

ETSI TS 119 312 V2.1.1 (2026-06)



TECHNICAL SPECIFICATION

Electronic Signatures and Trust Infrastructures (ESI); Cryptographic Suites

ReferenceRTS/ESI-0019312v211

Keywordsdigital signature, security, trust services

ETSI650 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

Important notice

The present document can be downloaded from the
[ETSI Search & Browse Standards](#) application.

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format on [ETSI deliver](#) repository.

Users should be aware that the present document may be revised or have its status changed,
this information is available in the [Milestones listing](#).

If you find errors in the present document, please send your comments to
the relevant service listed under [Committee Support Staff](#).

If you find a security vulnerability in the present document, please report it through our
[Coordinated Vulnerability Disclosure \(CVD\)](#) program.

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.

No recommendation as to products and services or vendors is made or should be implied.

No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.

In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2026.
All rights reserved.

Contents

Intellectual Property Rights	6
Foreword.....	6
Modal verbs terminology.....	6
Introduction	6
1 Scope	7
2 References	7
2.1 Normative references	7
2.2 Informative references.....	9
3 Definition of terms, symbols, abbreviations and notations	10
3.1 Terms.....	10
3.2 Symbols.....	11
3.3 Abbreviations	11
3.4 Notations	12
4 Use of ECGG Agreed Mechanisms and Maintenance of the present document	13
5 Hash functions.....	13
5.1 General	13
5.2 Recommendations for SHA hash functions.....	13
5.2.1 SHA-512/256.....	13
5.2.2 Hash Functions for use in post-quantum contexts	14
6 Signature schemes	14
6.1 Introduction	14
6.2 Signature algorithms.....	14
6.2.1 General.....	14
6.2.2 Signature algorithms	15
6.2.2.1 RSA.....	15
6.2.2.2 DSA.....	15
6.2.2.3 EC based DSA algorithms	15
6.2.2.4 Edwards-Curve DSA (EdDSA) algorithms.....	15
6.2.2.5 ML-DSA	16
6.2.2.6 SLH-DSA.....	16
6.2.2.7 LMS and XMSS.....	17
6.3 Key generation	17
6.4 Hybrid Cryptographic Schemes	17
6.4.1 Digital Signature Hybrid Modes.....	17
6.4.2 Recommended Hybrid Suites	17
7 Signature suites	18
7.1 Introduction	18
7.2 General	18
7.3 Signature suites	18
8 Hash functions and key sizes suitability end dates.....	19
8.1 Introduction	19
8.2 Basis for the recommendations	19
8.3 Void.....	20
8.4 Recommended end dates for key sizes	20
8.5 Post-Quantum Cryptography Migration	21
9 Life time and resistance of hash functions and keys	21
9.1 General notes.....	21
9.2 Time period resistance for hash functions.....	21
9.3 Time period resistance for signer's key	22
9.4 Time period resistance for trust anchors.....	22
9.5 Time period resistance for other keys.....	22

10	Practical ways to identify hash functions and signature algorithms.....	22
10.1	General	22
10.2	Hash function and signature algorithm objects identified using OIDs	23
10.2.1	Introduction.....	23
10.2.2	Hash functions	23
10.2.3	Elliptic curves	23
10.2.4	Signature algorithms	24
10.2.5	Signature suites	25
10.3	Hash function and signature algorithm objects identified using URIs	25
10.3.1	Hash functions	25
10.3.2	Signature algorithms	26
10.3.3	Signature suites	26
10.4	Void.....	26
Annex A (normative): Algorithms for various data structures.....		27
A.1	Introduction	27
A.2	CAAdES and PAdES	27
A.3	XAdES	28
A.4	Signer's certificates.....	29
A.5	CRLs.....	29
A.6	OCSP responses	29
A.7	CA certificates.....	30
A.8	Self-signed certificates for CA issuing CA certificates.....	30
A.9	TSTs based on IETF RFC 3161	31
A.10	TSU certificates.....	31
A.11	Self-signed certificates for CAs issuing TSU certificates	32
Annex B (informative): Signature maintenance		33
Annex C (informative): Machine processable formats of the Algo Paper.....		34
C.1	JSON file location	34
C.2	XML file location.....	34
Annex D (informative): Discontinued algorithms.....		35
Annex E (informative): Bibliography.....		37
History		38

List of tables

Table 1: Hash Functions.....	13
Table 2: Digital Signature Algorithms	14
Table 3: Elliptic Curve Parameters.....	15
Table 3.1: ML-DSA parameter sets.....	16
Table 3.2: SLH-DSA parameter sets.....	16
Table 3.3: Algorithm-level hybrid combinations	17
Table 4: List of signature suites	19
Table 5: Void.....	20
Table 6: Recommended end dates for RSA key sizes	20
Table 7: Recommended end dates for DSA key sizes.....	20
Table 8: Void.....	21
Table 9: Void.....	21
Table 10: Void.....	21
Table 11: OIDs of suitable hash functions	23
Table 12: OIDs of suitable elliptic curves	23
Table 13: OIDs of suitable signature algorithms.....	24
Table 14: OIDs of suitable signatures suites	25
Table 15: URIs of suitable hash functions	26
Table 16: URIs of suitable signature suites	26
Table A.1: Hash functions and signature algorithms for PAdES and CAAdES	28
Table A.2: Hash functions and signature algorithms for XAdES.....	28
Table A.3: Algorithms for signer public keys and CA issuing keys	29
Table A.4: Algorithms for CRL issuer public keys.....	29
Table A.5: Algorithms for OCSP responders.....	30
Table A.6: Algorithms for certification authorities	30
Table A.7: Algorithms for self-signed certificates	30
Table A.8: Algorithms for timestamps	31
Table A.9: Algorithms for timestamping units.....	31
Table D.1: Discontinued cryptographic hash functions	35
Table D.2: Discontinued signature algorithm and key size combinations.....	36
Table D.3: Discontinued signature suites (special cases).....	36

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 IPR online database](#).

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™**, **LTE™** and **5G™** logo 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.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Electronic Signatures and Trust Infrastructures (ESI).

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.

Introduction

Selection of the cryptographic suites to apply for digital signatures is an important business parameter for products and services implementing digital signatures. The present document provides guidance on selection of cryptographic suites with particular emphasis on interoperability. The present document is based on the specified agreed cryptographic mechanisms of the European Cybersecurity Certification Group (ECCG) [14]. The ECCG Sub-group on Cryptography is in charge of providing requirements and evaluation procedures related to cryptographic aspects of European cybersecurity certification schemes. To avoid conflicts between the evaluation of security products for qualified trust services and the recommendation given in the present document, the ETSI Technical Committee Electronic Signatures and Trust Infrastructures (ESI) decided to refer for the trust services defined in Regulation (EU) No 910/2014 [i.12] consisting of creation, verification, and validation of electronic signatures, electronic seals and electronic time stamps, electronic registered delivery services, electronic archiving, electronic ledgers and certificates related to those services to the ECCG Agreed Cryptographic Mechanisms. Other standardization bodies, security agencies and supervisory authorities of the Member States have published guidance documents with partially overlapping scope, not referenced in the present document.

1 Scope

The present document lists cryptographic suites used for the creation and validation of digital signatures, electronic timestamps, electronic registered delivery services, electronic archiving, electronic ledgers and related certificates. The present document builds on the agreed cryptographic mechanisms from the European Cybersecurity Certification Group (ECCG) [14].

It incorporates requirements for the transition to Post-Quantum Cryptography (PQC), including hybrid cryptographic schemes and the phasing out of legacy mechanisms in accordance with the security assessment of ECCG Agreed Mechanisms and Regulation (EU) No 910/2014 [i.12].

In contrast to previous versions of the present document, specific end dates are provided and aligned with the PQC migration phases. The present document works on the assumption that the validity period (i.e. between notBefore and notAfter) of (qualified) end-entity certificates issued by trust services providers is typically three years.

The present document focuses on interoperability issues and does not duplicate security considerations given by other standardization bodies, security agencies or supervisory authorities of the Member States. It instead provides guidance on the selection of concrete cryptographic suites that use agreed mechanisms. The use of ECCG agreed mechanisms is meant to help ensure a high level of security in the recommended cryptographic suites, while the focus on specific suites of mechanisms is meant to increase interoperability and simplify design choices.

There is no normative requirement on selection among the alternatives for cryptographic suites given here but for all of them normative requirements apply to ensure security and interoperability.

The present document also provides guidance on hash functions, (digital) signature schemes and (digital) signature suites to be used with the data structures used in the context of digital signatures and seals. For each data structure, the set of algorithms to be used is specified.

NOTE 1: Migration scheduling, transition phases, and compliance deadlines for the transition to Post-Quantum Cryptography is planned to be addressed in the applicable post-quantum migration document currently under development, not in the present document. The present document provides mechanism suitability classifications (R/L) that are independent of deployment-phase timelines.

NOTE 2: The present technical specification is known to be incomplete with respect to addressing the needs for the EUDI-Wallet according to Article 5a of the Regulation (EU) No. 910/2014 [i.12] as well as certain trust services including preservation services according to Article 34 and Article 40 as well as electronic registered delivery services according to Article 43 and Article 44. Therefore, an update of the present document is already in preparation.

2 References

2.1 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.

Referenced documents which are not found to be publicly available in the expected location might be found in the [ETSI docbox](#).

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

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

- [1] [NIST FIPS Publication 180-4 \(August 2015\)](#): "Secure Hash Standard (SHS)".
- [2] [NIST FIPS Publication 186-5 \(2023-02\)](#): "Digital Signature Standard (DSS)".
- [3] [IETF RFC 8017 \(November 2016\)](#): "PKCS #1: RSA Cryptography Specifications Version 2.2".

- [4] [ISO/IEC 14888-3:2018](#): "IT Security techniques — Digital signatures with appendix — Part 3: Discrete logarithm based mechanisms".
- [5] [IETF RFC 5639 \(March 2010\)](#): "Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation".
- [6] Void.
- [7] [IETF RFC 3279 \(April 2002\)](#): "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile".
- [8] [IETF RFC 4055 \(June 2005\)](#): "Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile".
- [9] [IETF RFC 5753 \(January 2010\)](#): "Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS)".
- [10] [IETF RFC 6931 \(April 2013\)](#): "Additional XML Security Uniform Resource Identifiers (URIs)".
- NOTE: Obsoleted by IETF RFC 9231.
- [11] W3C® Recommendation 11 April 2013: "[XML Encryption Syntax and Processing Version 1.1](#)".
- [12] [IETF RFC 3161 \(August 2001\)](#): "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)".
- [13] [IETF RFC 6960 \(June 2013\)](#): "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP".
- [14] [ECCG ACM 2.0](#): "European Cybersecurity Certification Group (ECCG): Agreed Cryptographic Mechanisms", Version 2.0, April 2025.
- [15] [NIST FIPS Publication 202 \(August 2015\)](#): "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions".
- [16] [IETF RFC 5480 \(March 2009\)](#): "Elliptic Curve Cryptography Subject Public Key Information".
- [17] Void.
- [18] Void.
- [19] [IETF RFC 5758](#): "Internet X.509 Public Key Infrastructure: Additional Algorithms and Identifiers for DSA and ECDSA".
- [20] [IETF RFC 9231](#): "Additional XML Security Uniform Resource Identifiers (URIs)".
- [21] [IETF RFC 9688](#): "Use of the SHA3 One-Way Hash Functions in the Cryptographic Message Syntax (CMS)".
- [22] [NIST SP 800-186](#): "Recommendations for Discrete Logarithm-based Cryptography: Elliptic Curve Domain Parameters".
- [23] [IETF RFC 8032 \(January 2017\)](#): "Edwards-Curve Digital Signature Algorithm (EdDSA)".
- [24] [IETF RFC 8410 \(August 2018\)](#): "Algorithm Identifiers for EdDSA, Ed448, and Ed25519".
- [25] Void.
- [26] [NIST FIPS Publication 204 \(August 2024\)](#): "Module-Lattice-Based Digital Signature Standard".
- [27] [NIST FIPS Publication 205 \(August 2024\)](#): "Stateless Hash-Based Digital Signature Standard".
- [28] [IETF RFC 8554](#): "Leighton-Micali Hash-Based Signatures".
- [29] [IETF RFC 8391](#): "XMSS: eXtended Merkle Signature Scheme".

