ETSI TS 133 434 V16.3.1 (2022-04)

LTE;
5G;
Security aspects of Service Enabler Architecture Layer (SEAL) for verticals
(3GPP TS 33.434 version 16.3.1 Release 16)

Reference
RTS/TSGS-0333434vg31

Keywords
5G,LTE,SECURITY

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Legal Notice

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities. These shall be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Contents

Foreword

This Technical Specification 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 specifies the security features and mechanisms to support the Service Enabler Architecture Layer (SEAL) in 5G. Specifically security architecture, functional model(s), security aspects of SEAL reference points (e.g. SEAL-UU, etc.), Key Management (KM) procedures, Identity Management (IdM) procedures and SEAL access authentication and authorization for supporting efficient use and deployment of vertical applications over the 3GPP systems are specified.

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 23.434: "Service Enabler Architecture Layer for Verticals (SEAL); Functional architecture and information flows".

[3] IETF RFC 6749: "The OAuth 2.0 Authorization Framework".

[4] IETF RFC 6750: "The OAuth 2.0 Authorization Framework: Bearer Token Usage".

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

[6] 3GPP TS 33.310: "Network Domain Security (NDS); Authentication Framework (AF)".

[7] 3GPP TS 23.401: "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access".

[8] 3GPP TS 23.501: "System Architecture for the 5G System; Stage 2".

[9] IETF RFC 7521: "Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants".

[10] IETF RFC 7523: "JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants".

[11] IETF RFC 7797: " JSON Web Signature (JWS) Unencoded Payload Option ".

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

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

[14] 3GPP TS 33.210: " 3G security; Network Domain Security (NDS); IP network layer security".

[15] 3GPP TS 33.222: "Generic Authentication Architecture (GAA); Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS)".

[16] 3GPP TS 33.501: "Security architecture and procedures for 5G system".

[17] 3GPP TS 29.122: "T8 reference point for Northbound Application Programming Interfaces (APIs)".

[18] 3GPP TS 33.122: "Security aspects of Common API Framework (CAPIF) for 3GPP northbound APIs".

3 Definitions of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in 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 TR 21.905 [1].

For the purposes of the present document, the terms and definitions given in TS 23.434 [2] apply.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in 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 TR 21.905 [1].

SEAL	Service Enabler Architecture Layer for Verticals
SIM-C	SEAL Identity Management Client
SIM-S	SEAL Identity Management Server
SKM-C	SEAL-Key Management Client
SKM-S	SEAL Key Management Server
VAL	Vertical Application Layer

4 SEAL security requirements

4.1 VAL user authentication and authorization

[SEAL-SEC-4.1-a] All users of the VAL Service shall be authenticated.

[SEAL-SEC-4.1-b] The VAL Client and the VAL Server shall mutually authenticate each other prior to providing the VAL UE with the VAL Service User profile and access to user-specific services.

[SEAL-SEC-4.1-c] The transmission of configuration data and user profile data between an authorized VAL server in the network and the VAL UE shall be confidentiality protected, integrity protected and protected from replays.

[SEAL-SEC-4.1-d] The VAL service should take measures to detect and mitigate DoS attacks to minimize the impact on the network and on VAL users.

[SEAL-SEC-4.1-e] The VAL service shall provide a means to support confidentiality of VAL user identities.

[SEAL-SEC-4.1-f] The VAL service shall provide a means to support confidentiality of VAL signalling.

4.2 Inter-domain

[SEAL-SEC-4.2-a] VAL systems should take measures to protect themselves from external attacks at the system border.

5 Procedures

5.1 Security for the SEAL interfaces

5.1.1 Security for the Application plane interfaces

5.1.1.1 SEAL-X1

As defined in TS 23.434 [2], the SEAL-X1 reference point, exists between the key management server and the group management server and uses HTTP-1 as defined in 3GPP TS 23.434 [2] for the transport and routing of security related information to the group management server. The SEAL-X1 shall be protected using HTTPS as defined in [3], [4] and [5]. The profile for TLS implementation and usage shall follow the provisions given in 3GPP TS 33.310 [6], annex E.

5.1.1.2 SEAL-X2

The SEAL-X2 reference point enables the group management server to interact with the location management server as defined in 3GPP TS 23.434 [2]. The SEAL-X2 shall be protected using HTTPS as defined in [3], [4] and [5]. The profile for TLS implementation and usage shall follow the provisions given in 3GPP TS 33.310 [6], annex E.

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 shall be protected using HTTPS as defined in [3], [4] and [5]. The profile for TLS implementation and usage shall follow the provisions given in 3GPP TS 33.310 [6], annex E.

5.1.1.4 KM-UU and KM-S

The KM-UU reference point is used between the Key Management Client and Key Management Server. The security mechanism of SEAL-UU shall also be used for KM-UU.

The security established between the KM Server and the KM client should be end-to-end. When this is not possible, then all client related material transferred between the KM server and KM client should be end-to-end protected with a mechanism that is out of scope of this specification.

The KM-S reference point is a direct HTTP connection used between the VAL server and the key management server and shall be protected with the same mechanism used for the SEAL-S reference point.

5.1.1.5 SEAL-UU

A SEAL client interacts with a SEAL server over the generic SEAL-UU reference point as defined in 3GPP TS 23.434 [2].. This interface shall be protected using HTTPS as defined in [3], [4] and [5] when using HTTP. The profile for TLS implementation and usage shall follow the provisions given in TS 33.310 [6], annex E.

5.1.1.6 VAL-UU

The VAL client interacts with VAL server over VAL-UU reference point as defined in 3GPP TS 23.434 [2].

NOTE: Security mechanism for the VAL-UU reference point is out of scope of present document.

5.1.1.7 SEAL-C

The VAL client interacts with a SEAL client over the SEAL-C reference point as defined in 3GPP TS 23.434 [2]. This reference point resides fully within the UE and therefore, security of this interface is left to the manufacturer and is out of scope for the present document.

