O-RAN.WG11.TS.Security-Protocols-Specification.0-R004-v11.00

Technical Specification

Copyright © 2025 by the O-RAN ALLIANCE e.V.

The copying or incorporation into any other work of part or all of the material available in this specification in any form without the prior written permission of O-RAN ALLIANCE e.V. is prohibited, save that you may print or download extracts of the material of this specification for your personal use, or copy the material of this specification for the purpose of sending to individual third parties for their information provided that you acknowledge O-RAN ALLIANCE as the source of the material and that you inform the third party that these conditions apply to them and that they must comply with them.

O-RAN ALLIANCE e.V., Buschkauler Weg 27, 53347 Alfter, Germany

Register of Associations, Bonn VR 11238, VAT ID DE321720189

O-RAN Work Group 11 (Security Work Group)

Security Protocols Specifications

# Contents

Contents 2

Foreword 5

Modal verbs terminology 5

1 Scope 6

2 References 6

2.1 Normative references 6

2.2 Informative references 9

3 Definition of terms, symbols and abbreviations 9

3.1 Terms 9

3.2 Symbols 9

3.3 Abbreviations 9

4 Security protocols specifications for O-RAN compliant implementation 10

4.1 SSH 10

4.1.1 General requirements 10

4.1.2 Required ciphers 10

4.2 TLS 12

4.2.1 General Requirements 12

4.2.2 TLS Protocol Profile 12

4.2.3 TLS Certificate Profile 12

4.3 Support NETCONF over secure Transport 15

4.4 DTLS 15

4.4.1 General requirements 15

4.4.2 DTLS Protocol Profile 16

4.4.3 Certificate Profile 16

4.5 IPSec 16

4.5.1 Overview 16

4.5.2 Parallel usage of IPsec and other secure transport protocols 17

4.5.3 Responder mode and Initiator/Responder mode support 17

4.6 CMPv2 17

4.7 OAuth 2.0 18

4.7.1 Overview 18

4.7.2 Basic Parametrization 18

4.7.3 OAuth2.0 Access token API for O-RAN Elements and interfaces 20

4.8 LDAP 27

4.8.1 General requirements 27

4.9 MACsec 27

4.9.1 General requirements 27

4.9.2 MACsec Profile 27

5 Cryptographic operations 28

6 Secure File Transfer protocols 30

6.1 General 30

6.2 SFTP 30

6.2.1 General Requirements 30

6.3 FTPES 30

6.3.1 General Requirements 30

6.4 HTTPS 30

6.4.1 General Requirements 30

Annex A (normative): OpenAPI specifications 31

A.1 OAuth2.0 API 31

A.1.1 Introduction 31

A.1.2 OAuth 2.0 Access Token API 31

Annex (informative): Change history/Change request (history) 34

**List of Figures**

[Figure 4.7.2.3‑1: Access Token request process 19](#_Toc184047775)

[Figure 4.7.2.4‑1: Service access request based on token verification 20](#_Toc184047776)

[Figure 4.7.3.1‑1: Resource structure for access token 21](#_Toc184047777)

[Figure 4.7.3.2‑1: Access Token Request and Response 22](#_Toc184047778)

[Figure 4.7.3.3‑1: Service request using access token 26](#_Toc184047779)

**List of Tables**

[Table 4.2.3‑1: TLS Client and Server Certificate Profile 13](#_Toc184047780)

[Table 4.7.3.1‑1: Resources and methods overview of the Access Token API 20](#_Toc184047781)

[Table 4.7.3.2.1‑1: Data structures supported by the POST request body on this resource 22](#_Toc184047782)

[Table 4.7.3.2.1-2: Data structures supported by the POST Response Body on this resource. 22](#_Toc184047783)

[Table 4.7.3.2.1-3: Headers supported by the 200 Response Code on this resource. 23](#_Toc184047784)

[Table 4.7.3.2.2.1‑1: Definition of type AccessTokenReq 23](#_Toc184047785)

[Table 4.7.3.2.2.2‑1: Definition of type AccessTokenRsp. 24](#_Toc184047786)

[Table 4.7.3.2.2.3‑1: Definition of type AccessTokenErr 24](#_Toc184047787)

[Table 5‑1: Cryptographic Operations for O-RAN 28](#_Toc184047788)

# Foreword

This Technical Specification (TS) has been produced by WG11 of the O-RAN Alliance.

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

version xx.yy.zz

where:

xx: the first digit-group is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc. (the initial approved document will have xx=01). Always 2 digits with leading zero if needed.

yy: the second digit-group is incremented when editorial only changes have been incorporated in the document. Always 2 digits with leading zero if needed.

zz: the third digit-group included only in working versions of the document indicating incremental changes during the editing process. External versions never include the third digit-group. Always 2 digits with leading zero if needed.

# 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 O-RAN Drafting Rules (Verbal forms for the expression of provisions).

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

# Scope

The present document specifies security protocols as to be used for O-RAN compliant implementation.

# References

## Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non‑specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, O-RAN cannot guarantee their long-term validity.

The following referenced documents are necessary for the application of the present document.

1. Void.

1. [IETF RFC 4252](https://www.rfc-editor.org/info/rfc4252): "The Secure Shell (SSH) Authentication Protocol".
2. [OpenSSH](https://www.openssh.com/specs.html): "OpenSSH Specifications".
3. Void.

1. [IANA](https://www.iana.org/assignments/ssh-parameters/ssh-parameters.xhtml): "Secure Shell (SSH) Protocol Parameters".
2. O-RAN ALLIANCE TS: "O-RAN Management Plane Specification".
3. Void.

