



DPC: 24/30492042 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: 24 April 2024

Origin: International

Latest date for receipt of comments: 16 June 2024

Project No. 2021/03094

Responsible committee: IST/6 Data communications

Interested committees:

Title: Draft BS ISO/IEC 9594-12 Draft Information technology — Open systems interconnection. Part 12: The Directory: Key management and public-key infrastructure establishment and maintenance.

Please notify admin.start@bsigroup.com if you are aware of any keywords that might assist in clarifying or identifying the standard or of 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 16 June 2024

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 <https://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	Comment (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 9594-12.2

Information technology — Open systems interconnection —

Part 12:

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

ICS: 35.100.70

ISO/IEC JTC 1/SC 6

Secretariat: **KATS**

Voting begins on:
2024-04-22

Voting terminates on:
2024-06-17

This document is circulated as received from the committee secretariat.

THIS DOCUMENT IS A DRAFT CIRCULATED FOR COMMENTS 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.

Reference number
ISO/IEC DIS 9594-12.2:2024(en)

Full document details: 24/30492042 DC

 **Documents**
Document Sharing Hub
© ISO/IEC 2024



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2024

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

SECTION 1 – GENERAL	8
1 Scope.....	8
2 Normative references	8
2.1 Identical Recommendations International Standards	8
2.2 Recommendations	8
2.3 Others	8
3 Definitions.....	9
3.1 OSI reference model security architecture definitions	9
3.2 Baseline identity management terms and definitions	9
3.3 Terms defined in public-key and attribute certificate frameworks	9
3.4 Key management and public-key infrastructure establishment and maintenance definitions.....	10
4 Abbreviations	11
5 Conventions	12
6 Cybersecurity consideration for communication networks	13
6.1 The challenge of large ICT networks	13
6.2 Connection-mode communication.....	14
6.2.1 General	14
6.2.2 Association establishment phase	15
6.2.3 Data transfer phase	15
6.2.4 Association termination phase.....	15
6.3 Symmetric key requirements	15
6.4 Security services.....	15
SECTION 2 – CRYPTOGRAPHIC ALGORITHMS	17
7 Introduction to cryptographic algorithms	17
8 Symmetric-key algorithms.....	18
8.1 Stream ciphers vs. block ciphers	18
8.2 Advanced encryption standard	19
8.2.1 General	19
8.2.2 The octet substitution layer	21
8.2.3 The Shift Rows layer.....	22
8.2.4 The mix columns layer.....	22
8.2.5 The add round key layer.....	23
8.3 Modes of advance encryption standard	24
8.3.1 Overview of AES modes.....	24
8.3.2 Advanced encryption standard – electronic codebook (AES-ECB) mode.....	25
8.3.3 Advanced encryption standard - cipher block chaining (AES-CBC).....	25
8.3.4 Cipher feedback mode.....	26
8.3.5 Output feedback mode.....	26
8.3.6 Counter mode	28
8.3.7 Padding.....	29
9 Hash algorithms	29
9.1 General	29
9.2 The SHA-2 series of hash algorithms.....	30
9.2.1 Referenced specifications.....	30
9.2.2 The SHA-2 model	30
9.2.3 The SHA-2 characteristics.....	31
9.2.4 SHA-2 formal specification.....	31
9.3 The KECCAK algorithms	32
9.4 SHA-3 characteristics	33
9.5 SHA-3 formal specifications	34
9.5.1 SHA3-224 specification	34
9.5.2 SHA3-256 specification	34
9.5.3 SHA3-256 specification	34

9.5.4	SHA3-512 specification	34
9.5.5	SHAKE-128 and SHAKE-256 specifications	34
9.5.6	SHAKE-128-len and SHAKE-256-len specifications	34
9.5.7	cSHAKE.....	35
10	Asynchronous encryption	35
10.1	General	35
10.2	The RSA public-key encryption algorithm	35
10.2.1	General	35
10.2.2	Key generation	35
10.2.3	Security considerations.....	36
10.2.4	RSA asynchronous encryption/decryption	36
11	Public-key algorithms including digital signature algorithms	37
11.1	General	37
11.2	The RSA digital signature system	38
11.2.1	General	38
11.2.2	RSA signature scheme with appendix - probabilistic signature scheme (RSASSA-PSS).....	38
11.2.3	Signature generation and verification.....	38
11.3	The DSA public-key algorithm	39
11.4	The elliptic curve digital signature algorithm (ECDSA).....	39
11.4.1	General	39
11.4.2	Defines curves	39
11.4.3	Key generation	40
11.4.4	Security considerations.....	40
11.4.5	Digital signature generation	40
11.4.6	Digital signature verification.....	41
11.4.7	Formal specifications of digital signature algorithms	41
11.5	Edwards-curve digital signature algorithm.....	41
11.5.1	General	41
11.5.2	Defined curves.....	42
11.5.3	Point encoding and decoding	42
11.5.4	Key pair generation	42
11.5.5	Digital signature generation	43
11.5.6	Signature verification	43
11.5.7	Security issues	43
11.5.8	Formal specifications of digital signature algorithms	43
11.6	XMSS	44
11.7	LMS.....	45
12	Key establishment algorithms.....	45
12.1	Introduction	46
12.2	Key transport	46
12.2.1	RSA key transport	46
12.2.2	Key wrapping	46
12.3	The Diffie-Hellman key agreement method.....	47
12.3.1	General	47
12.3.2	Finite field cryptography Diffie-Hellman key agreement.....	47
12.3.3	Elliptic curve Diffie-Hellman key agreement	48
12.3.4	X25519 and X448 Diffie-Hellman key agreement	48
12.3.5	Classic McEliece	40
12.3.7	Man-in-the-middle attack.....	48
12.4	Key establishment mechanism	41
12.5	Key derivation functions	49
12.5.1	General	49
12.5.2	HMAC-based extract-and-expand key derivation function	49
13	Authenticated encryption with associated data (AEAD) algorithms.....	49
13.1	General	49
13.2	Advanced encryption standard – Galois/counter mode (AES-GCM).....	50
13.3	Advanced encryption standard (AES)- Counter with CBC-MAC (CCM).....	50

