



DPC: 23 / 30447023 DC

BSI Group Headquarters

389 Chiswick High Road London W4 4AL

Tel: +44 (0)20 8996 9000

Fax: +44 (0)20 8996 7400

www.bsigroup.com

Date: 05 January 2023

Origin: International

Latest date for receipt of comments: 18 February 2023

Project No. 2021/03094

Responsible committee: IST/6 Data communications

Interested committees:

Title: Draft BS EN ISO/IEC 9594-12 Information technology — Open systems

interconnection

Part 12: The Directory: Public key infrastructure establishment and maintenance

Please notify the secretary if you are aware of any keywords that might assist in classifying or identifying the standard or if the content of this standard

- i) has any issues related to 3rd party IPR, patent or copyright
- ii) affects other national standard(s)
- iii) requires additional national guidance or information

**WARNING: THIS IS A DRAFT AND MUST NOT BE REGARDED OR USED AS A BRITISH STANDARD.
THIS DRAFT IS NOT CURRENT BEYOND 18 February 2023**

This draft is issued to allow comments from interested parties; all comments will be given consideration prior to publication. No acknowledgement will normally be sent. **See overleaf for information on the submission of comments.**

No copying is allowed, in any form, without prior written permission from BSI except as permitted under the Copyright, Designs and Patent Act 1988 or for circulation within a nominating organization for briefing purposes. Electronic circulation is limited to dissemination by e-mail within such an organization by committee members.

Further copies of this draft may be purchased from BSI Shop <http://shop.bsigroup.com> or from BSI Customer Services, Tel: +44(0) 20 8996 9001 or email cservices@bsigroup.com. British, International and foreign standards are also available from BSI Customer Services.

Information on the co-operating organizations represented on the committees referenced above may be obtained from <http://standardsdevelopment.bsigroup.com>

Introduction

This draft standard is based on national and international discussions. Your comments on this draft are invited and will assist in the preparation of the consequent standard.

For international standards, comments will be reviewed by the relevant UK national committee before sending the consensus UK vote and comments to the international committee, which will then decide appropriate action. If the international standard is approved, it is usual for the text to be published as a British Standard.

For national standards, comments will be reviewed by the relevant UK national committee and the resulting standards published as a British Standard.

UK Vote

Please indicate whether you consider the UK should submit a negative (with supporting technical reasons) or positive vote on this draft. Please indicate if you are aware of any reason why this draft standard should not be published as a British Standard.

Submission of Comments

- **Annotated drafts are not acceptable and will be rejected.**
- All comments should be submitted online at <https://standardsdevelopment.bsigroup.com/>. You will need to register in order to comment.

Template for comments and secretariat observations

Date: xx/xx/20xx	Document: ISO/DIS xxxx
------------------	------------------------

1	2	(3)	4	5	(6)	7
MB	Clause No./ Subclause No./Annex (e.g. 3.1)	Paragraph/Figure/ Table/Note	Type of comment	Commend (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
	3.1	Definition 1	ed	Definition is ambiguous and needs clarifying.	Amend to read '...so that the mains connector to which no connection...'	
	6.4	Paragraph 2	te	The use of the UV photometer as an alternative cannot be supported as serious problems have been encountered in its use in the UK.	Delete reference to UV photometer.	

DRAFT INTERNATIONAL STANDARD

ISO/IEC DIS 9594-12

ISO/IEC JTC 1/SC 6

Secretariat: **KATS**

Voting begins on:
2022-12-26

Voting terminates on:
2023-03-20

Information technology — Open systems interconnection —

Part 12:

The Directory: Key management and public-key infrastructure establishment and maintenance

ICS: 35.100.70

THIS DOCUMENT IS A DRAFT CIRCULATED FOR COMMENT AND APPROVAL. IT IS THEREFORE SUBJECT TO CHANGE AND MAY NOT BE REFERRED TO AS AN INTERNATIONAL STANDARD UNTIL PUBLISHED AS SUCH.

IN ADDITION TO THEIR EVALUATION AS BEING ACCEPTABLE FOR INDUSTRIAL, TECHNOLOGICAL, COMMERCIAL AND USER PURPOSES, DRAFT INTERNATIONAL STANDARDS MAY ON OCCASION HAVE TO BE CONSIDERED IN THE LIGHT OF THEIR POTENTIAL TO BECOME STANDARDS TO WHICH REFERENCE MAY BE MADE IN NATIONAL REGULATIONS.

RECIPIENTS OF THIS DRAFT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.

This document is circulated as received from the committee secretariat.



Reference number
ISO/IEC DIS 9594-12:2022(E)
Documents
AVA Document Sharing Hub
© ISO/IEC 2022



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2022

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

1	Scope	1
2	Normative references	1
2.1	Identical Recommendations International Standards	1
2.2	Recommendations	1
2.3	Others	1
3	Definitions	2
3.1	OSI reference model security architecture definitions	2
3.2	Baseline identity management terms and definitions	2
3.3	Terms defined in public-key and attribute certificate frameworks	2
3.4	Key management and public-key infrastructure establishment and maintenance definitions	3
4	Abbreviations	4
5	Conventions	6
6	Cyber security consideration for communication networks	6
6.1	The challenge of large ICT networks	6
6.2	Connection-mode communication	7
6.2.1	General	7
6.2.2	Association establishment phase	8
6.2.3	Data transfer phase	8
6.2.4	Association termination phase	9
6.3	Symmetric key requirements	9
7	Introduction to cryptographic algorithms	10
8	Hash algorithms	11
8.1	General	11
8.2	The SHA-2 series of hash algorithms	12
8.2.1	Referenced specifications	12
8.2.2	The SHA-2 model	12
8.2.3	The SHA-2 characteristics	13
8.2.4	SHA-2 formal specification	13
8.3	The KECCAK algorithms	14
8.4	SHA-3 characteristics	16
8.5	SHA-3 formal specifications	17
8.5.1	SHA3-224 specification	17
8.5.2	SHA3-256 specification	17
8.5.3	SHA3-256 specification	17
8.5.4	SHA3-512 specification	17
8.5.5	SHAKE-128 and SHAKE-256 specifications	17
8.5.6	SHAKE-128-len and SHAKE-256-len specifications	17
9	Public-key algorithms and digital signature algorithms	17
9.1	General	17
9.2	The RSA public-key algorithm	19
9.2.1	General	19
9.2.2	Key generation	19
9.2.3	Security considerations	20
9.2.4	RSA asynchronous encryption/decryption	20
9.2.5	Signature generation and verification	20
9.3	The DSA public-key algorithm	21
9.4	The elliptic curve digital signature algorithm (ECDSA)	21
9.4.1	General	21
9.4.2	Defines curves	22
9.4.3	Key generation	22
9.4.4	Security considerations	22