1. [IANA](https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml): "Transport Layer Security (TLS) Parameters".
2. Void.
3. O-RAN ALLIANCE TS: "O-RAN Operations and Maintenance Interface Specification" .
4. [IETF RFC 6668](https://www.rfc-editor.org/info/rfc6668): "SHA-2 Data Integrity Verification for the Secure Shell (SSH) Transport Layer Protocol".
5. [IETF RFC 8268](https://www.rfc-editor.org/info/rfc8268): "More Modular Exponentiation (MODP) Diffie-Hellman (DH) Key Exchange (KEX) Groups for Secure Shell (SSH)".
6. [IETF RFC 8308](https://www.rfc-editor.org/info/rfc8308): "Extension Negotiation in the Secure Shell (SSH) Protocol".
7. [IETF RFC 8332](https://datatracker.ietf.org/doc/html/rfc8332): "Use of RSA Keys with SHA-256 and SHA-512 in the Secure Shell (SSH) Protocol".
8. [IETF RFC 8709](https://www.rfc-editor.org/info/rfc8709): "Ed25519 and Ed448 Public Key Algorithms for the Secure Shell (SSH) Protocol".
9. Void.
10. Void.
11. [IETF RFC 8446](https://www.rfc-editor.org/info/rfc8446): "The Transport Layer Security Protocol (TLS) v1.3".
12. Void.
13. 3GPP TS 33.210: "Network Domain Security (NDS); IP network layer security".
14. Void.
15. [IETF RFC 6066](https://www.rfc-editor.org/info/rfc6066): "Transport Layer Security (TLS) Extensions: Extension Definitions".
16. Void.
17. [IETF RFC 6083](https://www.rfc-editor.org/info/rfc6083): "Datagram Transport Layer Security (DTLS) for Stream Control Transmission Protocol (SCTP)".
18. Void.
19. 3GPP TS 33.310: Network Domain Security (NDS); Authentication Framework (AF)”.
20. [IETF RFC 4301](https://www.rfc-editor.org/info/rfc4301): "Security Architecture for the Internet Protocol".
21. 3GPP TS 33.401: "3GPP System Architecture Evolution (SAE); Security architecture".
22. 3GPP TS 33.501: "Security architecture and procedures for 5G system".
23. Void.
24. [IETF RFC 4303](https://www.rfc-editor.org/info/rfc4303): "IP Encapsulating Security Payload (ESP)".
25. Void.
26. [IETF RFC 4306](https://www.rfc-editor.org/info/rfc4306): "Internet Key Exchange (IKEv2) Protocol".
27. [IETF RFC 7296](https://www.rfc-editor.org/info/rfc7296): "Internet Key Exchange Protocol Version 2 (IKEv2)".
28. [IETF RFC 6749](https://www.rfc-editor.org/info/rfc6749): "The OAuth 2.0 Authorization framework".
29. [IETF RFC 7519](https://www.rfc-editor.org/info/rfc7519): "JSON Web Token (JWT)".
30. [IETF RFC 7515](https://www.rfc-editor.org/info/rfc7515): "JSON Web Signature (JWS)".
31. [IETF RFC 4210](https://www.rfc-editor.org/info/rfc4210): "Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)".
32. [IETF RFC 4211](https://www.rfc-editor.org/info/rfc4211): "Internet X.509 Public Key Infrastructure Certificate Request Message Format".
33. [NIST](https://csrc.nist.gov/Projects/Cryptographic-Standards-and-Guidelines): "Cryptographic Standards and Guidelines".
34. Void.
35. [NIST](https://csrc.nist.gov/pubs/fips/186-5/final): "Digital Signature Standard (DSS) (FIPS PUB 186)".
36. Void.

1. [NIST](https://csrc.nist.gov/pubs/fips/197/final): "Advanced Encryption Standard (AES) (FIPS PUB 197)".
2. Void.
3. Void.
4. [NIST](https://csrc.nist.gov/publications/fips#202): "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Function (FIPS PUB 202)".
5. [NIST SP 800-131A Rev. 2](https://csrc.nist.gov/pubs/sp/800/131/a/r2/final): "Transitioning the Use of Cryptographic Algorithms and Key Lengths".
6. Void.
7. [IETF RFC 8017](https://www.rfc-editor.org/info/rfc8017): "PKCS #1: RSA Cryptography Specifications Version 2.2".
8. Void.
9. Void.
10. Void.
11. [IETF RFC 6125](https://www.rfc-editor.org/info/rfc6125): "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)".
12. [IETF RFC 7633](https://www.rfc-editor.org/info/rfc7633): "X.509v3 Transport Layer Security (TLS) Feature Extension".
13. [CA/Browser Forum](https://cabforum.org/uploads/CA-Browser-Forum-BR-1.8.0.pdf): "Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates".
14. Void.
15. Void.
16. Void.
17. [IETF RFC 7093](https://www.rfc-editor.org/info/rfc7093): "Additional Methods for Generating Key Identifiers Values".
18. Void.
19. [IETF RFC 6960](https://www.rfc-editor.org/info/rfc6960): "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP".
20. [IETF RFC 3647](https://www.rfc-editor.org/info/rfc3647): "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework".
21. [IETF RFC 5280](https://www.rfc-editor.org/info/rfc5280): "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile".
22. [SFTP](https://www.sftp.net/specification): "SFTP Standards".
23. [IETF RFC 4217](https://www.rfc-editor.org/info/rfc4217): "Securing FTP with TLS (FTPES)".
24. 3GPP TS 28.532: "Management and orchestration; Generic management services".
25. 3GPP TS 28.537: "Management and orchestration; Management capabilities".
26. [IETF RFC 9110](https://www.rfc-editor.org/info/rfc9110): "HTTP Semantics (HTTPS)".
27. [IETF RFC 4253](https://www.rfc-editor.org/info/rfc4253): "The Secure Shell (SSH) Transport Layer Protocol".
28. [IETF RFC 9325](https://www.rfc-editor.org/info/rfc9325): "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)".
29. [IETF RFC 3279](https://www.rfc-editor.org/info/rfc3279): "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile".
30. ITU‑T Recommendation X.509: "Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks".

1. [IETF RFC 4511](https://www.rfc-editor.org/info/rfc4511): "Lightweight Directory Access Protocol (LDAP): The Protocol".
2. [IETF RFC 4513](https://www.rfc-editor.org/info/rfc4513): ["Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms"](https://www.rfc-editor.org/rfc/rfc4513).
3. [IETF RFC 9147](https://www.rfc-editor.org/info/rfc9147): "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3".
4. 3GPP TS 29.501: "5G System; Principles and Guidelines for Services Definition".
5. [IETF RFC 6750](https://www.rfc-editor.org/info/rfc6750): "The OAuth 2.0 Authorization Framework: Bearer Token Usage".
6. IEEE Std 802.1AE-2018: "IEEE Standard for Local and metropolitan area networks — Media Access Control (MAC) Security".

## Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non‑specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, O-RAN cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

1. 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".
2. Void.
3. Void.
4. [NIST SP 800-52 Rev.2](https://csrc.nist.gov/pubs/sp/800/52/r2/final): "Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations".
5. Void.
6. [BSI TR-02102-1](https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.pdf?__blob=publicationFile&v=7): "Cryptographic Mechanisms: Recommendations and Key Lengths".
7. [NIST SP 800-38A](https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf): "Recommendation for Block Cipher Modes of Operation: Methods and Techniques".
8. [NIST SP 800-186](https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-186.pdf): "Recommendations for Discrete Logarithm-based Cryptography: Elliptic Curve Domain Parameters".
9. [NIST SP 800-57 Part 1 Rev. 5](https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf): "Recommendation for Key Management: Part 1 – General".
10. Void.
11. [IETF RFC-7589](https://www.rfc-editor.org/info/rfc7589): "Using the NETCONF Protocol over Transport Layer Security (TLS) with Mutual X.509 Authentication".

# Definition of terms, symbols and abbreviations

## Terms

Void

## Symbols

Void

## Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [i.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 [i.1].

