ETSI TS 119 312 V1.4.1 (2021-08)

TECHNICAL SPECIFICATION

Electronic Signatures and Infrastructures (ESI); Cryptographic Suites

Reference

RTS/ESI-0019312v141

Keywords

e-commerce, electronic signature, security,
trust services

ETSI

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

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

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

Contents

Intellectual Property Rights

Essential patents

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

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

Trademarks

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

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

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Electronic Signatures and 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 SOG-IS Crypto Evaluation Scheme [14]. The SOG-IS Crypto WG is in charge of providing requirements and evaluation procedures related to cryptographic aspects of Common Criteria security evaluations of IT products. To avoid conflicts between the evaluation of security product for qualified trust services and the recommendation given in the present document, the ETSI Technical Committee Electronic Signatures and Infrastructures (ESI) decided to refer for the trust services [i.12], article 3 (16a) consisting of creation, verification, and validation of electronic signatures, electronic seals and electronic time stamps, electronic registered delivery services and certificates related to those services to the SOG-IS Crypto Evaluation Scheme [14].

Other standardization bodies, security agencies and supervisory authorities of the Member States have published guidance documents with partially overlapping scope, for instance (but not limited to) France [i.2] and Germany [i.14].

1 Scope

The present document lists cryptographic suites used for the creation and validation of digital signatures and electronic time stamps and related certificates. The present document builds on the agreed cryptographic mechanisms from SOG-IS [14]. It may be used also for electronic registered delivery services in the future.

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 SOG-IS 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.

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 at https://docbox.etsi.org/Reference/.

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] FIPS Publication 180-4 (August 2015): "Secure Hash Standard (SHS)", National Institute of Standards and Technology.

[2] FIPS Publication 186-4 (July 2013): "Digital Signature Standard (DSS)", National Institute of Standards and Technology.

[3] IETF RFC 8017 (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 (2010): "Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation".

[6] ANSI X9.62 (2005): "Public Key Cryptography for the Financial Services Industry, The Elliptic Curve Digital Signature Algorithm (ECDSA)".

[7] IETF RFC 3279 (2002): "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile".

NOTE: Updated by IETF RFC 4055, IETF RFC 4491, IETF RFC 5480 and IETF RFC 5758.

[8] IETF RFC 4055 (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 (2010): "Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS)".

[10] IETF RFC 6931 (2013): "Additional XML Security Uniform Resource Identifiers (URIs)".

[11] W3C® Recommendation 11 April 2013: "XML Encryption Syntax and Processing Version 1.1".

NOTE: Available at https://www.w3.org/TR/2013/REC-xmlenc-core1-20130411.

[12] IETF RFC 3161 (2001): "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)".

NOTE: Updated by IETF RFC 5816.

[13] IETF RFC 6960 (2013): "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP".

NOTE: Updates IETF RFC 2560 and IETF RFC 6277.

[14] SOG-IS Crypto Working Group: "SOG-IS Crypto Evaluation Scheme - Agreed Cryptographic Mechanisms" Version 1.2, January 2020.

NOTE: Available at https://www.sogis.org/uk/supporting_doc_en.html.

[15] FIPS Publication 202 (August 2015): "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions", National Institute of Standards and Technology.

NOTE: Available at https://dx.doi.org/10.6028/NIST.FIPS.202.

[16] IETF RFC 5480 (2009): "Elliptic Curve Cryptography Subject Public Key Information".

[17] NIST: "Computer Security Objects Register (CSOR)".

NOTE: Available at https://csrc.nist.gov/groups/ST/crypto_apps_infra/csor/algorithms.html.

[18] IETF RFC 3526: "More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)".

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

[i.1] European Network of Excellence in Cryptology, ECRYPT Report on Algorithms, Key Size and Protocols Report (2018), ECRYPT - Coordination & Support, Action D5.4.

[i.2] Agence nationale de la sécurité des systèmes d'information: "Référentiel Général de Sécurité version 2.0" (2014-06).

NOTE: Annex B1 (version 2.03 of 2014-02) is available at https://www.ssi.gouv.fr/uploads/2014/11/RGS_v-2-0_B1.pdf.