14	Integrity check value (ICV) algorithms	51
14.1	Introduction	51
14.2	Keyed-hash message authentication code (HMAC).....	51
14.3	Cipher-based message authentication code (CMAC).....	51
14.3.1	General	51
14.3.2	Formal specification	52
14.4	KECCAK message authentication code (KMAC).....	52
14.4.1	General	52
14.4.2	Input message constructions.....	52
14.4.3	KMAC formal specifications	53
14.5	Advance encryption standard (AES) - Galois message authentication code (GMAC) algorithm	53
15	Security strength	54
16	Random number generation and entropy	54
17	Post-quantum cryptography considerations	55
17.1	Mathematic behind post-quantum cryptography	55
17.2	Expected attack by quantum computers	55
17.3	NIST selection of post-quantum cryptographic algorithms	56
17.4	Classic McEliece	48
18	Hardware security modules.....	56
	SECTION 3 – PUBLIC-KEY INFRASTRUCTURE.....	57
19	Introduction to the section	57
20	Public-key certificates.....	57
20.1	Content of a basic public-key certificate	57
20.2	Content of a basic public-key certificate	57
20.2.1	Structure	57
20.1.3	Distinguished name considerations.....	57
20.1.4	Version component.....	58
20.1.5	Serial number component.....	58
20.1.6	Signature component.....	58
20.1.7	Issuer component.....	58
20.1.8	Validity component.....	59
20.1.9	Subject component	59
20.1.10	Subject public-key information	59
20.1.11	Issuer unique ID and subject unique ID	59
20.2	Extensions.....	59
20.2.1	Use of extensions for public-key certificates	59
20.2.2	Basic constraints extension	59
20.2.3	Key usage extension.....	60
20.2.4	Subject directory attribute extension type	60
20.2.5	Subject alternative name extension	61
20.2.6	Authority information access extension.....	61
21	Trust establishment	61
21.1	General	61
21.2	Single public-key infrastructure domain	61
21.3	Trust establishment between two public-key infrastructure domains	62
21.4	A worldwide federated public-key infrastructure.....	63
21.5	Trust anchor compromise.....	63
22	Identity verification in a machine-to-machine environment.....	63
22.1	General	63
22.2	The two-PKIs approach.....	64
22.3	Use of the shared secret approach	64
22.4	Operational PKI establishment.....	65
23	PKI configurations	65
23.1	Introduction	65

23.2	PKI components	65
24	Certification path processing	65
24.1	Certification path	65
24.2	Name chaining.....	66
24.3	Key identifier chaining	67
25	Certification path validation	67
25.1	Overview	67
25.2	Validating certification path	68
25.3	Validating individual public-key certificates	69
A.1	Scope of appendix	95
A.2	Public-key certificate components.....	95
A.3	Length of distinguished names.....	95
A.4	Representation of validity component.....	96
A.5	Use of Extensions	96
A.6	Size of object identifiers.....	96
A.7	Size of hash digests	97
A.8	Length of public-key certificate	97
B.1	Introduction to basic mathematic	72
B.1.1	Scope of annex	72
B.1.2	The prime number, the semiprime and the co-prime concepts	72
B.1.3	Greatest common divisor.....	72
B.1.4	The logarithm concept.....	72
B.1.5	Operation on matrices	72
B.1.6	Least common multiple	73
B.1.7	Bitwise logical operations	73
B.2	Modular arithmetic	74
B.3	Group theory	75
B.3.1	Introduction	75
B.3.2	Notation	76
B.3.3	Additive group of integers.....	76
B.3.4	Multiplicative group of integers.....	76
B.3.5	Cyclic Groups.....	77
B.3.6	The discrete logarithm problem	77
B.3.7	Generalized discrete logarithm problem	78
B.3.8	Subgroup	78
B.3.9	The order of group and order of element	78
B.3.10	Ways to resolve or attack the discrete logarithm problem	78
B.3.11	Euler's phi function.....	71
B.4	Finite fields	79
B.4.1	General	79
B.4.2	Prime fields.....	79
B.4.3	Binary fields GF(2 ^m)	80
B.5	Overview of Elliptic curve cryptography	81
B.5.1	Reasons for using elliptic curve cryptography	81
B.5.2	Overview of curve types.....	82
B.5.3	Variants of the Weierstrass form.....	82
B.5.4	The Montgomery curves	82
B.5.5	The twisted Edwards curves.....	82
B.6	Elliptic curve cryptography for short-Weierstrass form.....	83
B.6.1	Definition of curves based on the Weierstrass equation	83
B.6.2	Defining group over elliptic curve	84
B.7	Montgomery elliptic curve cryptography.....	87
B.7.1	Introduction	87
B.7.2	Curve25519 and Ed25519.....	87
B.7.3	Curve448 and Ed448.....	87
B.7.4	The Montgomery curves	87
B.7.5	The Edwards curves	88

B.8	Conversion techniques.....	88
B.8.1	General	88
B.8.2	Bitstring-to-integer conversion and binary length of integer	88
B.8.3	Bitstring-to-octet string conversion.....	88
B.9	Miscellaneous formulae.....	88
B.9.1	Introductions.....	88
B.9.2	The Euclidean algorithm	88
B.9.3	The extended Euclidean algorithm	89
B.9.3	Fermat's little theorem	90
B.9.4	Lagrange's theorem.....	90
B.9.5	Euler's phi function.....	90
B.10	Endianness (big endian vs. little endian).....	90
B.11	Selected attacks on cryptographic algorithms	90
B.11.1	Side-channel attack.....	90
B.11.2	Square root attack	90

INTERNATIONAL STANDARD ISO/IEC 9594-12
RECOMMENDATION ITU-T X.508**Information technology – Open Systems Interconnection – 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 a somewhat more extended description 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 network (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

- Recommendation ITU-T X.501 (2019) | ISO/IEC 9594-2:2020, Information technology – Open Systems Interconnection – The Directory: Models.
- Recommendation ITU-T X.509 (2019) | ISO/IEC 9594-8:2020, Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks.
- Recommendation ITU-T X.510 (2020) | ISO/IEC 9594-11:2020, Information technology – Open Systems Interconnection – The Directory: Protocol specifications for secure operations.
- Recommendation ITU-T X.520 (2019) | ISO/IEC 9594-6:2020, Information technology – Open Systems Interconnection – The Directory: Selected attribute types.

2.2 Recommendations

None

2.3 International Standards

- ISO/IEC 10116:2017, *Information technology — Security techniques — Modes of operation for a n-bit block cipher*.
- ISO/IEC 14888-4:2024, *Information security — Digital signatures with appendix — Part 4: Stateful hash-based mechanisms*.

2.4 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 186-5 (2019), *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) certificate revocation list (CRL);
- d) certificate serial number;
- e) certificate validation;

- f) certification authority (CA);
- g) certification path;
- h) cross-certificate;
- i) end entity;
- j) end-entity public-key certificate;
- k) key agreement;
- l) private key;
- n) public key;
- m) relying party;
- n) trust anchor;
- o) trust anchor information.