CA Certification Authority

CMP Certificate Management Protocol

DTLS Datagram Transport Layer Security

DH Diffie–Hellman

DPD Dead Peer Detection

ESP Encapsulating Security Payload

FTPES Explicit SSL File Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

IKE Internet Key Exchange

IPsec Internet Protocol Security

NAT Network Address Translation

NE Network Element

NETCONF Network Configuration Protocol

PKI Public Key Infrastructure

RA Registration Authority

SA Security Association

SFTP SSH File Transfer Protocol

SPD Security Policy Database

SSH Secure Shell

TLS Transport Layer Security

# Security protocols specifications for O-RAN compliant implementation

## SSH

### General requirements

O-RAN interfaces that implement authentication, confidentiality and integrity using SSH shall:

* support SSHv2 [2] and [70];
* disable by default cryptographically insecure ciphers as specified in clauses 4.1.2.1, 4.1.2.3, and 4.1.2.4;
* enable an O-RAN deployer to configure SSH to offer less secure ciphers using standard SSH configurations to enable backward compatibility with older SSH implementations;
* enable remote shell access only if required by the interface. If remote shell access is enabled, disable remote login as root or equivalent highest privileged user. Such access shall be limited to the local system console only. Root user shall not be allowed to login to the system remotely.

Entities providing O-RAN components that support SSH for authentication, confidentiality or integrity shall:

* stay current with SSH [5];
* provide an upgrade path for changes to the SSH protocol and ciphers [5].

### Required ciphers

O-RAN requires the following ciphers when using SSH. For more information see [2], [5], [11], [12], [13], [14] and [15]. See Security clause 5.4 of the O-RAN Working Group 4 Management Plane Specification [6] for the M-plane mandated SSH ciphers.

#### Key agreement

Note that this document uses the IANA cipher naming convention [5].

* Required
  1. ecdsa-sha2-nistp256
  2. ecdsa-sha2-nistp384
  3. ecdsa-sha2-nistp521
  4. ssh-ed25519
* Optional
  1. ssh-ed448
* Shall not be implemented
  + ssh-rsa
  + ssh-dss

#### Symmetric algorithms for encrypting transferred data

* Required
  1. aes256-gcm
  2. aes128-gcm
  3. aes256-ctr
  4. aes192-ctr
  5. aes128-ctr

#### Key exchange algorithms (KexAlgorithms)

* Required
  1. ecdh-sha2-nistp521
  2. ecdh-sha2-nistp384
  3. ecdh-sha2-nistp256
  4. diffie-hellman-group-exchange-sha256
  5. curve25519-sha256
* Shall not be implemented
  1. Diffie-hellman-group1-sha1

#### Message authentication codes (MACs)

* Required
  1. hmac-sha2-512-etm
  2. hmac-sha2-512
  3. hmac-sha2-256-etm
  4. hmac-sha2-256
  5. umac-128
* Shall not be implemented
  1. hmac-sha1

## TLS

### General Requirements

O-RAN interfaces that implement one-way authentication, mutual authentication, confidentiality, integrity, and replay protection using Transport Layer Security (TLS) shall be compliant with [18], allow negotiation of both TLS 1.2 [18] and TLS 1.3 [18], and not allow negotiation to any version of SSL or any version of TLS below 1.2. Further, O-RAN interfaces using TLS may allow TLS 1.2 to be disabled by configuration ([i.4], section 3.1).

NOTE: In one-way TLS the server sends a Certificate message, a CertificateVerify message, and a Finished message to the client so that the client can authenticate the server by validating the server’s certificate and possession of the private key associated with the certificate. In mutual TLS (mTLS) the server sends a CertificateRequest message indicating that the client must authenticate itself with a certificate. The server then sends a Certificate message, a CertificateVerify message, and a Finished message to the client. Upon receiving the server's messages, the client responds with its Authentication messages, namely Certificate and CertificateVerify, and Finished. For full details of the authentication steps in TLS see [18].

See clause 5.4 of the O-RAN Working Group 4 Management Plane Specification for the Open Fronthaul M-plane mandated TLS requirements [6].

#### Void

Void

### TLS Protocol Profile

TLS 1.2 used on all interfaces except the Open Fronthaul interfaces shall support the TLS 1.2 profiles as defined by 3GPP TS 33.210 [20] clause 6.2.3.

TLS 1.3 used on all interfaces except the Open Fronthaul interfaces shall support the TLS 1.3 profiles as defined by 3GPP TS 33.210 [20] clause 6.2.2.

Use of a cipher suite in TLS shall be configurable.

Additional cipher suites recommended by IANA [52] may be supported.

### TLS Certificate Profile

The present clause addresses the certificate profile requirements for the TLS entities that may behave either as client, server, or both. The structure of the TLS certificate profile described in this clause under Table 4.2.3‑1 provides requirements based on the 3GPP – TS 33.310 [26], CA-Browser Forum [56] and NIST Special Publication 800-52.2 [i.4]. The TLS entity certificate profile shall be applied to all nodes and interfaces that use the TLS protocol for secured communication in the O-RAN network except the Open Fronthaul interfaces.

* The TLS entity certificates shall adhere to the certificate profile outlined in the Table 4.2.3‑1, based on their respective roles.
* The CA responsible for handling the certificate signing request shall ensure compliance in accordance with the certificate profile specified in clause 4.2.3.
* The common rules for all certificates defined in 3GPP- TS 33.310 [26], clause 6.1.1 and 6.1.3a shall apply.
* The CA-Browser Forum [56] has the following requirement for the certificate operational and key usage periods. Certificates issued on or after 1 September 2020 shall not have a Validity Period greater than 397 days.
* Subject: The subject field follows the X.501 type Name as defined in clause 4.1.2.6 of IETF RFC 5280 [64].
* Certificate Policies**:** Certificate policies shall be crafted using the guidelines defined in IETF RFC 3647 [63].
* subjectAltName shall (for TLS server certificates) contain at least one FQDN(Hostname) and shall not contain only IP Address as described in IETF RFC 6066 [22]. As per IETF RFC 6125 [54], an application service shall be identified by a name or names carried in the subject field (i.e., a CN-ID) and/or in one of the following identifier types within subjectAltName entries (DNS-ID, SRV-ID. URI-ID).
* Optional and non-critical TLS Feature Extension: IETF RFC 7633 [55] defines a certificate extension that indicates that clients expect stapled OCSP responses with a value of “status\_request (5)” for the certificate and aborts the handshake ("hard-fail") if such a response is not available. As per IETF RFC 9325 [71] TLS servers should support the following as a best practice: OCSP IETF RFC 6960 [62] and OCSP stapling using the status\_request extension defined in IETF RFC 6066 [22]. The exact mechanism for embedding the status\_request extension differs between TLS 1.2 and TLS 1.3. As a matter of local policy, server operators may request that CAs issue must-staple IETF RFC 7633 [55] certificates for the server and/or for client authentication.
* Extensions: Mandatory Critical Key Usage
  + digitalSignature for both TLS client and Server certificates. This applies for RSA signature certificate, ECDSA signature certificate, or DSA signature certificate.
  + When using RSA IETF RFC 3279 [72] with TLS 1.2, the keyEncipherment shall be set.
  + When using DH [72] or ECDH [72] with TLS 1.2, the keyAgreement shall be set.