2.2 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, ETSI cannot guarantee their long-term validity.

The following referenced documents may be useful in implementing an ETSI deliverable or add to the reader's understanding, but are not required for conformance to the present document.

- [i.1] European Network of Excellence in Cryptology: "Algorithms, Key Size and Protocols Report (2018)", ECRYPT - Coordination & Support, Action D5.4.
- [i.2] Void.
- [i.3] Void.
- [i.4] Void.
- [i.5] ISO/IEC 10118-3:2018: "Information technology — Security techniques — Hash functions — Part 3: Dedicated hash functions".

NOTE: This ISO Standard duplicates the standardization from FIPS Publication 180-5 [1].

- [i.6] ETSI TS 101 733 (V2.2.1) (04-2013): "Electronic Signatures and Infrastructures (ESI); CMS Advanced Electronic Signatures (CAAdES)".
- [i.7] ETSI TS 101 903 (V1.4.2) (12-2010): "Electronic Signatures and Infrastructures (ESI); XML Advanced Electronic Signatures (XAdES)".
- [i.8] ETSI TS 102 778 (parts 1 to 6): "Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles".
- [i.9] IETF RFC 5280 (2008): "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile".
- [i.10] W3C® Recommendation 15 March 2001: "[Canonical XML Version 1.0](#)".
- [i.11] W3C® Recommendation 18 July 2002: "[Exclusive XML Canonicalization Version 1.0](#)".
- [i.12] [Regulation \(EU\) No 910/2014](#) of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC.

NOTE: The eIDAS Regulation has been amended by Regulation (EU) 2024/1183 of the European Parliament and of the Council.

- [i.13] [OID Repository](#).

NOTE: This OID repository is a kind of wiki where any user can add any information about any OID. It is not an official registration authority for OIDs and should be handled with care. Nevertheless it provides usually the link to corresponding official registration authority.

- [i.14] Void.
- [i.15] ETSI EN 319 422 (V1.1.1) (03-2016): "Electronic Signatures and Infrastructures (ESI); Time-stamping protocol and time-stamp token profiles".
- [i.16] Void.
- [i.17] ETSI EN 319 122 (parts 1 and 2): "Electronic Signatures and Infrastructures (ESI); CAAdES digital signatures".

- [i.18] ETSI EN 319 132 (parts 1 and 2): "Electronic Signatures and Infrastructures (ESI); XAdES digital signatures".
- [i.19] ETSI EN 319 142 (parts 1 and 2): "Electronic Signatures and Trust Infrastructures (ESI); PAdES digital signatures".
- [i.20] ETSI EN 319 102-1: "Electronic Signatures and Trust Infrastructures (ESI); Procedures for Creation and Validation of AdES Digital Signatures; Part 1: Creation and Validation".
- [i.21] [LegiFrance](#): "Avis relatif aux paramètres de courbes elliptiques définis pas l'État français". In: Journal Officiel 0241 (October 2011), p. 17533.
- [i.22] ETSI TS 119 172-1: "Electronic Signatures and Infrastructures (ESI); Signature Policies; Part 1: Building blocks and table of contents for human readable signature policy documents".
- [i.23] [Fukang Liu et al.](#): "Analysis of RIPEMD-160: New Collision Attacks and Finding Characteristics with MILP".
- [i.24] [Marc Stevens et al.](#): "The first collision for full SHA-1".
- [i.25] [Thorsten Kleinjung et al.](#): "Factorization of a 768-bit RSA modulus".
- [i.26] Void.
- [i.27] Void.
- [i.28] [NIS Cooperation Group](#): "A Coordinated Implementation Roadmap for the Transition to Post-Quantum Cryptography", Part 1, Version: 1.1, EU PQC Workstream (11.06.2025).
- [i.29] IETF RFC 9881: "Internet X.509 Public Key Infrastructure -- Algorithm Identifiers for the Module-Lattice-Based Digital Signature Algorithm (ML-DSA)".
- [i.30] IETF RFC 9882: "Use of the ML-DSA Signature Algorithm in the Cryptographic Message Syntax (CMS)".
- [i.31] IETF RFC 9909: "Internet X.509 Public Key Infrastructure -- Algorithm Identifiers for the Stateless Hash-Based Digital Signature Algorithm (SLH-DSA)".
- [i.32] IETF RFC 9814: "Use of the SLH-DSA Signature Algorithm in the Cryptographic Message Syntax (CMS)".
- [i.33] NSA: "Commercial National Security Algorithm (CNSA) Suite 2.0 Cybersecurity Advisory", National Security Agency, September 2022 (updated April 2024).
- [i.34] ETSI TS 102 176-1 (V2.1.1): "Electronic Signatures and Infrastructures (ESI); Algorithms and Parameters for Secure Electronic Signatures; Part 1: Hash functions and asymmetric algorithms", ETSI, 2011-07.

NOTE: Withdrawn; superseded by ETSI TS 119 312.

3 Definition of terms, symbols, abbreviations and notations

3.1 Terms

For the purposes of the present document, the following terms apply:

AdES (digital) signature: digital signature that is either a CAdES signature, or a PAdES signature or a XAdES signature

CAdES signature: digital signature that satisfies the requirements specified within ETSI EN 319 122 (parts 1 and 2) [i.17]

cryptographic suite: combination of a signature scheme with a padding method and a cryptographic hash function

(digital) signature: data associated to, including a cryptographic transformation of, a data unit that:

- a) allows to prove the source and integrity of the data unit;
- b) allows to protect the data unit against forgery; and
- c) allows to support signer non-repudiation of signing the data unit.

hash function: As defined in ISO/IEC 10118-3 [i.5].

hybrid cryptographic scheme: cryptographic scheme combining a classical cryptographic algorithm and a post-quantum cryptographic algorithm, such that the security of the combination relies on the security of at least one of the component schemes

legacy mechanism: mechanism deployed on a large scale, currently offering a security level of at least 100 bits and considered to provide an acceptable short-term security but which should be phased out as soon as practical because no longer fully reflecting the state of the art and suffering from some security assurance limitations

NOTE: As defined in **ECCG Agreed Cryptographic Mechanisms** [14].

PADES signature: digital signature that satisfies the requirements specified within ETSI EN 319 142 (parts 1 and 2) [i.19]

recommended mechanism: mechanism, that fully reflects the state of the art in cryptography, currently offers a security level of at least 125 bits, supported by strong security arguments and can be said to provide an adequate level of security against all presently known or conjectured threats even considering the generally expected increases in computing power

NOTE: As defined in **ECCG Agreed Cryptographic Mechanisms** [14].

security level: number of operations necessary for an adversary to successfully break the security provided by the mechanism, expressed as a base 2 logarithm

NOTE 1: Security level is expressed as a base 2 logarithm, e.g. 100 bits of security means that 2^{100} operations are necessary.

NOTE 2: As defined in **ECCG Agreed Cryptographic Mechanisms** [14].

signature policy: set of rules for the creation and validation of a signature, that defines the technical and procedural requirements for signature creation and validation, in order to meet a particular business need, and under which the signature can be determined to be valid

signature scheme: triplet of three algorithms composed of a signature creation algorithm, a signature verification algorithm and a key generation algorithm

XAdES signature: digital signature that satisfies the requirements specified within ETSI EN 319 132 (parts 1 and 2) [i.18]

3.2 Symbols

For the purposes of the present document, the following symbols apply:

FR Identifier for Elliptic Curves defined by ANSSI

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ACM	Association for Computing Machinery
ANSSI	Agence Nationale de la Sécurité des Systèmes d'Information (National Agency for Security of Information Systems)
CA	Certification Authority

CMS	Cryptographic Message Syntax
CNSA	Commercial National Security Algorithm (Suite)
CRL	Certificate Revocation List
CRQC	Cryptographically Relevant Quantum Computer
CSOR	Cryptographic Algorithm Object Registration
DSA	Digital Signature Algorithm
EC	Elliptic Curve
ECCG	European Cybersecurity Certification Group
ECDSA	Elliptic Curve Digital Signature Algorithm
EC-SDSA-opt	optimized Elliptic Curve Schnorr Digital Signature Algorithm
EdDSA	Edwards-Curve Digital Signature Algorithm
ESI	Electronic Signatures and Trust Infrastructure

NOTE: A Technical Committee of ETSI.

FIPS	Federal Information Processing Standard
GF	Galois Field
HSM	Hardware Security Module
IETF	Internet Engineering Task Force
ISO	International Organization for Standardization
LMS	Leighton-Micali Signature(s)
NIST	National Institute of Standards and Technology
NSA	National Security Agency
OCSP	Online Certificate Status Protocol
OID	Object Identifier
PKCS	Public-Key Cryptography Standards
PKI	Public Key Infrastructure
PKIX	Public-Key Infrastructure (X.509)
PQC	Post-Quantum Cryptography
PSS	Probabilistic Signature Scheme
RFC	Request For Comments
RNG	Random Number Generator
RSA	Rivest, Shamir and Adleman algorithm
SHA	Secure Hash Algorithm
SLH	Stateless Hash-based (as in SLH-DSA, FIPS 205)
SOG-IS	Senior Officials Group Information Systems Security
TST	Time-Stamp Token
TSU	Time-Stamping Unit
URI	Uniform Resource Identifier
URN	Uniform Resource Number
XML	eXtensible Markup Language
XMSS	eXtended Merkle Signature Scheme

3.4 Notations

The requirements identified in the present document include the following notations for the classification of mechanisms as legacy mechanisms or recommended mechanisms:

L: denotes a legacy mechanism with a deprecation/phasing out date of 31.12.2034 and which might be extended with future releases of the present document.

NOTE: In contrast to [14] and to reflect the assumed typical validity period of end-entity certificates issued by trust service providers as laid out in the Scope, a default of three years is added to all the end dates in the present document.

L[yyyy]: denotes a legacy mechanism with a deprecation/phasing out date no later than 31.12.yyyy, where yyyy is an integer expressing a year.

L[yyyy+]: denotes a legacy mechanism with a deprecation/phasing out date of 31.12.yyyy, where yyyy is an integer expressing a year and which might be extended with future releases of the present document.

NOTE: L is semantically equivalent to L[2034+].

R: denotes a recommended mechanism which has no defined end date, yet.