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

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

- 3.4.1 brute-force attack:** For cryptographic algorithms it is an exhaustive search of all possible combinations until a workable combination is found.
- 3.4.2 cipher:** A cryptographic algorithm for performing encryption and decryption.
- 3.4.3 crypto agility:** The ability to adapt to more secure cryptographic algorithms without significant changes to system structure.
- 3.4.4 cryptanalysis:** study of cryptographic algorithms to identify weaknesses and information leakages.
- 3.4.5 cryptographic algorithm:** Well-defined computational procedure that takes variable inputs, often including a cryptographic key, and produces an output.
- 3.4.6 encryption:** The process of converting information or data into a code, especially to prevent unauthorized access.
- 3.4.7 extractor:** An algorithm that transforms a weakly random distribution into an almost uniform distribution.
- 3.4.8 forward cipher functions:** A block cipher function that converts a block of plain text to a block of ciphertext.
- 3.4.9 fully qualified domain name:** A domain name that specifies its exact location in the tree hierarchy of the domain name system (DNS)
- 3.4.10 integer factoring:** Breaking a (large) integer down into prime factors when multiplied equal the original integer.
- 3.4.11 inverse cipher function:** A block cipher function that converts a block of ciphertext to a block of plain text.
- 3.4.12 key agreement:** A pairwise key-establishment algorithm in which the resulting ciphertext shared secret is a function of information supplied by both participants.
- 3.4.13 key establishment:** The procedure that results in secret keying material shared between communicating entities using either key agreement or key transport.
- 3.4.14 key transport:** Secure transfer of a symmetric key to a peer communication partner.
- 3.4.15 mode of operation:** A way of using a block cipher for encryption/decryption.
- 3.4.16 national trust anchor:** A trust anchor established by a recognized national organization that is considered trustworthy by all entities in the country.
- 3.4.17 nonce:** A value that is used only once within a specified context.
- 3.4.18 public-key certificate owner:** An entity that is the subject of a public-key certificate.
- 3.4.19 public-key infrastructure (PKI) domain:** A PKI under a single management.

- 3.4.20 security strength:** A measure expressed in bits of the expected number of operations of some kind required to crack a cryptographic algorithm.
- 3.4.21 semiprime:** An integer being the product of two prime integers.
- 3.4.22 signatory:** The entity that digitally signs a document or a message.
- 3.4.23 Shor's algorithms:** A quantum algorithm for factoring a (large) integer.
- 3.4.24 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.
- 3.4.25 symmetric key:** A cryptographic key used for both encryption and description of data.
- 3.4.26 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
ASN.1	Abstract Syntax Notation One
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
FFC	Finite Field Cryptography
FQDN	Fully Qualified Domain Name
GCD	Greatest Common Divisor
HBS	Hash-Based Signature
HKDF	Hash-based Key Derivation Function
HMAC	Hash Message Authentication Code
HSM	Hardware Security Module
HSS	Hierarchical Signature Scheme
IANA	Internet Assigned Numbers Authority
ICV	Integrity Check Value
IETF	Internet Engineering Task Force
IIoT	Industrial Internet-of-Things
IoT	Internet of Things
IP	Internet Protocol
KDC	Key Distribution Centre
KEM	Key Establishment Mechanism

LCM	Least Common Multiple
LMS	Leighton-Micali Signature
MAC	Message Authentication Code
M2M	Machine-to-Machine
NIST	National Institute of Standards and Technology
OAEP	Optimal Asymmetric Encryption Padding
OCSF	Online Certificate Status Protocol
OSI	Open Systems Interconnection
OTS	One-Time Signature
PKI	Public-Key Infrastructure
PQC	Post-Quantum Cryptography
PRNG	Pseudo Random Number Generator
QKD	Quantum Key Distribution
RBG	Random Bit Generators
RDN	Relative Distinguished Name
RFC	Request For Comments
RSA	Rivest-Shamir-Adelman
RSAS	Rivest-Shamir-Adelman Encryption System
RSASSA	Rivest-Shamir-Adelman Signature Scheme with Appendix
SHA	Secure Hash Algorithm
SHAKE	Secure Hash Algorithm KECCAK
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
UUID	Universal Unique Identifier
WOTS+	Winternitz One-Time Signature Plus
XMSS	eXtended Merkle Signature Scheme
XOF	eXtendable-Output Function
XOR	eXclusive OR

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 **bold 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 **bold Courier New** typeface. Mathematical terms and equations are presented in the *Cambria Math* italic typeface.

To indicate that a string is a hexadecimal string, it is prefixed 0x.

The comma (',') is used as the decimal sign.

6 Cybersecurity consideration for communication networks

6.1 The challenge of large ICT networks

Large networks provide a particular challenge from a cybersecurity 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.

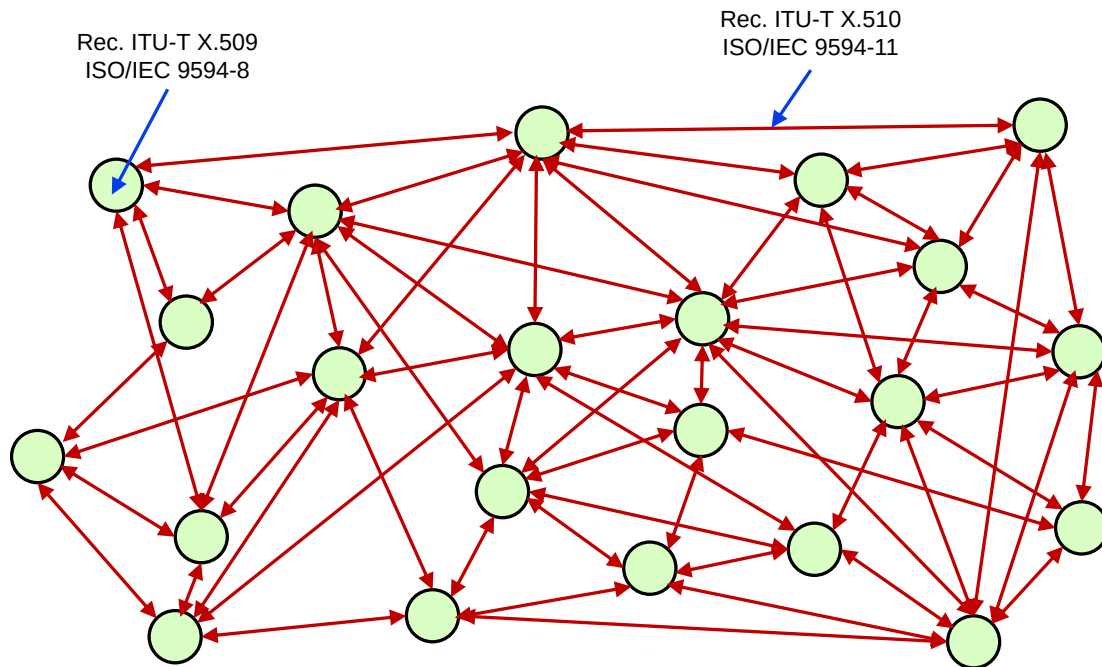


Figure 1 – Large interconnected ICT network

A large network is characterized by having many communication channels as illustrated in Figure 1. A network may not be fully interconnected as illustrated in the figure, but e.g., having a hierarchical structure.

