the ECS sends EAP Success message to the SMF including the GPSI.

8. The SMF sends a Namf\_Communication\_N1N2MessageTransfer to the AMF with the received information.

9. The AMF forwards NAS SM PDU Session Establishment Response message along with EAP Success and the GPSI to the UE/EEC.

10. The EEC sends Edge Enabler Client registration request to the EES.

11. The EES should verify the mapping between the EEC ID and GPSI. Then the EES requests to validate the authorization of the EEC from the ECS with the EEC ID. The ECS will check whether the EEC has been authorized to access to the EES for edge computing service with GPSI corresponding to the EEC ID.

12. If the EEC is authorized, the ECS responses to the EES with the service authorization response message.

13. After successful service authorization verification, the EES sends Edge Enabler Client registration response to the EEC.

### 6.6.3 Solution evaluation

This solution reuses the secondary authentication procedure to address the security requirement for the key issue #1. This solution provides the authentication between the EEC and the EES in case of the ECS is deployed by the ECSP.

In this solution, the authentication of EEC by the EES is based on the authentication of UE with user ID (EAP ID), the EES needs to check the mapping relationship between the UE ID and the EEC ID additionally.

Note 1: The mapping relationship between the UE ID used during the secondary authentication and the EEC ID is not addressed here.

Note 2: It is not addressed here how the EEC is authenticated with the relationship between EEC ID and UE ID.

The SMF checks whether the UE should be authenticated by the ECS or not.

The ECS plays the role of EDN-AAA server and support secondary authentication.

## 6.7 Solution #7: Authentication and Authorization with the Edge Data Network

### 6.7.1 Solution overview

The solution addresses the following key issues:

- Key issue #1: Authentication and Authorization between EEC and EES

- Key issue #2: Authentication and Authorization between EEC and ECS

- Key issue #6: Transport security for the EDGE-1-9 interfaces

The solution is based on the KAMF generated during the primary authentication. The network function that receives a registration request is querying the previous network function for authentication and the key for setting up an IPsec SA. Messages are protected with a MAC-I, which is also used to authenticate the UE.

The preferred ECS deployment scenario of the solution is, when the ECS is located in the serving network or hosted by a 3rd party service provider, since the services are to be hosted close to the UE's access point of attachment, to achieve an efficient service delivery through the reduced end-to-end latency and load on the transport network. In case of special roaming scenarios where the ECS is only located in the HPLMN while the UE is in a VPLMN, the KECS is then derived from the VPLMN KAMF.

NOTE: Edge Computing typically applies to non-roaming and LBO roaming scenarios.

In this solution it is assumed that the EEC ID is known to the UE and configured in the subscription profile. If the EEC ID is not configured in the UE/AMF, the AMF cannot identify the corresponding UE profile, thus the EEC ID needs to be provisioned in advance of the procedure.

6.7.2 Solution details



Figure 6.7.2-1: Authentication and Authorization with the Edge Data Network

1. The UE performs normal primary authentication and registration to the network. The UE is MEC capable and may indicate this in the MEC capabilities to the AMF during the registration procedure.

2. The UE establishes a PDU Session for IP connectivity.

3. If the UE is MEC capable and intends to use the MEC service, then the UE generates a MEC Key Identifier (MKI) for the MEC service and derives a key KECS for authentication with the ECS from the AMF key KAMF taking the MKI as input to the KDF. The lifetime of the keys may depend on the MEC application and configuration in the UE, or be synchronized with KAMF change. The MKI is assumed to be a counter. If the UE wants to use different MEC services at the same or different ECS, EES and EAS, the MKI is used to identify the particular key and to ensure the keys are different per service. The UE and AMF initialize the CounterECS when the KECS is derived and the counter is stored for the lifetime of the KECS.

4. The UE sends an Application Authentication Request with a MKI, MAC-IECS , GUAMI and a EEC ID to the ECS. The MAC-IECS is computed in a similar way as e.g. the SoR-MAC-IAUSF as defined in Annex A.17 of TS 33.501 [7]. The MAC-IECS is based on the payload of the Application Authentication Request, which form the input Application Authentication Request Data, a counter of the ECS messages CounterECS, and the key KECS to the KDF. The MAC-IECS is identified with the 128 least significant bits of the output of the KDF. The UE monotonically increment CounterECS for each additional calculated MAC-IECS. GUAMI is provided by the ME (from the allocated GUTI) to the EEC along with the KECS. The GUAMI has the form according to 23.003 of   
 <GUAMI> = <MCC><MNC> <AMF Region ID><AMF Set ID><AMF Pointer>

With this information it is possible to find the same AMF with the NEF as also selected by the gNB/N3IWF, since also in a similar case of IDLE mode mobility, the new gNB where the UE switches into CONNECTED mode needs to select the same AMF as well. There is no privacy issue with the GUAMI since it is a pure network entity identifier and not identifying any UE. The GUAMI is already transmitted over the air interface as part of the 5G-GUTI.

NOTE 1: In case the registration request is sent to the ECS just before an AMF relocation, the request will be routed to the old AMF, which rejects the request.

5. The UE is not authenticated at the ECS and the ECS sends a Key Request including the entire Application Registration Request to the NEF, which is selected based on EEC ID. The NEF selection is specified in TS 23.502 and the ECS may determine the IP address(es)/port(s) of the NEF by performing a DNS query using the EEC ID, or by using a locally configured NEF identifier/address.

6. The NEF authorizes the request from the ECS and identifies the AMF based on the GUAMI. The NEF stores the contact of the ECS (e.g. IP address, source NAI of the ECS etc.) with the EEC ID in order to route the answer from the AMF back to the ECS.

7. The NEF forwards the Key Request including the entire Application Authentication Request to the AMF.

8. The AMF derives the key KECS from the AMF key KAMF, taking the MKI as input to the KDF and verifies the MAC-IECS of the Application Authentication Request, i.e. it computes with the key KECS the MAC-I over the Application Authentication Request payload in the similar way as the UE and compares the result with the MAC-IECS included in message. If both are identical, the message can be authenticated to be sent by the UE, and the AMF monotonically increments CounterECS.

9. The AMF sends a Key Response to the ECS, including the result of the authentication as well as the KECS.

10. Based on the authentication result the ECS decides whether to accept or to reject the Application Authentication Request from the UE. The ECS sends the Application Authentication Response message to the UE including the authentication result and protects the message with a MAC-IECS based on the received key KECS in a similar way as the UE protected the payload of the message in step 4. The ECS associates the KECS to the requested service based on the MKI.

11. The UE verifies the MAC-IECS and if authentication result and verification of the message are successful, then the UE establishes an IPsec SA between the UE and ECS by using the ECS key KECS. All messages are now confidentiality and integrity protected by the IPsec tunnel.

12. The UE derives the key KEES from the key KECS using a MEC Key Distinguisher flag and the MKI as input to the KDF. In case different EES are used for different services, the UE uses a MKIEES as the MKI in a similar way as for the ECS, i.e. incrementally counting the EES requests.

13. The UE sends an Application Registration Request with the MKI and a MAC-IEES to the EES. The MAC-IEES is computed based on the payload of the Application Registration Request, which form the input Application Registration Request Data, and the key KEES to the KDF. The MAC-IEES is identified with the 128 least significant bits of the output of the KDF.

14. The UE is not authenticated at the EES and the EES sends a Key Request with the MKI to the ECS. The selection of the ECS may be based on the UE ID.

15. The ECS identifies the UE based on the UE ID and derives the key KEES in a similar way as the UE in step 10. The MKI is used to identify the particular key and to ensure the keys are different per service at the same or different EES. The ECS verifies the MAC-IEES of the Application Registration Request, i.e. it computes with the key KEES the MAC-I over the Application Registration Request payload in the similar way as the UE and compares the result with the MAC-IEES included in message. If both are identical, the message can be authenticated to be sent by the UE.

16. The ECS sends a Key Request Response to the EES, including the result of the authentication as well as the KEES.

17. Based on the authentication result the EES decides whether to accept or to reject the Application Registration Request from the UE. The EES sends the Application Registration Response message to the UE including the authentication result and protects the message with a MAC-IEES based on the received key KEES in a similar way as the UE protected the payload of the message in step 15. The EES associates the KEES to the requested service based on the MKI.

18. The UE verifies the MAC-IEES and if authentication result and verification of the message are successful, then the UE establishes an IPsec SA between the UE and EES by using the EES key KEES. All messages are now confidentiality and integrity protected by the IPsec tunnel.

NOTE 2: The authentication and authorization between AC and EAS is out of scope of the present document.

The following steps 19 – 25 only present a possible procedure to be used in application layer for completeness.

19. The UE derives the key KEAS from the key KEES using a MEC Key Distinguisher flag and the MKI as input to the KDF. In case different EAS are used for different services, the UE uses a MKIEAS as the MKI in a similar way as for the ECS, i.e. incrementally counting the EAS requests.

20. The UE sends an Application Registration Request with a MAC-IEAS to the EAS. The MAC-IEAS is computed based on the payload of the Application Registration Request, which form the input Application Registration Request Data, and the key KEAS to the KDF. The MAC-IEAS is identified with the 128 least significant bits of the output of the KDF.

21. The UE is not authenticated at the EAS and the EAS sends a Key Request with the MKI to the EES. The selection of the EES may be based on the UE ID.

22. The EES identifies the UE based on the UE ID and derives the key KEAS in a similar way as the UE in step 17. The MKI is used to identify the particular key and to ensure the keys are different per service at the same or different EAS. The EES verifies the MAC-IEAS of the Application Registration Request, i.e. it computes with the key KEAS the MAC-I over the Application Registration Request payload in the similar way as the UE and compares the result with the MAC-IEAS included in message. If both are identical, the message can be authenticated to be sent by the UE.

23. The EES sends a Key Request Response to the EAS, including the result of the authentication as well as the KEAS.

24. Based on the authentication result the EAS decides whether to accept or to reject the Application Registration Request from the UE. The EAS sends the Application Registration Response message to the UE including the authentication result and protects the message with a MAC-IEAS based on the received key KEAS in a similar way as the UE protected the payload of the message in step 22. The EAS associates the KEAS to the requested service based on the MKI.

25. The UE verifies the MAC-IEAS and if authentication result and verification of the message are successful, then the UE establishes an IPsec SA between the UE and EAS by using the EAS key KEAS. All messages are now confidentiality and integrity protected by the IPsec tunnel.

### 6.7.3 Solution evaluation

The solution is based on the KAMF generated during the primary authentication. The network function that receives a registration request is querying the previous network function for authentication and the key for setting up an IPsec SA. Messages are protected with a MAC-I, which is also used to authenticate the UE.

The AMF needs to provision the GUAMI to the UE. To authenticate requests from the UE at the ECS, the ECS queries the AMF to verify the received MAC-I and to retrieve the KECS. The AMF needs to understand the Key Request (including EEC ID, MKI, etc.) from the ECS and support the generation of KECS. The KECS is used to establish an IPsec SA between the UE and ECS.

To authenticate requests from the UE at the EES, the EES queries the ECS to verify the received MAC-I and to retrieve the KEES. The KEES is used to establish an IPsec SA between the UE and EES.

The UE needs to send in all MEC registration requests the MAC-I for authentication.

The UE needs to support the generation of MKI, KECS, KEES.

EEC ID verification is performed in the AMF based on the information in the key request from the EEC. If the subscription information from the UDM also contains specific information about the ECS, the AMF can perform ECS authorization as well.

## 6.8 Solution #8: Authentication between EEC and EES

### 6.8.1 Solution overview

This solution addresses the security requirement for the Authentication between EEC and EES in key issue #1.

In this solution, UE knows to use AKMA with EES via interact with ECS before communication with EES. If the EES deployed by MNO is considered to be trusted by the operator, the EES interacts directly with AAnF. Otherwise, the EESs not allowed by the operator to access directly the Network Functions should use the NEF to interact with AAnF.