4 Use of ECCG Agreed Mechanisms and Maintenance of the present document

In order to avoid duplicated effort, the assessment of the security of underlying cryptographic schemes is delegated to the ECCG document [14].

The ECCG Agreed Mechanisms distinguishes between **legacy mechanisms** (schemes and parameter selections which may enjoy wide deployment, but do not represent the current state of the art in cryptography) and **recommended mechanisms** (schemes and parameters which do represent the current state of the art in cryptography). The present document uses the notion of "recommended" and "legacy" primitives in the same way as in [14].

In general, only ECCG recommended mechanisms and key sizes or cryptographic suites using these cryptographic mechanisms and key sizes should be used to generate new signatures and seals (including certificate signatures). ECCG legacy mechanisms may, however, still be used for this purpose when this is necessary to ensure interoperability with existing infrastructures as long as they remain agreed. For the reader's convenience, the classification of mechanisms as legacy or recommended is repeated in the present document.

The maintenance activities will follow the maintenance procedure of ECCG Agreed Mechanisms [14] with revisions on a regular basis synchronized with ECCG publication cycles.

In the case of new attacks, the immediate need to remove an algorithm could arise, and a new revision of the present document will be published as soon as possible.

5 Hash functions

5.1 General

The list of hash functions in Table 1 shall be used. The functions shall be implemented as per the reference listed in Table 1 and shall follow the recommendations provided in the ECCG Agreed Cryptographic Mechanisms [14]. The present document provides additional recommendations in the following clauses.

Table 1: Hash Functions

Short hash function name	References	R/L
SHA-224	FIPS Publication 180-4 [1]	L[2028]
SHA-256	FIPS Publication 180-4 [1]	R
SHA-384	FIPS Publication 180-4 [1]	R
SHA-512	FIPS Publication 180-4 [1]	R
SHA3-256	FIPS Publication 202 [15]	R
SHA3-384	FIPS Publication 202 [15]	R
SHA3-512	FIPS Publication 202 [15]	R

5.2 Recommendations for SHA hash functions

5.2.1 SHA-512/256

SHA3-256 or SHA-512 should be used instead of SHA-512/256.

NOTE: The difference to SHA-256 is the bigger inner state, which gives a better collision resistance.

5.2.2 Hash Functions for use in post-quantum contexts

For digital signatures using classical algorithms (e.g. ECDSA, RSA): SHA-256, SHA-384, and SHA-512 should be used.

For hybrid digital signatures and long-term archiving (> 20 years): SHA-384, SHA-512, SHA3-384, and SHA3-512 should be used in order to provide at least 192 bits of classical security.

NOTE: SHA-256 may be considered insufficient for long-term archiving use cases due to its classical security margin of 128 bits being below the minimum classical security level of 192 bits recommended by the ECCG Agreed Cryptographic Mechanisms for post-quantum mechanisms. Quantum collision-finding algorithms do not reduce the collision resistance of SHA-256 and thus do not constitute the basis for this recommendation.

6 Signature schemes

6.1 Introduction

NOTE: A signature scheme consists of three algorithms: a key generation algorithm, a signature creation algorithm and a signature verification algorithm. The two latter are identified hereafter as a pair of algorithms. Each pair has its own name.

6.2 Signature algorithms

6.2.1 General

The list of signature algorithms given in Table 2 shall be used. The algorithms shall be implemented as per the reference listed in Table 2 and shall follow the recommendations provided in the ECCG Agreed Cryptographic Mechanisms [14], where applicable. The present document provides additional recommendations and requirements in the following clauses.

Table 2: Digital Signature Algorithms

Short signature algorithm name	References	R/L
RSA-PKCS#1v1_5	IETF RFC 8017 [3]	L
RSA-PSS	IETF RFC 8017 [3]	R
DSA	FIPS Publication 186-5, ISO/IEC 14888-3 [4]	R
ECDSA	FIPS Publication 186-5 [2]	R
EdDSA	IETF RFC 8032 [23]	R (see note 1)
ML-DSA	NIST FIPS Publication 204 [26]	R
SLH-DSA	NIST FIPS Publication 205 [27]	R
EC-SDSA-opt	ISO/IEC 14888-3 [4]	L
LMS	IETF RFC 8554 [28]	R (conditional, see note 2)
XMSS	IETF RFC 8391 [29]	R (conditional, see note 2)
NOTE 1: EdDSA is not an agreed digital signature scheme under the ECCG Agreed Cryptographic Mechanisms [14]. The "R" status assigned to EdDSA in Table 2 reflects the recommendation of the present document independently of the ECCG ACM.		
NOTE 2: LMS and XMSS are stateful hash-based signature schemes. Their use requires that the implementer establishes and maintains secure state management procedures to prevent private key reuse. Failure to manage state correctly may result in complete compromise of the signing key. The use of LMS and XMSS is therefore restricted to deployment environments where state management can be strictly enforced.		

NOTE: The notation given in parentheses in previous versions of the present document regarding SOG-IS document has been aligned with ECCG terminology.

6.2.2 Signature algorithms

6.2.2.1 RSA

The RSA algorithm with the padding scheme RSASSA-PSS [3], section 8.1 shall be used (ECCG recommended mechanism). RSA with the legacy padding scheme RSASSA-PKCS-v1_5 [3], section 8.2, may be used (ECCG legacy mechanism). The key length shall be selected according to clause 8.

The public exponent e shall be an odd positive integer such that $2^{16} < e < 2^{256}$.

In use cases where protection against attacks leveraging quantum computers is required (e.g. where the long-term validity of the signature extends beyond 2030), RSA shall be used in a hybrid scheme as specified in clause 6.4, or shall be replaced by a quantum-safe algorithm.

NOTE: RSA is vulnerable to attacks leveraging quantum computers (e.g. Shor's algorithm).

6.2.2.2 DSA

The DSA algorithm may be used (ECCG recommended mechanism) if the key length is chosen according to clause 8.

In use cases where protection against attacks leveraging quantum computers is required, DSA shall be used in a hybrid signature scheme as specified in clause 6.4, or shall be replaced by a quantum-safe algorithm.

NOTE: DSA is included to support widely used libraries (e.g. Bouncy Castle) for interoperability.

6.2.2.3 EC based DSA algorithms

The ECDSA algorithm shall be used (ECCG recommended mechanism). Key lengths are implicitly given by the named curves listed below.

ECDSA shall be used (ECCG recommended mechanisms) only if the elliptic curves are selected from the following Table 3.

In use cases where protection against attacks leveraging quantum computers is required, ECDSA shall be used in a hybrid signature scheme as specified in clause 6.4, or shall be replaced by a quantum-safe algorithm.

NOTE: EC-SDSA-opt is considered legacy L due to lack of support in widely used cryptographic libraries.

When used, the algorithms shall be as specified by the references provided in Table 3, derived from [14], page 26.

Table 3: Elliptic Curve Parameters

Curve family	Short curve name	References	R/L
FR	FRP256v1	ANSSI [i.21]	R
Brainpool	brainpoolP256r1	IETF RFC 5639 [5]	R
	brainpoolP384r1	IETF RFC 5639 [5]	R
	brainpoolP512r1	IETF RFC 5639 [5]	R
NIST	P-256	NIST Special Publication 800-186 [22]	R
	P-384	NIST Special Publication 800-186 [22]	R
	P-521	NIST Special Publication 800-186 [22]	R

6.2.2.4 Edwards-Curve DSA (EdDSA) algorithms

The Edwards-Curve Digital Signature Algorithm (EdDSA) including variants Ed25519 and Ed448 (as defined in IETF RFC 8032 [23]) shall be used as recommended mechanisms. EdDSA provides deterministic signatures and resistance to side-channel attacks. Ed25519 provides approximately 128 bits of security; Ed448 provides approximately 224 bits of security.

In use cases where protection against attacks leveraging quantum computers is required, EdDSA shall be used in a hybrid signature scheme as specified in clause 6.4, or shall be replaced by a quantum-safe algorithm.

NOTE: EdDSA provides deterministic signatures and is not vulnerable to side-channel attacks that exploit biased random values during signing. However, EdDSA may be vulnerable to fault attacks, and its resistance to other side-channel attacks depends on the specific implementation. The ACM Note 49-DSARandom, which concerns randomness requirements for DSA-type algorithms, does not apply to EdDSA.

6.2.2.5 ML-DSA

The Module-Lattice-Based Digital Signature Standard (ML-DSA) as specified in NIST FIPS 204 [26] shall be used as a recommended mechanism. Only pure ML-DSA as specified in NIST FIPS 204 [26] shall be used. HashML-DSA shall not be used.

The ML-DSA parameter sets listed in Table 3.1 shall be used.

Table 3.1: ML-DSA parameter sets

ML-DSA parameter set	Security level	R/L
ML-DSA-44	≈ 128 bits	R (see note)
ML-DSA-65	≥ 192 bits	R
ML-DSA-87	≥ 256 bits	R
NOTE: ML-DSA-44 provides a security level of approximately 128 bits, which meets the minimum threshold for recommended mechanisms in the present document. Its use is acceptable where ML-DSA-65 or ML-DSA-87 is not feasible. Implementers should prefer ML-DSA-65 or ML-DSA-87 for new deployments.		

The use of ML-DSA in a hybrid signature scheme as specified in clause 6.4 is recommended; migration scheduling is planned to be addressed in the applicable post-quantum migration document currently under development.

NOTE 1: HashML-DSA is excluded because: (a) it removes the non-resignability property of pure ML-DSA; (b) it introduces algorithm-confusion risk through an externally supplied hash-algorithm identifier; (c) it requires public-key commitment to a specific signing mode at certificate issuance time; and (d) it is excluded from IETF RFC 9881 [i.29] (X.509), IETF RFC 9882 [i.30] (CMS), and NSA CNSA 2.0 [i.33].

NOTE 2: For HSM-based or remote signing scenarios involving large messages, the External Mu pre-computation method - comprising computation of the 64-byte message representative μ in a separate cryptographic module as permitted under FIPS 204 Algorithm 7 - constitutes a conformant implementation of pure ML-DSA and does not require HashML-DSA.

6.2.2.6 SLH-DSA

The Stateless Hash-Based Digital Signature Standard (SLH-DSA) as specified in NIST FIPS 205 [27] shall be used as a recommended mechanism.

The SLH-DSA parameter sets listed in Table 3.2 shall be used.

Table 3.2: SLH-DSA parameter sets

SLH-DSA parameter set	Hash variant	Security level	R/L
slh-dsa-sha2-192s / slh-dsa-sha2-192f	SHA-2	192 bits	R
slh-dsa-shake-192s / slh-dsa-shake-192f	SHAKE	192 bits	R
slh-dsa-sha2-256s / slh-dsa-sha2-256f	SHA-2	256 bits	R
slh-dsa-shake-256s / slh-dsa-shake-256f	SHAKE	256 bits	R

For PKIX certificates and certificate-related structures: HashSLH-DSA as specified in IETF RFC 9909 [i.31] is permitted as a recommended mechanism, subject to the following conditions:

- 1) The pre-hash function used shall have a collision resistance of at least 192 bits.
- 2) The key-pair shall be committed to either pure SLH-DSA or HashSLH-DSA at the time of key-pair generation, as the two variants use distinct object identifiers and distinct Verify() procedures.