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 depicted in Figure 2 and is discussed in some details below.

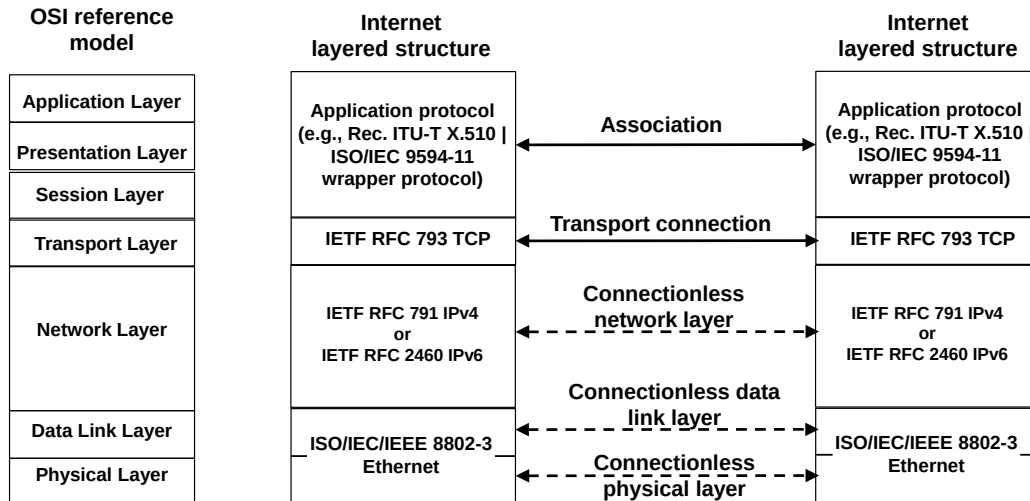


Figure 2 – Layered communication model

The Internet layered model and its relationship with the Open Systems Interconnection (OSI) reference model is illustrated in Figure 2. The OSI reference model consists of seven layers with one or more protocols operating at each layer for given communication system. Internet Engineering Task Force (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.

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

- For 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.
- For connectionless 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 connectionless 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 association.

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

- association establishment phase;
- data transfer phase; and
- 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 is also needed.
- 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 the same 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 an error condition requiring the association to be aborted. 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 or whether authenticated encryption is used.

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 look 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 into play. 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 and secure fashion.

6.4 Security services

The following security services may be required for an association:

- a) Confidentiality: Information is kept secret for all but the authorized parties.
- b) Authentication: The sender of the message is authentic.
- c) Integrity: Message cannot be changed during transmission without detection.
- d) Nonrepudiation: The sender of a message cannot deny the creation of the message. This is especially important for financial transactions.

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) symmetric-key algorithms as discussed in clause 8;
- b) hash algorithms as discussed in clause 9;
- c) asymmetric encryption in clause 10;
- d) public-key algorithms and digital signature algorithms as discussed in clause 11;
- e) key establishment algorithms as discussed in clause 12;
- f) authenticated encryption with associated data (AEAD) algorithms as discussed in clause 13;
- g) integrity check value (ICV) algorithms, also called message authentication code (MAC) algorithms, are discussed in clause 14.

The concept of security strength is handled in clause 15 while the importance of a good random number generator is discussed in clause 16.

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, it 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. That specification also introduces the concept of parameterized data types that together with information object classes allows for specifying versatile protocol specifications easy migration of cryptographic algorithms.

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 complete the specification. 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 algorithms to be used requires only the object identifiers, 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 were intermixed, which makes algorithm negotiation cumbersome.

Cryptographic algorithms are defined in many different specifications, typically in international standards, 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 Block cipher methods

8.1 Stream ciphers vs. block ciphers

In symmetric key cryptographic, the same key, called a symmetric key, is used by the two partners in their mutual communication. Security is only maintained if the two partners protect the shared symmetric key. A symmetric key is in some documents called a secret key.

Symmetric keys are used for two major types of ciphers, namely block cipher and stream cipher.

Stream ciphers process data one in small unit (typically an octet or even a bit) at a time. This allows for ciphers to process an arbitrary amount of data without padding.

Block ciphers process blocks of data at a time with a fixed block size. If the length of the data to be encrypted is not a multiple of the block size, the data may have to be padded, i.e., before encryption, dummy octets is added to make a multiple of the cipher's block size. These octets are then stripped off during the decryption phase. For der details see clause 8.3.7

8.2 Block cipher methods and block cipher operation modes

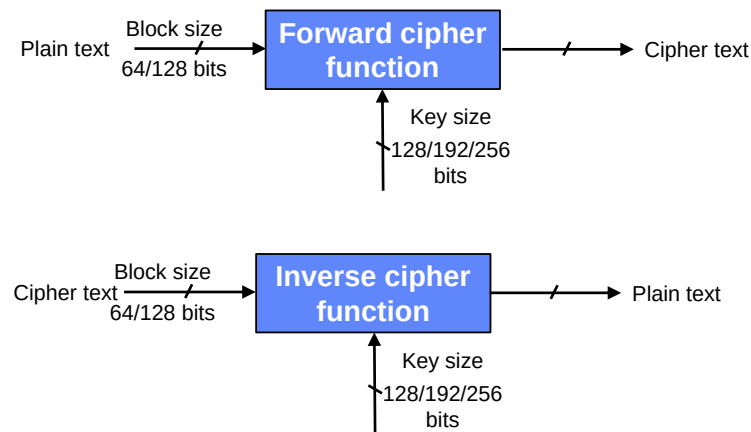


Figure 3 – Forward and inverse cipher functions

Block cipher methods typically operate on 64 or 128-bit blocks.

A block cipher method is a specification on how an encryption/decryption is performed on a single block. How the blocks are combined for a complete encryption/decryption of a message is called a cipher operation mode or just cipher mode and its specification is independent of the cipher method.

Figure 3 illustrates the concept of a cipher method.

The data to be encrypted/decrypted is therefore split-up in 64 or 128-bit blocks, and the blocks are encrypted/decrypted one-by-one.