5.1.1.8 SEAL-S

The VAL server interacts with SEAL server over SEAL-S reference point as defined in 3GPP TS 23.434 [2]. The protection of this interface shall be supported according to NDS/IP as specified in TS 33.210 [14].

When CAPIF is used as specified in TS 23.434 [2], the security mechanism for CAPIF specified in TS 33.122 [18] shall be followed. CAPIF core function shall choose the appropriate CAPIF-2e security method as defined in the sub-clause 6.5.2 in TS 33.122 [18] for mutual authentication and protection of the SEAL server – VAL server interface. Before invoking the API exposed by the SEAL server, the VAL server as API invoker shall negotiate the security method (TLS-PSK, PKI or TLS with OAuth token) with CAPIF core function and ensure the SEAL server has information to authenticate the VAL server.

5.1.1.9 SEAL-E

A SEAL server interacts with another SEAL server over SEAL-E reference point as defined in 3GPP TS 23.434 [2]. The protection of this interface shall be supported according to NDS/IP as specified in TS 33.210 [14].

5.1.2 Security for the Signalling control plane interfaces

5.1.2.1 Security for HTTP interfaces

In order to authenticate the HTTP-1 reference point, authentication mechanisms shall be performed between the HTTP client and VAL UE using either certificate based authentication or pre-shared key based authentication. Certificate based authentication shall follow in annex B of 3GPP TS 33.222 [15], and the profiles given in 3GPP TS 33.310 [6]. The usage of pre-shared key based ciphersuites is specified in the TLS profile given in 3GPP TS 33.310 [6], annex E.

The HTTP-1 reference point exists between the VAL UE and the HTTP proxy. The HTTP-2 exists between the HTTP proxy and HTTP server. The HTTP-3 reference point exists between the HTTP proxies in different networks. The HTTP interfaces shall be protected using TLS. The profile for TLS implementation and usage shall follow the provisions given in 3GPP TS 33.310 [6], annex E.

5.1.3 Security for the network domain interfaces

A VAL UE shall perform the authentication and security mechanisms as specified in 3GPP TS 33.501 [16] for 5G network access security.

To ensure security of the interfaces between network entities within a trusted domain and between trusted domains, 3GPP TS 33.210 [14] shall be applied to secure signalling messages on the reference points unless specified otherwise. SEG as specified in 3GPP TS 33.210 [14] may be used in the trusted domain to terminate the IPsec tunnel.

5.2 User authentication and authorization

5.2.1 VAL user authentication

Figure 5.2.3-1 shows the Identity Management functional model which consists of the SEAL Identity Management Server (SIM-S) and SEAL Identity Management Client (SIM-C) of the UE. The IM-UU reference point between the SIM-S and SIM-C shall provide the interface for user authentication and shall support OpenID Connect 1.0 [5] and OAuth 2.0 [9], [10] to obtain an access token for the VAL UE.

5.2.2 SEAL service authorization

SEAL Service Authorization procedure shall validate the VAL user to access the SEAL services. In order to gain access to SEAL services, the SEAL client shall present an access token to the SEAL server for each service of interest. If the access token is valid, then the client shall be granted to use the service.

5.2.3 Identity management functional model

The SEAL Identity Management Server (SIM-S) and the SEAL Identity Management Client (SIM-C) provide the endpoints for VAL user authentication as shown in the SEAL Identity Management functional model in figure 5.2.3-1.

The reference point IM-UU utilizes Uu reference point as described in 3GPP TS 23.401 [7] and 3GPP TS 23.501 [8]. IM-UU shall support OpenID Connect 1.0 [5] and OAuth 2.0 [9] for VAL user authentication.

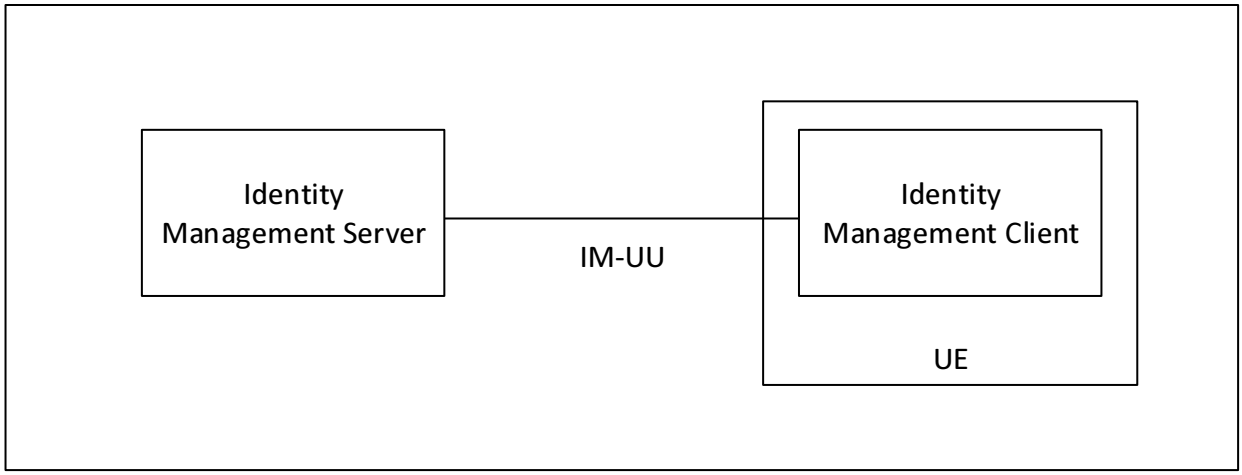

Figure 5.2.3-1: Functional model for SEAL Identity Management

In order to support VAL user authentication, the SIM-S shall be provisioned with the VAL user ID and VAL service IDs (usage of VAL user ID and VAL service ID is described in clause 7 of TS 23.434 [2]). A mapping between the VAL user ID and VAL service ID(s) shall be created and maintained in the SIM-S. When a VAL user wishes to authenticate for the VAL services, the VAL user ID and credentials are provided via the UE Identity management client to the SIM-S as per OpenID Connect 1.0 [5]. The SIM-S receives and shall verify the VAL user ID and credentials. If verification is successful, then the SIM-S returns an ID token, refresh token and access token to the UE Identity management client. The SIM-C shall learn the user's VAL service ID(s) from the ID token. Table A shows the SEAL specific tokens and their usage.