### 6.8.2 Solution details

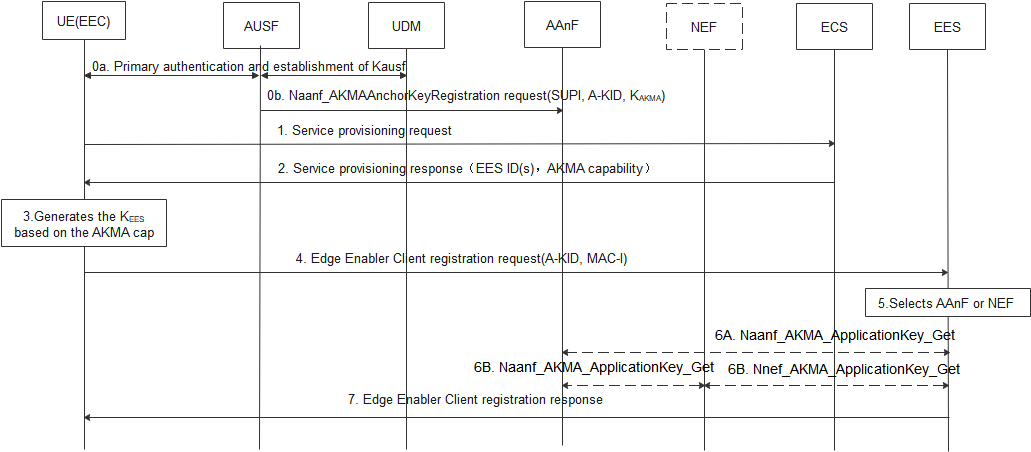


Figure 6.8.2-1 Authentication between the EEC and EES based AKMA

0a. UE performs primary authentication with the network. Then KAUSF is shared between UE and AUSF in Home network. If the AUSF receives the AKMA indication from the UDM, the AUSF should generate the AKMA Anchor Key (KAKMA) and the A-KID from KAUSF after the primary authentication procedure is successfully completed.

0b. After AKMA key material is generated, the AUSF should send the generated A-KID, and KAKMA to the AAnF.

1-2. The UE initiates the service provisioning procedure with the ECS. The ECS provides Edge Enabler Server Information (EES ID (i.e. FQDN or IP address(es) of EES), AKMA capability) to the UE. The AKMA capability indicates the EES support to use AKMA.

3. When the UE determines to communicate with EES, if the UE supports AKMA, the UE derives the AKMA key and the KEES(i.e. Kaf) as specified in TS 33.535 [6] based on the received AKMA capability.

4. The UE computes the MAC-I over the request message using the KEES and sending Edge Enable Client registration request with A-KID and MAC-I.

NOTE 1: TLS based on AKMA PSK solution is defined in other solutions.

5. Upon receiving the request, the Edge Enabler Server discovers the AAnF or NEF.

NOTE 2: In the case of architecture without CAPIF support, the EES is locally configured with the API termination points for the service. In the case of architecture with CAPIF support, the EES obtains the service API information from the CAPIF core function via the Availability of service APIs event notification or Service Discover Response as specified in TS 23.222 [9].

6. The EES contacts AAnF directly or via NEF to obtains the corresponding key KEES of the UE (as defined in TS 33.535 [6])..

7. The EES verifies the MAC-I using the KEES, when the verification is succeed, and if the UE is authorized to perform the operation. The EES computes MAC-I over the response message using KECS and sends Edge Enable Client registration response with the MAC-I to the UE.

### 6.8.3 Solution evaluation

This solution requires the ECS to provide the AKMA capability of EES(s) to the UE, then the UE can determine to use AKMA when establish connection with EES.

This solution also requires that the EEC and EES supports the AKMA.

Authentication of apps in the UE before accessing to the key is out of scope of this solution.

For the case of multiple EECs in UE, if the EECs connects to different EESes, separate application key KAF is used. If the EECs connects to the same EES, the same application key KAF is used, how to resolve the key collision is not defined in this solution.

## 6.9 Solution #9: Authentication and authorization between EEC and ECS based on AKMA

### 6.9.1 Introduction

This solution addresses the key issue #2. It is assumed that the key used for authentication between EEC and ECS is negotiated based on AKMA. Then, the EEC should initiate the service provisioning request with EEC ID included. To prevent EEC ID impersonation, the ECS should verify the authenticity of UE's EEC ID before performing authorization based on the EEC ID. Considering the ECS can determine the authenticity of UE's A-KID based on AKMA procedure, it can confirm the authenticity of UE's EEC ID in case the association between A-KID and EEC ID can be verified. This solution further transforms the association between A-KID and EEC ID to the association between A-KID and GPSI based on the pre-configured association between EEC ID and GPSI in ECS. Afterwards, the ECS interworks with 5GC to verify the association between A-KID and GPSI.

After successful verification, the ECS may retrieve the edge computing related profile for the EEC from the 3GPP Core Network or from its local database for edge computing. Then, the ECS can determine the EEC's authorization based on the profile.

### 6.9.2 Solution details



Figure 6.9.2-1: Authentication/Authorization between Edge Enabler Client and ECS

Pre-conditions:

- The EEC and ECS have shared A-KID and KAF via AKMA (as specified in TS 33.535 [1]).

- The ECS or the 5GC is configured with the edge computing related profile for the EEC.

- The ECS and the 5GC share an UE identifier (i.e. GPSI) to identify the EEC.

- The ECS stores the association between EEC ID and UE identifier. This association is pre-configured in the ECS by the ECS administrator.

Step 1: UE initiates the service provisioning procedure with EEC ID included (as specified in clause 8.3 in TS 23.558 [2]).

Step 2: The ECS retrieves GPSI from EEC ID according to the preconfigured association.

Step 3: In order to prove the authenticity of the UE's GPSI, the ECS sends an association check request to UDM (if the ECS is located out of 5GC, the request should be sent via NEF), including the GPSI and A-KID.

Step 4: In order to verify the association of GPSI and A-KID, the UDM first contacts the AUSF to obtain the corresponding SUPI of the A-KID. Afterwards, the UDM verifies the association of the GPSI and A-KID according to the association between SUPI and GPSI.

Step 5: The UDM sends the association verification response back to the ECS.

Step 6: On successful verification, the ECS retrieves the edge computing related profile for the EEC either from the 5GC or from its local database. Afterwards, the ECS can determine the EEC's authorization based on EEC's profile.

Step 7: The ECS sends the provisioning response back to EEC.

### 6.9.3 Solution Evaluation

This solution fully addresses the security requirement for authentication between EEC and ECS in the key issue #2 based on the AKMA.

This solution requires the ECS or the 5GC is configured with the edge computing related profile for the EEC and the ECS and the 5GC share an UE identifier (i.e. GPSI) to identify the EEC

This solution also requires that the EEC and ECS support the AKMA. The solution assumes the association between EEC ID and UE ID (i.e. GPSI) which is pre-configured in the ECS by the ECS administrator and verifies the association during service provisioning which requires updates to the service of UDM and AAnF.

This solution does not apply to the case when there are multiple EECs in one UE.

The authentication of EEC before giving the Kaf in the UE is out of scope.

## 6.10 Solution #10: Authentication and Authorization between the Edge Enabler Client and the Edge Configuration Server

### 6.10.1 Introduction

The following solution addresses the security requirement for the key issue #2 on Authentication and Authorization between the EEC and the ECS.

In clause 8.3.2.3 of TS 23.558 [2], before the service provisioning procedure, the Edge Enabler Client should been authorized to communicate with the Edge Configuration Server. From the security perspective, two security requirements are specified for the access of UE to Edge Data Network.

It needs to ensure that only authorized UE in the PLMN can access to the Edge Configuration Server.

It needs to ensure that only edge computing service authorized UE can access to the Edge Data Network.

This solution proposes a solution to address how the Edge Enabler Client is authorized to achieve the URI or address information of Edge Configuration Server after authentication when the Edge Configuration Server is deployed by the MNO. It reuses the primary authentication for the service authentication. The service authorization is performed via the 5GC.

Based on the primary authentication procedure, the AMF will allocate the Edge Configuration Server information to the UE. Then the Edge Enabler Client can use this URI information of the Edge Configuration Server to communicate with the Edge Configuration Server.

### 6.10.2 Solution details



Figure 6.10.2-1: Authentication and Authorization between the Edge Enabler Client and the Edge Configuration Server

The procedure assumes that the Edge Configuration Server is deployed by the MNO.   
The authentication between the Edge Enabler Client and the Edge Configuration Server reuses the 5G primary authentication procedure in the PLMN. The UE can obtain information to access the Edge Configuration Server via the registration accept message. The Authorization information of the Edge Enabler Client is retrieved from the UDM in the 5GC.

1. The UE sends a Registration Request with edge computing service capability.

2. The UE and the 5G core network preform the primary authentication procedure as described in clause 6.1.2 of TS 33.501[7].

3. The AMF sends UE the Registration Accept message with suitable ECS address information.

4. The Edge Enabler Client sends service provisioning request to the ECS.

5. The ECS verifies the binding between the EEC ID and the GPSI. Then the ECS obtains service authorization of the Edge Enabler Client from the UDM via the NEF. The UDM stores an indicator whether the user is allowed to use edge computing service and the allowed ECS list. The indicator can be used as the service authorization information for the EEC. The subscription expiration time and the binding between the EEC ID and the GPSI may also be stored in the UDM. When the ECS sends request, the GPSI is included in the request message.

Note: It is not addressed here how the ECS verifies the binding between EEC ID and GPSI, and how the UDM stores the binding between EEC ID and GPSI.

6. The UDM will check whether the user has been authorized to access to the ECS for edge computing service with GPSI corresponding to the EEC ID. If the user is authorized, the UDM responses with the service authorization response message to the ECS via the NEF.

7. The ECS sends service provisioning response to the Edge Enabler Client.

### 6.10.3 Solution Evaluation

This solution addresses the security requirements of key issue #2 on Authentication and Authorization between the EEC and the ECS.

## 6.11 Solution #11: Authentication between EEC and ECS

### 6.11.1 Solution overview

This solution addresses the security requirement for the Authentication between EEC and ECS in key issue #2.

In this solution, UE knows to use AKMA with ECS via interactions with 3GPP network before communication with ECS. If the ECS deployed by MNO is considered to be trusted by the operator, the ECS interacts directly with AAnF. Otherwise, the ECSs not allowed by the operator to access directly the Network Functions should use the NEF to interact with AAnF.

### 6.11.2 Solution details

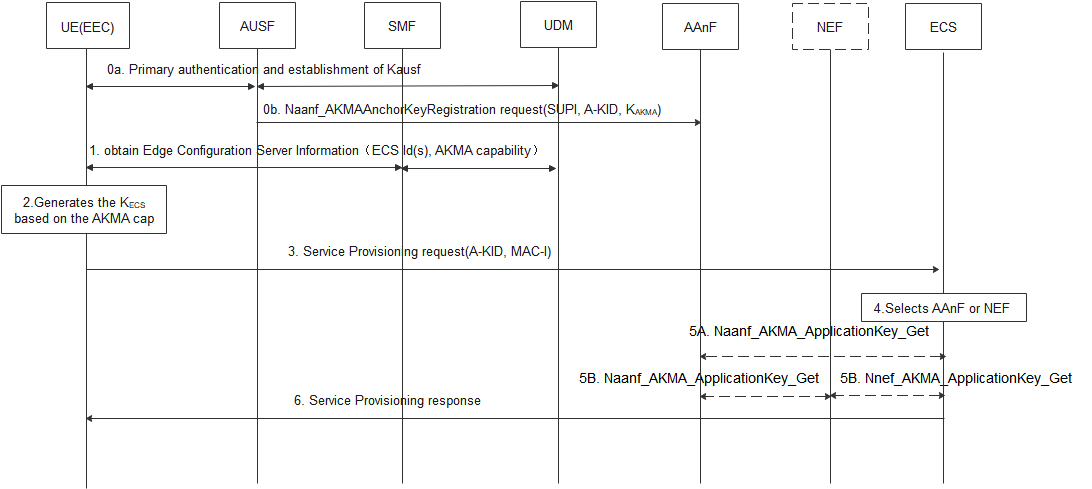


Figure 6.11.2-1 Authentication between the EEC and ECS based AKMA

0a. UE performs primary authentication with the network. Then KAUSF is shared between UE and AUSF in Home network. If the AUSF receives the AKMA indication from the UDM, the AUSF should generate the AKMA Anchor Key (KAKMA) and the A-KID from KAUSF after the primary authentication procedure is successfully completed.

0b. After AKMA key material is generated, the AUSF should send the generated A-KID, and KAKMA to the AAnF.

1) During PDU Session Establishment, the SMF provides Edge Configuration Server Information (ECS ID (i.e. FQDN or IP address(es) of ECS), AKMA capability) to the UE. The AKMA capability indicates the ECS support to use AKMA.

2) When the UE determines to communicate with ECS, if the UE supports AKMA, the UE derives the AKMA key and KECS(i.e. KAF) as specified in TS 33.535 [6] based on the received AKMA capability.

3) UE computes the MAC-I over the request message using the KECS and sends service provision request with A-KID, MAC-I.

NOTE 1: TLS based on AKMA PSK solution is defined in other solutions.

4) Upon receiving the request, the Edge Configuration Server discovers the AAnF or NEF.

NOTE 2 : In the case of architecture without CAPIF support, the ECS is locally configured with the API termination points for the service. In the case of architecture with CAPIF support, the ECS obtains the service API information from the CAPIF core function via the Availability of service APIs event notification or Service Discover Response as specified in TS 23.222 [9].