For CMS-based AdES signature formats (CAAdES, PAdES, XAdES): only pure SLH-DSA shall be used, consistent with IETF RFC 9814 [i.32].

The use of SLH-DSA in a hybrid signature scheme as specified in clause 6.4 is optional.

NOTE 1: HashSLH-DSA commits the key to a specific signing mode at the time of key-pair generation, because HashSLH-DSA and pure SLH-DSA use distinct object identifiers and distinct Verify() procedures. Pure SLH-DSA is recommended for new deployments to avoid this constraint.

NOTE 2: IETF RFC 9909 [i.31] permits HashSLH-DSA for X.509 structures due to CRL and certificate size constraints on HSMs. IETF RFC 9814 [i.32] excludes it from CMS because the signed-attributes mechanism resolves the message-size constraint without the associated drawbacks.

6.2.2.7 LMS and XMSS

The Leighton-Micali Signature scheme (LMS) as specified in IETF RFC 8554 [28] and the eXtended Merkle Signature Scheme (XMSS) as specified in IETF RFC 8391 [29] shall be used as recommended mechanisms, subject to the condition specified in note below.

The use of LMS or XMSS in a hybrid signature scheme as specified in clause 6.4 is optional.

LMS and XMSS are stateful hash-based signature schemes. Their security depends on the property that no two signatures are ever generated using the same one-time key. Implementers shall establish, document, and enforce secure state management procedures to prevent private key reuse. Failure to manage state correctly may result in complete compromise of the signing key. The use of LMS and XMSS is therefore restricted to deployment environments where state management can be strictly enforced, such as within a certified Hardware Security Module (HSM).

NOTE: LMS and XMSS are specifically suitable for use cases such as firmware signing and time-stamping where the signing entity is a controlled system component with well-defined state management capabilities.

6.3 Key generation

The key generation shall follow the recommendations and requirements in their normative references of Table 2 and the ECCG Agreed Cryptographic Mechanisms.

6.4 Hybrid Cryptographic Schemes

6.4.1 Digital Signature Hybrid Modes

For hybrid signatures, implementations shall combine a classical signature and a post-quantum signature. Acceptance requires both signatures to be valid.

6.4.2 Recommended Hybrid Suites

The recommended combinations for hybrid signature schemes are listed in Table 3.3.

Table 3.3: Algorithm-level hybrid combinations

Classical component	PQC component	Use case
RSA-PSS ($\geq 3\ 000$ bit)	ML-DSA-65	General purpose, certificates
RSA-PSS ($\geq 3\ 000$ bit)	ML-DSA-87	High-security, long-term
ECDSA (P-256/P-384)	ML-DSA-65	CAdES/XAdES/PAdES
ECDSA (P-384/P-521)	ML-DSA-87	High-security, long-term
EdDSA (Ed25519)	ML-DSA-65	General purpose
EdDSA (Ed448)	ML-DSA-87	High-security
ECDSA or EdDSA	SLH-DSA (Level 3 or Level 5)	Time-stamps, firmware signing

The hybrid requirement of the present document may be satisfied by:

- a) algorithm-level hybrids: two independent signature primitives computed and verified separately over the same message data; or

- b) protocol-level hybrids: mechanisms provided by the encapsulating protocol or data format, such as the signed-attributes mechanism in CMS (applicable to CAdES and PAdES) or multi-signature constructions in XAdES.

Both approaches are considered conformant for the purposes of the present document.

NOTE: The specific encoding and interoperability requirements for hybrid signature schemes are outside the scope of the present document and is planned to be addressed in the applicable profile standards and the post-quantum migration document currently under development.

7 Signature suites

7.1 Introduction

NOTE: The primary criteria for inclusion of an algorithm in the present document are:

- the algorithm is considered as agreed on by ECGG [14];
- the algorithm is commonly used; and
- the algorithm can easily and unambiguously be referenced (for example by means of an OID).

7.2 General

NOTE 1: A cryptographic signature suite is a combination of message encoding functions including a hash function and a defined signature scheme using a standardized signature algorithm. A signature suite consists therefore of the following components:

- a message encoding method including the hash function; and
- a signature algorithm and its associated parameters.

NOTE 2: To allow signing of more or less arbitrarily long messages, a signature suite uses a hash function, so that the signing/verification algorithms operate on a fixed-size hash of the message. An important issue is to tie the hash function to the signature scheme. Without this, the weakest available hash function can define the overall security level.

Due to possible interactions which can influence security of signatures, algorithms and parameters for secure signatures shall be used only in predefined combinations referred to as the signature suites.

7.3 Signature suites

Table 4 reflects the combination of the recommended hash functions and signature algorithms.

Whereas the signature suites based on elliptic curves can be implemented in principle with any recommended curve, only those combinations are recommended by the present document where the output length of the hash function is the same as the key size of the corresponding elliptic curve.

NOTE 1: In case of RSA the use of SHA-384 gives no security advantage over SHA-512, because it is a truncated derivation of the SHA-512 algorithm. Nevertheless, it is included here for reasons of compatibility.

NOTE 2: If in case of elliptic curves the output length of the hash function is greater than the key size n , then the leftmost n bits of the hash function output block is used in the calculations using the hash function output during the generation or verification of a digital signature output (FIPS Publication 186-5 [2], page 37).

NOTE 3: For pure ML-DSA and pure SLH-DSA, the hash function is intrinsic to the signature algorithm and is determined by the parameter set; no external hash-algorithm identifier is required. For HashML-DSA and HashSLH-DSA, the hash function is an explicit parameter supplied externally by the caller and is not fixed by the parameter set; the use of these pre-hash variants is subject to the restrictions defined in clauses 6.2.2.5 and 6.2.2.6.

The signature suites listed in Table 4 shall be used.

Table 4: List of signature suites

Entry name of the signature suite	Entry name for the hash function	Entry name for the signature algorithm	R/L
sha224-with-rsa	SHA-224	RSA-PKCSv1_5	L[2028]
sha256-with-rsa	SHA-256	RSA-PKCSv1_5	L
sha384-with-rsa	SHA-384	RSA-PKCSv1_5	L
sha512-with-rsa	SHA-512	RSA-PKCSv1_5	L
rsa-pss with mgf1SHA2-Identifier	SHA-256, SHA-384 or SHA-512	RSA-PSS	R
rsa-pss with mgf1SHA3-Identifier	SHA3-256, SHA3-384 or SHA3-512	RSA-PSS	R
sha2-with-dsa	SHA-256, SHA-384, SHA-512	DSA	R
sha3-with-dsa	SHA3-256, SHA3-384, SHA3-512	DSA	R
sha2-with-ecdsa	SHA-256, SHA-384 or SHA-512	ECDSA	R
sha3-with-ecdsa	SHA3-256, SHA3-384 or SHA3-512	ECDSA	R
sha2-with-ecdsdsa	SHA-256, SHA-384 or SHA-512	EC-SDSA-opt	L
sha3-with-ecdsdsa	SHA3-256, SHA3-384 or SHA3-512	EC-SDSA-opt	L
ed25519	SHA-512 (internal)	Ed25519	R
ed448	SHAKE256 (internal)	Ed448	R
ml-dsa-44	SHAKE256 (internal)	ML-DSA-44	R
ml-dsa-65	SHAKE256 (internal)	ML-DSA-65	R
ml-dsa-87	SHAKE256 (internal)	ML-DSA-87	R
slh-dsa-sha2-192s	SHA-256 (internal)	SLH-DSA-SHA2-192s	R
slh-dsa-sha2-192f	SHA-256 (internal)	SLH-DSA-SHA2-192f	R
slh-dsa-shake-192s	SHAKE256 (internal)	SLH-DSA-SHAKE-192s	R
slh-dsa-shake-192f	SHAKE256 (internal)	SLH-DSA-SHAKE-192f	R
slh-dsa-sha2-256s	SHA-512 (internal)	SLH-DSA-SHA2-256s	R
slh-dsa-sha2-256f	SHA-512 (internal)	SLH-DSA-SHA2-256f	R
slh-dsa-shake-256s	SHAKE256 (internal)	SLH-DSA-SHAKE-256s	R
slh-dsa-shake-256f	SHAKE256 (internal)	SLH-DSA-SHAKE-256f	R

NOTE 4: EC-SDSA-opt suites are considered legacy L due to lack of support in widely used cryptographic libraries (see clause 6.2.2.3).

8 Hash functions and key sizes suitability end dates

8.1 Introduction

In this clause recommendations are provided regarding the use of hash functions given in clause 5 and the key sizes to be used with the algorithms mentioned in clause 6. This clause is structured as follows:

- Clause 8.2 explains the considerations on which the recommendations are based.
- In clause 8.4, key sizes end dates are recommended.
- In clause 8.5, the migration timeline for the transition to Post-Quantum Cryptography is defined.

8.2 Basis for the recommendations

NOTE 1: The recommendations for algorithm and parameter strengths are characterized by taking a reasonable margin above minimum key lengths based on both extrapolations of current trends as well as estimations based on the necessary computing power needed to break a given algorithm. Such extrapolations are made in the ECCG Agreed Cryptographic Mechanisms [14]. Similar assessments can be found also elsewhere in the literature, e.g. in the ECRYPT report on algorithms, key size and protocols report (2018) [i.1].

NOTE 2: There are no rigorous security proofs for the components of signature schemes (hash function, signature algorithm, RNG), basically all security statements rely on results about the most effective attacks known at the time of writing of the present document. The possibility of a complete break of such a component (like, e.g. a fast universal factorization algorithm against RSA or a quantum computer running Shor's algorithm) that renders it useless can theoretically not completely be excluded. The ECCG document introduces specific measures to mitigate the "harvest now, decrypt later" threat posed by future quantum computers.

NOTE 3: Stability of the requirements in the present document is highly desirable for reasons of planning reliability. However, the transition to Post-Quantum Cryptography (PQC) requires adjustments to the lifecycle of classical algorithms. The following tables contain recommendations for the lifetime of keys and were chosen according to the ECCG Agreed Cryptographic Mechanisms plus the default validity period of end-entity certificates.

ECCG recommended mechanisms should provide at least 125 bits of security against offline attacks. 100 bits of security may be used by ECCG legacy mechanisms, but they provide a lower security margin.

8.3 Void

Table 5: Void

8.4 Recommended end dates for key sizes

The parameters defined in Table 6 and Table 7, derived from ECCG Agreed Cryptographic Mechanisms [14], should be used.

The key size (security parameter) for RSA is the bit length of the modulus n .

Table 6: Recommended end dates for RSA key sizes

Key size ($\log_2(n)$ in bits)	End date	Recommendation
$\geq 1\,900$ and $< 3\,000$	2026-12-31	L[2026]
$\geq 3\,000$	n/a	R

RSA keys with a length of at least 1 900 bits and less than 3 000 bits shall not be used to issue new certificates after 2026-12-31. Certificates based on such keys that were issued on or before 2026-12-31 shall have a validity period ending no later than 2028-12-31. The same key pair shall not be used to obtain a new certificate after expiry, regardless of the cryptographic parameters of that new certificate. After 2026-12-31, only RSA keys with a length of at least 3 000 bits or signature schemes marked as Recommended (R) in the present document shall be used for newly issued certificates.