8.3 Advanced encryption standard

8.3.1 General

Advanced encryption standard (AES) is currently one of the most used symmetric-key algorithm in some parts of the world.

All internal operations of AES are based on Galois field arithmetic as discussed in clause B.4.

AES is a block cipher with a 128-bit block size (16 octet block size). Three key sizes are defined: a 128-bit size, a 192-bit size and a 256-bit size.

The basic AES specifies the details for how the individual blocks are processed using forward cipher/inverse cipher.

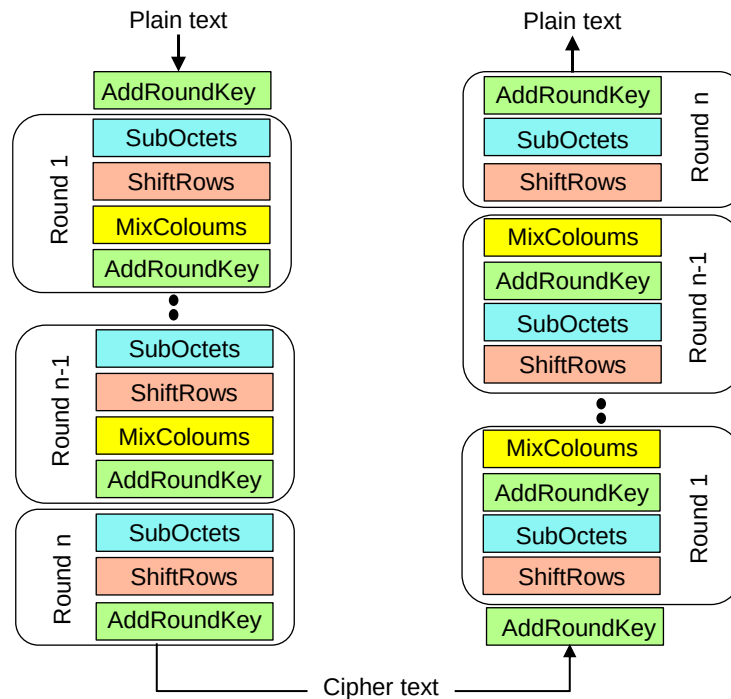


Figure 4 – AES encryption and decryption of block

Figure 4 illustrates the procedure for the encryption and decryption of a block of data. It consists of several rounds where there are four different steps for each round except the last round, where there are only three steps. The mix column layer is missing:

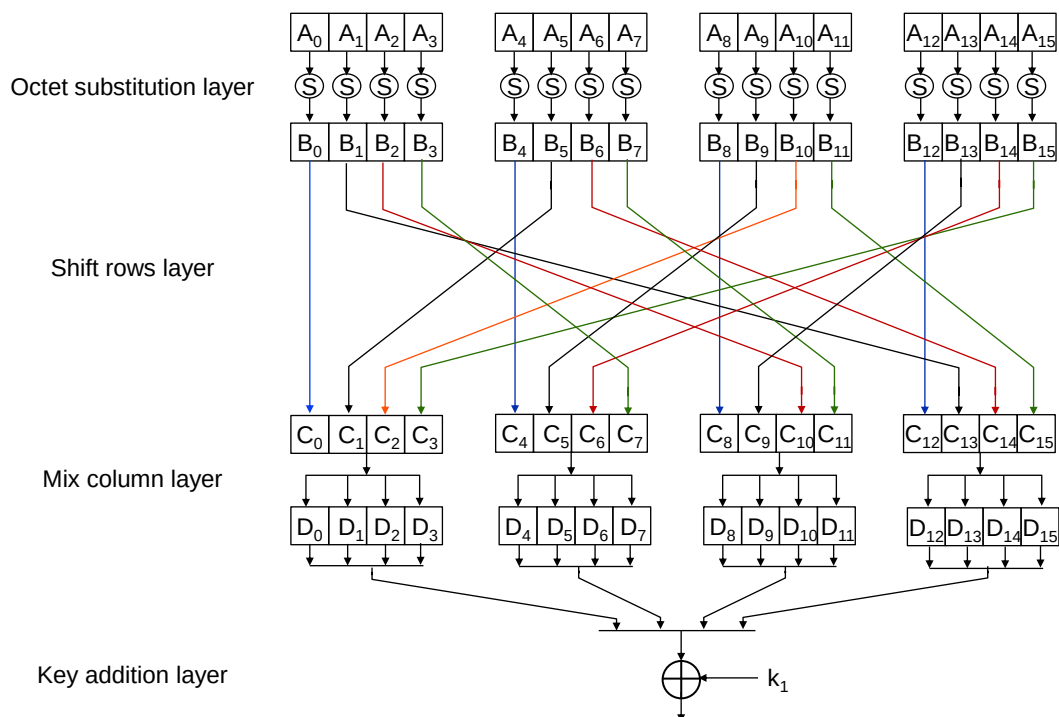


Figure 5 – Single round of block AES encryption

Figure 5 gives an overview of the encryption of a block of data, also called a data path, As also described in Figure 4, it consists of four layers.

The four layers are;

- a) Octet substitution layer;

- b) Shift row layer;
- c) Mix column layer; and
- d) Key addition layer.

The octet substitution layer provides confusion, i.e., each bit of the ciphertext should depend on several parts of the key, obscuring the relationship between the two.

The shift row and mix column layers provide diffusion, i.e., if a single bit of the plaintext is changed, then (statistically) half of the bits in the ciphertext should change, and similarly, if we change one bit of the ciphertext, then approximately one half of the plaintext bits should change.

At the beginning of AES encryption and decryption and at the very end, a subkey is added. This is called key whitening.

The number of rounds is dependent on the key size as shown in table 4.

Table 4 – Number of rounds in AES as function of key size

Key size	Number of rounds (n)
128 bits	10
192 bits	12
256 bits	14

The AES operations are octet oriented, i.e., the operations are performed on a complete octet.

The 128-bit block is split into 16 octets. The 16 octets block is modified as it moved through the different layers. The 16 octets may be conveniently represented by a 4×4 array of octet as shown in Figure 6, This array reflexes at every step the status of the block.

$$\begin{bmatrix} B_0 & B_4 & B_8 & B_{12} \\ B_1 & B_5 & B_9 & B_{13} \\ B_2 & B_6 & B_{10} & B_{14} \\ B_3 & B_7 & B_{11} & B_{15} \end{bmatrix}$$

Figure 6 – State array

The 4 octets in a column of the state array matrix is constitute a word and the state matrix holds 4 words.

In the following, each layer is shortly described.