5A-5B. The ECS contacts AAnF directly or via NEF to obtain the corresponding key KECS (KAF) of the UE(as defined in TS 33.535).

6. The ECS verifies the MAC-I based on the KECS, when the verification is succeed, and if the UE is authorized to perform the operation. Then the ECS computes MAC-I over the response message using KECS, and sends a service provisioning response with the MAC-I to the UE.

### 6.11.3 Solution evaluation

This solution addresses the security requirement for authentication between EEC and ECS in the key issue #2 based on the AKMA.

This solution requires the 5GC to provide the AKMA capability of ECS(s) to the UE, then the UE can determine to use AKMA when establish connection with ECS.

This solution also requires that the EEC and ECS supports the AKMA.

Authentication of apps in the UE before accessing to the key is out of scope of this solution.

The key isolation between multiple EECs connected to the same ECS is not addressed in this solution.

## 6.12 Solution #12: Onboarding and authentication/authorization framework for Edge Enabler Server and Edge Configuration Server

### 6.12.1 Introduction

This solution addresses the security requirement for the Onboarding of EES with ECS, as described in Key issue 3. The solution proposes a framework and procedure that the Edge Enabling Server and the Edge Configuration Server follows to secure and authenticate the Registration, update, and deregistration of the Edge Enabling Server to the Edge Configuration Server. As per [2], ECS can be owned by MNO or ECSP. ECSP can have its own authentication/authorization independent of MNO. ECSP may also have its own EES. The Edge Configuration Server (ECS) can be deployed in the MNO domain or can be deployed in 3rd party domain by the service provider in which one Edge Enabling Client may communicate with one or more Edge Configuration Server (ECS)(s) concurrently. One Edge Enabling Server may concurrently connect to one or more Edge Configuration Server with a separate EDGE-6 reference point interface. The Edge enabling server that is configured with multiple Edge Configuration Server (ECS) endpoint addresses (es) may perform the service registration, updates, or deregistration procedures per the Edge Configuration Server (ECS) of each Edge Configuration Server (ECS) multiple times. In this context, the Security Context of each of the EDGE-6 interfaces needs to be separate from each other as the trust domain may be different. In this solution, the trust relationship is based on a business relationship for each EDGE-6 interface described above.

As a prerequisite to this procedure (step 1), the solution assumes that Onboarding credential information is obtained by EES within the same PLMN domain or from a third party domain. EES uses onboarding credentials to authenticate and establish a secure TLS communication with the Edge Configuration Server during the registration process. The credential information includes details of the Edge Configuration Server Address and Root CA certificate, and it may also include an onboarding token (e.g. OAuth 2.0 access token). Security profiles for TLS implementation and usage follow the provisions given in TS 33.310 [13], Annex E and F.

Note: ECS address that is not belonging to the credentials, is out of scope of the present document.

### 6.12.2 Solution details



Figure 6.12.2-1: Authentication/Authorization framework for EES with ECS

Step 1-2: The Edge Enabling Server and Edge Configuration Server should establish a secure session based on TLS (Server-side certificate authentication). The Edge Enabling Server should use the credential information obtained in step 1 to establish the TLS session with the Edge Configuration Server.

Step 3: After the successful establishment of the TLS session, the Edge Enabling Server should send an Edge Enabler Server Registration message to the Edge Configuration Server along with the credential (OAuth access token) and EES Profile. The Edge Enabling Server generates the key pair {Private Key, Public key} and provides the public key along with the Onboard Edge Enabling Server request.

Step 4: The Edge Configuration Server should validate the enrolment credential (OAuth token). After successful verification of credentials (OAuth Token), Edge Configuration Server may generate Edge Enabling Server's certificate on its own, for the assigned Edge Enabling Server identity and public key. For subsequent authentication procedures with the Edge Configuration Server, the Edge Enabling server may use this certificate to establish a secure connection and authentication with the Edge configuration Server. When the third party issues edge Enabling Server's client certificate, then in Step 3, the Edge Enabling Server can include the certificate in the Onboard Edge Enabling Server request message. If the Edge Configuration Server trusts the issuer of the Edge Enabling Server's client certificate, then the Edge Configuration Server includes the provided certificate in the Edge Enabling Server's profile in step 4. It is up to the Edge Computing Service Provider domain policy to accept the third party's client certificates.

Step 5: The Edge Configuration Server should respond with a Registration response message. The response should include the Edge Configuration Server assigned Edge Enabling Server Registration ID, Edge Enabling Server Authentication and authorization information (if generated in step 4), Edge Enabling Server's certificate.

### 6.12.3 Solution evaluation

EDGE-6 interface is protected using TLS. TLS provides integrity protection, replay protection, and confidentiality protection over the EDGE-6 interface. An O-Auth token mechanism provides authorization for EES authorization with ECS. The solution provides a mutual authentication mechanism and authorization mechanism between EES and ECS to register and update the server profile information.

With the above analysis, the solution meets the security requirements for Key issue 3.

## 6.13 Solution #13: Transport security for EDGE-1-9 interfaces

### 6.13.1 Introduction

This solution addressed the transport security requirements for EDGE-1-9 interfaces in key issue #6. Generally, NDS/IP should be used here for data protection.

### 6.13.2 Solution details

#### 13.2.0 General

There are three types of interfaces related with edge application architecture defined in TS 23.558 [2]. Hence, the transport security will be discussed separately within three subclauses.

#### 6.13.2.1 Type A

Interfaces of type A (Between UE and Edge servers) are as follows:

- EDGE-1: between EEC and EES

- EDGE-4: between EEC and ECS

- EDGE-5: between EEC and Application Client(s)

For the EDGE-1 and EDGE-4, it is proposed to use the TLS specified in TS 33.210 [12] if HTTP protocol is selected.

NOTE: For the other protocols of EDGE-1 and EDGE-4, the protection protocol can be defined during the normative phase.

According to TS 23.558 [2], details of the EDGE-5 is out of scope of that document. As the EDGE-5 is an internal interface within the UE, the security of EDGE-5 can be left for implementation.

#### 6.13.2.2 Type B

Interfaces of Type B (Between 3GPP core and Edge servers) are as follows:

- EDGE-2: between 3GPP Core network and EES

- EDGE-7: between 3GPP Core network and EAS

- EDGE-8: between 3GPP Core network and ECS

How to protect the interface between 3GPP Core network and EES/ECS/EAS, depends on the functionality, which will be performed on this interface.

As defined in TS 23.558 clause 6.4.2, it says.

*EDGE-2 reference point enables interactions between the Edge Enabler Server and the 3GPP Core Network. It supports:*

*a) access to 3GPP Core Network functions and APIs for retrieval of network capability information, e.g. via SCEF and NEF APIs as defined in 3GPP TS 23.501 [2], 3GPP TS 23.502 [3], 3GPP TS 29.522 [4], 3GPP TS 23.682 [17], 3GPP TS 29.122 [5], and with the EES acting as a trusted AF in 5GC (see 3GPP TS 23.501 [2] clause 5.13, 3GPP TS 23.503 [12]).*

*NOTE: EDGE-2 reference point reuses 3GPP reference points or interfaces of EPS or 5GS considering different deployment models.*

Similarly, EDGE-7/8 is used to support the same functionality as the EDGE-2.

Therefore, For EDGE-2/7/8:

- if the NEF APIs is selected, security aspects of Network Exposure Function including the protection of NEF-AF interface and support of CAPIF defined in TS 33.501 clause 12 [10] can be reused here to protect the EDGE-2/7/8 interfaces, i.e. use of TLS.

- if the SCEF APIs is selected, the Security procedures for reference point SCEF-SCS/AS defined in TS 33.187 clause 5.5 [11] can be reused here, i.e. use of TLS.

NOTE: Transport security protection of EDGE-2/7/8 can take the other deployment models in the future.

#### 6.13.2.3 Type C

Interfaces of type C (Between Edge servers) are as follows:

- EDGE-3: between EAS and EES. The supported functionalities include EAS registration, de-registration, etc.

- EDGE-6: between EES and ECS. The supported functionalities include EES registration.

- EDGE-9: between EES(s). The supported functionalities include discovery of target EAS.

As all the exchanged data of EDGE-3/6/9 is in the application layer, transport security protection on the SBI interface can be reused here. Hence TLS should be used as specified in TS 33.210 [12], unless security is provided by other means, e.g. physical security. A SEG may be used to terminate the NDS/IP IPsec tunnels.

For the EDGE-3, if the CAPIF capability is consumed by the EAS, the interface security defined in the TS 33.501 clause 12 can be reused here to protect the CAPIF related data transferred in the EDGE-3 interfaces, i.e. TLS should be used.

NOTE 1: Regardless of whether TLS is used or not, NDS/IP as specified in TS 33.210 [12] and TS 33.310 [13] can be used for network layer protection.

### 6.13.3 Solution Evaluation

This solution address all the types of interface in the key issue #6.

For the type A interface (EDGE-1/4), TLS can be reused if HTTP protocol is selected.

For the type B interface (EDGE-2/7/8), security aspects of Network Exposure Function can be reused if NEF APIs is selected and the security procedures for reference point SCEF-SCS/AS can be reused if SCEF APIs is selected.

For the type C interface (EDGE-3/6/9), transport security protection on the SBI interface can be reused.

## 6.14 Solution #14: Protection of Network Information Provisioning to Local AF directly

### 6.14.1 Solution overview

This solution addresses the security requirement for the case that the UPF exposes information to local AF directly in the key issue 7.

NOTE: The interface between local UPF and local AF is N6 and how to deliver the information on N6 is out of scope.

### 6.14.2 Solution details

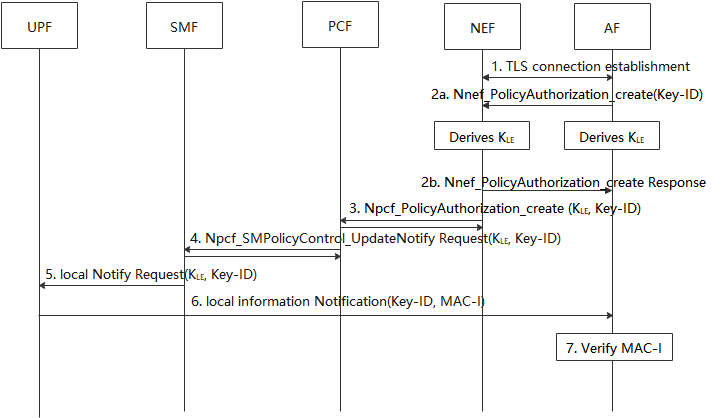


Figure 6.14.2-1: Protection of Network Information Provisioning to Local AF

1. AF establish a TLS session with the NEF, to secure the communication between the AF and NEF.

2a-2b. The AF generates a Key-ID for the UE and derives a KLE. The Key-ID is used to identify the KLE, and the KLE is used to protect the message transmission between UPF and AF. The AF provides the Key-ID to NEF in the request service.

3-4. The NEF initiates the policy authorization with PCF, including the K-ID and KLE received from the AF. PCF initiates the PDU session modification procedure as defined in clause 4.3.3.2 of TS 23.502 [3] and provides the Key-ID and KLE to the SMF.

5. SMF sends the notification information with Key-ID and KLE to the UPF.

6. When the QoS monitoring information is received from RAN, the UPF generates MAC-I over the message using the KLE to prove its authenticity. The UPF sends the message including Key-ID and MAC-I to AF.

7. The AF retrieves the KLE based on the received Key-ID and verify the MAC-I.

### 6.14.3 Solution evaluation

This solution fully addresses the security requirement for the case that UPF exposes the network information to local AF directly described in Key issue#7.

This solution is based on the KLE generated between the NEF and AF. The AF derives the Key-ID for the KLE and provides the Key-ID to NEF during network information provisioning subscription procedure. And the NEF provides the Key-ID and KLE via PCF to SMF and further to UPF.

When the UPF determines to provision the network information to AF via N6, the UPF uses the KLE to protect the network information and provides the Key-ID to AF for retrieving the KLE and verifying the MAC-I.

## 6.15 Solution #15: Network capability re-exposure via Edge Enabler Server

### 6.15.1 Introduction

In clause 5.8 of the present document, it describes the key issue on service capability APIs exposed by the Edge Enabler Server to the Edge Application Server(s). It states that the Edge Application Server(s) should be authenticated and authorized, otherwise attackers would potentially be able to perform unauthorized access or trigger DoS attacks .This solution is proposed to address the security requirement for authentication and authorization in EES capability exposure in key issue #8.

In this solution, it is proposed the Edge Enabler Server check whether the EAS is allowed to be authorized for the service capability re-exposure by the 5GC and the UE (EES capabilities and 3GPP Core Network capabilities) via the NEF.