9.4.5	Signature generation and verification	23
9.5	Edwards-curve digital signature algorithm	23
9.5.1	General	23
9.5.2	Defined digital signature algorithms	23
9.5.3	Key generation	24
9.5.4	Security issues	24
9.5.5	Signature generation and verification	24
10	Key establishment algorithms	24
10.1	Introduction	24
10.2	Key transport	25
10.2.1	RSA key transport	25
10.3	Key agreement	25
10.3.1	The Diffie-Hellman key agreement method	25
10.4	Key derivation functions	27
10.4.1	General	27
10.4.2	HMAC-based extract-and-expand key derivation function	27
11	Symmetric-key algorithms	28
11.1	Stream ciphers vs. block ciphers	28
11.2	Advanced encryption standard	28
11.2.1	General	28
11.2.2	The octet substitution layer	30
11.2.3	The Shift Rows layer	31
11.2.4	The mix columns layer	32
11.2.5	The add round key layer	32
11.3	Modes of advance encryption standard	32
11.3.1	Overview of AES modes	32
11.3.2	Advanced encryption standard – electronic codebook (AES-ECB) mode	33
11.3.3	Advanced encryption standard - cipher block chaining (AES-CBC)	33
11.3.4	Cipher feedback mode	34
11.3.5	Output feedback mode	34
11.3.6	Counter mode	34
12	Authenticated encryption with associated data (AEAD) algorithms	35
12.1	General	35
12.2	Advanced encryption standard – Galois/counter mode (AES-GCM)	35
12.3	Advanced encryption standard (AES)- Counter with CBC-MAC (CCM)	35
13	Integrity check value (ICV) algorithms	36
13.1	Introduction	36
13.2	Keyed-hash message authentication code (HMAC)	36
13.3	Cipher-based message authentication code (CMAC)	37
13.4	КЕССАК message authentication code (KMAC)	37
13.4.1	General	37
13.4.2	Input message constructions	38
13.4.3	KMAC formal specifications	38
13.5	Advance encryption standard (AES) - Galois message authentication code (GMAC) algorithm	39
14	Security strength	39
15	Random number generation and entropy	40
16	Post-quantum considerations	40
17	Hardware security modules	41
18	Introduction to the section	42
19	Public-key certificates	42
19.1	Content of a basic public-key certificate	42
19.1.1	Structure	42

19.1.2	Distinguished name considerations.....	42
19.1.3	Version component.....	43
19.1.4	Serial number component.....	43
19.1.5	Signature component.....	43
19.1.6	Issuer component.....	44
19.1.7	Validity component.....	44
19.1.8	Subject component.....	44
19.1.9	Subject public-key information.....	44
19.1.10	Issuer unique ID and subject unique ID.....	44
19.2	Extensions.....	45
19.2.1	Use of extensions for public-key certificates.....	45
19.2.2	Basic constraints extension.....	45
19.2.3	Key usage extension.....	45
19.2.4	Subject directory attribute extension type.....	45
19.2.5	Subject alternative name extension.....	46
19.2.6	Authority information access extension.....	46
20	Trust establishment.....	46
20.1	General.....	46
20.2	Single public-key infrastructure domain.....	47
20.3	Trust establishment between two public-key infrastructure domains.....	48
20.4	A worldwide federated public-key infrastructure.....	48
20.5	Trust anchor compromise.....	49
21	Identity verification in a machine-to-machine environment.....	49
21.1	General.....	49
21.2	The two-PKIs approach.....	49
21.3	Use of the shared secret approach.....	50
21.4	Operational PKI establishment.....	50
22	PKI configurations.....	50
22.1	Introduction.....	50
22.2	PKI components.....	51
23	Certification path processing.....	51
23.1	Certification path.....	51
23.2	Name chaining.....	52
23.3	Key identifier chaining.....	52
24	Certification path validation.....	53
24.1	Overview.....	53
24.2	Validating certification path.....	54
24.3	Validating individual public-key certificates.....	55
	Annex A Size of Public-key certificates.....	57
	Annex B Basic mathematic concepts for cryptographic algorithms.....	61
	Bibliography.....	77

Information technology — Open systems interconnection —

Part 12:

The Directory: Key management and public-key infrastructure establishment and maintenance

SECTION 1 – GENERAL

1 Scope

This Recommendation | International Standard supplements Rec. ITU-T X.509 | ISO/IEC 9594-8 and Rec. ITU-T X.510 | ISO/IEC 9594-11 by providing extended analysis of cryptographic algorithms and guidance in establishment and maintenance of a public-key infrastructure (PKI).

It is outside the scope of this Recommendation | International Standard to define new cryptographic algorithms, but it is within scope to discuss already-defined cryptographic algorithms that provide optimal protection, including protection in the future with powerful quantum computers.

This Recommendation | International Standard specifies how public-key infrastructure (PKI) may be adapted to support machine-to-machine (M2M) communication, e.g., Smart grid and Internet of Things (IoT) to allow interworking.

This Recommendation | International Standard specifies the procedures for establishment and maintenance of a PKI supporting new areas, such as intelligent electricity net (smart grid), industrial internet of things.

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

2.2 Recommendations

None

2.3 Others

IETF RFC 8017 (2016), *PKCS #1: RSA Cryptography Specifications Version 2.2*.

IETF RFC 8032 (2017), *Edwards-Curve Digital Signature Algorithm (EdDSA)*.

NIST FIPS 186-4 (2013). *Digital Signature Standard (DSS)*.

NIST FIPS 198-1 (2008), *The Keyed-Hash Message Authentication Code (HMAC)*.

NIST SP 800-38C (2004), *Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality*.

NIST SP 800-38D (2007), *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*.

NIST SP 800-56A, Revision 3 (2018), *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography*.

NIST SP 800-185 (2016), *SHA-3 Derived Functions: cSHAKE, KMAC, TupleHash and ParallelHash*.

NIST SP 800-208 (2020), *Recommendation for Stateful Hash-Based Signature Schemes*.

3 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply:

3.1 OSI reference model security architecture definitions

Terms defined in Rec. ITU-T X.800 | ISO 7498-2:

- a) asymmetric (encipherment);
- b) authentication exchange;
- c) authentication information;
- d) confidentiality;
- e) credentials;
- f) cryptography;
- g) data origin authentication;
- h) decipherment;
- i) digital signature;
- j) encipherment;
- k) key;
- l) peer-entity authentication;
- m) symmetric (encipherment).

3.2 Baseline identity management terms and definitions

The following term is defined in Rec. ITU-T X.1252:

- a) trust.

3.3 Terms defined in public-key and attribute certificate frameworks

The following terms are defined in Rec. ITU-T X.509 | ISO/IEC 9594-8:

- a) CA certificate;
- b) certificate policy;

- c) certification practice statement (CPS);
- d) certificate revocation list (CRL);
- e) certificate serial number;
- f) certificate validation;
- g) certification authority (CA);
- h) certification path;
- i) cross-certificate;
- j) end entity;
- k) end-entity public-key certificate;
- l) key agreement;
- m) private key;
- n) public key;
- o) relying party;
- q) trust anchor;
- r) trust anchor information.