Table 5.2.3-1: VAL UE authentication token

Token Type	Consumer of the Token	Description
ID token	VAL UE client(s)	Contains the VAL service ID for at least one authorized VAL service.
Access token	SKM-S, SEAL service server(s)	Short-lived token (definable in the SIM-S) that conveys the UE's identity. This token contains the VAL service ID for at least one authorized service.
Refresh token	SIM-S (Authorization Server)	Allows VAL UE to obtain a new access token without forcing user to log in again.

To support the VAL service identity functional model, the VAL service ID(s):

- Shall be provisioned into the SEAL Identity management database and mapped to VAL UE IDs.
- Shall be provisioned into the SEAL Key management server (SKM-S) and mapped to UE specific key material.

5.2.4 Authentication framework

Figure 5.2.4-1 describes the VAL Authentication Framework using the OpenID Connect protocol. It describes the steps by which a VAL UE authenticates to the SIM-S, resulting in a set of credentials delivered to the UE uniquely identifying the VAL service ID(s). The authentication framework supports extensible user authentication solutions based on the VAL service provider policy (shown as step 3). User authentication methods in support of step 3 (e.g. biometrics, secureID, etc.) are possible but not defined here.

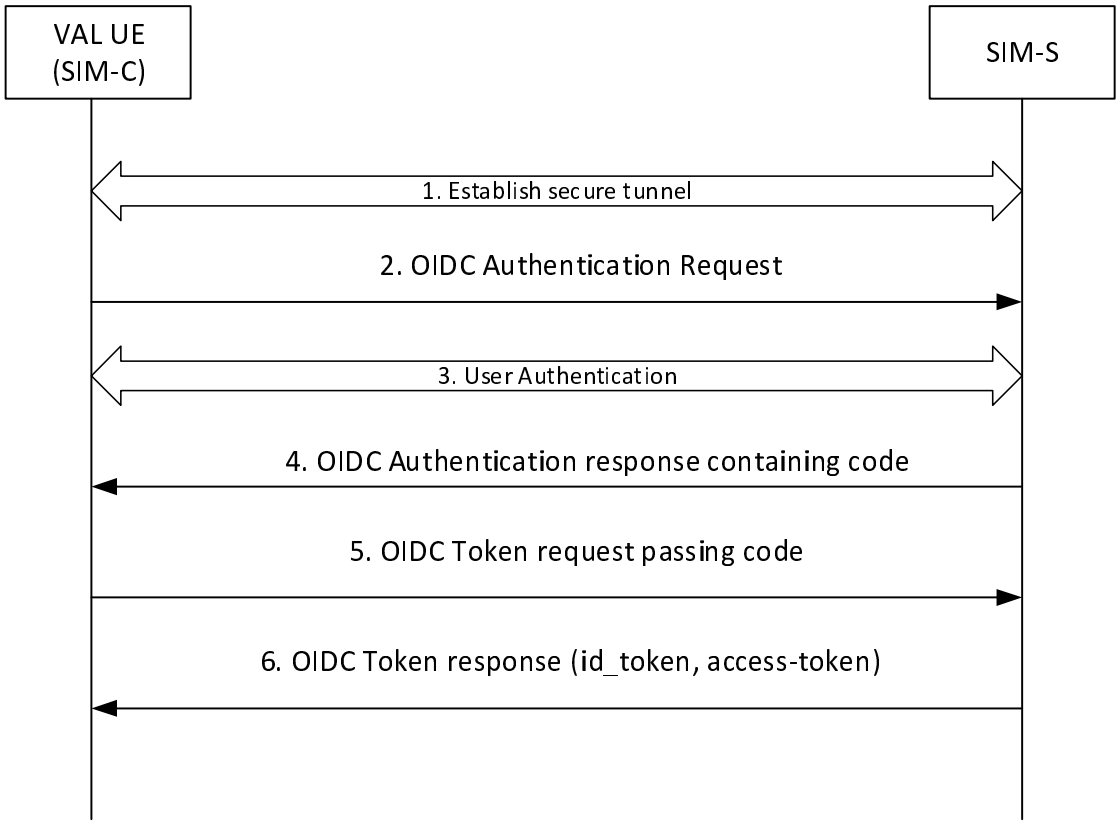

Figure 5.2.4-1: OpenID Connect (OIDC) flow supporting VAL user authentication

Step 1: VAL UE establishes a secure tunnel with the SIM-S.

Step 2: VAL UE sends an OpenID Connect Authentication Request to the SIM-S. The request may contain an indication of authentication methods supported by the UE.

Step 3: User Authentication is performed between VAL UE and the SIM-S.

NOTE: The primary credentials for user authentication (e.g. biometrics, secureID, OTP, username/password) are based on VAL service provider policy. The method chosen by the VAL service provider for authentication and authorization is neither defined nor limited by the present document, it depends on the Vertical services and authentication and authorization methods supported by it.

Step 4: SIM-S sends an OpenID Connect Authentication Response to the UE containing an authorization code.

Step 5: UE sends an OpenID Connect Token Request to the SIM-S, passing the authorization code.

Step 6: SIM-S sends an OpenID Connect Token Response to the UE containing an ID token and an access token (each which uniquely identify the user of the VAL service or key management service). The ID token is consumed by the UE to personalize the VAL client for the VAL user, and the access token is used by the UE to communicate and authorize the identity of the VAL user to the VAL server(s) and the VAL services.