### 6.15.2 Solution details

1. The Edge Application Server sends API request message to the Edge Enabler Server with the UE Identifier, related service exposure information and the Edge Application Server ID.

2. If the service can be provided by EES directly, the Edge Enabler Server will check whether the EAS has the authorization token, and it will provide the exposure information to the authorized EAS as specified in clause 8.6.2 in TS 23.558[2]. If the request service is provided EES indirectly, as the EES has not been authorized by 5G core network for the service exposure bases on SLA, the Edge Enabler Server should request the NEF to verify whether the service is allows to be exposed to the EAS by the EES with EAS ID. The authentication and authorization between the EES and the NEF reuses the mechanisms specified in clause 12 in TS 33.501[7].

3. The NEF sends the verification response with the information whether the exposure is allowed or not for the request EAS.

4. If the EAS is authorized, the Edge Enabler Server responds to the Edge Application Server with service exposure information to the EAS.



Figure 6.15.2-1: The authorization for service capability APIs exposure to EAS

### 6.15.3 Solution Evaluation

This solution addresses the security requirement for authentication and authorization in EES capability exposure in key issue #8.

## 6.16 Solution #16: EEC authentication and authorization framework with ECS and EES

### 6.16.1 Introduction

This solution addresses the security requirement for authentication/authorization between EEC and ECS/EES in the key issue #1, and key issue #2. In this solution, the AKMA, and TLS is reused as the building blocks for the EEC authentication framework if the EDGE-4 is deployed based on the UP connection. NAS and SBI interface protection is reused if the EDGE-4 is deployed based on the CP connection. For the authorization, the Oauth is selected for the authorization between EEC and EES.

### 6.16.2 Solution details



Figure 6.16.2-1: EEC authentication and authorization framework with ECS and EES

It is assumed that AKMA, TLS, and OAuth is supported by the UE, ECS, and EES.

Step 1-3. UE with EEC functionality registers in the 5G network, and retrieves ECS information (i.e. ECS Id and AKMA capability) from 5GC. The AKMA capability indicates the ECS support to use AKMA.

There are two options for the authentication and data protection between EEC and ECS.

Option A (The EDGE-4 is deployed based on the UP connection):

Step 4-6. The UE determines to use AKMA based on the received AKMA capability, AKMA defined in the TS 33.535 [6] is reused here to negotiate the pre-shared key KECS between UE and ECS. Here the A-KID sent from the UE is authenticated by the ECS based on the AKMA mechanism.

Step 7a. EEC and ECS establish the TLS security tunnel based on the pre-shared key KECS. The authentication is fulfilled based on the TLS.

Option B (The EDGE-4 is deployed based on the CP connection):

Step 7b. The protection between EEC and ECS relies on the NAS and SBI interface protection. The authentication is implicitly performed as the primary authentication.

NOTE 1: CP connection option between EEC and ECS is not addressed in the present document.

NOTE 2: The following EDGE-4 data could be protected based on Option A or Option B according to its connection option.

Step 8. EEC sends the Provisioning request message to the ECS, including its EEC ID, application info.

Step 9. ECS determines the EES info (including EES Id, AKMA capability), and generate the Oauth token with the following claims, i.e. EEC ID, ECS ID, EEC info, the authorized services.

NOTE 3: Whether the EEC ID could use the edge service can be authorized based on the local policy. The token for consuming the ECS service may not be needed.

Step 10. ECS sends the token and EES info back to the EEC via the Provisioning response message.

Step 11. Similar with step 4-6, pre-shared key KEES is negotiated between EEC and EES.

Step 12. EEC and EES establish the TLS security tunnel based on the pre-shared key KEES. The authentication is fulfilled based on the TLS.

Step 13. EES sends the EEC registration/ discovery request message to the EES, including the token.

Step 14. EES authorizes the EEC based on the verification of the token.

Step 15. If the verification successes, the EES sends the EEC registration/ discovery response message back to the EEC.

### 6.16.3 Solution Evaluation

This solution fully addresses the security requirement for authentication between EEC and ECS in the key issue #1 based on the AKMA and TLS, and fully address the security requirement for authentication and authorization between EEC and ECS in key issue #2 based on AKMA, TLS and Oauth.

This solution requires that the EEC, ECS, and EES support the AKMA, TLS and Oauth mechanism. Similar with the SBA authorization, the ECS plays the authorization server role, EEC plays the EES service consumer, and the EES plays the service producer role within the Oauth architecture.

How to authenticate the EEC ID is not captured in this solution.

The key isolation issue between multiple EECs connected to the same ECS is not addressed in this solution.

## 6.17 Solution #17: EEC/EES/ECS authentication and transport protection with TLS

### 6.17.1 Solution overview

This solution addresses the Key Issues

- KI#1 "Authentication and Authorization between EEC and EES",

- KI#2 "Authentication and Authorization between EEC and ECS",

- KI#3 "Authentication and Authorization between EES and ECS", and

- KI#6 "Transport security for the EDGE-1-9 interfaces".

It proposes

- To use TLS as specified in RFC 5246 [25] and RFC 8446 [19] for authentication and transport protection of the EDGE-1 (EEC-EES), EDGE-3 (EAS-EES), EDGE-4 (EEC-ECS), EDGE-6 (EES-ECS) and EDGE-9 (EES-EES) interfaces,

- To use token-based authentication as another solution option for the authentication of the EEC by the ECS and EES,

- To use an existing challenge-response protocol like e.g. HTTP Digest as specified in RFC 7616 [24] with AKMA pre-shared key for authentication of the GPSI used in communication between EEC and EES/ECS, and

- To use the IP address to GPSI translation API as another solution option for the authentication of the GPSI.

### 6.17.2 Solution details

#### 6.17.2.1 Authentication and transport protection for the EDGE-1, EDGE-3, EDGE-4, EDGE-6 and EDGE-9 interfaces

This solution proposes to align the protection of the EDGE-1, EDGE-3, EDGE-4, EDGE-6 and EDGE-9 interfaces with similar mechanisms in existing 3GPP security specifications. It seems that especially the security mechanisms in TS 33.434 [23], i.e. the security mechanisms for SEAL, are applicable here. In TS 33.434 [23], the security mechanisms are different for the signalling control plane and for the application plane interfaces. For the signalling control plane, TS 33.434 [23] specifies that HTTPS nees to be used, e.g. in clause 5.1.1.3 IM-UU:

"IM-UU reference point is used between the identity management client and the identity management server. The IM-UU between the Identity Management client and the Identity management server is protected using HTTPS as defined in [3], [4] and [5]. The profile for TLS implementation and usage follows the provisions given in 3GPP TS 33.310 [6], annex E."

EDGE-1, EDGE-3, EDGE-4, EDGE-6 and EDGE-9 are the interfaces between EEC, EES, ECS and EAS. They can be seen as control plane interfaces for the application traffic between Application Client and EAS. Hence it seems reasonable that the security mechanisms should align with the signalling control plane security mechanisms in TS 33.434 [23]. However, the application protocol for the EDGE interfaces is not yet determined. Although HTTP is common practice, it seems premature to specify the usage of HTTPS. Instead it is proposed to use TLS. If HTTP is chosen as application protocol, then this solution proposes to use HTTPS.

Summing up, the proposed security mechanism for EDGE-1, EDGE-3 EDGE-4, EDGE-6 and EDGE-9 is:

"EDGE-1, EDGE-3, EDGE-4, EDGE-6 and EDGE-9 are protected using TLS as specified in RFC 5246 [25] and RFC 8446 [19]. The profile for TLS implementation and usage follow the provisions given in 3GPP TS 33.310 [13], annex E."

Regarding the identifiers used on these interfaces, TS 23.558 [2], clause 7.2, specifies different identifiers that could be relevant to this solution. More specifically:

- EDGE-1: TLS client is EEC (identified by EECID), TLS server is EES (identified by EESID).

- EDGE-3: TLS client is EAS (identified by EASID), TLS server is EES (identified by EESID).

- EDGE-4: TLS client is EEC (identified by EECID), TLS server is ECS (identifier not specified in TS 23.558 [2]).

- EDGE-6: TLS client is EES (identified by EESID), TLS server is ECS (identifier not specified in TS 23.558 [2]).

- EDGE-9: TLS client is EES (identified by EESID), TLS server is EES (identified by EESID).

Note 1 : TS 23.558 does not specify an identifier for the ECS.

Note 2: TS 23.558 specifies that the EASID identifies the application on the EAS, not the specific EAS. For example, all Edge SA6Video Servers will share the same EASID.

Another solution for the authentication of the EEC by the ECS and EES is the usage of tokens instead of TLS certificate of the EEC. For this option, the following solution is proposed:

- Solution for the interface EDGE-4: The authentication of the ECS and the transport security of the interface are realized by using TLS with server authentication using the server's certificate issued by CAs in the PKI. For the first authentication of the EEC by the ECS, the token, including the EEC ID, provided by the ECSP of the EEC or by a trusted new entity (that could or could not be collocated with the ECSP) to the EEC is used. In the case of provision of token by the ECSP, it is assumed that there is a business relationship between the ECSPs of the EEC and ECS, ECSP of the EEC provisions an initial access token to the EEC, and the ECS can verify the token. After the authentication of the EEC, the ECS provides a token to the EEC in the initial access to be used for the next establishment of the communication between them. In the other accesses than the initial access, the ECS decides on whether a new access token is necessary or not, considering information such as the expiration time of the token.

- Solution of the interface EDGE -1: The authentication of the EES and the transport security of the interface are realized by using TLS with server authentication using the server's certificate issued by CAs in the PKI. For the authentication of the EEC by the EES, the EEC first gets a token from the ECS for this purpose and sends the token to the EES. It is assumed that there is a business relationship between the ECSPs of the ECS and EES and the EES can verify the token.

This token based solution is based on the federated identity concept. In this concept, the client authenticates itself towards to the identity provider and the authentication result is asserted to the relying party. NIST defined guidelines on digital identity and federation and assertions in [33] and [34]. One of the assertion technologies that can be used to realize federated identity is Open ID Connect [32] tokens as stated in [34]. Also, this token based solution for authentication is not new in 3GPP and already used in some architectures such as CAPIF, Mission Critical Security and SEAL architectures. Open ID Connect usage in SEAL architecture [23] can be taken as an example implementation of this technology.

#### 6.17.2.2 Authentication of the GPSI in EEC-EES/ECS communication

TS 23.558 [2] specifies different interactions between EEC and EES/ECS that use the UE ID for identifying the UE. The UE ID is specified in clause 7.2.6 of TS 23.558 [2]. The only example for the UE ID is the GPSI.

The GPSI also requires authentication. This solution proposes to use AKMA for the generation of a shared key KECUEID = KAF between the UE and the EES/ECS, i.e. AKMA AF. The EEC and EES/ECS can then use the KECUEID for authentication of the GPSI.

In order to use the shared KECUEID for authentication of the GPSI towards the EES/ECS, a modern but simple existing challenge-response protocol seems most appropriate. If HTTP is used as application protocol, HTTP Digest as specified in RFC 7616 [24] would be a good candidate.

The identifier used for the KAF is the A-KID in AKMA where A-KID is a temporary identifier. To verify the GPSI the following steps are executed. It is a pre-condition that the AAnF needs to be preconfigured with whether an AF can retrieve a specific GPSI.

1. The EEC sends GPSI in addition to the A-KID to EES/ECS if the GPSI is available to the EEC. EES/ECS verifies the GPSI received from EEC with the one locally configured (if available).

2. The EES/ECS send the A-KID and an indicator requesting GPSI, the application information associated with the expected GPSI and the AF\_ID to the AAnF via NEF or directly depending on the location of EEC/ECS.

The EES/ECS can retrieve the application information from the EEC or from a local configuration and policy.

3. The AAnF fetches the GPSI associated with the application information from the UDM based on the SUPI which is part of AKMA context in the AAnF.

The UDM can have UE's subscription data provisioned with multiple GPSIs, each of which could be associated with specific application information.

4. Depending on the location of the EES/ECS (inside the operator's network or outside) the AAnF or NEF checks whether the EES/ECS is authorized to get the GPSI and whether the EES/ECS is authorized to get the GPSI associated with the provided application information, based on the configured local policy. If the check is successful, the AAnF/NEF provides the KAF and the GPSI to the EES/ECS. Otherwise sends a related failure message.

NOTE 1: The format of application information needs to be defined by 3GPP and aligned with the application function. The definition of it is left to normative work.

NOTE 2: Analysis can be done in user consent study. Also, the same principle with the IP address to GPSI translation API, where GPSI is sent to the servers, can be applied here.