As per the certificate profile requirements described in clause 4.2.3 and guidelines outlined in IETF RFC 5280 [64], the following table captures the certificate profile for the TLS entity that may behave as a client, server or both.

Table 4.2.3‑1: TLS Client and Server Certificate Profile

|  |  |  |  |
| --- | --- | --- | --- |
| **ORAN TLS Client and Server Certificate Profile** | | | |
| Version | | v3 | |
| Serial Number | | Unique Positive Integer in the context of the issuing CA | |
| Subject | | EXAMPLE: (C=<Country>), O=<Organization Name>, CN=<Common Name such as server domain name>. | |
|  | |
| Validity Period | | 1 year or less | |
| Signature | | See 3GPP TS 33.310 [26], clause 6.1.1 for the list of supported signature algorithms. | |
| Subject Public Key Info | | See 3GPP TS 33.310 [26], clause 6.1.1 for the list of supported public key types. | |
| **Extensions** | **Mandatory** | **Criticality** | **Value** |
| keyUsage | TRUE | TRUE | digitalSignature for TLS clients and servers |
| extendedKeyUsage | FALSE | FALSE | id-kp-clientAuth TLS clients |  |
| id-kp-serverAuth for TLS servers |  |
| Entities that may be both client and server will have both OIDs set. |  |
| certificatePolicies | FALSE | FALSE | If added, then, it should be populated with a CP object identifier(OID). These OIDs correspond to specific policies established by the certificate issuer |  |
| authorityKeyIdentifier | FALSE | FALSE | This is same as the subjectKeyIdentifier of the Issuer’s certificate. CA utilizes the method as defined in clause 2 of IETF RFC 7093 [60]. |  |
| subjectKeyIdentifier | FALSE | FALSE | This is calculated by the issuing CA utilizing method as defined in clause 2 of IETF RFC 7093 [60]. |  |
| cRLDistributionPoint | TRUE | FALSE | distributionPoint |  |
| According to IETF RFC 5280 [64] this indicates if the CRL is available for retrieval using access protocol and location with HTTP URI or LDAP. |  |
| subjectAltName | TRUE | TRUE | Multiple subjectAltNames are permitted as defined above |  |
| authorityInfoAccess | FALSE | FALSE | id-ad-caIssuers |  |
| According to IETF RFC 5280 [64] id-ad-caIssuers describes the referenced description server and the access protocol and location, for example, using one or multiple HTTP and/or LDAP URIs. The referenced CA issuers description is intended to aid certificate users in the selection of a certification path that terminates at a point trusted by the certificate user |  |
| id-ad-ocsp |  |
| According to IETF RFC 5280 [64] id-ad-ocsp defines the location of the OCSP responder using HTTP URI. |  |
| TLS feature extension | FALSE | FALSE | id-pe-tlsfeature: "status\_request(5)" |  |
| This can be used according to IETF RFC 7633 [55] to prevent downgrade attacks that are not otherwise prevented by the TLS protocol; IETF RFC 7633[55] also defines a certificate extension that indicates that clients expects stapled OCSP responses with a value of "status\_request(5) " for the certificate and aborts the handshake ("hard-fail") if such a response is not available. |  |

## Support NETCONF over secure Transport

TLS requirements for use with NETCONF on the O1 interface [10] are specified in clause 4.2.

TLS requirements for use with NETCONF on the Open Fronthaul M-plane are specified in clause 5.4 of the O-RAN WG4 Management Plane Specification [6].

## DTLS

### General requirements

O-RAN interfaces that implement mutual authentication, integrity protection, replay protection and confidentiality protection using Datagram Transport Layer Security (DTLS) shall:

* support DTLS 1.2 [76];
* support DTLS for Stream Control Transmission Protocol [24];

and should support DTLS 1.3 [76].

DTLS 1.0 shall not be supported.

NOTE 1: IETF RFC 6083 [24] specifies a user message limit of approximately 16k bytes, which shall be compared to the unlimited user message size that exists when SCTP is used without DTLS. There could be applications messages exceeding the limit, preventing the use of DTLS over SCTP, so enforcing unsecured SCTP.

### DTLS Protocol Profile

For DTLS 1.2, all TLS 1.2 related provisions in clause 4.2.2 shall apply.

For DTLS 1.3, all TLS 1.3 related provisions in clause 4.2.2 shall apply.

### Certificate Profile

Certificate profile defined in clause 4.2.3 shall apply.

## IPSec

### Overview

O-RAN and 3GPP interfaces that implement authentication, confidentiality and integrity using IPsec shall support IPsec according to [27]. The supported IPsec capabilities in this clause follow [20] for interworking purposes and further apply requirements given in [26], [28], and [29].

#### Supported IPsec capabilities

Services that SHALL be supported (see also [29], section 9.8.2):

* Confidentiality, can be enabled/disabled (via ENCR\_NULL)
* integrity protection, always enabled
* data origin authentication, always enabled
* anti-replay protection, can be enabled/disabled
* extended sequence numbers, can be enabled/disabled

Protocol that SHALL be supported: ESP ([31], as profiled by [20])

* IPsec mode: Tunnel mode
* Copying the DSCP value from the inner IP-header to the outer IP-header for encrypted packets
* for encrypted packets the DSCP value of the inner packet will be copied to the outer packet
* NAT traversal

ESP encryption transforms that SHALL be supported (including authenticated encryption transforms, see also [20], section 5.3.3). ESP authentication transforms that SHALL be supported according to [20], subsection 5.3.4.

* IKE endpoint Identification that SHALL be supported, according to [26], subsection 6.2.1b: IP address
* Fully Qualified domain name (FQDN) (if DNS is supported

Authentication that SHALL be supported:

* X.509v3 digital certificates provided by a Certificate Authority solution

Authentication that MAY be supported:

* Pre-shared Keys

NOTE: Pre-shared keys should not be used.

Key exchange that SHALL be supported:

* IKEv2 [33] profile as described in [26]
* certificate-based authentication according to [26]
* certificates according to the profile described by [26]
* DH group 19
* (Optional) DH group 20

Security Association multiplicity that SHALL be supported:

* multiple IKE SAs (multiple IPsec tunnels)
* multiple IPsec SAs
* multiple IPsec SAs per IPsec tunnel

IKE SA(s) and IPSEC SA(s) SHALL be regularly re-established.