5.2.5 Authorization framework

Authorization framework is shown in figure 5.2.5-1. A secure HTTP tunnel using HTTPS between VAL UE and VAL server shall be established before VAL service authorization. Subsequent VAL service authorization messaging make use of this tunnel. The service clients in the VAL UE present the access tokens to the VAL server over HTTP. The VAL server authorizes the user for the requested services only if the access token is valid. The procedures may be repeated as necessary to obtain additional VAL user authorizations.

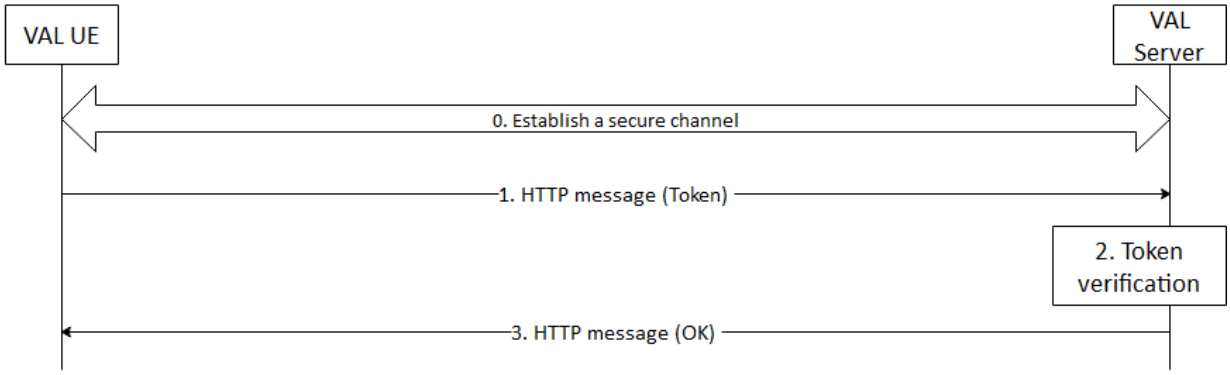

Figure 5.2.5-1: VAL User Service Authorization

After the VAL UE establishing a secure connection with the VAL server, the VAL UE sends an HTTP message containing the access token to the VAL server where service authorization is requested. The VAL server receives the message and validates the access token. If the access token is valid, The VAL server positively acknowledges the request. The VAL server may provide service related information to the VAL UE at this time.

5.2.6 VAL service authorization

The VAL service authorization procedure shall validate the VAL user authorized to access the VAL services. In order to gain access to VAL services, the VAL client shall present an access token to the VAL server for each VAL service of interest (see clause 5.2.5). If the access token is valid, then the VAL client shall be granted use of the requested VAL service.

5.3 SEAL key management procedure

5.3.1 General

To enable security for VAL services, a SEAL KM client (located in either a SEAL UE or VAL server) may request key material applicable to a particular VAL service, VAL client or user.

Prior to making a key management request to the SEAL KMS (SKM-S), the VAL client or VAL user shall be authenticated by the SEAL identity management service (clause 5.2). In addition, secure connections shall be established between the SEAL client and the SKM-S (reference point KM-UU) and the VAL server and the SKM-S (reference point KM-S) prior to any associated key management requests.

As a result of the SEAL identity management authentication procedure, an access token scoped for key management services is provisioned to the SEAL UE. This access token is provided with each and every key management request to the SKM-S.

A VAL server is provisioned with an access token scoped for SEAL key management services and is provided with each and every key management request to the SKM-S. The method for provisioning this access token into the VAL server is out of scope of the present document.

Figure 5.3.1-1 shows the SEAL key management procedure. A SKM client may send a SEAL KM Request message to the SKM-S. The SKM-S validates and processes the request and responds with a SEAL KM Response message. The response contains key management material specific to the SEAL service or the VAL server request, or alternatively, an error code if the SKM-S encounters a failure condition.

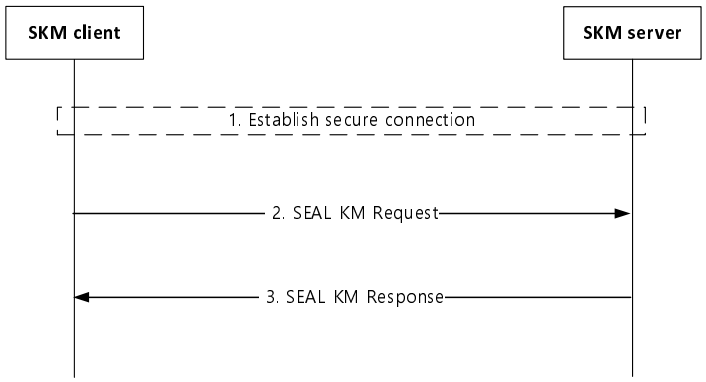

Figure 5.3.1-1: SEAL key management procedure

The procedure in figure 5.3.1-1 is described here:

1. The SKM-C establishes a direct HTTPS connection to the SKM-S. Steps 2 and 3 are within this secure connection.
2. The SKM-C sends a SEAL KM Request message to the SKM-S. The request contains the authorization credentials obtained during authentication and message content specified in clause 5.3.2.
3. The SKM-S authorizes the request and if valid, sends a SEAL KM Response message containing the requested key material (or error code) as specified in clause 5.3.3.

As a successful result of this procedure, the VAL UE or VAL Server has securely obtained service specific key material for use within the VAL system.

5.3.2 SEAL KM Request message

A SKM-C may send a SEAL KM Request message to the SKM-S. This request shall be protected (via the HTTPS tunnel) and shall contain the access token acquired during the SEAL identity management authentication procedure (clause 5.2).

The content of the SEAL KM Request is shown in table 5.3.2-1.