3.4 Key management and public-key infrastructure establishment and maintenance definitions

The following terms are defined in the Recommendation | International Standard:

3.4.1 extractor: An algorithm that transform a weakly random distribution into an almost uniform distribution-

3.4.2 brute-force attack: For cryptographic algorithms it is an exhaustive search of all possible combinations until a workable combination is found.

3.4.3 crypto agility: The ability to adapt to more secure cryptographic algorithms without significant changes to system structure.

3.4.4 fully qualified domain name: A domain name that specifies its exact location in the tree hierarchy of the domain name system (DNS)

3.4.5 integer factoring: Breaking a (large) integer down into prime factors when multiplied equal the original integer.

3.4.6 key agreement: A pairwise key-establishment algorithm in which the resulted shared secret is a function of information supplied by both participants.

3.4.7 key establishment: The procedure that results in secret keying material shared between communicating entities using either key agreement or key transport.

3.4.8 key transport: Secure transfer of a symmetric key to a peer communication partner.

3.4.9 national trust anchor: A trust anchor established by a recognized national organization that is considered trustworthy by all entities in the country.

3.4.10 nonce: A value that is used only once within a specified context.

3.4.11 public-key certificate owner: An entity that is the subject of a public-key certificate.

3.4.12 public-key infrastructure (PKI) domain: A PKI under a single management.

3.4.12 security strength: A measure expressed in bits of the expected number of operations of some kind required to crack a [cryptographic algorithm](#).

3.4.13 semiprime: An integer being the product of two prime integers.

3.4.14 signatory: The entity that digitally signs a document or a message.

3.4.15 Shor's algorithms: A quantum algorithm for factoring a (large) integer.

3.4.16 subject name: The name supplied in the **subject** component of a public-key certificate, and which is used to reference the entity holding the public-key certificate in question.

3.4.17 verifier: The entity that verifies a digital signature.

4 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply:

AEAD Authenticate Encryption with Associated Data

NOTE – Some specifications use the term Authenticate Encryption with Additional Data

AES Advanced Encryption Standard

CA Certification Authority

CBC Cipher Block Chaining

CRL Certificate Revocation List

CTR Counter

DNS Domain Name System

DSA Digital Signature Algorithm

ECC Elliptic Curve Cryptography

ECDH Elliptic Curve Diffie-Hellman

ECDLP Elliptic Curve Discrete Logarithm Problem

ECDSA Elliptic Curve Digital Signature Algorithm

EdDSA Edwards-curve Digital Signature Algorithm

FQDN Fully Qualified Domain Name

GCD	Greatest Common Divisor
HKDF	Hash-based Key Derivation Function
HMAC	Hash Message Authentication Code
HSM	Hardware Security Module
IANA	Internet Assigned Numbers Authority
ICV	Integrity Check Value
IIoT	Industrial Internet-of-Things
IoT	Internet of Things
IP	Internet Protocol
KDC	Key Distribution Centre
LCM	Least Common Multiple
MAC	Message Authentication Code
M2M	Machine-to-Machine
NIST	National Institute of Standards and Technology
OCSP	Online Certificate Status Protocol
OSI	Open Systems Interconnection
PKI	Public-Key Infrastructure
PRNG	Pseudo Random Number Generator
QKD	Quantum Key Distribution
RBG	Random Bit Generators
RDN	Relative Distinguished Name
RSA	Rivest-Shamir-Adelman
SHA	Secure Hash Algorithm
SHAKE	Secure Hash Algorithm KECCAK
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
XOF	Extendable-Output Function
XOR	eXclusive OR
UUID	Universal Unique Identifier

5 Conventions

This Recommendation | International Standard has been developed according to the rules for presentation of ITU-T | ISO/IEC common text of September 2014.

The term "Specification" (as in "this Specification") shall be taken to mean this Recommendation | International Standard.

The term "shall" is used when expressing requirement on implementation referencing this Specification. The term "must" is used to express requirements that by necessity has to be satisfied, e.g., expressing mathematical relationships.

This Specification makes extensive use of the abstract syntax notation one (ASN.1) for the formal specification of data types and values, as it is specified in Rec. ITU-T X.680 | ISO/IEC 8824-1, Rec. ITU-T X.681 | ISO/IEC 8824-2, Rec. ITU-T X.682 | ISO/IEC 8824-3, and Rec. ITU-T X.683 | ISO/IEC 8824-4.

This Recommendation | International Standard presents ASN.1 notation in the **courier new** typeface. When ASN.1 types and values are referenced in normal text, they are differentiated from normal text by presenting them in the **Courier New** typeface. Mathematical terms and equations are presented in the *Combria Math* typeface.

6 Cyber security consideration for communication networks

6.1 The challenge of large ICT networks

Large networks provide a particular challenge from a cyber security point of view, as it is necessary to have scalable specifications. Smart grid networks and Industrial Internet-of-Things (IIoT) networks are examples of large networks potentially including entities with constrained capabilities. This is the underlying subject for this Specification. A simplified example of such a network is illustrated in [Figure 1](#).