5. The EES/ECS checks whether the GPSI sent by the EEC and the GPSI received from the AAnF /NEF are same or not. If the check is successful, the KAF (KECUEID) is used for authentication as mentioned above.

Another solution option to authenticate the GPSI is the usage of IP address to GPSI translation API (UE identifier API).

1. The EEC sends GPSI to the EES/ECS if the GPSI is available to the EEC.

2. The EES/ECS invokes the UE identifier API inputting the UE IP address and the application information associated with the expected GPSI. Note that, the EES/ECS can retrieve the application information from the EEC or from local configuration and policy.

3. The NEF authorizes that the EES/ECS is entitled to use the UE identifier API and the application information to get the application specific GPSI, based on local policy. If the GPSI is not available in the NEF, then NEF first locates the SUPI from the UE IP address and then fetches the GPSI associated with the application information from the UDM.

NOTE 3: How the NEF locates the UE ID from the UE IP address is out of scope of the present document..

Note that, the UDM can have UE's subscription data provisioned with multiple GPSIs, each of which could be associated with specific application information.

4. The NEF sends the application specific GPSI to the EES/ECS after successful authorization.

5. The EES/ECS checks whether the GPSI sent by the EEC and the GPSI received from the NEF are same or not.

### 6.17.3 Solution evaluation

This solution addresses the following key issues:

- KI#1 "Authentication and Authorization between EEC and EES",

- KI#2 "Authentication and Authorization between EEC and ECS",

- KI#3 "Authentication and Authorization between EES and ECS", and

- KI#6 "Transport security for the EDGE-1-9 interfaces".

For KI#6, it is proposed to use TLS which is one of the commonly chosen and also preferred technology in existing mechanisms such as SEAL.

For authentication of servers, it is proposed to use TLS certificates of the servers (EES and ECS).

For authentication of EEC, two alternative options are proposed: 1) usage of EEC TLS certificate 2) usage of tokens provisioned by the provider of the EEC or another trusted entity for the onboarding of the EEC, provided by the ECS for the access to the EES. These are common methods used in mobile application worlds. Also token based solution for onboarding is an existing mechanism used in CAPIF framework for onboarding the API invokers. When token based solution is compared with TLS certificate based solution, it seems that token based solution is the most appropriate one because for mobile applications it is easier to handle tokens than to handle certificates.

For authentication of GPSI of the UE where the EEC runs, two alternative solutions are proposed: 1) Usage of AKMA and HTTP digest protocol 2) IP address to GPSI translation (UE identifier API). Since IP address to GPSI translation API seems enough for authentication of GPSI, option #2 can be preferred to have a simple solution. IP address to GPSI translation can work only when the access is over 3GPP. If the EEC sends the GPSI to the EES/ECS, the procedures to configure the GPSI(s) in the UE as well as their respective services needs to be specified. If the EEC sends the GPSI to the EES/ECS, then the EEC needs to be aware of which GPSI to provide to which Ua\* protocol in the case of multiple GPSIs are available to the EEC.

Overall, solution #17 proposes the usage of common existing mechanisms to address the key issues.

The AKMA based options in the solution require some updates to the AKMA technical specification TS 33.535, including a pre-configuration in AAnF with whether an AF can retrieve a specific GPSI, for the retrieval of the application specific GPSI and authorization of the AFs before giving the application specific GPSI to the AFs.

Analysis about user consent can be done in user consent study. It should be noted that this issue is not specific to this solution and is mainly related to the requirements, so it is valid for all solutions proposed for the requirements. Also, the same principle with the IP address to GPSI translation API can be considered here.

## 6.18 Solution #18: Authentication and Authorization Framework for EDGE-4 interfaces using Primary authentication and proxy interface

### 6.18.1 Introduction

The solution addresses the following key issue:

- Key issue #2: Authentication and Authorization between EEC and ECS

This solution enables authentication and authorization (Proxy AA) with an ECS during registration after primary authentication successful completion. The solution is based on the KAMF generated during the primary authentication.

### 6.18.2 Solution details



Figure 6.18.2-1: Authentication and Authorization with the Edge Data Network

1. The UE performs normal primary authentication and registration to the network. The UE is MEC capable and may indicate this in the MEC capabilities to the AMF during the registration procedure.

2. The UE establishes a PDU Session for IP connectivity (Additional information IE in UL NAS transport message with request type PDU Session Establishment request includes EEC ID). If the UE is MEC capable, then the UE and the AMF derive a key KProxy for authentication with the ECS from the AMF key KAMF. AMF pushes the EEC ID and KProxy to the Proxy AA network function in one of the options. Proxy AA network function maintains a mapping of EEC ID and KProxy.

Note: In case of Option 2, Proxy AA can identify the serving AMF by identifying "Allocated AMF for the registered UE" field in the "UE context in AMF data" as per TS 23.502 [5] 5.2.3.3.1.

Note 1: It is not addressed in the present document how proxy AA gets UE context.

Note 2: Whether the Kamf can be used to derive the Kecs in case ECS is deployed by the home network is not addressed in the present document.

Note 3: It is not addressed in the present document how this solution works if the EEC ID is not unique across different UEs.

3. The UE sends an Application Registration Request with a MAC-IProxy to the ECS. The MAC-IProxy is computed similarly as, e.g. the SoR-MAC-IAUSF, as defined in Annex A.17 of TS 33.501. The MAC-IProxy is based on the Application Registration Request's payload, which forms the input Application Registration Request Data, and the key KProxy to the KDF.

4. a. The UE is not authenticated at the ECS, and the ECS sends a Verify Request including the Application Registration Request with the MAC-IProxy to the Proxy AA through NEF, which then either verifies by retrieving context its own stored mapping (step 2 option 1) or it sends a key request to AMF by selecting serving AMF based on UE ID the serving AMF and forwards the message to this AMF.

4. b. The AMF replies with KProxy to Proxy AA, which then stores this in its database. Proxy AA verifies the MAC-IProxy of the Application Registration Request, i.e. it computes with the key KProxy the MAC-I over the Application Registration Request payload the UE and compares the result with the MAC-IProxy included in the message. If both are identical, the message can be authenticated to be sent by the UE.

4. c. Proxy AA Devices KECS from KProxy.

4.d. The Proxy AA sends a Key Response to the ECS, including the result of the authentication and the KECS.

5. Based on the authentication result, the ECS decides whether to accept or to reject the Application Registration Request from the UE. The ECS sends the Application Registration Response message to the UE, including the authentication result, and protects the message with a MAC-IECS based on the received key KECS in a similar way as the UE protected the payload of the message.

6. The UE derives KECS from KProxy and verifies the MAC-IECS. The rest of the procedure will proceed from step 10 of solution 6.7 in 33.839.

### 6.18.3 Solution Evaluation

This solution addresses the security requirement of key issue #2: Authentication and Authorization between EEC and ECS.

## 6.19 Solution #19: Authentication/authorization between UE and Edge Data Network based on the secondary authentication

### 6.19.1 Introduction

This solution addresses the security requirement for authentication/authorization between UE and Edge Data Network in the key issue #4.

### 6.19.2 Solution details

The Edge Data Network can be regarded as a particular Data Network in the edge computing scenario. Therefore, secondary authentication defined in the TS 33.501 [7] clause 11 can be reused here for authentication/authorization between UE and Edge Data Network. A high level of the procedure is given in the following figure.



Figure 6.19.2-1: Initial EAP Authentication with an external AAA server

Re-authentication defined in the TS 33.501 [7] clause 11.1.3 also applies here.

### 6.19.3 Solution Evaluation

The proposed solution meets all the requirement of Key issue #4.

The secondary authentication is reused for authentication and authorization between UE and Edge Data Network. Hence, there is no impact of the existing security procedures.

## 6.20 Solution #20: Authentication and authorization in EES capability exposure based on CAPIF

### 6.20.1 Introduction

This solution addresses the security requirement for authentication and authorization in EES capability exposure in the key issue #8.

### 6.20.2 Solution details

As defined in the TS 23.558 [2] clause 8.7.3, it says that

"*The Edge Enabler Server may re-expose the network capabilities of the 3GPP core network to the Edge Application Server(s) as per the CAPIF architecture specified in 3GPP TS 23.222 [6].*

*Depending on the deployment models (centralized or distributed) employed,*

*- the Edge Enabler Server assumes the role of the API exposing function (may also acts as the API topology hiding entry) as described in 3GPP TS 23.222 [6]; and*

*- the Edge Application Server assumes the role of an API invoker.*"

Therefore, CAPIF functional security model defined in the TS 33.122 [16] can be reused here for the authentication and authorization.



Figure 6.20.2-1: CAPIF functional security model

For the edge computing scenario, the EES can be regarded the API exposing function in API provider domain1 or API provider domain 2, which depends on the EES deployment. And EAS can be regarded as the API invoker.

Therefore, authentication and authorization mechanism defined in the TS 33.122 clause 6.5 could be reused here to meet the authentication and authorization requirement between EES and EAS. In general, the authentication could be realized based on the TLS-PSK, TLS-PKI, or TLS with OAuth token.

For the TLS-PSK, and TLS-PKI, the CAPIF core function as an NF managed by the operator could forward the pre-shared key or the security information related with TLS-PKI to the EAS. Then EAS could authenticate with the EES. Then, the following authorization procedure is referring to the TS 23.222 [9] clause 8.17.

For the TLS with OAuth token, the authentication could be based on the TLS, and the authorization could be based on the Oauth mechanism.

### 6.20.3 Solution Evaluation

This solution fully addresses the security requirement for authentication and authorization in EES capability exposure in the key issue #8. This solution reuses the existing authentication and authorization in EES capability exposure based on CAPIF.

## 6.21 Solution #21: security for the interface between the SMF and LDNSR

### 6.21.1 Solution overview

This solution addresses the transport security requirements for the interaction message between the SMF and LDNSR in key issue #9. In SA2, it was agreed that the interface between the SMF and LDNSR is based on SBI.

### 6.21.2 Solution details

In clause 13.1 of TS 33.501[7], it defines how to protect the interaction message transferred in SBI, and it is proposed to reuse the mechanism on the interface between the SMF and LDNSR.

TLS should be used for the interaction message between the SMF and LDNSR as specified in clause 6.2 of TS 33.210[12], unless security is provided by other means, e.g. physical security.

NOTE 1: Regardless of whether TLS is used or not, NDS/IP as specified in TS 33.210 [12] and TS 33.310 [13] can be used for network layer protection.

NOTE 2: If interfaces are trusted (e.g. physically protected), it is for the PLMN-operator to decide whether to use cryptographic protection.

### 6.21.3 Solution evaluation

This solution fully addresses the transport security requirements for the interaction message between the SMF and LDNSR in key issue #9. This solution reuses the security mechanism on the SBI interface between the SMF and LDNSR. Hence, no new mechanisms is required.

## 6.22 Solution #22: Authorization during Edge Data Network change

### 6.22.1 Introduction

This solution addresses the security requirement for authentication/authorization during Edge Data Network change in the key issue #10.

For the Edge Data Network (EDN) change scenario, the edge service consumed by the UE will be relocated from the source EDN to the target EDN. If the secondary authentication for authorization is performed between UE and the source EDN, the issue is whether the secondary authentication between UE and target EDN is required or not. Considering the authorization requirement defined in key issue #10 and seamless change required in the key issue #2 of TR 23.748 [3], this solution gives out a simple authorization method between UE and target EDN, maintaining the seamless change requirement.

### 6.22.2 Solution details



Figure 6.22.2-1: Authorization during the EDN relocation

Pre-requisite:

- The S-EDN stores the signing key used for token computation. And the T-EDN stores the verification key used for token verification. The two keys could be pre-configured or shared by the same service application or DN-AAA.

- The T-EDN trusts that the UE has already been authorized by the S-EDN, if the token including the UE ID, the IDs of S-EDN and T-EDN provided by the S-EDN is successfully verified.

Step 1. UE sends the Registration request to the AMF and registers in the network.

Step 2. UE initiates the PDU session1 establishment procedure. It is assumed that secondary authentication is performed during the PDU session establishment procedure.

Step 3. SMF detects that EDN relocation is required, and determines the T-EDN info.

NOTE 1: Void

NOTE 2: EDN relocation detection and T-EDN info determination will be decided in the TR 23.748 [3], and are out of scope of the present document.

Step 4. SMF selects the T-UPF.

Step 5. SMF performs the N4 session configuration with the T-UPF.

Step 6. SMF sends the Authorization request to the S-EDN via S-UPF, including the GPSI, T-EDN info.

Step 7. S-EDN assures that the UE identified by the GSPI is already successfully authorized, then generates an authorization token, based on the signing key. Here, the token includes the ID of the S-EDN, the ID of the T-EDN, and the UE ID, where the UE ID could be the GPSI of the UE.