Table 5.3.2-1: Contents of a SEAL KM Request

Name	Description
Version	The version number of the SEAL key management request .
SKmsUri	The URI of the SKM-S to which the request is sent.
ServiceID	A string representing the VAL service/application related to the VAL client request.
ClientID	(Optional) A string representing the client. See note.
DeviceID	(Optional) A string representing the device. See note.
UserID	(Optional) A string representing the user. See note.
Date/Time	The Date and Time of the request. This number represents the number of seconds from 1970-01-01T0:0:0Z as measured in UTC.
NOTE: Only one of these fields may be present in any given SEAL KMS Request message.	

The identities listed in table 5.3.2-1 map to SEAL identities defined in 3GPP TS 23.434 [2]. Namely, the ServiceID maps to the VAL service identity (VAL service ID), the ClientID maps to the VAL client or client on the VAL server, the DeviceID maps to the VAL UE identity (VAL UE ID), and the UserID maps to the VAL user identity (VAL user ID).

The ‘Version’ field identifies the version of the SEAL KM Request message. The current version is defined as "1.0.0".

The ‘Date/Time’ field is used primarily as an anti-replay mechanism for SEAL key management requests and responses. If the ‘Date/Time’ field is significantly out of range (more than a few seconds), this could indicate a replay attack.

Upon receipt of a SEAL KM Request message, the SKM-S shall verify that:

- the access token is valid;
- the signature is valid;
- the SKmsUri is the SKM-S URI of the target SEAL KMS where the key information is stored; and
- the Date/Time is within a recent time window (e.g. 5 seconds).

If valid, the request is accepted and processed by the SKM-S. A standalone ServiceID, or a ServiceID in combination with a ClientID, DeviceID, or UserID may be present in the SEAL KM Request message. This combination may be used by the KMS to identify a specific key material record. Each key management record may be unique to a VAL application or VAL service. The format and content of a key management record is defined and securely provisioned into the SEAL KMS by the VAL application or VAL service owner/operator. The method used to provision the VAL service or VAL application key material into the KMS is out of scope for the present document. The method used to organize, manage, and maintain VAL service or VAL application key material within the KMS is out of scope of the present document.

5.3.3 SEAL KM Response message

The SEAL KM Response message is sent to the SKM-C in response to a SEAL KM Request message.

A successful SEAL key management procedure results in a SEAL KM Response message which typically includes a payload containing key management information uniquely applicable to the requested service, client or user. If an error occurs, an error code may be returned in the SEAL KM Response message.

The SEAL KM Response message shall be protected in transit via the HTTPS tunnel. The Payload within a SEAL KM Response message may be protected end-to-end between the SKM-C and SKM-S depending on the applicability of the underlying VAL service making the request. The method for securing a Payload end-to-end between the SKM-C and the SKM-S is outside the scope of the present document. The key material contents provided in a Payload are defined by the underlying VAL service and are outside the scope of the present document.

The content of a SEAL KM Response message is shown in table 5.3.3-1.

Table 5.3.3-1: Contents of a SEAL KM Response message

Name	Description
UserUri	URI of the user for which the response is intended.
SKmsUri	The URI of the SKM-S sending the response.
ServiceID	A string representing the VAL service/application related to the VAL client request. This is the same field as received in the SEAL KM Request message.
SKmsID	(Optional) The ID of the SKM-S providing the response message.
ClientID	(Optional) A string representing the client (see note)
DeviceID	(Optional) A string representing the device (see note)
UserID	(Optional) A string representing the user. (see note)
Date/Time	The Date and Time of the response. This number represents the number of seconds from 1970-01-01T0:0:0Z as measured in UTC.
ErrorCode	(Optional) Reason code indicating the failure of the requested action. If not present, the key management request is assumed to be successful.
Payload	(Optional) Key management payload specific to the VAL user, client or application. This field is not be present if an error occurs .
NOTE: If this field is present in the SEAL KM Request message then this field shall be present in the SEAL KM Response message and shall be the same value.	

The identities listed in table 5.3.3-1 are described in clause 5.3.2.

If the SKM-S does not encounter an error during processing of the SEAL KM Request message, the SEAL KM Response message carries a set of security parameters contained in the “Payload” field.

If the SKM-S encounters an error while processing the SEAL KM Request message, an error value described in table 5.3.3-2 shall be returned in the ‘ErrorCode’ field of the SEAL KM Response message and the ‘Payload’ field shall not be present.

In the event of an error, the user and/or the operator of the VAL service, UE, or client may be notified.

Table 5.3.3-2: 'ErrorCode' values

ErrorCode	Description	Maps To
01	Unspecified error	"500 Internal Server Error" as described in Table 5.2.6-1 of TS 29.122 [17]
02	Key Information not available for specified service, client, device or user.	"404 Not Found" as described in Table 5.2.6-1 of TS 29.122 [17]
03	Request rejected	"401 Unauthorized" as described in Table 5.2.6-1 of TS 29.122 [17]
04	Unable to validate request	"400 Bad Request" or "403 Forbidden" as described in Table 5.2.6-1 of TS 29.122 [17]
05-FF	Reserved	N/A

The selection of the key material returned in the Payload of a SEAL KM Response message is determined by the ServiceID and (optionally) the ClientID, DeviceID or UserID. The combination of the ServiceID with the ClientID, DeviceID or UserID allows the VAL service to request a more specific set of key material.

For example, if a ClientID is included in the SEAL KM Request message, the KMS may return a Payload that contains a set of client specific key material applicable to the ClientID within the requesting VAL service (ServiceID). If the DeviceID is included, the KMS may return a Payload that contains device specific key material applicable to the DeviceID within the requesting VAL service (ServiceID). If the UserID is included, the KMS may return a Payload that contains user specific key material applicable to that UserID within the requesting VAL service (ServiceID).

5.4 Security procedures for interconnection

Interconnection between a primary VAL system and a partner VAL system is specified in 3GPP TS 23.434 [2].

A VAL client shall perform user authorization only to VAL servers within their own VAL system. When communication is required by a VAL client from another interconnected VAL system, user authorization takes place in the serving VAL system and follows the VAL user service authorization procedures as defined in clause 5.2.