8.2.2 The octet substitution layer

In the octet substitution layer, each octet in substituted by another octet using the following procedure:

- a) The octet A is mapped in a polynomial element as described in clause B.4.3.3.
- b) The inverse of this element is calculated as described in clause B.4.3.6.
- c) the resulting polynomial is mapped into an octet B.
- d) This octet consisting of bits b_0 to b_7 used as a vector is then multiplied by the 8×8 matrix shown in Figure 7 and then added to the vector to the right. The result is a new octet B'. This is the octet with which the octet A is substituted.

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \text{ Mod } 2$$

Figure 7 – Affine mapping

This is quite complicated process for each octets for each round. The result of an octet substitute is only dependent on the bit pattern of the octet. Instead of doing the calculation for each octets in a block, all possible 256 different octet values can be pre-calculated once and arranged as shown in Figure 8. This table is called the S-box.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Figure 8 – The S-box

An octet consists of two hex digits. To access the table, the first hex digit is used to go down along the left side to find the right row and the second hex digit is used to go along the top to find the right column. At the intersection of the row and the column the substitution octet is located. As an example, the octet $C2_{16}$ is substituted with octet 25_{16} .

8.2.3 The Shift Rows layer

Diffusion: If one bit is flip in the input, the change to the output is unpredictable and there should be no apparent relationship between the outputs before and after.

The shift rows step is shown in Figure 5. As illustrated, a complete octet is shifted for each step and the octets in a in a four-octet block are shifted to four different blocks. Figure 9 is another way of describing the shift row procedure. It is less illustrative, but more suited from a processing point of view.

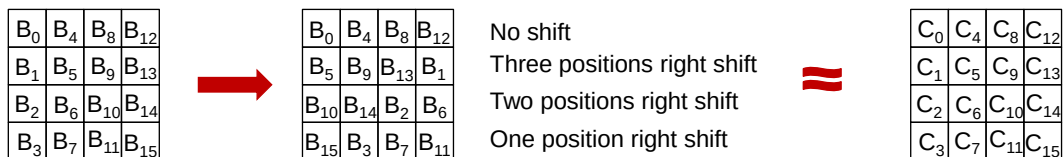


Figure 9 – Shift rows operation

8.2.4 The mix columns layer

$$\begin{bmatrix} D_0 \\ D_1 \\ D_2 \\ D_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \circ \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{bmatrix}$$

$$\begin{aligned} D_0 &= 02 \circ C_0 + 03 \circ C_1 + 01 \circ C_2 + 01 \circ C_3 \\ D_1 &= 01 \circ C_0 + 02 \circ C_1 + 03 \circ C_2 + 01 \circ C_3 \\ D_2 &= 01 \circ C_0 + 01 \circ C_1 + 02 \circ C_2 + 03 \circ C_3 \\ D_3 &= 03 \circ C_0 + 01 \circ C_1 + 01 \circ C_2 + 02 \circ C_3 \end{aligned}$$

Figure 10 – The mix column layer

Figure 10 how one of the first group of four octet are transformed in the mix column layer. The group as a vector is multiplied with a 4×4 matrix of fixed octet values as shown. Following the operation described in B.1.5, is seen that each resulting octet is a result of all four input octets. This is major contribution to the diffusion.

The other three groups are transformed in the same way.

The multiplications in Figure 10 are between two octets, which is something special. In B.3.3.3 it is described how it is possible to map between an octet and an element of the $GF(2^8)$ binary field. The fixed octet $01_{16} = \{0000\ 0001\}$ is then mapped to the 1 element $GF(2^8)$ binary field, $02_{16} = \{0000\ 0010\}$ is mapped the x element and $03_{16} = \{0000\ 0011\}$ is mapped to x+1 element. The octets C_0, C_1, C_2 and C_3 are mapped in a similar way.

8.2.5 The add round key layer

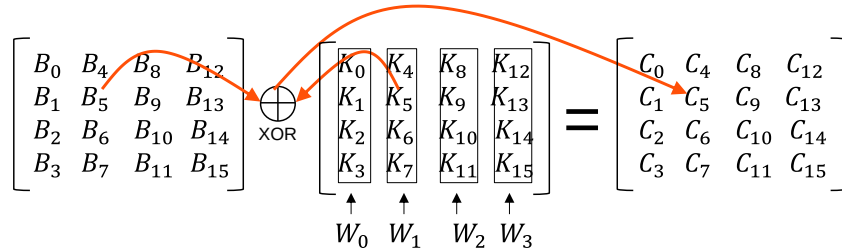


Figure 11 – State matrix XOR'ed with subkey

Before the start of the first round an add round key operation is performed and it is again performed at the end of each round. At each invocation of the add round key the state matrix is XOR'ed with a subkey also expressed in a 4×4 matrix as illustrated in Figure 11. The result is an updated state matrix. The XOR is performed by taking each element in the state matrix and XOR'ed with the corresponding element of the subkey matrix. Figure 11 also illustrates that a subkey is word oriented where a subkey consists of four word and a word consists of 4 octets.

A new subkey is needed for each add round key operation, which require key expansion of the symmetric key used for the cipher. The number of subkeys to be generated is the number of rounds plus one. As a subkey consists of four words, the number of words to be generated is then:

- 44 ($W_0 - W_{43}$) for 128 key size;
- 52 ($W_0 - W_{51}$) for 192 key size; and
- 60 ($W_0 - W_{59}$) for 256 key size.

The words for the subkeys are in arrays of n words, where n is the number of words in the symmetric key, i.e., $n = 4$ for a 128-bit key, $n = 6$ for a 192-bit key, and $n = 8$ for a 256-bit key. Array(0) is filled with the symmetric key.

Each following array of n words is generated from the array. This is illustrated in Figure 12 for a 128-bit symmetric key, but the principle is the same for other key sizes.