* When the full sequence number space of an IPSEC SA(s) is used, the transmission for that SA SHALL stop.
* 4.5.1.1.10 Dead-Peer-Detection (DPD) SHALL be supported as defined in [34].Each node SHALL support a traffic narrowing function for the traffic selector ([34]): If the traffic selector notified from the peer (e.g. Security GW or neighbouring node) is wider than the Local/Remote Address range in the SPD information set on the node side, the peers set the traffic selector values to the narrower range.

### Parallel usage of IPsec and other secure transport protocols

Implementations SHALL support IPSec configuration with one or more dedicated connections in parallel with other secure transport protocols.

EXAMPLE: It should be possible to run SSH or TLS connections within an IPsec tunnel or in parallel to IPsec tunnel(s).

### Responder mode and Initiator/Responder mode support

Implementations SHALL support a configurable option per IKE Security Association to use "Responder mode" instead of "Responder/Initiator mode".

Responder mode is applied to IKE SA establishment only. The introduction of "Responder mode" does not change the IKE SA rekeying behavior: Each node shall be able to operate as initiator and responder in IKE\_SA rekeying.

## CMPv2

Certificate Management Protocol version 2 (CMPv2) is based on Internet X.509 Public Key Infrastructure (PKI) Certificate Management Protocol (CMP) specified in IETF RFC 4210 [38] and 4211 [39].

3GPP TS 33.310 [26] specifies the use of CMPv2 used by base stations to obtain an operator-signed certificate using a secured communication based on the vendor-signed certificate in the base station and a vendor root certificate pre-installed in the CA/RA server.

Certificates may be installed at initial system initialization or obtained through certificate enrolment with the operator’s PKI. Certificate enrolment may be supported with the CMPv2 protocol between the Network Element (NE) and the operator’s CA as defined in 3GPP TS 33.310 [26]. A PNF that supports CMPv2 shall use the CMPv2 profile defined in 3GPP TS 33.310 [26], clause 9.5. The CA/RA server may include the trust anchor for the operator issued certificate and the appropriate certificate chains in the initialization response message.

## OAuth 2.0

### Overview

The authorization framework described in clause 4.7 of the present document uses the OAuth 2.0 framework as specified in IETF RFC 6749 [35]. It allows the service producers to authorize the requests from service consumers, and the service consumers to obtain authorization credentials (“token endpoint”).

Interfaces requiring the use of OAuth 2.0 for authorization purposes shall support the functionality according to clause 4.7 of the present document.

Following options have been introduced to guarantee interoperability and align with the existing OAuth 2.0 authorization framework in 3GPP [29] (i.e., Service Based Architecture).

### Basic Parametrization

#### General

Client access token authorization grants shall be supported with type Client Credentials Grant, as described in 4.4 of IETF RFC 6749 [35]. Mutual TLS authentication, as specified in the present document (clause 4.2.1), is the mechanism selected by O-RAN for this security procedure.

In addition, still aligned with Client Credential Grant as described in section 4.4. of IETF RFC 6749 [35], Client Id and Client Secret may be supported for client authentication. In this case, one-way TLS may be used (certificate on server side).

Access tokens shall be JSON Web Tokens (JWT) as described in IETF RFC 7519 [36] and shall be secured with digital signatures based on JSON Web Signature (JWS) as described in IETF RFC 7515 [37].

The ‘scope’ attribute as described in clause 3.3 of IETF RFC 6749 [35] shall be used to specify the allowed services per type of service producer. A more granular level of authorization may be defined by adding additional scope information with the token (e.g., to authorize specific operations, or access to particular resources or datasets), which requires to be verified by the service producer.

OAuth 2.0 roles, as defined in clause 1.1 of IETF RFC 6749 [35], are as follows:

* OAuth 2.0 Authorization server: new security function in O-RAN architecture; or provided by existing OAuth 2.0 infrastructure in the network.
* OAuth 2.0 client: every API service consumer in O-RAN system (e.g., rApp, xApps)
* OAuth 2.0 resource owner/server: every API service producer in O-RAN system (e.g., Near-RT RIC platform)

#### Registration process

OAuth 2.0 resource servers (API service producers) shall be registered with the OAuth 2.0 Authorization server. Service producers may include additional scope information related to specific allowed service operations and resources per client type.

Before initiating the authorization protocol, OAuth 2.0 clients shall be registered with the OAuth 2.0 Authorization server. To achieve that, information about the service consumer instance and its type shall be made available in the OAuth 2.0 Authorization server. The registration process is subject to implementation procedures of the operator, with the following consideration on authentication procedure:

OAuth 2.0 clients shall be capable to authenticate securely with the authorization server and client type shall be “confidential” as specified in clause 2.1 of IETF RFC 6749 [35].

Strong authentication mechanisms based on digital certificates shall be supported.

In addition, client authentication mechanism based on client Id and Client Secret may be supported.

#### Access Token request process

##### Introduction

After TLS mutual authentication procedure between the OAuth 2.0 client and OAuth 2.0 Authorization server, or one-way TLS authentication with server certificate and client authentication based on Client Id and Client Secret has been executed, the Authorization server exposes a ‘Token Endpoint’ where the Access Token request service can be requested by OAuth 2.0 client.

The following procedure depicts the procedure of the OAuth 2.0 client to obtain an access token from the OAuth 2.0 authorization server.



Figure 4.7.2.3‑1: Access Token request process

The OAuth 2.0 Client shall request an access token from the OAuth 2.0 Authorization server. For this operation the client shall send a HTTP POST request to the authorization server (‘Token endpoint’), as described in IETF RFC 6749 [35], clause 3.2. The message, i.e., the body of the HTTP POST request, shall include the identifier of the client instance making the request, the ‘scope’ parameter indicating the expected services, and optionally additional scope information with more granular information about resources and operations on the resources, and optionally the type of client (‘confidential’) and the expected OAuth 2.0 resource owner/server. The message may include as well information referring to instance(s) of specific resource owner(s)/server(s) if required.

##### Server requirements

* The OAuth 2.0 Authorization server shall verify that the input parameters in the access token (e.g., client type) match with the corresponding ones in the public certificate of the client, or those in the client profile previously registered.
* If the client is authorized, the Authorization server shall generate an access token with appropriate scope as defined in clause 3.3 of IETF RFC6749[35] included.
* The Authorization server shall digitally sign the generated access token based on a private key as described in IETF RFC 7515 [37]. If the client is not authorized, the Authorization server shall not issue an access token to that client. If the authorization is successful, the Authorization server shall send the access token to the client (‘200 OK’), otherwise it shall reply based on OAuth 2.0 error response defined in IETF RFC 6749 [35].
* The success response should include in addition the expiration time for the token as indicated in IETF RFC 6749 [35].
* The response shall include information to identify the resource owner(s)/server(s) if they differ from the related information included in the token request.