NOTE 1: The transition to Post-Quantum Cryptography and the recommendations in ECCG Agreed Mechanisms v2.0 [14] require expedited phasing out of RSA key sizes $< 3\,000$ bits by the end of 2026, overriding previous guidance.

The key sizes (security parameters) for DSA are the bit lengths of the prime p and q the order of a subgroup of the multiplicative group of the prime field $GF(p)$.

Table 7: Recommended end dates for DSA key sizes

Key size ($\log_2(p)$, $\log_2(q)$ in bits)	End date	Recommendation
$\log_2(p) \geq 1\,900$ and $< 3\,000$, $\log_2(q) \geq 200$ and < 250	2026-12-31	L[2026]
$\log_2(p) \geq 3\,000$, $\log_2(q) \geq 250$	n/a	R

NOTE 2: EdDSA (Ed25519, Ed448) uses fixed parameters and key sizes as defined in IETF RFC 8032 [23]. They are considered Recommended (R) without specific end dates in this clause.

Table 8: Void

Table 9: Void

Table 10: Void

8.5 Post-Quantum Cryptography Migration

The present document provides mechanism suitability classifications (R/L) that are independent of deployment-phase timelines. Migration scheduling, transition phases, and compliance deadlines for the transition to Post-Quantum Cryptography is planned to be addressed in the applicable post-quantum migration document currently under development referenced by the issuing authority.

NOTE: Guidance on PQC migration timelines for European trust services may be found in documents published by the NIS Cooperation Group [i.28] and ENISA.

9 Life time and resistance of hash functions and keys

9.1 General notes

NOTE 1: The hash functions and signature algorithms defined in the present document are suitable to be used in the context of advanced electronic signatures ETSI TS 101 733 [i.6], ETSI TS 101 903 [i.7], ETSI TS 102 778 [i.8], ETSI EN 319 122 [i.17], ETSI EN 319 132 [i.18] and ETSI EN 319 142 [i.19].

NOTE 2: The time period over which a given key needs to remain confidential depends on the usage of the key. More generally, the period of time over which a given mechanism needs to resist cryptanalytic attacks depends on the way it is being used. Determining this time period for a given mechanism allows one to then apply the figures provided in clause 9 to derive appropriate parameters.

9.2 Time period resistance for hash functions

Hash functions should remain suitable as long as a signature verification still needs to be done.

If not, a specific signature maintenance process shall be performed (see annex B for more information).

A hash function used to compute the hash of a certificate, which is not a self-signed certificate, should remain suitable during the validity period of that certificate.

A hash function used to compute the hash of a self-signed certificate shall resist during the validity period of that self-signed certificate.

NOTE 1: In the cases above, a hash function is used to produce a message digest to be signed. In these cases, the output length of the hash function will in general depend on the parameters of the signature scheme. However, this reasoning does not apply to all security critical roles that hash functions may fulfil in the context of trust services. A hash function used to compute the imprint of a message placed in a time-stamp token, for instance, is not used in combination of a signature scheme, but generates only part of the message to be signed. The length of its output is not dependent upon the size of the parameters of the signature scheme.

A hash function used to compute the imprint of a message placed in a time-stamp token should never be a legacy mechanism at the time of time stamp creation.

NOTE 2: If the signature suite that has been used by the signer is a recommended mechanism, the signature maintenance process can be minimized.

9.3 Time period resistance for signer's key

Signer's keys shall remain suitable during the certificate maintenance period (commonly called validity period from notBefore to notAfter) of the associated certificate.

NOTE 1: The focus is very often placed on the resistance of signer's keys.

NOTE 2: If they become weak due to progress in cryptographic research, revocation will be necessary, and there would be a large burden to re-issue new keys and certificates. However, there is no security breach after revocation.

NOTE 3: If a signer's key does not remain suitable during the validity period of its associated certificate, then the use of time-stamping is sufficient to provide adequate protection, if a time stamp using recommended mechanisms can be produced at a time when the signature suite retains at least legacy status.

9.4 Time period resistance for trust anchors

A trust anchor shall remain secure during the whole time period during which advanced electronic signature ETSI TS 101 733 [i.6], ETSI TS 101 903 [i.7], ETSI TS 102 778 [i.8], ETSI EN 319 122 [i.17], ETSI EN 319 132 [i.18] and ETSI EN 319 142 [i.19] needs to be verified.

NOTE 1: This can be longer than the life time of the associated certificate. If it becomes weak, it cannot be used anymore for immediate verifications. It can be used for subsequent verifications, if a specific maintenance process is performed before the trust anchor becomes insecure.

NOTE 2: This is an important difference to the estimation of the life time for signers' key.

9.5 Time period resistance for other keys

All other keys (TSU keys, CA keys, CRL issuer keys, OCSP responder keys) should resist during the validity period of the associated certificate and the certificates that rely on its validity.

Their security parameters shall then be chosen at least as strong as the corresponding parameters of the certified keys.

If they do not remain suitable for the foreseen time period, a maintenance process shall be applied before the algorithm is broken.

For these keys the same rule as for trust anchors in clause 9.4 applies.

10 Practical ways to identify hash functions and signature algorithms

10.1 General

Hash functions and signatures algorithms shall be referenced using an OID and/or a URN.

NOTE 1: Only the owner of the OID or the URN is allowed to define its meaning and thus the meaning of the algorithm, usually referencing another document.

NOTE 2: If such an OID/URN is not available the algorithm is unusable.

10.2 Hash function and signature algorithm objects identified using OIDs

10.2.1 Introduction

NOTE: All listed here OID can be found in the OID repository [i.13].

10.2.2 Hash functions

The hash functions shall be identified using the OIDs in Table 11.

Table 11: OIDs of suitable hash functions

Short object name	OID	References
id-sha224	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistalgorithm(4) hashalgs(2) 4 }	IETF RFC 4055 [8]
id-sha256	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistalgorithm(4) hashalgs(2) 1 }	IETF RFC 4055 [8]
id-sha384	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistalgorithm(4) hashalgs(2) 2 }	IETF RFC 4055 [8]
id-sha512	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistalgorithm(4) hashalgs(2) 3 }	IETF RFC 4055 [8]
id-sha512-256	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistalgorithm(4) hashalgs(2) 6 }	IETF RFC 8017 [3]
id-sha3-256	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistalgorithm(4) hashalgs(2) 8 }	IETF RFC 9688 [21]
id-sha3-384	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistalgorithm(4) hashalgs(2) 9 }	IETF RFC 9688 [21]
id-sha3-512	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistalgorithm(4) hashalgs(2) 10 }	IETF RFC 9688 [21]

10.2.3 Elliptic curves

The signature algorithms shall be identified using the OIDs in Table 12.

Table 12: OIDs of suitable elliptic curves

Short object name	OID	References
FRP256v1	{iso(1) member-body(2) fr(250) type-org(1) 223 101 256 1 }	ANSSI [i.21]
brainpoolP256r1	{iso(1) identified-organization(3) teletrust(36) algorithm(3) signatureAlgorithm(3) ecSign(2) ecStdCurvesAndGeneration(8) ellipticCurve(1) versionOne(1) brainpoolP256r1(7)}	IETF RFC 5639 [5]
brainpoolP384r1	{iso(1) identified-organization(3) teletrust(36) algorithm(3) signatureAlgorithm(3) ecSign(2) ecStdCurvesAndGeneration(8) ellipticCurve(1) versionOne(1) brainpoolP384r1(11)}	IETF RFC 5639 [5]
brainpoolP512r1	{iso(1) identified-organization(3) teletrust(36) algorithm(3) signatureAlgorithm(3) ecSign(2) ecStdCurvesAndGeneration(8) ellipticCurve(1) versionOne(1) brainpoolP512r1(13)}	IETF RFC 5639 [5]
P-256 (secp256r1)	{iso(1) member-body(2) us(840) ansi-X9-62(10045) curves(3) prime(1) 7 }	IETF RFC 5480 [16]
P-384 (secp384r1)	{iso(1) identified-organization(3) certicom(132) curve(0) 34 }	IETF RFC 5480 [16]
P-521 (secp521r1)	{iso(1) identified-organization(3) certicom(132) curve(0) 35 }	IETF RFC 5480 [16]

10.2.4 Signature algorithms

The signature algorithms shall be identified using the OIDs in Table 13.

Table 13: OIDs of suitable signature algorithms

Short object name	OID	References
rsaEncryption	{ iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 1 }	IETF RFC 3279 [7]
id-dsa	{ iso(1) member-body(2) us(840) x9-57(10040) x9cm(4) 1 }	IETF RFC 3279 [7]
id-ecPublicKey	{ iso(1) member-body(2) us(840) 10045 2 1 }	IETF RFC 5753 [9]
id-Ed25519	{ iso(1) identified-organization(3) thawte(101) 112 }	IETF RFC 8410 [24]
id-Ed448	{ iso(1) identified-organization(3) thawte(101) 113 }	IETF RFC 8410 [24]
id-ml-dsa-44	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) sigAlgs(3) 17 }	NIST FIPS Publication 204 [26]
id-ml-dsa-65	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) sigAlgs(3) 18 }	NIST FIPS Publication 204 [26]
id-ml-dsa-87	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) sigAlgs(3) 19 }	NIST FIPS Publication 204 [26]
id-slh-dsa-sha2-192s	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) sigAlgs(3) 22 }	NIST FIPS 205 [27]
id-slh-dsa-sha2-192f	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) sigAlgs(3) 23 }	NIST FIPS 205 [27]
id-slh-dsa-sha2-256s	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) sigAlgs(3) 24 }	NIST FIPS 205 [27]
id-slh-dsa-sha2-256f	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) sigAlgs(3) 25 }	NIST FIPS 205 [27]
id-slh-dsa-shake-192s	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) sigAlgs(3) 28 }	NIST FIPS 205 [27]
id-slh-dsa-shake-192f	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) sigAlgs(3) 29 }	NIST FIPS 205 [27]
id-slh-dsa-shake-256s	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) sigAlgs(3) 30 }	NIST FIPS 205 [27]
id-slh-dsa-shake-256f	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) sigAlgs(3) 31 }	NIST FIPS 205 [27]

10.2.5 Signature suites

The signature suites shall be identified using the OIDs in Table 14.