Step 8. S-EDN sends the Authorization response message to the SMF via S-UPF, including the token.

Step 9. (Optional) According to the TR 23.748 [3] clause 4.2, SSC #2 and SSC #3 mode could be used to optimize traffic routing here for distributed anchor point and multiple PDU session cases.

In case of SSC #2 mode, the SMF will release the existing PDU session, and trigger the UE to establish the PDU session2 for the T-EDN according to TS 23.502 [5] clause 4.3.5.1, by sending the PDU Session Release Command message to the UE which contains the PDU Session ID and Cause indicating that a PDU Session re-establishment to the same DN is required. In case of SSC #3 mode, according to TS 23.502 [5] clause 4.3.5.2, the SMF will trigger the UE to establish a new PDU session by sending the N1 SM container (PDU Session Modification Command (Cause, PCO (PDU Session Address Lifetime value)))) where the Cause indicates that a PDU Session re-establishment to the same DN is required.

Step 10. (Optional) If a new PDU session is required for the T-EDN, then UE initiates the PDU session2 establishment procedure for the T-EDN. Otherwise, steps 9, 13 and 14 are skipped.

NOTE 3: The detailed procedure on how the PDU session2 establishment is triggered refers to the TR 23.748 [3].

Step 11. SMF send the Authorization request to the T-EDN via T-UPF, including the token.

Option A: For the Distributed anchor point and Multiple PDU session scenario, step 9 and step 10 will be performed. The SMF could send the Authorization request to the T-EDN, if the SMF in the new PDU session is not changed. The SMF could use the SUPI, DNN and S-NSSAI received from AMF to check whether the UE is the same, or not.

Option B: For the Session Breakout case, Step 9 and Step 10 are optional. The SMF could send the authorization request to the T-EDN.

Step 12. T-EDN verifies the authorization token based on the verification key.

NOTE 4: The token generation and verification could refer to the token mechanism defined in TS 33.501 [7] for SBA.

Step 13. If the authorization verification successes, T-EDN sends the Authorization response message to the SMF via T-UPF, including the success indication.

Step 14. After receiving the success indication, the SMF proceeds with the following PDU session establishment procedure.

Step 15. SMF sends the PDU session2 establishment response to the UE.

### 6.22.3 Solution Evaluation

This solution fully addresses the security requirement for authentication/authorization during Edge Data Network change in the key issue #10. This solution reuses the Oauth token based mechanism. In this solution, the S-EDN plays the role of the authorization server, which generates the Oauth token for the SMF if the original security authentication is successfully performed with the SMF. And SMF will forward the token to the T-EDN.

The solution defines that the S-EDN should compute the authorization token, and send it to the T-EDN. Then T-EDN should have the key to verify the received token. The implemented token in CT4 including the claims, computation, and verification could be reused here, to avoid any further discussion on the implementation of Oauth.

For the Distributed Anchor Point and Multiple PDU sessions scenario, this solution applies if the SMF is not changed during the re-anchoring (SSC#2 and SSC#3).

For the Session breakout scenario, this solution applies.

The impact of the solution is that it requires the S-EDN and T-EDN to support the Oauth mechanism, and SMF to redirect the authorization information to T-EDN.

## 6.23 Solution #23: Authentication and Authorization between EEC and ECS/EES

### 6.23.1 Solution overview

This solution uses AKMA to addresses the KI#1 "Authentication and Authorization between EEC and EES" and KI#2 "Authentication and Authorization between EEC and ECS". Besides, a mechanism is proposed for authentication of the GPSI used in communication between EEC and EES/ECS

### 6.23.2 Solution details

Authentication between EEC and ECS/EES is achieved via AKMA. As GPSI is used as a user identifier for interactions between EEC and EES/ECS, the GPSI also requires authentication. To verify the GPSI the following steps are executed:

0. The EEC and ECS/EES complete the AKMA procedures and derive the shared A-KID and KAF.

1. The EEC sends GPSI to ECS/EES in provisioning request.

2. To verify the received GPSI, the EES/ECS sends the A-KID and the GPSI to the AAnF via NEF or directly depending on the location of EEC/ECS.

3. The AAnF fetches the GPSI from the UDM based on the SUPI which is a part of AKMA context identified by the A-KID in the AAnF.

4. The AAnF checks whether the GPSI sent by the EEC and the GPSI received from the UDM are same or not. If the check is successful, the AAnF provides validated result to the EES/ECS. Otherwise AAnF sends a failure message.

### 6.23.3 Solution evaluation

As the verification of GPSI is performed after the AKMA procedure, the solution does not impact the AKMA. The 5GC does not send the GPSI to ECS/EES. The only impact is enhancing the AAnF with GPSI validation service, which requires two more communication rounds that adding the GPSI to existing AKMA messages.

The UE needs to be aware of which GPSI to provide to which Ua\* protocol in the case of multiple GPSIs are available.

The AF has to be authorised by the AAnF to receive GPSI or validate GPSI.

In the rear case that there is a primary authentication between the AKMA and GPSI validation protocols, the AF could require a new A-TID from the UE for the GPSI validation.

The requirements on the UE are:

- Only the authorized application in the UE can get the GPSI.

NOTE 1: How this requirement is satisfied is out of scope of 3GPP.

NOTE 2: Analysis about user consent can be done in an user consent study.

Summing up, this solution meets the requirements of KI#1 and KI#2.

## 6.24 Solution #24: Using TLS with AKMA to protect edge interfaces

### 6.24.1 Solution overview

This solution addresses the EDGE-1 and EDGE-4 parts of key issue #6.

### 6.24.2 Solution details

#### 6.24.2.1 General

The solution proposes to use TLS with AKMA generated keys to protect EDGE-1 and EDGE-4. Profile of the Ua protocols for GBA (TS 33.220 [8]) are provided in the following clauses as examples to describe that protection.

NOTE 1: Standardisation of profile for TLS with AKMA is under discussion. It is not proposed to update the below profile in the present document if there is a specific work item for that work.

The solution proposed differs from solution #16 in not requiring an exchange of messages between EEC and ECS/EES to agree on the shared AKMA generated key before creating the TLS session.

NOTE 2: EEC authentication is not in the scope of this solution.

#### 6.24.2.2 Shared key-based UE authentication with certificate-based AF authentication

##### 6.24.2.2.1 General

The clause provides details needed to define solution similar to the Ua protocol given in 3GPP TS 33.222 [26] rather than providing changes to Ua protocol given in TS 33.222 [26].

##### 6.24.1.2.2 Procedures

The procedures are similar to those given in clause 5.3.0 of TS 33.222 [26] with the AKMA AF taking the role of the NAF from GBA (see TS 33.220 [8]), with the following changes.

At step 2, the client adds the constant string "3gpp-akma" to the "User-Agent" HTTP header as product tokens as specified in IETF RFC 2616 [27].

At step 3, if the AF selects AKMA for deriving the key, then the AF includes the "3GPP-bootstrapping-akma" within the WWW-Authenticate header field.

At step 5 given AKMA has been selected for keying, the client sends a response with an Authorization header field where Digest is inserted using the A-KID as username. KAF is used as password in the Digest calculation.

At step 6 given AKMA has been selected for keying, the AF verifies the value of the password attribute using KAF retrieved from AAnF using the A-KID received as username attribute in the query. If the AF is not able to obtain the AF-specific key when using AKMA mode, the AF responds with an appropriate error message not containing the realm attributes from step 3.

#### 6.24.2.3 Shared key-based mutual authentication between UE and AF

##### 6.24.2.3.1 General

The clause provides details needed to define solution similar to the Ua protocol given in 3GPP TS 33.222 [26] rather than providing changes to Ua protocol given in TS 33.222 [26].

##### 6.24.2.3.2 Procedures

The procedures are similar to those given in clause 5.4.0 of TS 33.222 [26] with the AKMA AF taking the role of the NAF from GBA (see TS 33.220 [8]), with the following changes.

At step 2, the AF includes a constant string "3GPP-AKMA" as a PSK-identity hint to indicate that AKMA based keying is supported.

At step 3, the UE may use an AKMA generated key if support was indicated by the AF. To use as AKMA generated key, the UE derives the TLS premaster secret from KAF and sends a ClientKeyExchange message including a PSK identity consisting of "3GPP-AKMA" and the A-KID.

At step 4, if the AF receives the "3GPP-AKMA" prefix and the A-KID in the ClientKeyExchange messages it fetches the AF specific shared secret (KAF) from the AAnF using the A-KID. The AF derives the TLS premaster secret from the AF specific key (KAF).

### 6.24.3 Solution evaluation

This is a straightforward method for providing the security for these interfaces which will be usable for many other applications. The solution requires the UE and ECS/ESS to support TLS using AKMA. Editor's Note: Impact of multiple ECC in UE is FFS.

## 6.25 Solution #25: Practical authorization during Edge Data Network change

### 6.25.1 Introduction

This solution addresses the security requirement for authentication/authorization during Edge Data Network change in the key issue #10.

For the Edge Data Network (EDN) change scenario, the edge service consumed by the UE will be relocated from the source EDN to the target EDN. If the secondary authentication for authorization is performed between UE and the source EDN, the issue is whether the secondary authentication between UE and target EDN is required or not. Considering the authorization requirement defined in key issue #10 and seamless change required in the key issue #2 of TR 23.748 [3], this solution gives out a practical authorization method between UE and target EDN based on the assumed trust mode between 5GC (SMF) and AF, maintaining the seamless change requirement.

### 6.25.2 Solution details



Figure 6.25.2-1: Authorization during the EDN relocation

Pre-requisite:

- Trust mode between SMF and T-EDN is assumed in this solution. The T-EDN trusts the authorization information provided by the operator, i.e. the success indication.

Step 1. UE sends the Registration request to the AMF and registers in the network.

Step 2. UE initiates the PDU session1 establishment procedure. It is assumed that secondary authentication is performed during the PDU session establishment procedure. SMF store the UE app ID using during the secondary authentication.

Step 3. SMF detects that EDN relocation is required, and determines the T-EDN info.

NOTE 1: EDN relocation detection and T-EDN info determination will be decided in the TR 23.748 [3], and are out of scope of the present document.

Step 4. (Optional) In case of SSC #2 mode, the SMF will release the existing PDU session, and trigger the UE to establish the PDU session2 for the T-EDN according to TS 23.502 [5] clause 4.3.5.1, by sending the PDU Session Release Command message to the UE which contains the PDU Session ID and Cause indicating that a PDU Session re-establishment to the same DN is required. In case of SSC #3 mode, according to TS 23.502 [5] clause 4.3.5.2, the SMF will trigger the UE to establish a new PDU session by sending the N1 SM container (PDU Session Modification Command (Cause, PCO (PDU Session Address Lifetime value)))) where the Cause indicates that a PDU Session re-establishment to the same DN is required.

Step 5. (Optional) If a new PDU session is required for the T-EDN, then UE initiates the PDU session2 establishment procedure for the T-EDN. Otherwise, steps 5 are skipped.

Step 6. SMF selects the T-UPF and sends the Authorization request to the T-EDN via T-UPF, including the UE app ID, GPSI, success indication, where success indication is to indicate the UE has already been successfully authenticated during the secondary authentication with S-EDN.

Option A: For the Distributed anchor point and Multiple PDU session scenario, step 4 and step 5 will be performed. The SMF could send the Authorization request to the T-EDN, if the SMF in the new PDU session is not changed. The SMF could use the SUPI, DNN and S-NSSAI received from AMF to check whether the UE is the same, or not.

Option B: For the Session Breakout case, Step 9 and Step 10 are optional. The SMF could send the authorization request to the T-EDN.

Step 7. T-EDN authorizes the UE according to UE app ID, GPSI, success indication, and assures that UE has already been successfully authenticated during the secondary authentication with S-EDN. Then the secondary authentication is not required.

Step 8. If the authorization verification successes, T-EDN sends the Authorization response message to the SMF via T-UPF.

Step 9. After receiving the success indication, the SMF proceeds with the following PDU session establishment procedure.

Step 10. SMF sends the PDU session2 establishment response to the UE.

### 6.25.3 Solution Evaluation

This solution assumes that a trust mode is established between the 5G network and EDN.

For the Distributed Anchor Point and Multiple PDU sessions scenario, this solution applies if the SMF is not changed during the re-anchoring (SSC#2 and SSC#3).

For the Session breakout scenario, this solution applies.

The impact of the solution is that it requires the SMF to forward the authorization information to T-EDN.

Note: Further evaluation is needed.