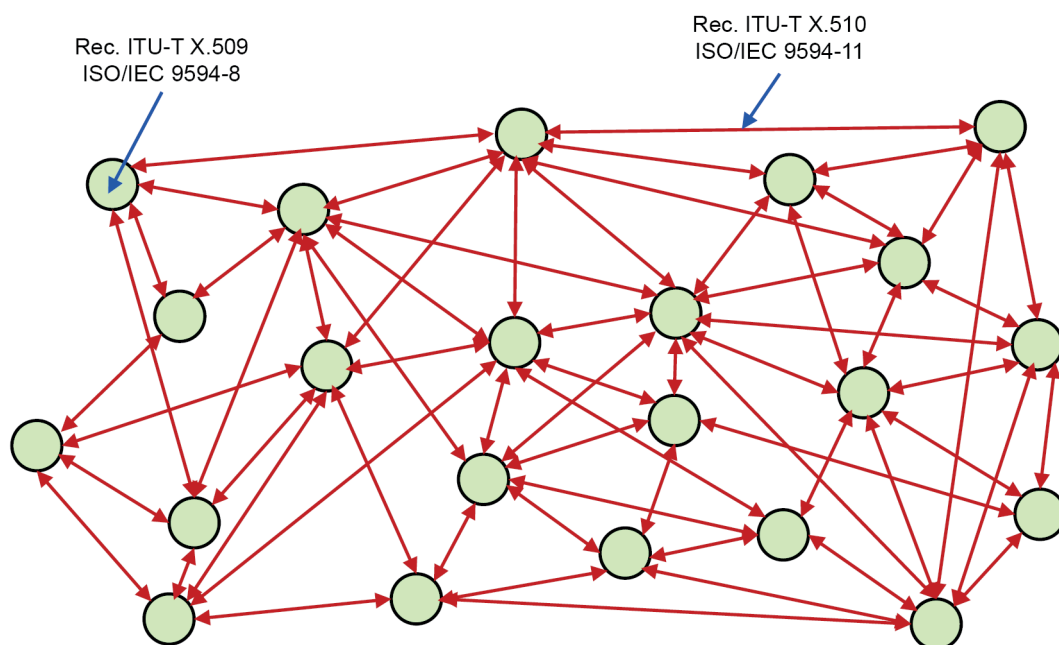


Figure 1 — Large interconnected ICT network

A large network is characterized by having many communication channels as illustrated in [Figure 1](#). The figure also illustrates the cryptographic relationship between Rec ITU-T X.509 | ISO/IEC 9594-8 and Rec ITU-T X.510 | ISO/IEC 9594-11. Rec ITU-T X.509 | ISO/IEC 9594-8 is the specification for

cryptographic capabilities associated with the entities, while Rec ITU-T X.510 | ISO/IEC 9594-11 is the specification for secure end-to-end communication between two entities.

The purpose of this Specification is to provide a bridge between the above two specifications to provide recommendations as to establish and maintain a PKI based on the specifications given by Rec ITU-T X.509 | ISO/IEC 9594-8 and to provide details behind cryptographic algorithms including the mathematics behind to the Rec ITU-T X.510 | ISO/IEC 9594-11 specification.

Security of a network is in general important. If the network is part of a critical infrastructure, security is even more crucial. This involves:

- On each connection the two partners need to authenticate each other and ensure integrity during initial handshaking by using asymmetric cryptography.
- The use of integrity check value (ICV, also called message authentication code (MAC) during data transfer requiring cryptographic symmetric keys.
- If confidentiality is required, additional cryptographic symmetric keys are required.

6.2 Connection-mode communication

6.2.1 General

To establish the scope of this Specification, the general Internet communication reference model is depict in [Figure 2](#) and is discussed in some details below.

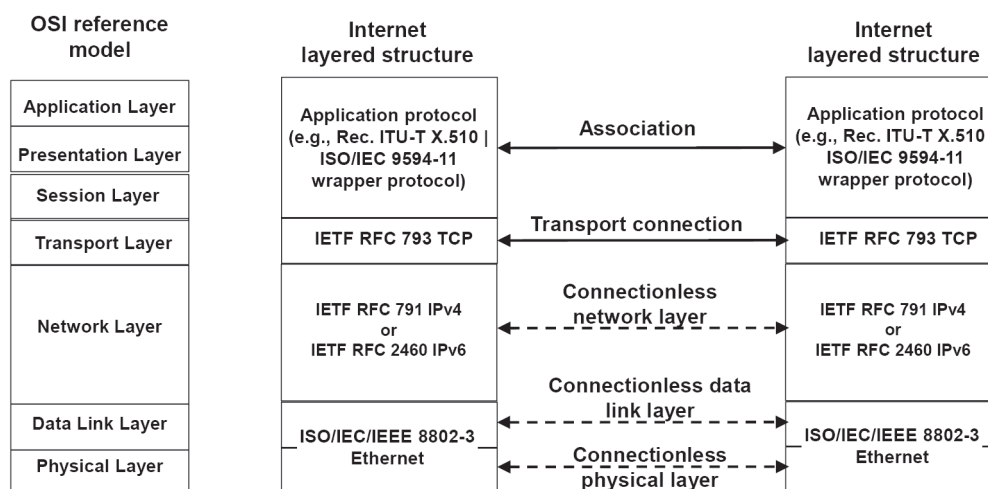


Figure 2 — Layered communication model

The Internet layered model and its relationship with the Open Systems Interconnection (OSI) reference model. The OSI reference model consists of seven layers with one or more protocols operating at each layer for given communication system. IETF in its definition of the internet communication reference model suite has collapsed the three upper layers of the OSI model into one application layer. The Internet communication reference model is illustrated by including a typical Internet protocol stack that provides support at each layer.

[Figure 2](#) also shows a simplified picture of the communication between peer layers. There are two different modes of operation for communications protocols:

- For a connection-oriented communication, explicit handshake is performed before the actual data transfer. The handshake is used to establish some common agreement and understanding before the actual data transfer.

- b) For a connection-less communication, data is transferred without a previously performed explicit handshake. There may be an implicit understanding, like both parties having the necessary software to handle the communication.

The three lower layers typically use connectionless communication. The Internet protocol (IP) version 4 is specified by IETF RFC 791, while IP version 6 is specified by IETF RFC 2460.

Data link layer and physical layer are typically using some LAN communication, such as ISO/IEC/IEEE 8802-3.

At the transport layer, the Transmission Control Protocol (TCP) specified by IETF RFC 793 is widely used for establishing a connection-oriented transport connection.

NOTE The user datagram protocol (UDP) specified in IETF RFC 768 is a connection-less protocol that may be used at the transport layer instead of TCP.

According to the OSI reference model, a connection established at one layer is used by the layer above for its communication. As there is no layer above the application layer, a connection is not established at the application layer, but is instead labelled as an association. For some this may be a subtle difference.

This Specification mostly considers connection-oriented communication at the application layer, where an association goes through three phases:

- a) association establishment phase;
- b) data transfer phase; and
- c) association termination phase.

During these three phases, cryptographic algorithms play specific roles as detailed by this Specification.

6.2.2 Association establishment phase

During the association establishment phase the following cryptographic algorithms may come into play:

- a) Each of the public-key certificate in the certification path (see [clause 21.1](#)) requires the use of one or two digital signature algorithms and one or two public-key algorithms, depending on whether algorithm migration is required or not.
- b) The handshake itself also requires the use of one or two digital signature algorithms depending on whether algorithm migration is required or not.
- c) Cryptographic algorithms to be used during the data transfer phase may be negotiated during the handshake phase. If confidentiality is required there may be a choice between authenticated encryption with associated data (AEAD) or not using AEAD (e.g., using AES-CBC). If not using AEAD is a possible choice, then negotiation of ICV algorithms may also be included.
- d) Depending on migration requirement, one or two key establishment algorithms are required for establishment of symmetric keys.

6.2.3 Data transfer phase

Unless the data transfer phase is rather short, the symmetric keys used during data transfer should be renewed at regular intervals. There are several reasons for renewal of the symmetric keys:

- a) There is less damage if a symmetric key is exposed.
- b) An adversary has less ciphertext available that was generated under one symmetric key.
- c) An adversary needs to recover several keys if interested in decrypting larger parts of a ciphertext.