#### Service access request based on token verification

Once the client is in possession of a valid access token, it may proceed with the request of service access towards the service producer (OAuth 2.0 resource owner/server).



Figure 4.7.2.4‑1: Service access request based on token verification

1. The OAuth 2.0 client requests service from the OAuth 2.0 resource owner/server. The OAuth 2.0 client shall include the access token in the request. The OAuth 2.0 client and OAuth 2.0 resource owner/server shall authenticate each other via mutual TLS, as defined in the present document (clause 4.2.1) In addition, one-way TLS authentication with certificate only on server side may be supported, as defined in the clause 4.2.1.
2. The OAuth 2.0 resource owner/server shall verify the token:
   1. It ensures the integrity of the token by verifying the signature using the OAuth 2.0 Authorization server’s public key.
   2. If the integrity check is successful, the OAuth 2.0 resource owner/server shall verify the claims.
3. If the verification is successful, the OAuth 2.0 resource owner/server shall execute the requested service and respond back to the OAuth 2.0 client. Otherwise, it shall send a proper error response. If the HTTP protocol is used, that error response shall be based on the OAuth 2.0 error response defined in IETF RFC 6749 [35].

### OAuth2.0 Access token API for O-RAN Elements and interfaces

#### Introduction

This API allows the API Consumer to request an access token based on the procedures for "Access Token request process" as defined in clause 4.7.2.3.

The API version number format shall follow the definitions in clause 4.3.1.1 of 3GPP TS 29.501 [77]. The MAJOR version field shall be 1, the MINOR version field shall be 0 and the PATCH version field shall be 0. Consequently, the <apiVersion> URI path segment shall be set to "v1".

The API is under development and consequently the API version shall include the pre-release version "alpha.1".

For the general token endpoint for O-RAN RESTful service APIs, the URI structure should follow the provisions specified below, with the <apiName> resource URI variable set to "oauth2".

The resources and methods overview of the access token API are addressed in Table 4.7.3.1‑1 and shown in Figure 4.7.3.1‑1.

Table 4.7.3.1‑1: Resources and methods overview of the Access Token API

|  |  |  |  |
| --- | --- | --- | --- |
| Resource Name | Resource URI | HTTP method | Service Operation |
| Access Token | …/token | POST | Access token request |

{apiRoot}/oauth2/<apiVersion>

/token

Figure 4.7.3.1‑1: Resource structure for access token

#### Service operations for access token request

The service operation is defined in IETF RFC 6749 [35] and represented in Figure 4.7.3.2-1.

In the present document, only the use of the "Client credentials grant type" (see clauses 1.3.4 and 4.4 of IETF RFC 6749 [35]) is supported.

@startuml

!pragma teoz true

skinparam ParticipantPadding 5

skinparam BoxPadding 10

skinparam defaultFontSize 12

skinparam lifelineStrategy solid

autonumber

participant "API Consumer" as AC

participant " Authorization Server " as AS

AC->AS: POST .../token (AccessTokenReq)

AS->AS: validate-credentials

alt Success

AS->AC: 200 Ok (AccessTokenRes)

else Failure

AS->AC: 400 Bad Request (AccessTokenErr)

end

Ein Bild, das Text, Screenshot, Schrift, Diagramm enthält.

Automatisch generierte Beschreibung

Figure 4.7.3.2‑1: Access Token Request and Response

1. The API Consumer sends an HTTP POST request to Authorization Server to request the access token from the Token Endpoint.
2. The Authorization Server authenticates the client as specified in clause 4.4.2 of IETF RFC 6749 [35] and validates the request as implied in clause 4.4.3 of IETF RFC 6749 [35].
3. On success, the Authorization Server returns a “200 OK” HTTP response along with an access token response structure.
4. On failure, the Authorization Server returns a “400 Bad Request” HTTP response and the message content contains additional error information.

##### Resource Standard Methods

The POST method and its request and response parameters are specified in IETF RFC 6749 [35].

This information is reproduced using the O-RAN scheme for RESTful API definitions below, with the request data structure represented in Table 4.7.3.2.1-1, the response data structures and response codes represented in Table 4.7.3.2.1-2 and the HTTP headers represented in Table 4.7.3.2.1-3.

Table 4.7.3.2.1‑1: Data structures supported by the POST request body on this resource

|  |  |  |  |
| --- | --- | --- | --- |
| Data Type | P | Cardinality | Description |
| AccessTokenReq | M | 1 | As content specified in IETF RFC 6749 [35]. See also clause 4.7.3.1.2.1. The content type is " application/x-www-form-urlencoded". |

Table 4.7.3.2.1-2: Data structures supported by the POST Response Body on this resource.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Type | P | Cardinality | Response codes | Description |
| AccessTokenRsp | M | 1 | 200 OK | As content specified in IETF RFC 6749 [35]. See also clause 4.7.3.1.2.2.  The content type is "application/json". |
| AccessTokenErr | M | 1 | 400 Bad request | As content specified in IETF RFC 6749 [35]. See also clause 4.7.3.1.2.3. The content type is "application/json". |

Table 4.7.3.2.1-3: Headers supported by the 200 Response Code on this resource.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| Cache-control | string | M | 1 | As content specified in IETF RFC 6749 [35]. |
| Pragma | string | M | 1 | As content specified in IETF RFC 6749 [35]. |

##### Data Model

The following clauses represent the message content to be used by the OAuth2.0 access token API as O-RAN data types in line with the provisions in IETF RFC 6749 [35].

###### Data type: AccessTokenReq

The AccessTokenReq data type represents an access token request to the Authorization Server. Its attributes are defined by IETF RFC 6749 [35] and reproduced in Table 4.7.3.2.2.1-1.

Table 4.7.3.2.2.1‑1: Definition of type AccessTokenReq

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute Name | Data type | P | Cardinality | Description |
| grant\_type | string | M | 1 | As content specified in clause 4.4.2 of IETF RFC 6749 [35].  In the present document, the only allowed value shall be "client\_credentials”. |
| client\_id | string | C | 0..1 | As content specified in IETF RFC 6749 [35]. See NOTE. |
| client\_secret | string | C | 0..1 | As content specified in IETF RFC 6749 [35]. See NOTE. |
| scope | string | O | 0..1 | As content specified in in clause 4.4.2 of IETF RFC 6749 [35]. |
| NOTE: As specified in clause 2.1 of IETF RFC 6749 [35], there are two options for authenticating the clients to the Authorization Server: HTTP basic authentication and access token request message content. The attributes "client\_id" and "client\_secret" shall be absent if HTTP basic authentication is used to convey the client credentials as per section 2.3.1 of IETF RFC6749 [35] and shall be present otherwise. | | | | |

###### Data type: AccessTokenRsp

The AccessTokenRsp data type represents an access token response from the Authorization Server. Its attributes are defined by IETF RFC 6749 [35] and reproduced in table 4.7.3.2.2.2-1.