## 6.26 Solution #26: GBA-based solution for EEC authentication and authorization framework with ECS and EES

### 6.26.1 Introduction

This solution addresses the security requirement for authentication/authorization between EEC and ECS/EES in the key issue #1, and key issue #2. In this solution, GBA as specified in 3GPP TS 33.220 [8], and TLS is reused as the building blocks for the EEC authentication framework if the EDGE-4 is deployed based on the UP connection. NAS and SBI interface protection is reused if the EDGE-4 is deployed based on the CP connection. For the authorization, the Oauth is selected for the authorization between EEC and EES.

### 6.26.2 Solution details

UE (EEC)

AMF

AUSF

BSF

ECS

EES

1. Primary authentication

KECS

KECS

5a. TLS based on KECS

5. SBI interface protection

3. NAS security established

5b. NAS protection

6. Provisioning request

7. Token generation

8. Provisioning response (Token)

9. GBA procedure between UE and EES

KEES

KEES

10. TLS based on KEES

11. EEC registration/discovery request (Token)

12. Authorization based on the token

13. EEC registration/discovery response

1. Registration req.

4. GBA procedure between UE and ECS

Option A

Option B

Figure 6.26.2-1 GBA-based EEC authentication and authorization framework with ECS and EES

It is assumed that GBA, TLS, and OAuth is supported by the UE, ECS, and EES.

Step 1-3. UE with EEC functionality registers in the 5G network.

There are two options for the authentication and data protection between EEC and ECS.

Option A (The EDGE-4 is deployed based on the UP connection):

Step 4. The UE determines to use GBA based on the received GBA capability, and GBA as defined in the TS 33.220 [2] is used here to negotiate the pre-shared key KECS between UE and ECS, where ECS plays the role of a NAF.

Step 5a. EEC and ECS establish the TLS security tunnel based on the pre-shared key KECS. The authentication is fulfilled based on the TLS.

Option B (The EDGE-4 is deployed based on the CP connection):

Step 5b. The protection between EEC and ECS relies on the NAS and SBI interface protection. The authentication is implicitly performed as the primary authentication.

NOTE 0: CP connection option between EEC and ECS depends on the conclusion of CT1 and SA6.

NOTE 1: the following EDGE-4 data could be protected based on Option A or Option B according to its connection option.

Step 6. EEC sends the Provisioning request message to the ECS, including its EEC ID, application info.

Step 7. ECS determines the EES info (including EES Id, GBA capability), and generate the OAuth token with the following claims, i.e. EEC ID, ECS ID, EEC info, the authorized services.

Editor's Note: It is ffs which identifier for the UE/EEC is included in the token.

NOTE 2: Whether the EEC ID could use the edge service can be authorized based on the local policy. The token for consuming the ECS service may not be needed.

Step 8. ECS sends the token and EES info back to the EEC via the Provisioning response message.

Step 9. Similar with step 4, pre-shared key KEES is negotiated between EEC and EES, where EES plays the role of a NAF.

Step 10. EEC and EES establish the TLS security tunnel based on the pre-shared key KEES. The authentication is fulfilled based on the TLS.

Step 11. EES sends the EEC registration/ discovery request message to the EES, including the token.

Step 12. EES authorizes the EEC based on the verification of the token.

Step 13. If the verification successes, the EES sends the EEC registration/ discovery response message back to the EEC.

### 6.26.3 Solution Evaluation

This solution fully addresses the security requirement for authentication between EEC and ECS in the key issue #2 based on GBA and TLS, and fully address the security requirement for authentication and authorization between EEC and EES in key issue #1 based on GBA, TLS and Oauth.

User identity privacy is addressed during GBA usage. GBA, as specified in 3GPP TS 33.220 [12], defines mechanisms to protect user identity privacy. Additionally, GBA may be run over a secure transport (e.g. Uu).

This solution requires that the EEC, ECS, and EES support GBA, TLS and OAuth mechanism. Similar with the SBA authorization, the ECS plays the authorization server role, EEC plays the EES service consumer, and the EES plays the service producer role within the OAuth architecture.

How to authenticate the EEC ID is not captured in this solution.

The key isolation issue between multiple EECs connected to the same ECS is not addressed in this solution.

## 6.27 Solution #27: Using TLS with Edge Security Service to protect edge interfaces

### 6.27.1 Solution overview

This solution addresses Key Issue #1, Key Issue #2, and the EDGE-1, EDGE-4 parts of Key Issue #6.

This solution relies on framework defined by GSMA to protect communications between UE and servers with the establishment of TLS sessions leveraging the presence of the USIM to provision server certificates in the User Equipment, confer clause 3 of GSMA IoT.04 [28].

To protect EDGE-1 and EDGE-4, the Edge Security Service plays the role of the IoT Security Service described in clause 3 of GSMA IoT.04 [28], the Edge Server Middleware plays the role of IoT Server Middleware, the SIM with Edge Security Applet plays the role of SIM with IoT Security Applet, and the Edge Device Middleware plays the role of IoT Device Middleware.

### 6.27.2 Solution details



Figure 6. 27.2-1: Using TLS with Edge Security Service to protect edge interfaces

**Pre-requisites:**

UE contains AC, EEC, and SIM with Edge Security Applet.

- The SIM with Edge Security Applet plays the role of SIM with IoT Security Applet defined in clause 3 of GSMA IoT.04 [28].

- To have SIM with Edge Security Applet, there is no need to change UICC on the field if this UICC supports asymmetric crypto and OTA features.

ECS contains Edge Server Middleware.

- The Edge Server Middleware plays the role of IoT Server Middleware described in clause 3 of GSMA IoT.04 [28].

EES contains Edge Server Middleware.

- The Edge Server Middleware plays the role of IoT Server Middleware described in clause 3 of GSMA IoT.04 [28].

- API between EEC and SIM with Edge Security Applet is defined in clause 3 of GSMA IoT.04 [28].

- API between Edge Security Service and Edge Server Middleware is defined in clause 3 of GSMA IoT.04 [28].

- APIs between Edge Security Service and SIM with Edge Security Applet corresponds to existing API defined for Over-The-Air (OTA) SIM management.

- EES profile contains EES certificate.

**Procedure:**

Step 1: The EEC sends Initial Provisioning request to the ECS in order to trigger client certificate provisioning. The requested client certificate is to be used to perform mutual authentication between the EEC and the ECS.

Step 2: The ECS sends to the Edge Security Service ECS certificate and Security Profile corresponding to the ECS. The Security Profile associated to the SIM contains all necessary data to establish TLS tunnel between the ECS and EEC thanks to the SIM with Edge Security Applet.

Step 3: The Edge Security Service checks whether ECS-related data have already been downloaded to SIM with Edge Security Applet, checks the client certificate validity, and determines whether ECS-related data have to be downloaded to the SIM with Edge Security Applet.

Step 4: The Edge Security Service sends response to the ECS indicating whether ECS-related data have to be downloaded to the SIM

Step 5: If the Edge Security Service response in step 4 indicates that no ECS-related data have to be downloaded to the SIM, then the ECS continues the procedure with step 9. Otherwise, the ECS continues the procedure with step 6.

Step 6: The ECS requests to the 3GPP core via Edge 8 to trigger the SIM with Edge Security Applet.

Step 7: The SIM with Edge Security Applet sends "Open channel" command.

Step 8: The Edge Security Service downloads to the SIM with Edge Security Applet client certificate enabling to establish TLS tunnel between EEC and the ECS, and optionally ECS certificate or root certificate.

Step 9: Establishment of TLS tunnel between the EEC and the ECS thanks to the SIM with Edge Security. There is mutual authentication between EEC and ECS.

Step 10: The EEC sends Service Provisioning Request to the ECS.

Step 11: The ECS sends Service Provisioning Response. Among the response data there are list of EEEs and Access Token.

Step 12: The EEC sends to the ECS a Client Certificate Provisioning request for selected EES. The requested client certificate is to be used to perform mutual authentication between the EEC and the selected EES.

Step 13: The ECS sends to the Edge Security Service a Security Profile corresponding to the EES, and EES certificate. The Security Profile associated to the SIM contains all necessary data to establish TLS tunnel between the EES and EEC thanks to the SIM with Edge Security Applet.

Step 14: The Edge Security Service checks whether EES-related data have already been downloaded to the SIM with Edge Security Applet, checks the client certificate validity, determines whether EES-related data have to be downloaded to the SIM with Edge Security Applet.

Step 15: the Edge Security Service sends response indicating whether EES-related data have to be downloaded to the SIM

Step16: If the Edge Security Service response in step 15 indicates that no EES-related data have to be downloaded to the SIM, then the ECS continues the procedure with step 19. Otherwise, the ECS continues the procedure with step 17.

Step 17: The ECS request to the 3GPP core via Edge 8 to trigger the SIM with Edge Security Applet.

Step 18: The SIM with Edge Security Applet sends "Open channel" command.

Step 19: The Edge Security Service downloads to the SIM with Edge Security Applet client certificate enabling to establish TLS tunnel between EEC and the EES, and optionally EES certificate or root certificate.

Step 20: Establishment of TLS tunnel between the EEC and EES thanks to the SIM with Edge Security. There is mutual authentication between EEC and EES.

Step 21: The EEC sends EEC Registration request to the EES. The request contains Access Token.

Step 22: Validation of Access Token by the ECS.

Step 23: The EES sends response to the EEC Registration Request.

### 6.27.3 Solution evaluation

The solution addresses Key Issue #1, Key Issue #2, and the EDGE-1, EDGE-4 parts of Key Issue #6.

The solution introduces an additional server (Edge Security Service) compared to architecture defined by SA6. The Edge Security Service is connected to the ECS via a new interface.

The solution impacts the ME.

The solution enables the provisioning of ECS and EES certificates in the User Equipment.

To have SIM with Edge Security Applet, there is no need to change UICC on the field if this UICC supports asymmetric crypto and OTA features.

## 6.28 Solution #28: Authentication between EEC and ECS based on AKMA

### 6.28.1 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.

### 6.28.2 Solution details

#### 6.28.2.1 Procedure

Timeline

Description automatically generated

Figure-6.28.2.1-1. 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.

Note 1: It is not addressed in the present document how EEC ID is authenticated.

Note 2: It is not addressed in the present document whether ECS could perform the authentication instead of AAnF.

#### 6.28.2.2 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.

#### 6.28.2.3 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.

### 6.28.3 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.

## 6.29 Solution #29: Using TLS with GBA to protect edge interfaces

### 6.29.1 Solution overview

This solution addresses the EDGE-1 and EDGE-4 parts of key issue #6. It also addresses security requirement for mutual authentication of Key Issue #1 and Key Issue #2.

### 6.29.2 Solution details

#### 6.29.2.1 General

The solution proposes to use TLS with GBA generated keys to protect EDGE-1 and EDGE-4.

GBA (Generic Bootstrapping Architecture), as specified in 3GPP TS 33.220 [8], is a generic mechanism enabling the establishment of shared keys between the UE and any Application Function (named Network Application Function in GBA description) thanks to 3GPP user authentication (AKA-based authentication).

ECS and EES play the role of NAFs (Network Application Function) in GBA.

#### 6.29.2.2 Shared key-based UE authentication with certificate-based AF authentication

##### 6.29.2.2.1 General

The EEC and ECS/EES can establish a TLS tunnel using GBA shared key as specified in clause 5.3.0 of 3GPP TS 33.222 [26].

#### 6.29.2.3 Shared key-based mutual authentication between UE and AF

##### 6.29.2.3.1 General

The EEC and ECS/EES can establish a TLS tunnel using GBA shared key as specified in clause 5.4.0 of 3GPP TS 33.222 [26].

### 6.29.3 Solution evaluation

This solution addresses the security requirement for mutual authentication between EEC and EES in Key Issue #1, the security requirements for mutual authentication between EEC and ECS in Key Issue #2, and the security requirements of Key Issue #6.

The solution requires that EEC, ECS and EES support GBA and TLS.

The GBA keys to establish TLS tunnel could be either Ks\_(ext/int)\_NAF keys, or keys derived from Ks\_(ext/int)\_NAF.

Note 1: It is not addressed in the present document whether GBA guideline may need to be revisited due to application technology evolution in the UE.

Note 2: Evaluation on the application authentication in the UE is not addressed in the present document.

## 6.30 Solution #30: An AKMA-based solution for authentication and interface protection between EEC and EES/ECS

### 6.30.1 Solution overview

This solution partially addresses key issue #1 and #2 from the authentication perspective and addresses the EDGE-1 and EDGE-4 protection of key issue #6. This solution presents a mechanism that can be placed on top of AKMA feature to prevent key collision by providing different keys to the clients.

### 6.30.2 Solution details

The procedure flow of the solution is depicted in Figure 6.30.2-1 and the steps are explained below.



Figure 6.30.2-1: A method for authentication and interface protection between EEC and EES/ECS

Step 0. UE and 3GPP network run primary authentication and derive KAKMA. UE learns the A-KID and KAKMA.