[i.3] Void.

[i.4] Void.

[i.5] ISO/IEC 10118-3 (2004): "Information technology -- Security techniques - Hash functions -- Part 3: Dedicated hash functions".

NOTE: This ISO Standard duplicates the standardization from FIPS Publication 180-4 [1].

[i.6] ETSI TS 101 733 (V2.2.1) (04-2013): "Electronic Signatures and Infrastructures (ESI); CMS Advanced Electronic Signatures (CadES)".

[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" (omits comments).

NOTE: Available at https://www.w3.org/TR/2001/REC-xml-c14n-20010315.

[i.11] W3C® Recommendation 18 July 2002: "Exclusive XML Canonicalization Version 1.0" (with comments).

NOTE: Available at https://www.w3.org/TR/2002/REC-xml-exc-c14n-20020718.

[i.12] Regulation (EU) No 910/2014 of the European Parliament and of the Council, July 2014.

[i.13] OID Repository.

NOTE 1: Available at http://oid-info.com.

NOTE 2: 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 handle with care. Nevertheless it provides usually the link to corresponding official registration authority.

[i.14] Bundesamt für Sicherheit in der Informationstechnik, BSI TR-02102: "Cryptographic Mechanisms, version" (2017-01).

NOTE: Available at https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Technische-Richtlinien/TR-nach-Thema-sortiert/tr02102/tr02102_node.html.

[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] ANSSI: "Publication d'un paramétrage de courbe elliptique visant des applications de passeport électronique et de l'administration électronique française", October 2011.

NOTE: Available at https://www.ssi.gouv.fr.

[i.17] ETSI EN 319 122 (parts 1 and 2): "Electronic Signatures and Infrastructures (ESI); CAdES 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 Infrastructures (ESI); PAdES digital signatures".

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

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].

legacy mechanism: mechanism deployed on a large scale, currently offering a security level for an acceptable short-term security but no longer representing the cryptographic state of the art

NOTE 1: As defined in [14].

NOTE 2: As a consequence, a validity period is defined for legacy mechanisms.

recommended mechanism: mechanism, that fully reflects the state of the art in cryptography, providing an adequate level of security against all presently known or conjectured threats even taking into account the generally expected increases in computing power

NOTE: As defined in [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

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:

ANSI American National Standards Institute
ANSSI Agence Nationale de la Sécurité des Systèmes d'Information (National Agency for Security of Information Systems)
CA Certification Authority
CMS Cryptographic Message Syntax
CRL Certificate Revocation List
CSOR Cryptographic Algorithm Object Registration
DER Distinguished Encoding Rules

NOTE: Syntax rules for ASN.1.

DSA Digital Signature Algorithm
EC Elliptic Curve
ECC Elliptic Curve Cryptography
ECDSA Elliptic Curve Digital Signature Algorithm
EC-DSA Elliptic Curve Digital Signature Algorithm
ESI Electronic Signatures and Infrastructure

NOTE: Technical Committee of ETSI

FIPS Federal Information Processing Standard
IETF Internet Engineering Task Force
ISO International Organization for Standardization
IT Information Technology
MGF Mask Generation Function

NIST	National Institute of Standards and Technology
OCSP	Online Certificate Status Protocol
OID	Object Identifier
PKCS	Public-Key Cryptography Standards
PSS	Probabilistic Signature Scheme
RFC	Request for Comments
RNG	Random Number Generator
RSA	Rivest, Shamir and Adleman algorithm
SHA	Secure Hash Algorithm
SOG-IS	Senior Officials Group Information Systems Security
TST	Time-Stamp Token
TSU	Time-Stamping Unit
URI	Uniform Resource Identifier
URN	Uniform Resource Number
WG	Working Group
XML	eXtensible Markup Language

4 Use of SOG-IS 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 SOG-IS document [14].

The SOG-IS Evaluation Scheme 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 [14].

In general, only SOG-IS 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). SOG-IS 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 the SOG-IS Crypto Evaluation Scheme [14] with revisions on a two-year base. This coincides with the established schedule in ETSI ESI.