VAL systems should protect themselves at the system border from external attackers.

Annex A (normative): OpenID connect profile for VAL

A.1 General

The information in this annex provides a normative description of the Authentication and Authorization framework based on the OpenID Connect 1.0 standard. Characterization of the ID token, access token, how to obtain tokens, how to validate tokens, and how to use the refresh token is explained.

The OpenID Connect 1.0 standard provides the source of the information contained in this annex. This annex profiles the OpenID Connect standard and includes the ID token and the access token, as well as the definition of VAL specific scopes for key management, VAL services, configuration management, and group management. This profile is compliant with OpenID Connect.

A.2 VAL tokens

A.2.1 ID token

A.2.1.1 General

The ID Token shall be a JSON Web Token (JWT) and contain the following standard and VAL token claims. Token claims provide information pertaining to the authentication of the VAL client by the SIM-S as well as additional claims. The following clause profiles the required standard and VAL claims for the VAL Connect profile.

A.2.1.2 Standard claims

These standard claims are defined by the OpenID Connect 1.0 specification and are REQUIRED for VAL implementation. Other claims defined by OpenID Connect are optional. The standards-based claims for a VAL Connect ID token are shown in table A.2.1.2-1.

Table A.2.1.2-1: ID token standard claims

Parameter	Description
Iss	REQUIRED. The URL of the SIM-S.
Sub	REQUIRED. A case-sensitive, never reassigned string (not to exceed 255 bytes), which uniquely identifies the VAL user within the VAL server provider's domain.
Aud	REQUIRED. The Oauth 2.0 client_id of the SIM-C
Exp	REQUIRED. Implementers MAY provide for some small leeway, usually no more than a few minutes, to account for clock skew (not to exceed 30 seconds)
iat	REQUIRED. Time at which the ID Token was issued. Its value is a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time.

A.2.1.3 VAL claims

The VAL Connect profile extends the OpenID Connect standard claims with the additional claims based on the VAL service.

A.2.2 Access token

A.2.2.1 Introduction

The access token is opaque to VAL clients and is consumed by the VAL resource servers. The access token shall be encoded as a JSON Web Token as defined in IETF RFC 7797 [11]. The access token shall include the JSON web digital signature profile as defined in IETF RFC 7515 [12].

A.2.2.2 Standard claims

VAL access tokens shall convey the following standards-based claims as defined in IETF RFC 7662 [13].

Table A.2.2.2-1: Access token standard claims

Parameter	Description
Exp	REQUIRED. Implementers MAY provide for some small leeway, usually no more than a few minutes, to account for clock skew (not to exceed 30 seconds).
Scope	REQUIRED. A JSON string containing a space-separated list of the authorization scopes associated with this token. The scope(s) contained here reflect the requested scope(s) from the Authentication Request (clause A.4.2.2).
client_id	REQUIRED. The identifier of the SIM-C making the API request as previously registered with the SIM-S.

A.2.2.3 VAL claims

The VAL profile extends the standard claims defined in IETF RFC 7662 [13] with the additional claims based on the VAL service.

A.3 SIM-C registration

Before a SIM-C can obtain ID tokens and access tokens (required to access VAL resource servers) it shall first be registered with the SIM-S of the service provider as required by OpenID Connect 1.0. The method by which this is done is not specified by this profile. For native SIM-C, the following information shall be registered:

- The client is issued a client identifier. The client identifier represents the client's registration with the authorization server, and enables the SIM-S to reference parameters associated with that client's registration when being requested for an access token by the SIM-C.
- Registration of the client's redirect URIs.

Other information about the SIM-C such as (for example): application name, website, description, logo image, legal terms to be consented to, may optionally be registered.

A.4 Obtaining tokens

A.4.1 General

Once a SIM-C has been successfully registered with the SIM-S of the VAL service provider, the SIM-C may request ID tokens and access tokens (as required to access VAL service servers). Only native SIM-C are defined here. The exact method in which a SIM-C requests the access token depends upon the client profile. The SIM-C profiles, along with steps required from them to obtain OAuth access tokens, are explained below.

A.4.2 Native SIM-C

A.4.2.1 General

This conforms to the Native Application profile of OAuth 2.0 as per IETF RFC 6749 [3].

SIM-C fitting the Native application profile utilize the authorization code grant type with the PKCE extension for enhanced security as shown in figure A.4.2.1-1.

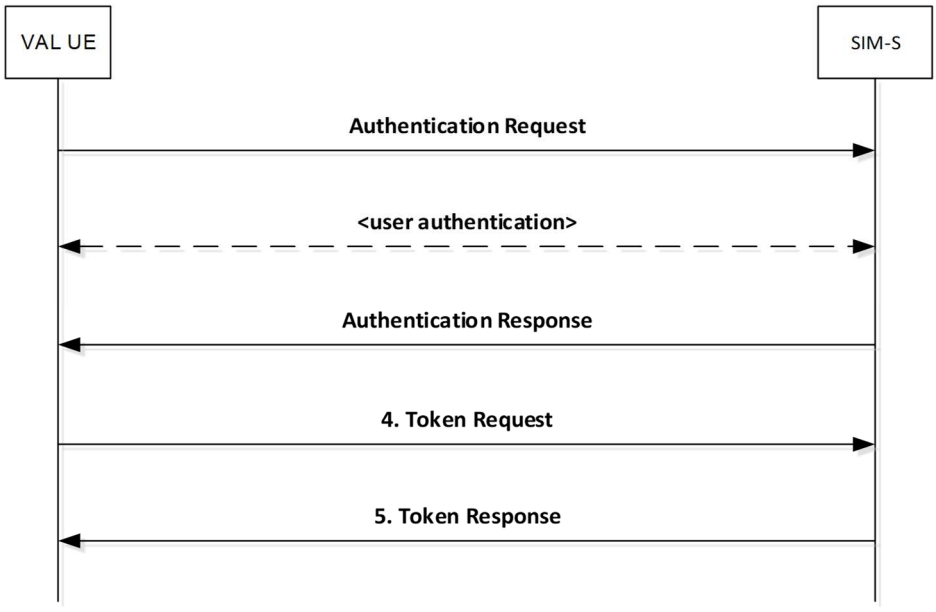

Figure A.4.2.1-1: Authorization Code flow

A.4.2.2 Authentication request

As described in OpenID Connect 1.0, the SIM-C constructs a request URI by adding the following parameters to the query component of the authorization endpoint's URI using the "application/x-www-form-urlencoded" format, redirecting the user's web browser to the authorization endpoint of the SIM-S. The standard parameters shown in table A.4.2.2-1 are required by this Connect profile. Other parameters defined by the OpenID Connect specification are optional.