The same key establishment algorithm used during association establishment may be used for rekeying.

6.2.4 Association termination phase

An association may be terminated either due to a normal release or due to error condition requiring an abort. To ensure authenticity and integrity of a release or abort, the same digital signature algorithm as for association establishment might be used.

6.3 Symmetric key requirements

If both ICV and encryption are required for a specific association, at least two symmetric keys are required per connection. If different keys are used in the two directions, which is recommended, four symmetric keys may be required per connection.

The wrapper protocol specified in Rec. ITU-T X.510 | IEC 9594-11 requires four or two symmetric keys per connection depending on whether encryption is required or not.

For security reasons, each association in a network requires its own set of symmetric keys different from those of other associations. Furthermore, the keys need to be renewed at certain intervals, e.g., every 24 hours. Traditional key distribution via a key distribution centre (KDC) is not useable in this environment. Quantum key distribution (QKD) does not seem like the answer either. Instead of a centralized approach, which is not scalable, a decentralized approach is necessary, where any two entities establish their own symmetric keys used in their mutual communication. It is here that PKI comes in. By having a public-key certificate together with the corresponding private key present in every entity makes it possible dynamically to produce the required symmetric keys in a decentralized fashion.

SECTION 2 – CRYPTOGRAPHIC ALGORITHMS

7 Introduction to cryptographic algorithms

Cryptography is the practice and study of techniques for secure communication in the presence of adversarial behaviour. Secure cryptographic algorithms are important components of a cyber security policy. They therefore play a major role within PKI and related technologies, such as transport layer security (TLS) as defined by IETF RFC 8446. Rec. ITU-T X.510 | ISO/IEC 9594-11 is depending on secure cryptographic algorithms.

Different types of cryptographic algorithms are required:

- a) hash algorithms as discussed in [clause 8](#);
- b) public-key algorithms and digital signature algorithms as discussed in [clause 9](#);
- c) key establishment algorithms as discussed in [clause 10](#);
- d) symmetric key algorithms as discussed in [clause 11](#);
- e) authenticated encryption with associated data (AEAD) algorithms as discussed in [clause 12](#);
- f) integrity check value (ICV) algorithms, also called message authentication code (MAC) algorithms, are discussed in [clause 13](#).

The concept of security strength is handled in [clause 14](#) while the importance of a good random number generator is discussed in [clause 15](#).

When the abstract syntax notation one (ASN.1) is used for specification of communication protocol syntaxes, cryptographic algorithms are identified by object identifiers. Protocols not using ASN.1 for syntax specification may use character strings to identify cryptographic algorithms. Such character strings are typically registered by the Internet Assigned Numbers Authority (IANA) to ensure uniqueness. Use of cryptographic algorithms by non-ASN.1 specifications is outside the scope of this Specification.

When ASN.1 is used as the syntax specification, a cryptographic algorithm is identified by an object identifier, but to fully identify a particular cryptographic algorithm, there may also be necessary to specify some additional parameters. Rec. ITU-T X.509 | ISO/IEC 9594-8 uses the following ASN.1 information object class for identifying cryptographic algorithms:

```
ALGORITHM ::= CLASS {
    &Type          OPTIONAL,
    &DynParms       OPTIONAL,
    &id             OBJECT IDENTIFIER UNIQUE }
WITH SYNTAX {
    [PARMS         &Type]
    [DYN-PARMS     &DynParms]
    IDENTIFIED BY &id }
```

NOTE A short introduction to the concept of ASN.1 information object class is given in Rec. ITU-T X.510 | ISO/IEC 9594-11.

The US based National Institute of Standards and Technology (NIST) has allocated object identifiers to several cryptographic algorithms. Other object identifiers are allocated by IETF RFCs. Rec. ITU-T X.510 | ISO/IEC 9594-11 has defined object identifier for cryptographic algorithms not defined anywhere else.

The procedure associated with a cryptographic algorithm is described in some detailed specification that is mostly relevant for implementors. The type of algorithm is identified by the object identifier given in the `&id` field. Some algorithms have as part of the procedure fixed parameters to be supplied to uniquely specify the procedure. The `&Type` field is used for supplying such fixed parameters. Some algorithms require parameters to be used during their deployment, such as a random number used

only once (also called nonce). Such parameters are called dynamic parameters. The `&DynParms` field is used for specifying requirement for dynamic parameters.

Negotiation of algorithm requires only the object identifier, and if present, the fixed parameters. During deployment of an algorithm, only the dynamic parameters, if any, are required as part of the deployment.

Prior to Rec. ITU-T X.509 (2019) | ISO/IEC 9594-8:2020, the **ALGORITHM** information object class did not include the `&DynParms` field meaning that fixed and dynamic parameter are intermixed, which makes algorithm negotiation cumbersome.

Cryptographic algorithms are defined in many different specifications, typically in IETF RFCs and in National Institute of Standards and Technology (NIST) specifications. For easy reference and for easy import into ASN.1 modules, [annex B](#) of Rec. ITU-T X.509 | ISO/IEC 9594-8 provides copy of relevant cryptographic algorithm definitions.

Current cryptographic algorithms that have dynamic parameters need to be updated using the extended **ALGORITHM** information object class. This requires assigning new object identifiers. Such updated specifications may be found in Rec. ITU-T X.510 | ISO/IEC 9594-11.

The security properties of a cryptographic algorithm depend on:

- The security parameters of the algorithm.
- The processing speed of computers, which determines the feasibility of cracking an algorithm by brute force, i.e., trying all possibilities. It is asserted that there may be adversaries with access to abundant computer power.
- An algorithm may have built-in weakness that allow skilled mathematicians to find ways to crack it not requiring a complete brute force attack.
- An algorithm may be implemented in such a way that it does not provide the full level of security it should theoretically offer. As an example, if a poor random number generator is used for generating cryptographic keys, some values are more likely than others. This may be used by an adversary to reduce the effort in cracking the key.

8 Hash algorithms

8.1 General

A hash algorithm is the specification on how to take an arbitrary message (bit string) and produce a fixed length hash value called a digest.

Secure hash algorithms are typically used together with other cryptographic algorithms and are useful for different purposes, including:

- a) Integrity protection: Digests may be used to detect whether messages have been changed since the digests were created (integrity check). This functionality is primarily part of digital signatures as discussed in [clause 9](#) or as part of integrity check value (ICV) as discussed in [clauses 13.2](#) and [13.4](#). Hash algorithms are also used for ensuring blockchain integrity (ensuring immutability).
- b) For generating fingerprints to generate a small representation of a larger data item, like a public-key certificate.
- c) Key derivation functions: This is a function that derives one or more secret keys from a shared secret value generated in some way, e.g., by use of the Diffie-Hellman method as discussed in [clause 10.3](#).
- d) For random number generation.