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 SOG-IS Agreed Cryptographic Mechanisms [14]. The present document provides additional recommendations in the following clauses.

Table 1: Agreed Hash Functions [14], p. 13

Short hash function name	References	R/L
SHA-224	FIPS Publication 180-4 [1]	L[2025]
SHA-256	FIPS Publication 180-4 [1]	R
SHA-384	FIPS Publication 180-4 [1]	R
SHA-512	FIPS Publication 180-4 [1]	R
SHA-512/256	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 SHA hash functions

5.2.1 SHA-512/256

SHA-512/256 should not be used if SHA3-256 or SHA-512 can be used instead without truncation.

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

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 SOG-IS Agreed Cryptographic Mechanisms [14]. The present document provides additional recommendations and requirements in the following clauses.

Table 2: Agreed Digital Signature Algorithms [14], p. 28

Short signature algorithm name	References	R/L
RSA-PKCS#1v1_5	IETF RFC 8017 [3]	L
RSA-PSS	IETF RFC 8017 [3]	R
DSA (FF-DLOG DSA)	FIPS Publication 186-4 [2], ISO/IEC 14888-3 [4]	R
EC-DSA (EC-DLOG EC-DSA)	FIPS Publication 186-4 [2]	R
EC-SDSA-opt (EC-DLOG EC-Schnorr)	ISO/IEC 14888-3 [4]	R
NOTE: The notation given in parentheses is given in the SOG-IS document [14].		

NOTE: Although EC-GDSA is a SOG-IS recommended mechanism for interoperability reasons the EC-GDSA algorithm is not listed in table 2 due to the low dissemination in trust services.

6.2.2 Signature algorithms

6.2.2.1 RSA

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

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

6.2.2.2 DSA

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

NOTE: The dissemination of DSA in trust services is low. Therefore it is suggested to use other more widely deployed algorithms unless it is the only alternative for interoperability. Due to this fact signature suites based on DSA are not listed in clause 7.3 and in annex A.

6.2.2.3 EC based DSA algorithms

The EC-DSA algorithm shall be used (SOG-IS recommended mechanism) if the key length is chosen according to clause 8.

EC-DSA and EC-SDSA-opt shall be used (SOG-IS recommended mechanisms) only if the elliptic curves are selected from the following table 3.

When used, the algorithms shall be as specified by the references provided in table 3.

Table 3: Agreed Elliptic Curve Parameters [14], p. 25

Curve family	Short curve name	References	R/L
FR	FRP256v1	ANSSI [i.16]	R
Brainpool	brainpoolP256r1	IETF RFC 5639 [5]	R
	brainpoolP384r1	IETF RFC 5639 [5]	R
	brainpoolP512r1	IETF RFC 5639 [5]	R
NIST	P-256	FIPS Publication 186-4 [2]	R
	P-384	FIPS Publication 186-4 [2]	R
	P-521	ISO/IEC 14888-3 [4]	R

For interoperability reasons only one version (EC-SDSA-opt) from the EC-XDSA Schnorr variants defined in ISO/IEC 14888-3 [4] is selected by the present document. EC-SDSA in the optimized version has the small advantage of minimal data transfer for smart cards.

NOTE: Due to former patent issues (the U.S. Patent 4,995,082 expired in February 2008) Schnorr signatures are not commonly used. Nevertheless they have the following advantages: firstly the signing equation is simpler (allowing for some optimizations) and secondly the hash function is applied to the concatenation of the ephemeral key and the data to be signed, i.e. it implements randomized hashing. With this property Schnorr signatures can be proved secure in the random oracle model. There is also a proof in the generic group model.

6.3 Key generation

The key generation shall follow the recommendations and requirements in their normative references of table 2.

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 [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 or SHA-512/256 gives no security advantage over SHA-512, because they are truncated derivations of the SHA-512 algorithm. Nevertheless they are 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 ([2], p.37).