Table 4.7.3.2.2.2‑1: Definition of type AccessTokenRsp.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute Name | Data type | P | Cardinality | Description |
| access\_token | string | M | 1 | As content specified in clause 5.1 of IETF RFC 6749 [35]. |
| token\_type | string | M | 1 | As content specified in clause 5.1 of IETF RFC 6749 [35]. |
| expires\_in | integer | O | 0..1 | As content specified in clause 5.1 of IETF RFC 6749 [35]. |
| scope | string | O | 0..1 | As content specified in clause 5.1 of IETF RFC 6749 [35]. |

###### Data type: AccessTokenErr

The AccessTokenErr data type represents an access token error response from the Authorization Server. Its attributes are defined by IETF RFC 6749 [35] and reproduced in table 4.7.3.2.2.3-1.

Table 4.7.3.2.2.3‑1: Definition of type AccessTokenErr

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute Name | Data type | P | Cardinality | Description |
| error | string | M | 1 | As content specified in clause 5.2 of IETF RFC 6749 [35], the following values are valid: "invalid\_request", "invalid\_client, invalid\_grant", "unauthorized\_client", "unsupported\_grant\_type", "invalid\_scope." |
| error\_description | string | O | 0..1 | As content specified in clause 5.2 of IETF RFC 6749 [35]. |
| error\_uri | string | O | 0..1 | As content specified in clause 5.2 of IETF RFC 6749 [35]. |

#### Service operations for Service access request

The service operation is as follows:

@startuml

!pragma teoz true

skinparam ParticipantPadding 5

skinparam BoxPadding 10

skinparam defaultFontSize 12

skinparam lifelineStrategy solid

autonumber

participant "API Consumer" as AC

participant " API Producer " as AP

AC->AP: GET.../(access\_token)

AP->AP: validate(access\_token)

alt Authorized

AP->AC: 200 Ok

else Access token invalid

AP->AC : 401 Unauthorized

else Insufficent scope

AP->AC : 403 Forbidden

else Other error response

AP-> AC: 4xx/5xx Error response

end

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Figure 4.7.3.3‑1: Service request using access token

1. The API Consumer shall access the protected resources by using access token for the service API being invoked as shown in Figure 4.7.3.2-1. The access token shall be included in the request as an HTTP header field as specified in clause 2.1 of IETF RFC 6750 [78].
2. The API Producer shall validate the access token and ensure that it has not expired and that its scope covers the requested resource.
3. If the token validation is successful, the API Producer shall allow access for the actual resource with the actual request and its parameters. The API Producer shall return the HTTP response as defined for the service API being invoked.
4. If the token validation fails due to invalid token or expired token, the API Producer shall return "401 Unauthorized" in line with the provisions in clause 3 of IETF RFC 6750 [78].
5. If the token validation is failed due to insufficient access rights to access the resource, the API Producer shall return “403 Forbidden" in line with the provisions in clause 3 of IETF RFC 6750 [78].
6. In case of any other error, the API Producer shall return the appropriate error as defined by the service API.

## LDAP

### General requirements

LDAP, or the Lightweight Directory Access Protocol, is a widely adopted, unbiased standardized communication protocol that enables the interaction and management of distributed directory data via an IP network. This protocol plays a crucial part in the creation of both intranet and internet applications, as it facilitates the sharing of detailed information about various resources, such as users, systems, networks, services, and applications, across the entire network.

O-RAN interfaces that implement Lightweight Directory Access Protocol (LDAP) shall:

* support LDAP version 3 as specified in [74] and LDAP with StartTLS as specified in[75].
* not support previous versions of LDAP.

StartTLS operation shall be initiated prior to any other LDAP operation and all LDAP data shall be sent only after successful establishment of TLS connection. Requirements for TLS in clause 4.2 shall apply.

## MACsec

### General requirements

The O-RAN interfaces that implement authentication, confidentiality and integrity using MACsec shall support MACsec capabilities according to **IEEE Std 802.1AE-2018** [79].

### MACsec Profile

The O-RAN interfaces using MACsec shall support the mandatory cipher suites as specified in IEEE Std 802.1AE-2018 [79] clause 14, Table 14-1—MACsec Cipher Suites.

# Cryptographic operations

The following table outlines the main cryptographic operations involved in the protection of (1) App/VNF/CNF packages for ensuring their integrity, authenticity and confidentiality (for sensitive artifacts) during delivery, onboarding, and instantiation phases, (2) the communication channel over O-RAN interfaces in terms of authenticity, confidentiality and integrity, and (3) the stored assets within O-RAN system. It contains the allowed list of algorithms, key sizes and standards according to BSI [i.6] and NIST [40], [48], [i.9] cryptographic guidelines.

Table 5‑1: Cryptographic Operations for O-RAN

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cryptographic operations | Algorithms | Key sizes | Applicable standards | Usage | Note |
| Signature generation and verification | RSASSA-PSS  RSA\_PKCS1\_V1\_5 | >= 2048-bits | FIPS 186 [42] | For ensuring the integrity and authenticity of Apps/VNF/CNF packages during delivery, onboarding, and instantiation phases | Existing code should use PKCS #1 v1.5 padding mode for compatibility only.  Use of null padding is not recommended. |
| Signature generation and verification | ECDSA NIST-approved curves (P-256, P-384, or P-521) | >= 256-bits | FIPS 186 [42], SP 800-186 [i.7] | Parameter key size for DSA shall not allow the following combination:  Bit lengths of L and N parameters  L = 2048, N = 224 (BSI TR-02102-1) |
| Symmetric Encryption/Decryption | AES\_128, AES\_192 and AES\_256 | 128, 192 and 256 bits | FIPS 197 [44], SP 800-38 [i.8] (operation modes) | For ensuring the confidentiality of sensitive artifacts |  |
| Asymmetric Encryption/Decryption | RSAES-OAEP | >= 2048-bits | IETF RFC 8017 [50] |  |
| Hashing | SHA-2 family (SHA- 224, SHA-256, SHA-384, SHA-512, SHA-512/224 and SHA-512/256)  SHA-3 family (SHA3-224, SHA3-256, SHA3-384, and SHA3-512) |  | FIPS 180 [44] for SHA2  FIPS 202 [47] for SHA 3 | For ensuring the integrity  of Apps/VNF/CNF packages |  |

The signature operation shall involve X.509-based PKI certificates [74].