Step 1. KAF is derived as defined in AKMA procedure. KEEC is derived from KAF and EEC-ID as KEEC = KDF(KAF, EEC-ID). KEEC and E(KAF, EEC-ID) are revealed to the EEC where E(KAF, EEC-ID) is the encryption of EEC-ID under the key KAF.

Step 2. The EEC sends a session establishment request to the EES/ECS, including the parameters A-KID and E(KAF, EEC-ID).

Step 3. The EES/ECS request KAF from the network. In this request, EES/ECS sends A-KID and AF-ID.

Step 4. The network executes the authentication and authorization for the EES/ECS as defined in the AKMA procedure and if the result is successful then derives the KAF such that KAF = KDF (KAKMA, EES-ID/ECS-ID).

Step 5. The network sends KAF and expire time for the key to the EES/ECS.

Step 6. Using the KAF key, the EES/ECS decrypts the encrypted EEC-ID and also derives KEEC such that KEEC = KDF(KAF, EEC-ID).

Step 7. The EEC and the EES/ECS use KEEC in Ua\* protocol, instead of KAF.

### 6.30.3 Solution evaluation

This solution provides a mechanism that can be placed on the top of AKMA feature to handle the case that there are multiple EEC in UE. The solution does not have any impact on AAnF and on the AKMA feature and can be implemented with a new Ua\* protocol for AKMA.

Note: Further evaluation is needed.

## 6.31 Solution #31: Enhancing TLS with GBA for usage with Edge

### 6.31.1 Solution overview

This solution address key issues #1 and #2 on authenticating the mobile subscription for the cases when access to the edge services is granted based on the used mobile subscription.

The enhancement to TLS with GBA is that all EEC are given different keys for use with the ECS/EES that is acting as a GBA NAF (see TS 33.220 [8]). A similar enhancement works for AKMA (see TS 33.535 [6]).

NOTE: It is not addressed in the present document whether this enhancement is needed.

### 6.31.2 Solution details

The solution proposes to use TLS with GBA exactly as described in TS 33.220 [8] and TS 33.222 [26] with the following exceptions:

- The Ks(\_int/ext)\_NAF is not used directly in the TLS or Digest protocol but an additional key, Ks\_NAF\_unique, is derived from Ks(\_int/ext)\_NAF and a random number. This new key, Ks\_NAF\_unique, is used directly in the TLS or Digest protocol exactly as Ks(\_int/ext)\_NAF.

- In the UE, the GBA client (that handles the Bootstrapping, the derivation of Ks(\_int/ext)\_NAF keys and passing the relevant key to the application) does not pass Ks(\_int/ext)\_NAF to the application but instead generates a random number and derive Ks\_NAF\_unique from Ks(\_int/ext)\_NAF and the generated random number. The GBA client then passes Ks\_NAF\_unique and the random number to the application in the UE.

NOTE 1: The generation of Ks\_NAF\_unique can be specified in the normative phase.

- The application in the UE uses Ks\_NAF\_unique as it would the Ks(\_int/ext)\_NAF and passes the random number to the NAF. For Digest this is done in the 'cnonce' information element and for TLS this is done as part of the random number sent in the ClientHello message.

NOTE 2: The random number can form only part of that IE as long as it clearly specified which part.

- The NAF fetches Ks(\_int/ext)\_NAF from the BSF as normal. The NAF uses the random number received to derive Ks\_NAF\_unique and uses this key as it would the Ks(\_int/ext)\_NAF.

A similar enhancement works for AKMA.

### 6.31.3 Solution evaluation

The solution provides a method of ensuring different keys are available for different EEC clients in the UE. It does not provide authentication of the EEC client. Compared to legacy TLS with GBA, there is some extra processing in the UE and NAF.

# 7 Conclusions

## 7.1 Conclusions for Key Issue #1

It is proposed to optionally use TLS with AKMA as specified in TS 33.535 [6] and TLS with GBA as specified in TS 33.222 [9] for authentication between EEC and EES.

NOTE : For authentication between EEC and ECS, additional TLS authentication methods can be used. Details of other TLS authentication methods that uses other than 3GPP subscription credential(s) (e.g. client certificate based TLS authentication) is out of scope of 3GPP.

For the Authorization issue between EEC and EES, it is suggested to use the access token generated by the ECS for the EES service authorization described in the step 2L-2M of solution #3, or the static authorization by the EES for the normative work.

For the GPSI Verification issue, it is proposed to use IP translation to verify the GPSI as described in solution #17.

Client authentication and key diversity will be addressed in the normative work.

Note: How to select between options, and which network/UE side elements need to support which options is not addressed in the present document.

## 7.2 Conclusions for Key Issue #2

It is proposed to optionally use TLS with AKMA as specified in TS 33.535 [6] and TLS with GBA as specified in TS 33.222 [9] for authentication between EEC and ECS.

NOTE 1: For authentication between EEC and ECS, additional TLS authentication methods can be used. Details of other TLS authentication methods that uses other than 3GPP subscription credential(s) (e.g. client certificate based TLS authentication) is out of scope of 3GPP.

For the Authorization issue between EEC and ECS, it is suggested to use the static authorization by the ECS for the normative work.

For the GPSI Verification issue, it is suggested to use IP translation to verify the GPSI as described in solution #17.

Client authentication and key diversity will be addressed in the normative work.

Note 2: How to select between options, and which network/UE side elements need to support which options is not addressed in the present document.

## 7.3 Conclusions for Key Issue #3

Solution #12 is endorsed for the normative phase for mutual authentication, authorization between Edge Configuration Server and the Edge Enabling Server to register and update the server profile information. It is also proposed to use static authorization for the ECS service authorization required by the EES.

NOTE: It is not addressed in the present document whether the security requirement of ECSP defined in the solution #12 is applicable.

## 7.4 Conclusions for Key Issue #4

Solution #19 that proposes to reuse the secondary authentication to solve key issue #4 is endorsed for conclusion. Since secondary authentication is already supported in TS 33.501 [7], no normative works is required.

## 7.5 Conclusions for Key Issue #5

Key issue #5 requires that the edge data network application user identifiers and credentials are protected in storage and in transit. It is concluded that no normative works is required, since protection in storage/in transit of user identifiers and credentials belonging to application layer is out of scope of this study..

## 7.6 Conclusions for Key Issue #6

There are three types of interface in the Key issue #6.

For the type A interface (EDGE-1/4), Solution #13 that was proposed to reuse TLS if HTTP protocol is selected.

Editor's Note: Further conclusion for the type A interface is FFS.

For the type B interface (EDGE-2/7/8), Solution #13 that was proposed to reuse security aspects of Network Exposure Function can be reused if NEF APIs is selected and the security procedures for reference point SCEF-SCS/AS can be reused if SCEF APIs is selected. No new normative work is required.

For the type C interface (EDGE-3/6/9), Solution #13 that was proposed to reuse transport security protection on the Web service API interface can be reused. No new normative work is required.

## 7.7 Conclusions for Key Issue #7

Key issue #7 described that security of Network Information Provisioning to Local Applications with low latency procedure is required for two cases, i.e. UPF exposes the network information to local AF via Local NEF, UPF exposes the network information to local AF directly.

For the first case, it is proposed to reuse the CAPIF /NEF security mechanism for the data exposure from UPF to the local AF via local NEF.

For the second case, the solution #14 proposed a new mechanism based on NEF for the key derivation between UPF and AF. However, considering that the Note added in TS 23.548, i.e. "NOTE 1: Local PSA UPF can expose the QoS monitoring results to local AF via N6. How to deliver the information on N6 is out of SA2 scope", it is proposed to not define any normative work for this issue in this release.

## 7.8 Conclusions for Key Issue #8

Solution #20 that was proposed to reuse the CAPIF functional security model for authentication and authorization in EES capability exposure, is endorsed for normative phase.

For the case that CAPIF is not used, it is suggested to use the mutually authenticated TLS defined in in RFC 5246 [36] and RFC 8446 [37]. The TLS profile follows the profile given in clause 6.2 of TS 33.210 [12], and the authorization is based on local authorization policy at the EES.

NOTE: Adaptations of the security protocol dependent on the lower layer protocols could be necessary, e.g. HTTPS instead of TLS if HTTP is used.

## 7.9 Conclusions for Key Issue #9

Solution #21 that was proposed to reuse the SBI based security for interaction message protection between the SMF and LDNSR, is endorsed for normative phase to include the security requirement of the new interface.

To address the security requirement of the secure discovery of EDGE services, Solution #1 that proposes to reuse security aspects on DNS for 5G defined in TS 33.501 [7] informative annex P.2, is endorsed for normative phase. No new normative work is required.

## 7.10 Conclusions for Key Issue #10

Secondary authentication and authorization, as specified in TS 33.501 [7] clause 11.1 and TS 23.502 [5] clause 4.3.2.3, addresses authorization of UEs when the Edge Data Network is relocated. No new normative work is required.

Annex A:  
Change history

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
| 2020-08 | SA3#100-e | S3-202073 |  |  |  | TR Skeleton | 0.0.0 |
| 2020-08 | Sa3#100-e | S3-202085 |  |  |  | Implemented S3-201903, S3-201832, S3-201750, S3-201669, S3-1668, S3-201833, S3-202074, S3-202117, S3-202116, S3-202115, S3-202151, S3-202063, S3-202119, and S3-202062. | 0.1.0 |
| 2020-10 | SA3#100bis-e | S3-202764 |  |  |  | Implemented S3-202729, S3-202521, S3-202620, S3-202697, S3-202318, S3-202319, S3-202320, S3-202321, S3-202733, S3-202743, S3-202744, S3-202731, S3-202742, S3-202781, S3-202779, S3-202778, S3-202777, S3-202776, S3-202759, S3-202758, S3-202757, S3-202756 | 0.2.0 |
| 2020-11 | SA3#101-e | S3-203436 |  |  |  | Implemented S3-203443, S3-203444, S3-203412, S3-203413, S3-203414, S3-203437, S3-203433, S3-203434, S3-203435, S3-202928, S3-203028, S3-203064, S3-203066, S3-203011, S3-203012, S3-203013, S3-203248, S3-203457, S3-203441, S3-203355, S3-203321, S3-203460, and S3-203461. | 0.3.0 |
| 2021-01 | SA3#102-e | S3-210654 |  |  |  | Implemented S3-210648, S3-210649, S3-210650, S3-210651. S3-210170. S3-210388, S3-210216, S3-210391, [S3-210633](https://www.3gpp.org/ftp/TSG_SA/WG3_Security/TSGS3_102e/Docs/S3-210633.zip), S3-210634, S3-210635, S3-210623, S3-210624, S3-210625, S3-210604, S3-210575, S3-210677, S3-210678, S3-210682, and S3-210692. | 0.4.0 |
| 2021-03 | SA3#102bis-e | S3-211321 |  |  |  | Implemented S3-210870. S3-210871, S3-210930, S3-210931, S3-210932, S3-210970, S3-211222, S3-211232, S3-211268, S3-211269, S3-211281, S3-211282, S3-211318, S3-211319, S3-211320, S3-211326, S3-211327, and S3-211337. | 0.5.0 |
| 2021-05 | SA3#103-e | S3-212240 |  |  |  | Implemented S3-212223, S3-212190. S3-211993, S3-211994, S3-212206, S3-212225, S3-212043, S3-212226, S3-212227, S3-212051, S3-212068, S3-212082, S3-212083, S3-212084, S3-212191, S3-211709, S3-212253, S3-212254, [S3-212234](https://www.3gpp.org/ftp/TSG_SA/WG3_Security/TSGS3_103e/Docs/S3-212234.zip), S3-212235, S3-212167, S3-212236, S3-212224, S3-212180, S3-211568, S3-212243, S3-212112. | 0.6.0 |
| 2021-08 | SA3#104-e | S3-213115 |  |  |  | Implemented S3-212678, S3-212680, S3-212937, S3-212939, S3-212951, S3-212952, S3-213090, S3-213112, S3-213116, S3-213117, S3-213131, S3-213135, S3-213211, and S3-213212. | 0.7.0 |
| 2021-10 | SA3#104-e-Ad-hoc | S3-213667 |  |  |  | Implemented S3-213441, S3-213624, S3-213626, S3-213627, S3-213628, S3-213631, S3-213632, S3-213662, and S3-213709. | 0.8.0 |
| 2021-11 | SA3#105-e | S3-214413 |  |  |  | Implemented S3-214066, S3-214347, S3-214385, S3-214386 | 0.9.0 |
| 2021-12 | SA#94e | SP-211403 |  |  |  | Presented for information and approval | 1.0.0 |
| 2021-12 | SA#94e |  |  |  |  | EditHelp/MCC review and upgrade to change control version | 17.0.0 |