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	SOGIS-recommended/ legacy ([14], p. 28)
sha224-with-rsa	SHA-224	RSA-PKCSv1_5	L
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 mgf1SHA-256Identifier	SHA-256	RSA-PSS	R
rsa-pss with mgf1SHA-384Identifier	SHA-384	RSA-PSS	R
rsa-pss with mgf1SHA-512Identifier	SHA-512	RSA-PSS	R
rsa-pss with mgf1SHA3-Identifier	SHA3-256, SHA3-384 or SHA3-512	RSA-PSS	R
sha224-with-ecdsa	SHA-224	EC-DSA	L
sha2-with-ecdsa	SHA-256, SHA-384 or SHA-512	EC-DSA	R
sha2-with-ecsdsa	SHA-256, SHA-384 or SHA-512	EC-SDSA-opt	R
sha3-with-ecdsa	SHA3-256, SHA3-384 or SHA3-512	EC-DSA	R
sha3-with-ecsdsa	SHA3-256, SHA3-384 or SHA3-512	EC-SDSA-opt	R

8 Hash functions and key sizes versus time

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 8.3, hash functions versus time are recommended.
- In clause 8.4, key sizes versus time are recommended.

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 SOG-IS Crypto Evaluation Scheme [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) that renders it useless can theoretically not completely be excluded but "breakthroughs" of that kind are regarded as improbable. In contrast to that certain unforeseen advances of moderate degree in analysing cryptographic algorithms are regarded as a realistic threat (see the SHA-1 issue, where a substantial progress was made in 2005 reducing the time complexity from 2^{80} to 2^{69} and breaking at last SHA-1 in 2017). The security margin chosen by the SOG-IS document is so that advances of this level are expected to be compensated without changing the parameters.

NOTE 3: Stability of the requirements in the present document is highly desirable for reasons of planning reliability. This means that if in e.g. 2021 a key length *y* is declared as suitable for 3 years, i.e. at least until the end of 2024, an updated version in e.g. 2022 normally still declares this key length *y* as sufficient at least until the end of 2024. The following tables contain recommendations for the lifetime of keys and were chosen according to the SOG-IS Crypto Evaluation Scheme [14].

An attempt was made to achieve roughly similar security for all the components. SOG-IS recommended mechanisms should provide at least 125 bits of security against offline attacks. 100 bits of security may be used by SOG-IS legacy mechanisms, but they provide a lower security margin.

8.3 Hash functions versus time

The hash functions listed in table 5 are expected to remain suitable during X years.

Table 5: Recommended hash functions for a resistance during X years

Entry name of the hash function	1 year	3 years	6 years
SHA-224	usable	usable	unusable
SHA-256	usable	usable	usable
SHA-384	usable	usable	usable
SHA-512	usable	usable	usable
SHA3-256	usable	usable	usable
SHA3-384	usable	usable	usable
SHA3-512	usable	usable	usable

8.4 Recommended key sizes versus time

The parameters defined in following tables should be used.

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

Table 6: Recommended parameters for RSA for a resistance during X years

Parameter	1 year	3 years	6 years
Key size ($\log_2(n)$)	≥ 1 900	≥ 1 900	≥ 3 000

NOTE 1: A recommendation for RSA of the form "Key size greater or equal *y* for a resistance during 3 years" means "Key size should be at least *y* for RSA keys with an intended life time of 3 years (i.e. until end of 2024)".

The security parameters for DSA are the bit length *pLen* of the field characteristic *p*. The groups from IETF RFC 3526 [18] are recommended.

Table 7: Recommended parameters for DSA for a resistance during X years

Parameter	1 year	3 years	6 years
pLen	2 048	2 048	3 072

The security parameters for EC-DSA and EC-SDSA-opt are commonly the bit length *pLen* of the field characteristic *p* and *qLen* of the order *q* of the generator of the elliptic curve. They are equal for all recommended elliptic curves, therefore there is only one entry here.

Table 8: Recommended parameters for EC-DSA and EC-SDSA-opt for a resistance during X years

Parameter	1 year	3 years	6 years
pLen = qLen	256, 384 or 512	256, 384 or 512	256, 384 or 512

Table 9 summarizes the recommendations from the above tables.