A good hash algorithm is designed to satisfy the following requirements:

- a) It is a one-way algorithm, meaning it is computationally infeasible to reverse the algorithm to find the message that maps to a specific digest.
- b) It is infeasible to modify a message without changing the digest.
- c) It is collision resistant, meaning it is computationally infeasible to find two distinct messages that map to the same digest.
- d) Even a small change in the input bit string causes an unpredictable change of the output digest. This will result in a verification failure when the hash algorithm is used with a digital signature algorithm or ICV.
- e) It is deterministic. When the same message is hashed multiple times, it results every time in the same digest. This is a characteristics that is required in contrast to encryption (see [clause 11.3.2](#)). This characteristic allows to verify a digital signature or ICV.
- f) Fast to calculate.

The first three properties are related to three types of hash algorithm attacks:

- a) Collision attack: The possibility to find two different messages that result in the same digest.
- b) Preimage attack: Given a digest value without knowing the corresponding message, it is possible to find a bit string having that digest. It is considerably more difficult to perform preimage attack than a collision attack.
- c) Second preimage attack: Given a message and its digest, it is possible to find another message that has the same digest. This is the attack that could be made on data being transmitted, where a man-in-the-middle attacker is trying to replace the data with some other data.

8.2 The SHA-2 series of hash algorithms

8.2.1 Referenced specifications

IETF RFC 6234 give a detailed specification of the SHA-2 hash algorithm. It also includes some programming aspects. NIST FIPS PUB 180-4 is another, but shorter reference.

IETF RFC 5754 provides information on the use of SHA-2 algorithms by cryptographic message syntax (CMS).

8.2.2 The SHA-2 model

NIST has in FIPS PUB 180-4 published specifications for so-call secure hash algorithms (SHAs) to replace previous hash algorithms not considered sufficiently safe. This Specification considers SHA-224, SHA-256, SHA-512, SHA-512/224 and SHA-512/256. SHA-1 is now considered unsafe and therefore not further considered by this Specification.

SHA-2 is now widely used and considered safe for the time being. However, current implementations of SHA-2 should plan migration to SHA-3

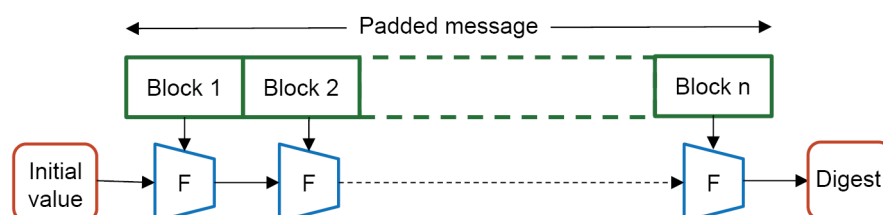


Figure 3 — SHA-2 generation

As indicated in [Figure 3](#), the SHA-2 procedure is block oriented, with two different block sizes as listed in [Table 1](#).

Messages are padded to a multiple of the block size. The last part of the padding is a 64-bit integer for a block size of 512 bits and 128-bit integer for block size of 1024 bits. This integer shall hold the length of the message to be hashed. For details on padding see IETF RFC 6234.

The creation of hash digest goes through several steps as illustrated in [Figure 3](#). The function F takes as input a block of the padded message and the output from the previous instance of the function F . It then manipulates these two inputs in a somewhat complicated way, as described in NIST FIPS PUB 180-4. The result is a bit string of the same size as the block size. It is then an input to the next invocation of the function F .

When handling the first block, there is not output from a previous invocation of the F -function and instead an initial value of the same size as the block size is used. The initial values are listed in NIST FIPS PUB 180-4.

The output of the last invocation of the F -function is the hash digest possibly truncated as required.

8.2.3 The SHA-2 characteristics

Table 1 — Characteristics of SHA-2 algorithms

Algorithm	Message size (bits)	Block size (bits)	Word size (bits)	Digest size (bits)	Security strength		
					Collision	Preimage	2nd preimage
SHA-224	$< 2^{64}$	512	32	224	112	224	Note 1
SHA-256	$< 2^{64}$	512	32	256	128	256	256-L(M)
SHA-384	$< 2^{128}$	1024	64	384	192	384	384
SHA-512	$< 2^{128}$	1024	64	512	256	512	512-L(M)
SHA-512/224	$< 2^{128}$	1024	64	224	112	224	224
SHA-512/256	$< 2^{128}$	1024	64	256	128	256	256

Note 1: Security strength is the minimum of 224 and $(256 - L(M))$
See NIST FIPS 202 Section A.1 for an explanation of $L(M)$

8.2.4 SHA-2 formal specification

8.2.4.1 General

The object identifiers for the SHA-2 algorithms are allocated by the NIST Computer Security Objects Register (see Bibliography).

The formal algorithm specifications for SHA-2 algorithms take no parameter. A NULL parameter shall not be included as a parameter in the formal specification. IETF RFC 5754 states that a conformant CMS shall not include the NULL parameter but should accept that a communication partner includes the NULL parameter.

8.2.4.2 SHA-224 and SHA-256 algorithms

SHA-224 and SHA-256 perform identical processing on message blocks and differ only on the initial value. Both create a 256 bits output, but for SHA-224, the output is truncated to 224 bits.

They may both be used to hash a message having a length of less than 2^{64} bits.

```
sha224 ALGORITHM ::= {
  IDENTIFIED BY { hashAlgs 4 } }
```

```
sha256 ALGORITHM ::= {
```

```
IDENTIFIED BY { hashAlgs 1 } }
```

```
hashAlgs OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16) us(840) organization(1)
gov(101) csor(3) nistAlgorithm(4) hashalgs(2) }
```

8.2.4.3 SHA-384 and SHA-512 algorithms

SHA-384 and SHA-512 perform identical processing on message blocks and differ only on the initial value. Both create a 512 bits output, but for SHA-384, the output is truncated to 384 bits.

They may both be used to hash a message having a length of less than 2^{128} bits.

```
sha384 ALGORITHM ::= {
  IDENTIFIED BY { hashAlgs 2 } }
```

```
sha512 ALGORITHM ::= {
  IDENTIFIED BY { hashAlgs 3 } }
```

8.2.4.4 SHA-512/224 and SHA-512/256 algorithms

SHA-512/224 and SHA-512/256 as for SHA-384 and SHA-512, except that different initialization values are used and the output for SHA-512/224 is truncated to 224 bits and the output for SHA-512/256 is truncated to 256 bits.

```
sha512-224 ALGORITHM ::= {
  IDENTIFIED BY { hashAlgs 5 } }
```

```
sha512-256 ALGORITHM ::= {
  IDENTIFIED BY { hashAlgs 6 } }
```

Using a 64 bits processor, it has been demonstrated that the SHA-224 and SHA-256 algorithms take more processing power than SHA-384 and SHA-512 algorithms, except for short messages. The SHA-512/224 and SHA-512/256 algorithms were introduced to take the benefit of a more efficient processing, but still have shorter hash digests in environments with size constraints.