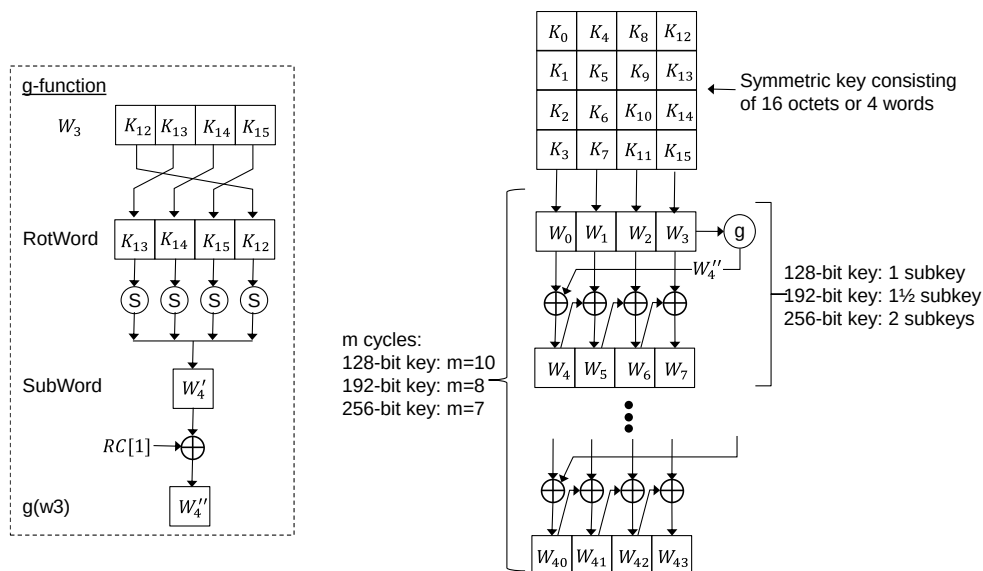


Figure 12 – Key expansion for 128-bit symmetric key

The first n words ($W_0 - W_{n-1}$) are filled with the symmetric key. The first word W_n of the following array is derived by taking W_{n-1} as input to a g -function to generate W_n'' which is then XORed with W_0 to generate W_n . The following words of second array are then $W_{n+1} = W_n \oplus W_1, \dots, W_{2n-1} = W_{2n-2} \oplus W_{n-1}$.

The remaining arrays are generated after the same principle.

The g -function for each circle takes the word $W_{n-1}, W_{2n-1}, \dots, W_{m \times n-1}$ as input. It then circles the octets one position to the left and thereby making the first octet the last octet. The result is called RotWord. Each octet of the RotWord is then substituted using the s-box to generate the SubWord $W_{i \times n}'$, where $i = 1, \dots, m$. Following that the SubWord is XORed with the round constant $RC[i]$ to generate the word $W_{i \times n}''$.

```

RC[1] = 0x0000 0001 0000 0000 0000 0000 0000 0000
RC[2] = 0x0000 0010 0000 0000 0000 0000 0000 0000
RC[3] = 0x0000 0100 0000 0000 0000 0000 0000 0000
RC[4] = 0x0000 1000 0000 0000 0000 0000 0000 0000
RC[5] = 0x0001 0000 0000 0000 0000 0000 0000 0000
RC[6] = 0x0010 0000 0000 0000 0000 0000 0000 0000
RC[7] = 0x0100 0000 0000 0000 0000 0000 0000 0000
RC[8] = 0x1000 0000 0000 0000 0000 0000 0000 0000
RC[9] = 0x0001 1011 0000 0000 0000 0000 0000 0000
RC[10] = 0x0011 0110 0000 0000 0000 0000 0000 0000

```

8.4 Operation modes for block cipher symmetric-key algorithms

8.4.1 Overview of block cipher operation mode

The cipher methods for block cipher methods considered above specify the encryption method of a single block. How the encrypted blocks are combined to provide cipher of a complete message is independent on the block cipher method.

The combination of a cipher method and an operation mode makes-up a symmetric key algorithm as used in communications protocol.

There are defined five modes of how the different blocks are combined to cover a total message:

- electronic codebook (ECB) mode;
- cipher block chaining (CBC) mode;
- cipher feedback (CFB) mode;
- output feedback (OFB) mode; and
- counter (CTR) mode.

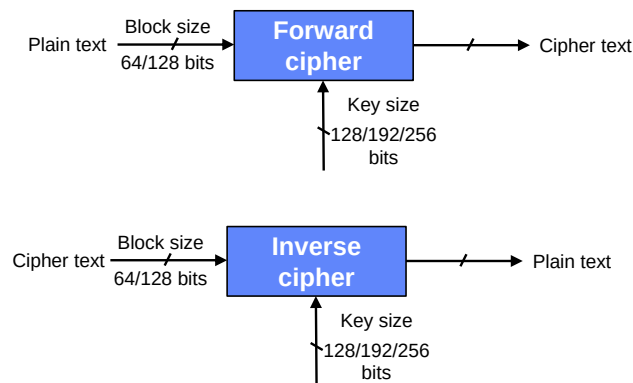


Figure 13 – Cipher operation on a single block

In the figures associated with the different EAS modes, the block shown in the figure 13 represents the complete encryption of a block as described for the five cipher modes described above.

ISO/IEC 10116:2017 gives a formal specification for the modes. NIST SP 800-38A also provides details.

Beyond the five modes listed above, the different cipher methods are also used for

- Authenticated encryption algorithms.
- Integrity check values (ICVs).

8.4.2 Electronic codebook (ECB) cipher mode

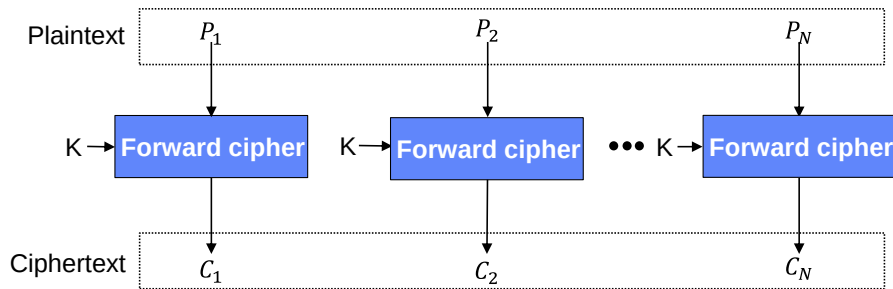


Figure 14 – Electronic codebook (ECB) cipher mode encryption

Figure 14 illustrates the encryption process while Figure 15 illustrates the decryption process.

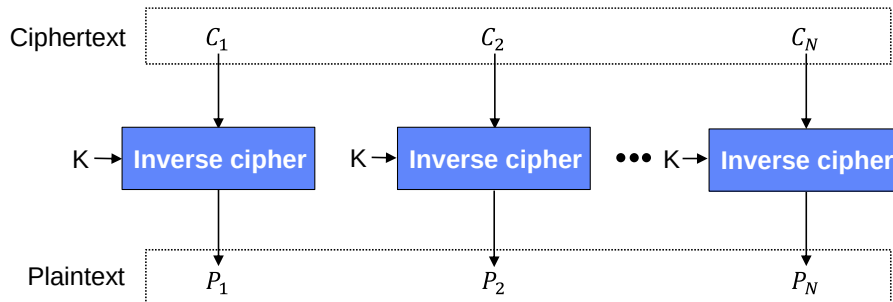


Figure 15 – Electronic codebook (ECB) cipher mode decryption

Electronic Code Book (ECB) is the simplest cipher operation mode.