Table 9: Recommended signature suites for algorithm resistance during X years (was table 12 in version 1.1.1)

Entry name of the signature suite	1 year	3 years	6 years
sha256-with-rsa	≥ 1 900	≥ 1 900	not recommended
sha384-with-rsa	≥ 1 900	≥ 1 900	not recommended
sha512-with-rsa	≥ 1 900	≥ 1 900	not recommended
rsa-pss with mgf1SHA-256Identifier	≥ 1 900	≥ 1 900	≥ 3 000
rsa-pss with mgf1SHA-384Identifier	≥ 1 900	≥ 1 900	≥ 3 000
rsa-pss with mgf1SHA-512Identifier	≥ 1 900	≥ 1 900	≥ 3 000
rsa-pss with mgf1SHA3-Identifier	≥ 1 900	≥ 1 900	≥ 3 000
sha256-with-dsa	2 048	2 048	3 072
sha512-with-dsa	2 048	2 048	3 072
sha224-with-ecdsa	legacy		not recommended
sha2-with-ecdsa	recommended		
sha2-with-ecsdsa	recommended		
sha3-with-ecdsa	recommended		
sha3-with-ecsdsa	recommended		

Table 10 provides the absolute dates for the recommendations from table 9.

Table 10: Recommended signature suites for a resistance up to year X

Entry name of the signature suite	2023	2024	2025	after 2025
sha256-with-rsa	≥ 1 900	≥ 1 900	not recommended	
sha384-with-rsa	≥ 1 900	≥ 1 900	not recommended	
sha512-with-rsa	≥ 1 900	≥ 1 900	not recommended	
rsa-pss with mgf1SHA-256Identifier	≥ 1 900	≥ 1 900	≥ 1 900	≥ 3 000
rsa-pss with mgf1SHA-384Identifier	≥ 1 900	≥ 1 900	≥ 1 900	≥ 3 000
rsa-pss with mgf1SHA-512Identifier	≥ 1 900	≥ 1 900	≥ 1 900	≥ 3 000
rsa-pss with mgf1SHA3-Identifier	≥ 1 900	≥ 1 900	≥ 1 900	≥ 3 000
sha256-with-dsa	2 048	2 048	2 048	3 072
sha512-with-dsa	2 048	2 048	2 048	3 072
sha224-with-ecdsa	legacy		not recommended	
sha2-with-ecdsa	recommended			
sha2-with-ecsdsa	recommended			
sha3-with-ecdsa	recommended			
sha3-with-ecsdsa	recommended			

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 fulfill 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

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

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

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 http://oid-info.com [i.13]. For example one gets the OID assigned for EC-SDSA in the optimized version by http://oid-info.com/get/1.0.14888.3.0.13.

10.2.2 Hash functions

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

Table 11

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 }	NIST CSOR [17]
id-sha3-256	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistalgorithm(4) hashalgs(2) 8 }	NIST CSOR [17]
id-sha3-384	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistalgorithm(4) hashalgs(2) 9 }	NIST CSOR [17]
id-sha3-512	{ joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistalgorithm(4) hashalgs(2) 10 }	NIST CSOR [17]

10.2.3 Elliptic curves

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

Table 12

Short object name	OID	References
FRP256v1	{iso(1) member-body(2) fr(250) type-org(1) 223 101 256 1}	ANSSI [i.16]
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

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]