8.3 The KECCAK algorithms

NIST chose the KECCAK algorithm as the new SHA-3 standard. KECCAK offers many benefits, such as efficient performance and good resistance against attacks.

KECCAK is best known as the underlying technique for the SHA-3 series of hash algorithms and the KMAC algorithms as described in [clause 13.4](#). The basic principles of Keccak are firstly presented here as they are the basis discussing SHA-3 and KMAC. The KECCAK algorithm is discussed in NIST FIPS PUB 202.

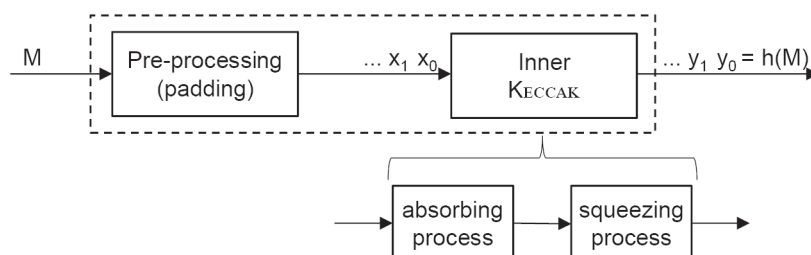


Figure 4 — High-level view on Keccak

[Figure 4](#) gives a high-level view of the KECCAK algorithm. A KECCAK turns a message M into something else. It may be turned into a hash digest, or an integrity check value (ICV) as discussed in [clause 13.4](#). Keccak requires that the message is split up in blocks of r bits, where r has value specific for a particular

usage of the KECCAK algorithm (see [Table 2](#) for an example). The message is therefore padded to a be multiple of r . As also illustrated, the individual blocks $x_0, x_1, \dots, x_i$ of the input message are going through a process that is being compared to a sponge. It sucks the input data and then squeezes the data out in a modified form, e.g., as a hash digest.

A KECCAK operates on a fixed number of bits of width b at a time and where $b > r$. The difference $b - r = c$, is called the capacity. b can take 7 different values in multiples of 25 bits, which can be written as $b = 25 \times 2^l$, $l = 0, 1, 2, 3, 4, 5, 6$, which means:

$$b \in \{25, 50, 100, 200, 400, 800, 1600\}$$

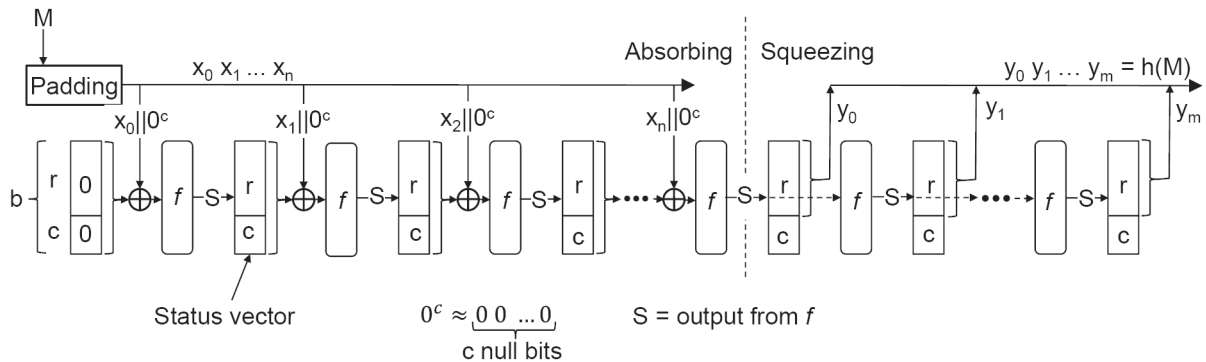


Figure 5 — Absorbing and sequencing phases of the sponge construction

[Figure 5](#) illustrates the absorbing and sequencing phases of the sponge construction.

The blocks of the padded input message M are fed into a permutation function one by one as illustrated in [Figure 5](#). Each input block x_i of size r is padded with c null bits to get a length of b bits. This padded input block is XOR'ed with a status vector also of b bits, and then fed into the permutation function f and the result is an update status vector, which is then again XOR'ed with the next padded input block x_{i+1} , etc. The first block x_0 is XOR'ed with an initial status vector of b zero bits. When the last input block has been processed, the absorbing phase is complete.

The output of the last permutation transformation of the absorbing phase is then the input to the squeezing phase. In contrast to a physical sponge, KECCAK can squeeze out an indefinite amount of data. As seen from [figure 5](#), the final r output of the absorbing phase yields the first part of the output data. If this output is longer than required, it is truncated. If the output is shorter than required, the permutation function is applied once more to create more output. This may continue until enough data is squeezed out.

The transformation is done by permutation. This permutation is done based on a structure of a three-dimensional array as illustrated in [Figure 6](#). The number of cells is the same as the width b , i.e., $r + b$. The figure illustrates the case with $b = 1600$. How the bits are allocated to the cells is described in NIST FIPS PUB 202.

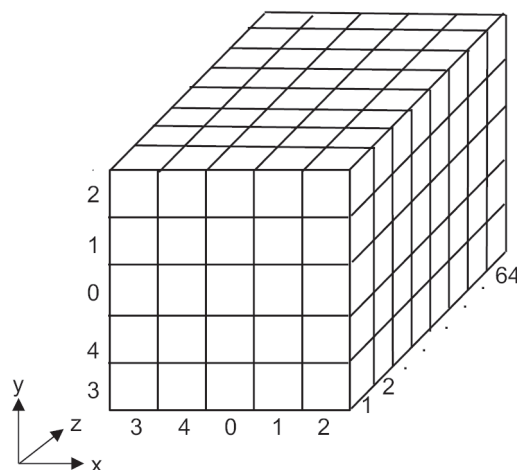


Figure 6 — State array

Within the state matrix, the bits may be moved around and/or changed by XOR operations in many ways in five different steps named after the Greek letters: Θ (Theta), ρ (Rho), π (Pi), χ (Chi) and ι (Iota). For details on these five steps, see NIST FIPS PUB 202.

The 5 steps are repeated in several rounds for each invocation of f . The number of round n_r is depending on the l defined above using the following formular: $n_r = 12 + 2l$. For $l = 6$, i.e., for, $b = 1600$, the number of rounds is 24 for each invocation of the permutation function.

How the output is extracted from the matrix to the status vector is also described in detail in NIST FIPS PUB 202.

NIST FIPS PUB 202 defines the function:

$\text{KECCAK}[c](M, d)$

where the following is fixed: $b = 1600$, $n_r = 24$, and where c is the capacity. M is the input bitstring and d is the requested output size in bits.