Table 14: OIDs of suitable signatures suites

Short object name	OID	References
sha256WithRSAEncryption	{ iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 11 }	IETF RFC 4055 [8]
sha512WithRSAEncryption	{ iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 13 }	IETF RFC 4055 [8]
id-RSASSA-PSS	{ iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 10 }	IETF RFC 4055 [8]
ecdsa-with-SHA224	{ iso(1) member-body(2) us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-Specified(3) 1 }	IETF RFC 5758 [19]
ecdsa-with-SHA256	{ iso(1) member-body(2) us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-Specified(3) 2 }	IETF RFC 5758 [19]
ecdsa-with-SHA384	{ iso(1) member-body(2) us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-Specified(3) 3 }	IETF RFC 5758 [19]
ecdsa-with-SHA512	{ iso(1) member-body(2) us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-Specified(3) 4 }	IETF RFC 5758 [19]
id-ecdsa-with-sha3-256	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) sigAlgs(3) 10 }	NIST CSOR [17]
id-ecdsa-with-sha3-384	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) sigAlgs(3) 11 }	NIST CSOR [17]
id-ecdsa-with-sha3-512	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) sigAlgs(3) 12 }	NIST CSOR [17]
id-Ed25519		IETF RFC 8410 [24]
id-Ed448		IETF RFC 8410 [24]
id-hash-slh-dsa-sha2-192s-with-sha512	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) sigAlgs(3) 37 }	NIST FIPS 205 [27]
id-hash-slh-dsa-sha2-192f-with-sha512	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) sigAlgs(3) 38 }	NIST FIPS 205 [27]
id-hash-slh-dsa-sha2-256s-with-sha512	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) sigAlgs(3) 39 }	NIST FIPS 205 [27]
id-hash-slh-dsa-sha2-256f-with-sha512	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) sigAlgs(3) 40 }	NIST FIPS 205 [27]
id-hash-slh-dsa-shake-192s-with-shake256	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) sigAlgs(3) 43 }	NIST FIPS 205 [27]
id-hash-slh-dsa-shake-192f-with-shake256	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) sigAlgs(3) 44 }	NIST FIPS 205 [27]
id-hash-slh-dsa-shake-256s-with-shake256	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) sigAlgs(3) 45 }	NIST FIPS 205 [27]
id-hash-slh-dsa-shake-256f-with-shake256	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) sigAlgs(3) 46 }	NIST FIPS 205 [27]
NOTE:	IETF RFC 4055 [8] defined a hash-independent OID for the RSASSA-PSS signature algorithm. The OID for the specific hash function used in these algorithms is included in the algorithm parameters. So it is applicable for SHA2 and SHA3.	

10.3 Hash function and signature algorithm objects identified using URIs

10.3.1 Hash functions

The hash functions shall be identified using the URIs in Table 15.

Table 15: URIs of suitable hash functions

Short object name	URI	References
sha224	http://www.w3.org/2001/04/xmldsig-more#sha224	IETF RFC 6931 [10]
sha256	http://www.w3.org/2001/04/xmlenc#sha256	W3C® Recommendation XML Encryption Syntax and Processing, April 2013 [11]
sha384	http://www.w3.org/2001/04/xmldsig-more#sha384	IETF RFC 6931 [10]
sha512	http://www.w3.org/2001/04/xmlenc#sha512	W3C® Recommendation XML Encryption Syntax and Processing, April 2013 [11]
sha3-256	http://www.w3.org/2007/05/xmldsig-more#sha3-256	IETF RFC 9231 [20]
sha3-384	http://www.w3.org/2007/05/xmldsig-more#sha3-384	IETF RFC 9231 [20]
sha3-512	http://www.w3.org/2007/05/xmldsig-more#sha3-512	IETF RFC 9231 [20]

10.3.2 Signature algorithms

NOTE: There is no need to define such URIs since XAdES uses the signature algorithms contained in X.509 certificates which are referenced using OIDs.

10.3.3 Signature suites

The signature suites shall be identified using the URIs in Table 16.

Table 16: URIs of suitable signature suites

Short object name	URI	References
rsa-sha224	http://www.w3.org/2001/04/xmldsig-more#rsa-sha224	IETF RFC 9231 [20]
rsa-sha256	http://www.w3.org/2001/04/xmldsig-more#rsa-sha256	IETF RFC 6931 [10]
rsa-sha384	http://www.w3.org/2001/04/xmldsig-more#rsa-sha384	IETF RFC 6931 [10]
rsa-sha512	http://www.w3.org/2001/04/xmldsig-more#rsa-sha512	IETF RFC 6931 [10]
rsapss-with-parameters	http://www.w3.org/2007/05/xmldsig-more#rsa-pss	IETF RFC 6931 [10]
rsapss-with-defaults-sha224	http://www.w3.org/2007/05/xmldsig-more#sha224-rsa-MGF1	IETF RFC 6931 [10]
rsapss-with-defaults-sha256	http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-MGF1	IETF RFC 6931 [10]
rsapss-with-defaults-sha384	http://www.w3.org/2007/05/xmldsig-more#sha384-rsa-MGF1	IETF RFC 6931 [10]
rsapss-with-defaults-sha512	http://www.w3.org/2007/05/xmldsig-more#sha512-rsa-MGF1	IETF RFC 6931 [10]
rsapss-with-sha3-224	http://www.w3.org/2007/05/xmldsig-more#sha3-224-rsa-MGF1	IETF RFC 6931 [10]
rsapss-with-sha3-256	http://www.w3.org/2007/05/xmldsig-more#sha3-256-rsa-MGF1	IETF RFC 6931 [10]
rsapss-with-sha3-384	http://www.w3.org/2007/05/xmldsig-more#sha3-384-rsa-MGF1	IETF RFC 6931 [10]
rsapss-with-sha3-512	http://www.w3.org/2007/05/xmldsig-more#sha3-512-rsa-MGF1	IETF RFC 6931 [10]
ecdsa-sha224	http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha224	IETF RFC 6931 [10]
ecdsa-sha256	http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256	IETF RFC 6931 [10]
ecdsa-sha384	http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha384	IETF RFC 6931 [10]
ecdsa-sha512	http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha512	IETF RFC 6931 [10]
ecdsa-sha3-256	http://www.w3.org/2021/04/xmldsig-more#ecdsa-sha3-256	IETF RFC 9231 [20]
ecdsa-sha3-384	http://www.w3.org/2021/04/xmldsig-more#ecdsa-sha3-384	IETF RFC 9231 [20]
ecdsa-sha3-512	http://www.w3.org/2021/04/xmldsig-more#ecdsa-sha3-512	IETF RFC 9231 [20]
ed25519	http://www.w3.org/2021/04/xmldsig-more#ed25519	IETF RFC 9231 [20]
ed448	http://www.w3.org/2021/04/xmldsig-more#ed448	IETF RFC 9231 [20]

NOTE 1: The URI rsapss-with-parameters allows also the parametrization with SHA-3.
NOTE 2: There are no URI defined for RSA with PKCS#1v1.5 padding and SHA-3.

10.4 Void

Annex A (normative): Algorithms for various data structures

A.1 Introduction

ETSI TS 101 733 [i.6], ETSI TS 101 903 [i.7], ETSI TS 102 778 [i.8], ETSI EN 319 122 [i.17], ETSI EN 319 132 [i.18] and ETSI EN 319 142 [i.19] define the formats of advanced (digital) signatures. These documents reference other documents defining various standardized data structures.

These other documents or companion documents define the algorithms which can be supported by the issuers of the data structures and the algorithms which will (for interoperability purposes) and can be supported by the users of the data structures:

- Signer Certificates (IETF RFC 5280 [i.9] and IETF RFC 3279 [7]).
- Certificate Revocation Lists (IETF RFC 5280 [i.9] and IETF RFC 3279 [7]).
- OCSP responses (IETF RFC 6960 [13]).
- Certification Authority Certificates (IETF RFC 5280 [i.9] and IETF RFC 3279 [7]).
- Self-signed certificates for CA certificates (IETF RFC 5280 [i.9] and IETF RFC 3279 [7]).
- Time-Stamping Tokens (TSTs) (IETF RFC 3161 [12] and ETSI EN 319 422 [i.15]).
- Time-Stamping Unit certificates (IETF RFC 3161 [12] and ETSI EN 319 422 [i.15]).
- Self-signed certificates for TSU Certificates (IETF RFC 5280 [i.9] and IETF RFC 3279 [7]).
- Attribute Certificates (Acs) (IETF RFC 5280 [i.9] and IETF RFC 3279 [7]).

For each data structure, the set of algorithms to be used is specified.

Since many of these documents have been published some years ago, they cannot be all up to date with the latest cryptographic advancements. In particular, some of the algorithms specified in the above documents exhibit weaknesses or, worse, are now broken. These algorithms are not listed in the following tables.

Despite outdated algorithms may be used in the verification of archive signatures, e.g. SHA-1, they are not mentioned in the following. The requirements of this annex apply to the date of issuance of the present document.

Algorithms which may be additionally supported by issuers or users are not indicated too.

For the transition to Post-Quantum Cryptography, the use of hybrid schemes (combining classical and post-quantum algorithms) should be considered for new implementations as indicated in clause 6.4. The tables below list both classical and post-quantum algorithms to facilitate this transition.

A.2 CAdES and PAdES

A CMS based digital signature (ETSI TS 101 733 [i.6]/ETSI EN 319 122 [i.17] and ETSI TS 102 778 [i.8]/ETSI EN 319 142 [i.19]) contains an identifier of the hash function that has been used (contained in the `digestAlgorithm` element from the `SignerInfo` data structure) and an identifier of the signature algorithm that has been used (contained in the `signatureAlgorithm` element from the `SignerInfo` data structure) which will be consistent with the identifier of the signature algorithm contained in the signer's certificate.

Requirements in Table A.1 apply to CAdES [i.6] and PAdES [i.8]. They apply both to the hash function and the signature algorithm.

Table A.1: Hash functions and signature algorithms for PAdES and CAdES

CAdES [i.6] and PAdES [i.8]	Issuers of AdES	Users of AdES
Hash functions	shall support SHA-256 should support SHA-512 should support SHA3-512	shall support SHA-256, SHA-384, SHA-512 should support SHA3-256, SHA3-384, SHA3-512
Signature algorithms	should support RSA-PSS, DSA, EdDSA or ECDSA should support ML-DSA or SLH-DSA (PQC)	shall support RSA-PKCS1v1_5, RSA-PSS , DSA, EdDSA,ECDSA shall support ML-DSA, SLH-DSA (PQC)
Legacy	may support RSA-PKCS1v1_5	should support EC-SDSA-opt (L)

NOTE 1: The use of EdDSA (Ed25519, Ed448) and Post-Quantum algorithms (ML-DSA, SLH-DSA) often implies specific internal hash functions or state handling defined by the algorithm OID. In these cases, the "Hash functions" row applies to the message digest algorithm used to hash the content before the signature primitive is applied (e.g. the message-digest attribute in CMS).

NOTE 2: During the transition to Post-Quantum Cryptography, hybrid schemes may require two independent signatures or a composite signature structure. In such cases, the requirements apply to both components of the hybrid suite.

NOTE 3: EC-SDSA-opt is considered legacy (L) and not recommended for new signatures due to limited support in widely deployed cryptographic libraries.

A.3 XAdES

ETSI TS 101 903 [i.7]/ETSI EN 319 132 [i.18] use a URI to reference the hash function in the ds:DigestMethod element. Since ETSI TS 101 903 [i.7]/ETSI EN 319 132 [i.18] are built upon XML DigSig, the algorithm requirements from XML DigSig [11] shall apply with the amendments defined in Table A.2.

Table A.2: Hash functions and signature algorithms for XAdES

XAdES [i.7]	Issuers of AdES	Users of AdES
Hash functions	shall support SHA-256 should support SHA-512 should support SHA3-512	shall support SHA-256, SHA-384, SHA-512 should support SHA3-256, SHA3-384, SHA3-512
Signature algorithms	should support RSA-PSS, DSA, EdDSA or ECDSA should support ML-DSA or SLH-DSA (PQC)	shall support RSA-PKCS1v1_5, RSA-PSS , DSA, EdDSA,ECDSA shall support ML-DSA, SLH-DSA (PQC)
Legacy	may support RSA-PKCS1v1_5	should support EC-SDSA-opt (L)

For canonicalization:

- 1) the following Canonical XML (omits comments) [i.10] should be used:
<http://www.w3.org/TR/2001/REC-xml-c14n-20010315>;
- 2) the following Canonical XML with Comments [i.11] may be used:
<http://www.w3.org/TR/2002/REC-xml-exc-c14n-20020718>.