10.2.5 Signature suites

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

Table 14

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]
id-dsa-with-sha224	{ joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) algorithms(4) sigAlgs (3) id-dsa-with-sha224(1) }	NIST CSOR [17]
id-dsa-with-sha256	{ joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) algorithms(4) sigAlgs(3) id-dsa-with-sha256(2) }	NIST CSOR [17]
ecdsa-with-SHA224	{ iso(1) member-body(2) us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-Specified(3) 1 }	ANSI X9.62 [6]
ecdsa-with-SHA256	{ iso(1) member-body(2) us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-Specified(3) 2 }	ANSI X9.62 [6]
ecdsa-with-SHA384	{ iso(1) member-body(2) us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-Specified(3) 3 }	ANSI X9.62 [6]
ecdsa-with-SHA512	{ iso(1) member-body(2) us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-Specified(3) 4 }	ANSI X9.62 [6]
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-dswa-dl-EC-SDSA-opt	{iso(1) standard(0) digital-signature-with-appendix(14888) part3(3) algorithm(0) id-dswa-dl ec-sdsa-opt(13) }	ISO/IEC 14888-3 [4]
NOTE 1: 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. NOTE 2: ISO/IEC 14888-3 [4] defined hash-independent OIDs for the EC-XDSA algorithms. So the OID for EC-SDSA-opt algorithm 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

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]

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

Short object name	URI	References
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]

NOTE: The URI rsapss-with-parameters allows also the parametrization with SHA-3.

10.4 Recommended hash functions and signature algorithms objects without a URN description

The signature suite using signature algorithm EC-DSA and a SHA3 hash function does not have a URN yet.

The signature algorithm EC-SDSA and therefore all signature suites based on it do not have an URN yet.

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.

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

CadES [i.6] and PadES [i.8]	***Issuers* of AdES**	***Users* of AdES**
Hash functions	shall support SHA-256 should support SHA-512	shall support SHA-256, SHA-384, SHA-512 should support SHA3
Signature algorithms	should support RSA-PKCS1v1_5 or RSA-PSS or EC-DSA or EC-SDSA	shall support RSA-PKCS1v1_5 shall support RSA-PSS shall support EC-DSA should support EC-SDSA

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	shall support SHA-256, SHA-384, SHA-512 should support SHA3
Signature algorithms	should support RSA-PKCS1v1_5 or RSA-PSS or EC-DSA	shall support RSA-PKCS1v1_5 shall support RSA-PSS shall support EC-DSA

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.

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**
Signer public keys	should support RSA or EC-DSA	shall support RSA shall support EC-DSA should support EC-SDSA
CA issuing keys	shall support RSA with SHA-256 or ECDSA with SHA-256	shall support RSA with SHA-256 or SHA-512 shall support EC-DSA with SHA-256

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

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 EC-DSA with SHA-224 shall support RSA with SHA-256 or SHA-512 shall support EC-DSA with SHA-256

NOTE: 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.

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 SHA-256 with RSA	shall support RSA with SHA-256 or SHA-512 shall support EC-DSA with SHA-256

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 with SHA-256	shall support RSA with SHA-256 and SHA-512 shall support EC-DSA with SHA-256
Issuer CA public keys	should support RSA with SHA-256 or SHA-512	shall support RSA with SHA-256 and SHA-512 shall support EC-DSA with SHA-256

NOTE: 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.

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: Self-signed certificates need to resist quite long (e.g. more than 10 years).

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 or SHA-512 should support EC-DSA with SHA-256 should support RSA with SHA3	shall support RSA with SHA-256 or SHA-512 shall support EC-DSA with SHA-256 should support RSA with SHA3

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 time stamps

Time-Stamping Tokens	TST requesters	TST issuers	TST verifiers
Hash function	shall support SHA-256	shall support SHA-256	shall support SHA-256
TST signature algorithms	shall support RSA with SHA-256 or SHA-512	shall support RSA with SHA-256 or SHA-512 should support EC-DSA with SHA-256	shall support RSA with SHA-256 or SHA-512 should support EC-DSA with SHA-256

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 time stamping units

TSU certificates	*Issuers* of TSU certificates	*Users* of TSU certificates
TSU public key	should support RSA with SHA-256 or SHA-512 should support EC-DSA with SHA-256	shall support RSA with SHA-256 or SHA-512 should support EC-DSA with SHA-256
Issuer CA public keys	shall support RSA with SHA-256 or SHA-512 should support EC-DSA with SHA-256	shall support RSA with SHA-256 or SHA-512 should support EC-DSA with SHA-256

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).

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.

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.

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

Machine processable formats (DER or XML encoded) are under development and may be included in a future version of the present document.

History

Document history		
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