The following recommendations should be considered:

* For the protection of cryptographic keys, Hardware Security Modules, or HSMs, should be used.
* Along with HSMs, the principle of least privilege should be used with keys, to ensure only users who need the key have access to it.
* In case a legacy system doesn't support ECDSA, RSA signing algorithms should be used instead. Otherwise, the use of the Elliptic curve signing algorithms is recommended.
* If libraries or frameworks are in use that don't support PSS padding scheme, one of the RSA PKCS1 algorithms should be used instead. Otherwise, the use of one of the RSA PSS algorithms is recommended.
* In general, the largest key size available for an algorithm family should be used:
* For RSA, the largest supported key size is 4096 bits.
* For ECDSA, the largest supported key size is 512 bits.
* For AES, the largest supported key size is 256 bits.

# Secure File Transfer protocols

## General

File Transfer management is required in several O-RAN use cases, e.g. for export of log files.

An O-RAN architectural element that implements file management shall support at least one of these secure file transfer protocols: SFTP, FTPES, HTTPS. This is aligned with, for example, how 3GPP specifies File Management in [67] and [68]

SFTP is authenticated with username/password, SSH keys or X.509 certificates. FTPES is authenticated with X.509 certificates. HTTPS is mutually authenticated with X.509 certificates.

## SFTP

### General Requirements

O-RAN architectural elements that implement secure file transfer using SFTP shall:

* Support secure connection and authentication using SSHv2 Authentication Protocol [2] with all O-RAN specific requirements from Clause 4.1 in this document
* Support SFTPv3 as the SFTP Industry Best Practice [65]

## FTPES

### General Requirements

O-RAN architectural elements that implement secure file transfer using FTPES shall:

* Support secure connection and authentication using TLS with all O-RAN specific requirements from Clause 4.2 in this document
* Support FTPES as defined in [66]

## HTTPS

### General Requirements

O-RAN architectural elements that implement secure file transfer using HTTPS shall:

* Support secure connection and authentication using TLS with all O-RAN specific requirements from Clause 4.2 in this document.
* Support HTTPS as defined in [69].

# Annex A (normative): OpenAPI specifications

## A.1 OAuth2.0 API

### A.1.1 Introduction

The Open API for the OAuth2.0 Access Token API specified in clause A.1.2 is based on the API definitions in clause 4.7.3.4.

### A.1.2 OAuth 2.0 Access Token API

openapi: 3.0.3

info:

title: OAuth 2.0 Access Token API

version: 1.0.0

description: |

API for OAuth 2.0.

© 2024, O-RAN ALLIANCE.

All rights reserved.

externalDocs:

description: O-RAN.WG11.SecProtSpecs-R003-v10.00

url: https://www.o-ran.org/specifications

servers:

- url: '{apiRoot}/oauth2/v1'

variables:

apiRoot:

description: apiRoot as defined in in clause 4.4.1 of 3GPP TS 29.501

default: https://example.com

paths:

/token:

post:

description: Token endpoint

operationId: AccessTokenRequest

tags:

- Access Token Request

requestBody:

required: true

content:

application/x-www-formurlencoded:

schema:

$ref: '#/components/schemas/AccessTokenReq'

responses:

'200':

description: Success case 200 OK

content:

application/json:

schema:

$ref: '#/components/schemas/AccessTokenRsp'

headers:

Cache-Control:

$ref: '#/components/headers/cache-control'

Pragma:

$ref: '#/components/headers/pragma'

'400':

description: Error in the Access Token Request

content:

application/json:

schema:

$ref: '#/components/schemas/AccessTokenErr'

components:

schemas:

AccessTokenReq:

description: Contains information related to the access token request

type: object

properties:

grant\_type:

description: Represents the Client Credentials grant type

enum: [ client\_credentials ]

type: string

client\_id:

description: >-

Represents the identity assigned to the client configured in the

authorization server

type: string

client\_secret:

description: >-

Represents the secret assigned to the client configured in the

authorization server

type: string

scope:

description: Represents the scope of the access request

type: string

required:

- grant\_type

AccessTokenRsp:

description: Contains information related to the access token response

type: object

properties:

access\_token:

description: Represents the access token issued by the authorization server

type: string

token\_type:

$ref: '#/components/schemas/token\_type'

expires\_in:

description: Represents the lifetime in seconds of the access token

type: integer

scope:

description: Represents the scope of the access request

type: string

required:

- access\_token

- token\_type

token\_type:

description: As described in clause 6.1.1 of IETF RFC 6750

type: string

enum:

- Bearer

AccessTokenErr:

description: Error returned in the access token response message

type: object

properties:

error:

type: string

enum:

- unauthorized\_client

- unsupported\_grant\_type

- invalid\_request

- invalid\_scope

- invalid\_client

- invalid\_grant

error\_description:

type: string

error\_uri:

type: string

required:

- error

headers:

cache-control:

required: true

schema:

type: string

enum:

- no-store

pragma:

required: true

schema:

type: string

enum:

- no-cache

# Annex (informative): Change history/Change request (history)

|  |  |  |
| --- | --- | --- |
| Date | Revision | Description |
| 2024.11.27 | 11.00 | More editorial fixes  Published as Final version 11.00 |
| 2024.11.27 | 10.04 | Disabled automatic update of styles in the document developer settings |
| 2024.11.27 | 10.03 | Further editorial changes |
| 2024.11.27 | 10.02 | Several minor editorial- and format corrections |
| 2024.11.27 | 10.01 | Added DTAG-0013, ATT.AO-0061, ACN.AO-0004 and ERI-0172 |
| 2024.07 | 10.00 | Published as Final version 10.00 |
| 2024.03 | 09.00 | Certificate Profile for O-RAN TLS entity  Reference Update  Published as Final version 09.00 |
| 2023.11 | 08.00 | Secure file transfer protocols added  SSH Update on no root remote login  ssh-ed448 changed to optional  Reference on SSH added  Published as Final version 08.00 |
| 2023.07 | 07.00 | Editorial alignments  TLS entity certificate profile  CMPv2 profile update  Introduction of one-way TLS authentication for OAuth2.0  Published as Final version 07.00 |
| 2023.03 | 06.00 | Cryptographic operations update  Published as Final version 06.00 |
| 2022.11 | 05.00 | TLS Cipher update  Published as Final version 05.00 |
| 2022.07.20 | 04.00 | Addition of CMPv2  Update of O-Cloud Image protocols  Addition of mTLS  Update of OAuth 2.0  Published as Final version 04.00 |
| 2021.11.08 | 03.00 | Update the O-RAN security protocols and specifications to include mandatory support for TLS 1.3  Published as Final version 03.00 |
| 2021.07.05 | 02.00 | Addition of DTLS and IPsec requirements.  Alignment of TLS 1.2 and TLS 1.3 profiles with 3GPP TS 33.210.  Typographical changes.  Published as Final version 02.00 |
| 2021.04.01 | 01.00 | Initial version of the document with requirements for TLS and SSH.  Published as Final version 01.00 |