NOTE 1: The use of EdDSA (Ed25519, Ed448) and Post-Quantum algorithms (ML-DSA, SLH-DSA) often implies specific internal hash functions or state handling defined by the algorithm OID. In these cases, the "Hash functions" row applies to the message digest algorithm used to hash the content before the signature primitive is applied (e.g. the message-digest attribute in CMS).

NOTE 2: EC-SDSA-opt is considered legacy (L) and not recommended for new signatures due to limited support in widely deployed cryptographic libraries.

NOTE 3: During the transition to Post-Quantum Cryptography, hybrid schemes (e.g. combining ECDSA with ML-DSA) may be required. Implementations should refer to clause 6.4 for hybrid composition rules.

A.4 Signer's certificates

A signer certificate contains a subject public key and is signed by a CA issuing key. IETF RFC 5280 [i.9] does not require to use any particular cryptographic algorithms. However, IETF RFC 3279 [7] does. The requirements in IETF RFC 3279 [7] shall apply to signer public keys and CA issuing keys with the amendments defined in Table A.3.

Table A.3: Algorithms for signer public keys and CA issuing keys

Signer certificates	Issuers of signer certificates	Users of signer certificates
Hash functions	should support RSA, DSA, EdDSA or ECDSA should support ML-DSA or SLH-DSA (PQC)	shall support RSA, DSA, EdDSA, ECDSA shall support ML-DSA, SLH-DSA
Signature algorithms	shall support RSA, EdDSA or ECDSA should support Hybrid (Classical + ML-DSA/SLH-DSA)	shall support RSA, DSA, EdDSA, ECDSA shall support ML-DSA, SLH-DSA
Legacy	may support RSA-PKCS1v1_5	should support EC-SDSA-opt (L)

With RSA and DSA, the hash functions SHA-256 and SHA-512 should be used instead of SHA-224 or SHA-384.

NOTE 1: The use of EdDSA (Ed25519, Ed448) implies specific subject public key information formats as defined in IETF RFC 8410 [24].

NOTE 2: For Post-Quantum Cryptography (ML-DSA, SLH-DSA), the subject public key information structures are defined in NIST FIPS 204 and NIST FIPS 205 (or the corresponding IETF RFCs).

NOTE 3: During the transition to Post-Quantum Cryptography, hybrid certificates containing both classical and post-quantum keys (or multiple certificates linked via specific extensions) are RECOMMENDED for CA issuing keys to ensure long-term security.

A.5 CRLs

A CRL is signed by a CRL Issuer. IETF RFC 5280 [i.9] does not require to use any particular cryptographic algorithms. However, IETF RFC 3279 [7] does. The requirements defined in IETF RFC 3279 [7] shall apply to CRL Issuer public keys with the amendments defined in Table A.4.

Table A.4: Algorithms for CRL issuer public keys

CRLs	Issuers of CRLs	Users of CRLs
CRL issuer keys	shall support RSA with SHA-256 should support EdDSA or ECDSA should support ML-DSA or SLH-DSA	shall support RSA, DSA, EdDSA, ECDSA shall support ML-DSA, SLH-DSA

NOTE 1: Because the usage of SHA-224 with RSA and DSA gives no advantage compared with SHA-256 neither in security nor in performance there is no requirement on SHA-224 support with these algorithms.

NOTE 2: For EdDSA (Ed25519, Ed448) and Post-Quantum algorithms (ML-DSA, SLH-DSA), the hash function is often intrinsic to the signature algorithm definition.

With RSA and DSA the hash functions SHA-256 and SHA-512 should be used instead of SHA-224 or SHA-384.

A.6 OCSP responses

An OCSP response is signed by an OCSP responder. The algorithm requirements from IETF RFC 6960 [13], clause 4.3 shall apply with the amendments defined in Table A.5. These requirements shall apply to the hash algorithm and the signature algorithm used by OCSP responders.

Table A.5: Algorithms for OCSP responders

OCSP response	Issuers of OCSP responses	Users of OCSP response
OCSP responder keys	shall support RSA with SHA-256 should support EdDSA or ECDSA should support ML-DSA or SLH-DSA	shall support RSA, DSA, EdDSA, ECDSA shall support ML-DSA, SLH-DSA

NOTE 1: For EdDSA (Ed25519, Ed448) and Post-Quantum algorithms (ML-DSA, SLH-DSA), the hash function is intrinsic to the signature algorithm definition or specified within the parameter set.

NOTE 2: During the transition to Post-Quantum Cryptography, OCSP responders may sign responses using hybrid schemes (e.g. classical signature + post-quantum signature) to ensure long-term non-repudiation.

A.7 CA certificates

A CA certificate contains a CA public key and is signed by a CA private key. For CA public keys (as subject) and CA public keys (as issuer), the algorithm requirements from IETF RFC 3279 [7] shall apply with the amendments defined in Table A.6.

Table A.6: Algorithms for certification authorities

CA certificates	Issuers of CA certificates	Users of CA certificates
Subject CA public key	should support RSA, EdDSA or ECDSA should support ML-DSA (PQC)	shall support RSA, DSA, EdDSA, ECDSA shall support ML-DSA, SLH-DSA
Issuer CA public keys	should support RSA, EdDSA or ECDSA should support Hybrid (Classical + ML-DSA)	shall support RSA, DSA, EdDSA, ECDSA shall support ML-DSA, SLH-DSA

NOTE 1: Because the usage of SHA-224 with RSA and DSA gives no advantage compared with SHA-256 neither in security nor in performance there is no requirement on SHA-224 support with these algorithms.

NOTE 2: For EdDSA (Ed25519, Ed448) and Post-Quantum algorithms (ML-DSA, SLH-DSA), the hash function is often intrinsic to the signature algorithm definition (e.g. SHA-512 for Ed25519, SHAKE256 for Ed448/ML-DSA).

NOTE 3: During the transition to Post-Quantum Cryptography, hybrid CA certificates (e.g. using X.509 alternative public key extensions or composite keys) are RECOMMENDED for Issuer CA keys to ensure long-term security of the PKI hierarchy.

With RSA and DSA, SHA-256 and SHA-512 should be used instead of SHA-224 or SHA-384.

A.8 Self-signed certificates for CA issuing CA certificates

A self-signed certificate contains a single root CA public key. For root CA public keys, the algorithm requirements from IETF RFC 3279 [7] shall apply with the amendments defined in Table A.7.

NOTE 1: Self-signed certificates need to resist quite long (e.g. more than 10 or 20 years). Therefore, the selection of algorithms and key sizes requires consideration of long-term security, including resistance against quantum computing attacks (see clause 8.5).

Table A.7: Algorithms for self-signed certificates

Self-signed certificates	Issuers of self-signed certificates	Users of self-signed certificates
Root CA public keys	shall support RSA with SHA-256/512 should support EdDSA or ECDSA should support ML-DSA or SLH-DSA (PQC)	shall support RSA, DSA, EdDSA, ECDSA shall support ML-DSA, SLH-DSA

NOTE 2: For EdDSA (Ed25519, Ed448) and Post-Quantum algorithms (ML-DSA, SLH-DSA), the hash function is intrinsic to the signature algorithm definition.

NOTE 3: To ensure long-term security during the transition to Post-Quantum Cryptography, issuers should consider using hybrid schemes (e.g. using X.509 alternative public key extensions to carry a PQC key alongside a classical key) for Root CAs.

With RSA and DSA, SHA-256 and SHA-512 should be used instead of SHA-224 or SHA-384.

A.9 TSTs based on IETF RFC 3161

The following requirements apply to hash functions and TST signature algorithms. The algorithm requirements from IETF RFC 3161 [12] shall apply with the amendments defined in Table A.8.

Table A.8: Algorithms for timestamps

Time-Stamping Tokens	TST requesters	TST issuers	TST verifiers
Hash function	shall support SHA-256 should support SHA-384, SHA-512	shall support SHA-256 should support SHA-384, SHA-512	shall support SHA-256, SHA-384, SHA-512
TST signature algorithms	shall support RSA, DSA, EdDSA, ECDSA	shall support RSA, EdDSA, ECDSA should support SLH-DSA, XMSS or LMS (PQC)	shall support RSA, DSA, EdDSA, ECDSA shall support SLH-DSA, XMSS, LMS

NOTE 1: For long-term archiving (> 20 years), the use of Post-Quantum Cryptography is required. XMSS (IETF RFC 8391 [29]) and LMS (IETF RFC 8554 [28]) are stateful hash-based signature schemes specifically recommended for use cases like firmware updates and time-stamping, where the state management can be securely handled by the TSU.

NOTE 2: For EdDSA (Ed25519, Ed448) and Post-Quantum algorithms (ML-DSA, SLH-DSA), the hash function is often intrinsic to the signature algorithm definition.

NOTE 3: In the context of PQC transition, hash functions with an output length of at least 384 bits (e.g. SHA-384, SHA-512) are RECOMMENDED.

A.10 TSU certificates

A TSU certificate contains a TSU public key and is signed by a CA private key. For TSU public keys (as subject) and CA public keys (as issuer), the algorithm requirements from IETF RFC 3279 [7] shall apply with the amendments defined in Table A.9.

Table A.9: Algorithms for timestamping units

TSU certificates	Issuers of TSU certificates	Users of TSU certificates
TSU public key	should support RSA, EdDSA or ECDSA should support SLH-DSA, XMSS or LMS (PQC)	shall support RSA, DSA, EdDSA, ECDSA shall support SLH-DSA, XMSS, LMS
Issuer CA public keys	should support RSA, EdDSA or ECDSA should support Hybrid (Classical + ML- DSA)	shall support RSA, DSA, EdDSA, ECDSA shall support ML-DSA, SLH-DSA

NOTE 1: XMSS (IETF RFC 8391 [29]) and LMS (IETF RFC 8554 [28]) are stateful hash-based signature schemes. They are specifically RECOMMENDED for Time-Stamping Units (TSUs) where state management can be strictly enforced by the cryptographic module, providing high security assurances against quantum computer attacks for long-term validation.

NOTE 2: For EdDSA (Ed25519, Ed448) and Post-Quantum algorithms (ML-DSA, SLH-DSA, XMSS, LMS), the hash function is often intrinsic to the signature algorithm definition.

NOTE 3: During the transition to Post-Quantum Cryptography, TSU certificates typically have shorter validity periods than CA certificates. However, the TSU public key itself (and the resulting time-stamps) is expected to remain verifiable for long periods. Therefore, the use of PQC-algorithms or hybrid schemes for the TSU key is critical.

With RSA and DSA, SHA-256 and SHA-512 should be used instead of SHA-224 or SHA-384.

A.11 Self-signed certificates for CAs issuing TSU certificates

A self-signed certificate contains a single root CA public key. For self-signed certificates for CAs issuing TSU certificates, the algorithm requirements from IETF RFC 3279 [7] shall apply with the amendments defined in Table A.7 (see clause A.8).