Table A.4.2.2-1: Authentication Request standard required parameters

Parameter	Values
response_type	REQUIRED. For native SIM-C the value shall be set to "code".
client_id	REQUIRED. The identifier of the SIM-C making the API request. It shall match the value that was previously registered with the SIM-S of the VAL service provider.
Scope	REQUIRED. Scope values are expressed as a list of space-delimited, case-sensitive strings which indicate which VAL resource servers the client is requesting access to. If authorized, the requested scope values will be bound to the access token returned to the client. The scope value "openid" is defined by the OpenID Connect standard and is mandatory, to indicate that the request is an OpenID Connect request, and that an ID token should be returned to the SIM-C. NOTE: Additional VAL service specific scopes need to be defined by VAL service specification and it is out of scope of the present document.
redirect_uri	REQUIRED. The URI of the SIM-C to which the SIM-S will redirect the SIM-C's user agent in order to return the authorization code to the SIM-C. The URI shall match the redirect URI registered with the SIM-S during the client registration phase.
State	REQUIRED. An opaque value used by the SIM-C to maintain state between the authorization request and authorization response. The SIM-S includes this value in its authorization response back to the SIM-C.
acr_values	REQUIRED. Space-separated string that specifies the acr values that the SIM-S is being requested to use for processing this authorization request, with the values appearing in order of preference. For minimum interoperability requirements, a password-based ACR value is mandatory to support. "3gpp:acr:password".
code_challenge	REQUIRED. The base64url-encoded SHA-256 challenge derived from the code verifier that is sent in the authorization request, to be verified against later.
code_challenge_method	REQUIRED. The hash method used to transform the code verifier to produce the code challenge. This profile current requires the usage of "S256"
NOTE: The order in which they are expressed does not matter.	

A.4.2.3 Authentication response

The authorization endpoint running on the SIM-S issues an authorization code and delivers it to the SIM-C. The authorization code is used by the SIM-C to obtain an ID token, access token and refresh token from the SIM-S. The authorization code is added to the query component of the redirection URI using the "application/x-www-form-urlencoded" format. The authorization code standard parameters are shown in table A.4.2.3-1.

Table A.4.2.3-1: Authentication Response standard required parameters

Parameter	Values
Code	REQUIRED. The authorization code generated by the authorization endpoint and returned to the SIM-C via the authorization response.
State	REQUIRED. The value shall match the exact value used in the authorization request. If the state does not match exactly, then the NGMI API client is under a Cross-site request forgery attack and shall reject the authorization code by ignoring it and shall not attempt to exchange it for an access token. No error is returned.

A.4.2.4 Access token request

In order to exchange the authorization code for an ID token, access token and refresh token, the SIM-C makes a request to the authorization server's token endpoint by sending the following parameters using the "application/x-www-form-urlencoded" format, with a character encoding of UTF-8 in the HTTP request entity-body. Note that client authentication is REQUIRED for native applications (using PKCE) in order to exchange the authorization code for an access token. Assuming that client secrets are used, the client secret is sent in the HTTP Authorization Header. The access token request standard parameters are shown in table A.4.2.4-1.

Table A.4.2.4-1: Access token request standard required parameters

Parameter	Values
grant_type	**REQUIRED. The value shall be set to "authorization_code".**
code	REQUIRED. The authorization code previously received from the SIM-S as a result of the authorization request and subsequent successful authentication of the VAL user.
client_id	REQUIRED. The identifier of the client making the API request. It shall match the value that was previously registered with the OAuth Provider during the client registration phase of deployment, or as provisioned via a development portal.
redirect_uri	REQUIRED. The value shall be identical to the "redirect_uri" parameter included in the authorization request.
code_verifier	REQUIRED. A cryptographically random string that is used to correlate the authorization request to the token request.

A.4.2.5 Access token response

If the access token request is valid and authorized, the SIM-S returns an ID token, access token and refresh token to the SIM-C in an access token response message; otherwise it will return an error.

The access token response standard parameters are shown in table A.4.2.5-1.

Table A.4.2.5-1: Access token response standard parameters

Parameter	Values
access_token	REQUIRED. This is the issued access token.
token_type	REQUIRED. This field shall be "bearer"
expires_in	REQUIRED. The lifetime in seconds of the access token.
Id_token	OPTIONAL. This is the issued id token.
Refresh_token	OPTIONAL. This is the issued refresh token.

The SIM-C may now validate the user with the ID token and configure itself for the user (e.g. by extracting the VAL service ID from the ID Token). The SIM-C then uses the access token to make authorized requests to the SIM resource servers on behalf of the end user.

A.5 Refreshing an access token

A.5.1 General

To protect against leakage or other compromise, access token lifetimes are typically short lived (though it is ultimately a matter of security policy & configuration by the service provider). Some client types can be issued longer-lived refresh tokens, which enable them to refresh the access token and avoid having to prompt the user for authentication again when the access token expires. Refresh tokens are available only to clients utilizing the authorization code grant type. Figure A.5.1-1 shows how Native SIM-C can use the refresh token as a grant type to obtain new access tokens.