8.4 SHA-3 characteristics

SHA-3 uses KECCAK with a width of 1600 bits. The defined SHA-3 algorithms are listed in [Table 2](#).

Table 2 — Characteristics of SHA-3 algorithms

Algorithm	Output size d (bits)	Rate r (block size)	Capacity c	Security strength in bits		
				Collision	Preimage	2nd preimage
SHA3-224	224	1152	448	112	224	224
SHA3-256	256	1088	512	128	256	256
SHA3-384	384	832	768	192	384	384
SHA3-512	512	576	1024	256	512	512
SHAKE128	256	1344	256	128	256	256
SHAKE256	512	1088	512	256	512	512
SHAKE128-len	d	1344	256	$\min(d/2, 128)$	$\geq \min(d, 128)$	$\min(d, 128)$
SHAKE256-len	d	1088	512	$\min(d/2, 256)$	$\geq \min(d, 256)$	$\min(d, 256)$

8.5 SHA-3 formal specifications

8.5.1 SHA3-224 specification

```
sha3-224 ALGORITHM ::= {
  IDENTIFIED BY { hashAlgs 7 } }
```

8.5.2 SHA3-256 specification

```
sha3-256 ALGORITHM ::= {
  IDENTIFIED BY { hashAlgs 8 } }
```

8.5.3 SHA3-384 specification

```
sha3-384 ALGORITHM ::= {
  IDENTIFIED BY { hashAlgs 9 } }
```

8.5.4 SHA3-512 specification

```
sha3-512 ALGORITHM ::= {
  IDENTIFIED BY { hashAlgs 10 } }
```

8.5.5 SHAKE-128 and SHAKE-256 specifications

```
shake128 ALGORITHM ::= {
  IDENTIFIED BY { hashAlgs 11 } }
```

```
shake256 ALGORITHM ::= {
  IDENTIFIED BY { hashAlgs 12 } }
```

8.5.6 SHAKE-128-len and SHAKE-256-len specifications

```
shake128-len ALGORITHM ::= {
  PARMS          ShakeOutputLen
  IDENTIFIED BY { hashAlgs 17 } }
```

```
shake256-len ALGORITHM ::= {
  PARMS          ShakeOutputLen
  IDENTIFIED BY { hashAlgs 18 } }
```

```
ShakeOutputLen ::= INTEGER -- Output length in bits
```

An extendable-output function (XOF) is a function on bit strings in which the output can be extended to any desired length. The two SHA-3 XOFs are named SHAKE128-len and SHAKE256-len.

9 Public-key algorithms and digital signature algorithms

9.1 General

In asymmetric cryptography, an entity is assigned two mathematically related keys, a private key, that the owning entity is assumed to protect from disclosure and a public key that may be more freely distributed. Although the two keys are mathematically interrelated, it is infeasible for a secure public-key algorithm to determine the private key from the corresponding public key, although it is theoretically possible.

If the public key is broken such that the private key is disclosed or the private key is otherwise compromised, it is possible to produce false digital signatures.

NIST has issued a guide on digital signatures: NIST FIPS PUB 186-5. In addition to specifying digital signature algorithms, it also includes public-key algorithm specifications.

Digital signature generation requires the deployment of both a hash algorithm and a public-key algorithm. Some public key algorithms allow a choice among hash algorithms. In this case, it is not sufficient to have a specification identifying the public-key algorithm. It is also necessary to define a digital signature algorithm for each combination of a public-key algorithm and a hash algorithm. See e.g., [clauses 9.2.4](#) and [9.4.5](#).

Some public-key algorithms have a specific hash algorithm as part of their specifications, which is also used for digital signature generation and verification. For such public-key algorithms there is no need for a special digital signature algorithm. See [clause 9.5](#).

Technically, a key pair used by a digital signature algorithm could also be used for other purposes, e.g., key establishment. However, it is highly recommended not to do that but to have different key pairs for different purposes. NIST FIPS PUB 186-5 explicitly forbids multiple uses key pairs.

Digital signatures may be generated over both stored and transmitted data.

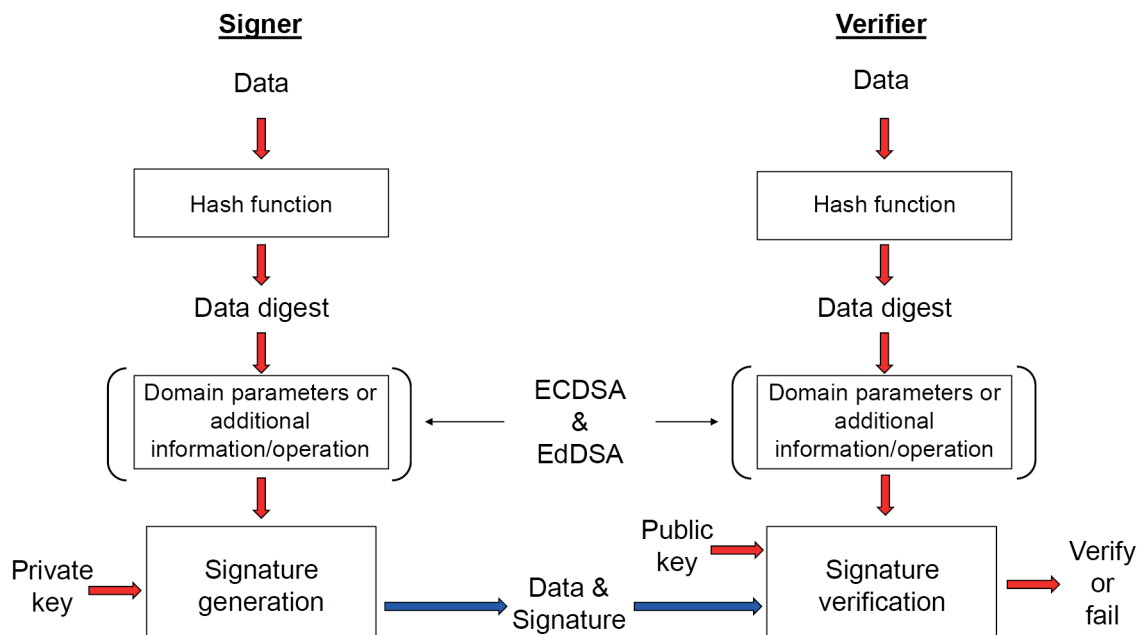


Figure 7 — Digital signature generation and verification

[Figure 7](#) illustrates the principle of digital signatures in general and it also illustrates the use of different cryptographic algorithms. Only the user that owns the private key, also called the signatory, can perform signature generation.

Digital signatures are designed such that they cannot be forged.

Digital signatures are sometimes compared with handwritten signatures, but there are differences. A digital signature is closely tied to the data being signed. Even a small change in the data will invalidate the signature.