NOTE 1: Self-signed certificates for TSU hierarchies often serve as trust anchors for long-term validation (e.g. Qualified Electronic Archiving) and therefore need to resist cryptanalytic attacks for very long periods (e.g. > 20 years).

NOTE 2: Consequently, the use of Post-Quantum Cryptography (specifically SLH-DSA or ML-DSA in hybrid modes) as defined in Table A.7 is strongly RECOMMENDED for new TSU Root CAs to ensure that timestamps remain verifiable beyond the entry of cryptographically relevant quantum computers.

Annex B (informative): Signature maintenance

An advanced (digital) signature (see ETSI TS 101 733 [i.6], ETSI TS 101 903 [i.7], ETSI TS 102 778 [i.8], ETSI EN 319 122 [i.17], ETSI EN 319 132 [i.18] and ETSI EN 319 142 [i.19]) can be verified according to a signature policy that meets the business needs.

A signature policy can include constraints about which algorithms and key lengths are deemed appropriate under that policy and/or define a time beyond which the algorithms/keys related to an advanced electronic signature should not be trusted anymore, unless additional security measures are taken.

It may be required to re-verify advanced signatures (this is called a subsequent verification) well beyond the time they were initially verified. At the time of re-verification, trust anchors and algorithms that were initially defined in the signature policy may not be secure anymore. Additional security measures need to be taken so that this can be accomplished.

Specific Consideration for Post-Quantum Cryptography: In particular, with the advent of Cryptographically Relevant Quantum Computers (CRQC), classical asymmetric algorithms (RSA, DSA, ECDSA, EdDSA) are expected to become insecure. For signatures requiring long-term validity (e.g. Qualified Electronic Archiving > 10 years), the signature maintenance process is expected to address this threat proactively.

It can also happen that some keys were secure at the time the initial verification of an advanced signature was performed, but due to some "accident" this is no more the case later on (e.g. due to a key compromise).

In both cases, it is possible to maintain the security of an advanced signature which has already been successfully verified. This can be achieved with security measures such as:

- the secure archival of both the definition of the signature policy (or an unambiguous reference to it) and all the data initially used to verify the advanced signature according to that signature policy; or
- the secure archival of both the definition of the signature policy and the addition to the advanced signature of other data (e.g. time-stamps) that will allow subsequent verifications.

PQC Transition Recommendation: During the transition to Post-Quantum Cryptography, it is **RECOMMENDED** that the maintenance process (e.g. applying Archive Time-Stamps) utilizes:

- 1) **Quantum-Safe Algorithms:** Use of hash-based signatures (e.g. SLH-DSA, XMSS, LMS) or hybrid schemes for the time-stamping authorities protecting the archive; and
- 2) **Strong Hash Functions:** Use of hash functions with larger outputs (e.g. SHA-384, SHA-512, SHA3-384) to resist quantum collision attacks.

These measures can be defined in the signature policy itself or "elsewhere" in a set of rules called a "signature maintenance policy" which will allow maintenance of the validity of advanced signatures.

A timely application of a signature maintenance process allows for re-verification of advanced signatures under a given signature policy even at a point in time where it is possible or likely that the algorithms and key lengths originally used will not be secure anymore. The sooner the process is applied, the better.

Annex C (informative): Machine processable formats of the Algo Paper

C.1 JSON file location

The file at https://forge.etsi.org/rep/esi/x19_312_crypto_suites/raw/v2.1.1/19312MachineReadable.json (19312MachineReadable.json) contains the JSON version of the present document.

NOTE: Independent of the present document, the latest version of the JSON file is linked to https://forge.etsi.org/rep/esi/x19_312_crypto_suites/-/blob/main/19312MachineReadable.json.

C.2 XML file location

The file at https://forge.etsi.org/rep/esi/x19_312_crypto_suites/raw/v2.1.1/19312MachineReadable.xml (19312MachineReadable.xml) contains the XML version of the present document.

NOTE: Independent of the present document, the latest version of the XML file is linked to https://forge.etsi.org/rep/esi/x19_312_crypto_suites/-/blob/main/19312MachineReadable.xml.

Annex D (informative): Discontinued algorithms

This annex lists algorithms that are not recommended anymore, not even with "legacy" status, and that were listed as recommended in earlier versions of the present document. The information provided here may be used as a basis for cryptographic constraints as specified by ETSI TS 119 172-1 [i.22], clause A.4.2.1, Table A.2 row p, for the purpose of validating electronic signatures in the past, typically based on proof-of-existence information (e.g. time-stamps), as for example specified in ETSI EN 319 102-1 [i.20], clause 5.

One way to determine an expiration date for a given algorithm, or combination of algorithm and key size, is to take into consideration:

- 1) the date of any known practical attack;
- 2) the publication date of the last specification recommending the algorithm, or combination of algorithm and key size;
- 3) the number of years of resistance stated by that specification; and
- 4) the publication date of the subsequent specification where it stopped being recommended;

as given in the tables below.

Note that the resistance periods listed in earlier versions of the present document have commonly been interpreted as relative to the date of signature creation or to the issuance of a certificate for a key, instead of relative to the publication date of the version of the present document containing the recommendation. The actual usage period therefore potentially extends beyond the date when the algorithm or key length stopped being recommended in a subsequent version of the present document. For example, RSA with 1 536 bits stopped being recommended in 2018-09, but a certificate for a 1536-bits RSA key may conceivably have been issued shortly before, with a validity period of 1 year in accordance with the previous recommendation, thus only ending a year later in 2019-09.

In tables D.1 to D.3, the last column indicates cryptographic constraints that can be used by default, in the absence of diverging application-specific interoperability or security considerations. The constraints are derived from a lenient interpretation of the resistance periods, unless overturned by the publication of a practical attack.

Table D.1: Discontinued cryptographic hash functions

Hash function	Last listed as recommended in	Resistance (a)	Not recommended since	First known practical attack	Suggested cryptographic constraint
RIPMD160	ETSI TS 102 176-1 V2.0.0 (2007-11)	3 years	ETSI TS 102 176-1 V2.1.1 (2011-07)	none [i.23]	< 2014-08-01
SHA-1	ETSI TS 102 176-1 V2.0.0 (2007-11)	1 year (b)	ETSI TS 102 176-1 V2.1.1 (2011-07)	February 2017 [i.24]	< 2012-08-01
SHA-224	ETSI TS 119 312 V1.5.1	Legacy (d)	ETSI TS 119 312 V2.1.1 (2026-06) (the present document)	none	< 2026-01-01
WHIRLPOOL	ETSI TS 102 176-1 V2.1.1 (2011-07)	6 years (c)	ETSI TS 119 312 V1.1.1 (2014-11)	none	< 2020-12-01
(a)	As last stated by the specification in the preceding column.				
(b)	Resistance for 3 years was listed as "unknown", and 6 years as "unusable".				
(c)	Resistance for up to 10 years was speculatively listed as "usable".				
(d)	Insufficient security margin against quantum attacks and low performance advantage over SHA-256.				

Table D.2: Discontinued signature algorithm and key size combinations

Algorithm	Key size	Last listed as recommended in	Resistance (a)	Not recommended since	First known practical attack	Suggested cryptographic constraint
DSA	1 024 bits	ETSI TS 102 176-1 V2.1.1 (2011-07)	1 year	ETSI TS 119 312 V1.1.1 (2014-11)	<i>none</i>	< 2015-12-01
RSA (b)	786 bits	ETSI TS 102 176-1 V1.2.1 (2005-07)	3 years	ETSI TS 102 176-1 V2.0.0 (2007-11)	August 2010 [i.25]	< 2010-08-01
RSA (b)	1 024 bits	ETSI TS 102 176-1 V2.0.0 (2007-11)	1 year (c)	ETSI TS 102 176-1 V2.1.1 (2011-07)	<i>none</i>	< 2019-10-01 (c)
RSA (b)	1 536 bits	ETSI TS 119 312 V1.1.1 (2014-11)	1 year	ETSI TS 119 312 V1.2.2 (2018-09)	<i>none</i>	< 2019-10-01
RSA (b)	< 3 000 bits	ETSI TS 119 312 V1.5.1 (2024-12)	Legacy	ETSI TS 119 312 V2.1.1	<i>none</i>	< 2026-01-01
ECDSA	163 bits	ETSI TS 102 176-1 V2.0.0 (2007-11)	1 year	ETSI TS 102 176-1 V2.1.1 (2011-07)	<i>none</i>	< 2012-08-01
ECDSA	224 bits	ETSI TS 119 312 V1.1.1 (2014-11)	3 years	ETSI TS 119 312 V1.2.2 (2018-09)	<i>none</i>	< 2021-10-01
EC-SDSA-opt	all	ETSI TS 119 312 V1.5.1 (2024-12)	N/A (d)	ETSI TS 119 312 V2.1.1	<i>none</i>	< 2026-06-01
(a) As last stated by the specification in the preceding column.						
(b) Regardless of padding scheme, i.e. for both PKCS#1-v1.5 and PSS.						
(c) RSA with 1 024 bits was still stated as being secure for up to 1 year in ETSI TS 102 176-1 V2.1.1 (2011-07), clause 9.3, note 2 and ETSI TS 119 312 V1.1.1 (2014-11) clause 9.3, note 5. This statement was removed with ETSI TS 119 312 V1.2.2 (2018-09).						
(d) Removed due to lack of support in widely deployed libraries (interoperability issues), not due to cryptographic weakness.						

Certain signature suites (i.e. combinations of hash algorithms and signature algorithms) had recommendations that did not match the combined minimum of the separate individual recommendations for the hash algorithm and signature algorithm. Such special-case recommendations are listed in Table D.3.

Table D.3: Discontinued signature suites (special cases)

Signature suite	Key size	Last listed as recommended in	Resistance (a)	Not recommended since	First known practical attack	Suggested cryptographic constraint
RSASSA-PSS with mgf1SHA-1Identifier	1 536 bits	ETSI TS 119 312 V1.1.1 (2014-11)	1 year	ETSI TS 119 312 V1.2.2 (2018-09)	<i>none</i>	< 2019-10-01
(a) As last stated by the specification in the preceding column.						

Annex E (informative): Bibliography

- [IETF RFC 3526](#): "More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)".
- [NIST FIPS Publication 203 \(August 2024\)](#): "Module-Lattice-Based Key-Encapsulation Mechanism Standard".
- [ETSI TR 103 619 \(V1.1.1\)](#): "CYBER; Migration strategies and recommendations to Quantum Safe schemes".
- [BSI TR-02102-1](#): "Cryptographic Mechanisms: Recommendations and Key Lengths", Version 2026-01.

History

Version	Date	Status
V1.1.1	March 2003	Publication as ETSI SR 002 176
V1.2.1	July 2005	Publication as ETSI TS 102 176-1 (Historical)
V2.0.0	November 2007	Publication as ETSI TS 102 176-1 (Historical)
V2.1.1	July 2011	Publication as ETSI TS 102 176-1 (Historical)
V1.1.1	November 2014	Publication
V1.2.1	May 2017	Publication
V1.2.2	September 2018	Publication
V1.3.1	February 2019	Publication
V1.4.1	August 2021	Publication
V1.4.2	February 2022	Publication
V1.4.3	August 2023	Publication
V1.5.1	December 2024	Publication
V2.1.1	June 2026	Publication