Figure A.5.1-1: Requesting a new access token

A.5.2 Access token request

To obtain an access token from the SIM-S using a refresh token, the SIM-C makes an access token request to the token endpoint of the SIM-S. The SIM-C does this by adding the following parameters using the "application/x-www-form-urlencoded" format, with a character encoding of UTF-8 in the HTTP request entity-body. The access token request standard parameters are shown in table A.5.2-1.

Table A.5.2-1: Access token request standard required parameters

Parameter	Values
grant_type	REQUIRED. The value shall be set to "refresh_token".
Scope	Space-delimited set of permissions that the SIM-C requests. Note that the scopes requested using this grant type shall be of equal to or lesser than scope of the original scopes requested by the SIM-C as part of the original authorization request.

If the SIM-C was provided with client credentials by the SIM-S, then the client shall authenticate with the token endpoint of the SIM-S utilizing the client credential (shared secret or public-private key pair) established during the client registration phase.

A.5.3 Access token response

In response to the access token request (above) the token endpoint on the SIM-S will return an access token to the SIM-C, and optionally another refresh token in an access token response message.

The access token response standard parameters are shown in table A.5.3-1.

Table A.5.3-1: Access token response standard parameters

Parameter	Values
access_token	REQUIRED. This is the issued access token.
token_type	REQUIRED. This field shall be "bearer"
expires_in	REQUIRED. The lifetime in seconds of the access token.
Id_token	OPTIONAL. This is the issued id token.
Refresh_token	OPTIONAL. This is the issued refresh token.

It is possible to configure the SIM-S to confirm that the user account is still valid each time the refresh token is presented, and to revoke the refresh token if not. This security practice is RECOMMENDED.

A.6 Using the token to access VAL resource servers

Connect for VAL shall initially support the bearer access token type. Access tokens of type "bearer" shall be communicated from the VAL or SEAL Clients in UE to VAL resource servers by including the access token in the HTTP Authorization Header, per IETF RFC 6750 [4].

The access token is opaque to the VAL or SEAL Clients in UE, meaning that the client does not have any knowledge of the access token itself. The client will be given some metadata corresponding to the access token, such as its expiration time, so that it does not send an expired access token to VAL resource servers. If the access token is presented to a VAL resource server and the scope is invalid or the token is expired or revoked, the VAL resource server should return an error message indicating such to the VAL or SEAL Clients in UE.

A.7 Token validation

A.7.1 ID token validation

The VAL or SEAL Clients in UE shall validate the ID token as per clause 3.1.3.7 of the OpenID Connect 1.0 specification [5].

A.7.2 Access token validation

VAL resource servers shall validate access tokens received from the VAL or SEAL Clients in UE according to IETF RFC 7797 [11].

A.8 Token revocation

In order to limit the time validity of a token, the "exp" and "expires_in" parameters may be used as a method of access token revocation. If either the "exp" or "expires_in" parameter is used as a method of access token revocation, then the following applies:

Within the standard claims of an access token, the "exp" parameter shall be used by the authorising server to determine whether or not the token is valid. If the current time is beyond the time specified by the "exp" parameter, the associated token shall no longer be considered valid and any requests made with an expired token shall be rejected by the authorising server.

Within the standard claims of an access token response, token exchange response or token response message, the "expires_in" parameter shall be used by the UE client(s) to determine validity of the associated token. If the current time is beyond the time specified by the "expires_in" parameter, the associated token shall no longer be considered valid and no client requests shall be made using the expired token. A refresh token may be used per clause A.5 to obtain a new access token.

A.9 SIM-S interface security

The support of Transport Layer Security (TLS) between the SIM-C in the VAL UE and the SIM-S is mandatory. The profile for TLS implementation and usage shall follow the provisions given in 3GPP TS 33.310 [6], annex E.

If PSK TLS based authentication is supported, the SIM-C in the VAL UE and the SIM-S shall support the TLS version, PSK ciphersuites and TLS Extensions as specified in the TLS profile given in 3GPP TS 33.310 [6], annex E. The usage of pre-shared key ciphersuites for TLS is specified in the TLS profile given in 3GPP TS 33.310 [6], annex E.

Annex B (informative): Change history

Change history							
Date	**Meeting**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
2019-11	SA3#97	S3-194627				Initial TS Skeleton proposal	0.0.0
2019-11	SA3#97	S3-194630				Implementation of documents agreed in the meeting SA3#97: S3-194628, S3-194629, S3-194631, S3-194632	0.1.0
2020-03	SA3#98e	S3-200449				Implementation of documents agreed in the meeting SA3#98e: S3-200164, S3-200451, S3-200452, S3-200167, S3-200492, S3-200493, S3-200494, S3-200495	0.2.0
2020-04	SA3#98bis-e	S3-200827				Implementation of documents agreed in the meeting SA3#98bis-e: S3-200835, S3-200836	0.3.0
2020-05	SA3#99e	S3-201401				Implementation of documents agreed in the meeting SA3#99e: S3-201396	0.4.0
2020-06	SA#88e	SP-200383				EditHelp review. Presented for information and approval	1.0.0
2020-07	SA#88e					Upgrade to change control version	16.0.0
2020-09	SA#89e	SP-200715	0001	-	F	KM Clarifications	16.1.0
2020-09	SA#89e	SP-200715	0002	1	F	TS 33.434 clean up	16.1.0
2021-03	SA#91e	SP-210112	0003	1	F	correction in clause 5.2.4	16.2.0
2022-03	SA#95e	SP-220224	0009	1	F	CAPIF usage for SEAL-S	16.3.0
2022-03	SA#95e	SP-220224	0011	1	F	Correcting SEAL-UU security	16.3.0
2022-03	SA#95e					Correccion in an error of implementation in 0009r1	16.3.1

History

Document history		
V16.0.0	July 2020	Publication
V16.1.0	November 2020	Publication
V16.2.0	April 2021	Publication
V16.3.1	April 2022	Publication