It has the advantage that forward cipher operations can be performed in parallel for encryption and that the inverse cipher operation can also be performed in parallel for decryption.

It has the disadvantage that the same plaintext and the same symmetric key always produce the ciphertext which may make it subject for cryptanalysis. In application where that could be a problem, this operation mode should not be used.

This mode requires padding as specified in clause 8.3.7.

The formal definitions of electronic codebook mode symmetric-key algorithms using the **ALGORITHM** information object class for different block ciphers are listed in Rec. ITU-T X.510 | ISO/IEC 9594-11.

8.4.3 Cipher block chaining (CBC) cipher mode

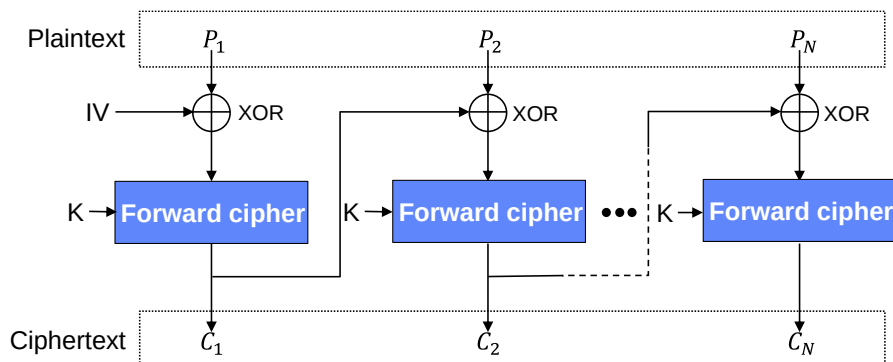


Figure 16 – Cipher block chaining (CBC) mode encryption

Figure 16 illustrates how the different blocks are processed during encryption.

In cipher block chaining, the first block of plain text is modified by 'XOR'ing it with a random generated bitstring of the same size as the block size, called the initialization vector. Forward cipher is then used to generate a ciphertext block. This ciphertext block is used to modify ('XOR'ing) the second block of plain text. Forward cipher is then used to

generate a ciphertext block. This second resulting block of ciphertext is then used to modify the third block of plaintext, etc, until all blocks have been handled. The concatenation of ciphertext blocks is the result of the encryption process.

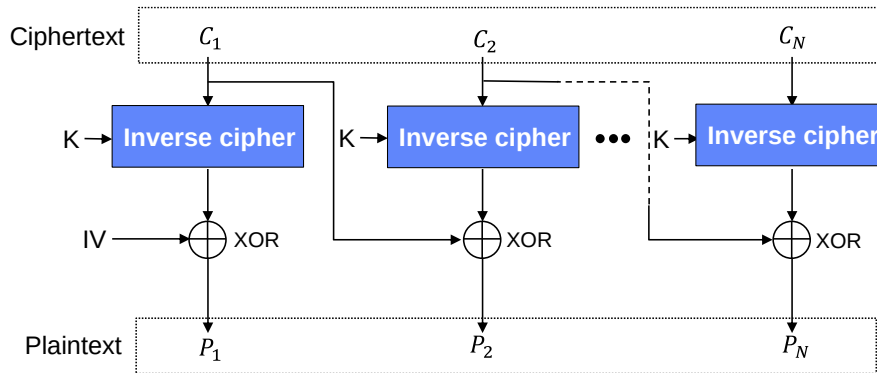


Figure 17– Cipher block chaining (CBC) mode decryption

The same initialization vector is also used for decryption. However, a new initialization vector is required for each invocation of the algorithm. As a result, when the same plaintext is encrypted a second time, the ciphertext will be different.

This mode requires padding as specified in clause 8.3.7.

The formal definitions of cipher block chaining mode symmetric-key algorithms using the **ALGORITHM** information object class for different block ciphers are listed in ITU-T X.510 | ISO/IEC 9594-11.

8.4.4 Cipher feedback (CFB) mode

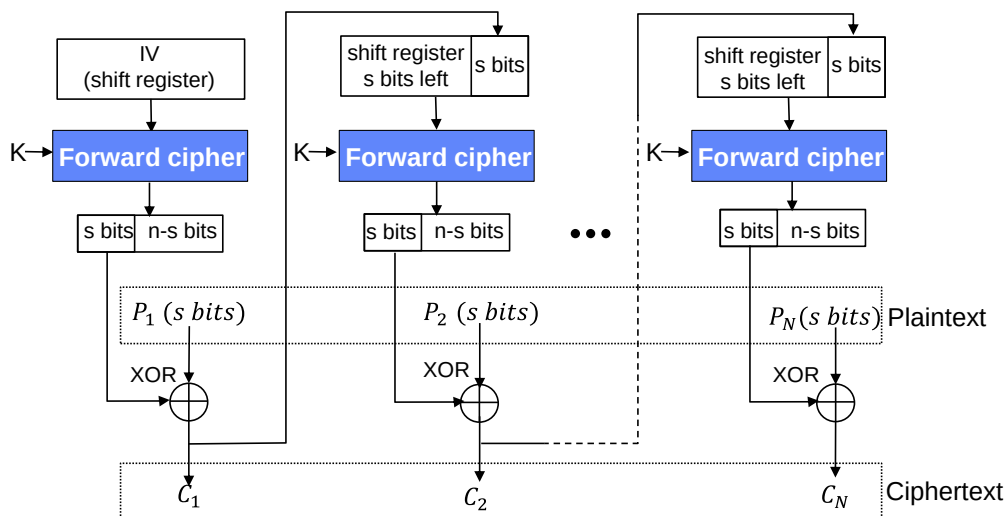


Figure 18 – Cipher feedback (CFB) mode encryption

Figure 18 illustrates the procedure for cipher feedback (CFB) mode encryption while Figure 19 illustrates decryption procedure. In both cases the generation of ciphertext from plaintext and visa versa in done in chunk of length s , which may be less than the block size n , e.g., 8 bits. If the message length is always a multiple of s , no padding is needed.

Firstly, the initialization vector (IV) is converted to ciphertext using forward encryption. s bits of the result are used in the following, while the remain bits ($n-s$) bits are discarded. The s bits are then XORed with the first s bits of the plaintext during encryption to generate ciphertext or with the first s bits of the ciphertext during decryption to generate plaintext.

To convert the next s bits, the least ($n-s$) bits of the IV is concatenated with the resulting s bit from the previous operation to generate a new block subject going through forward cipher to generate a block of which only s bits are uses. The procedure continues in this fashion to its completion.

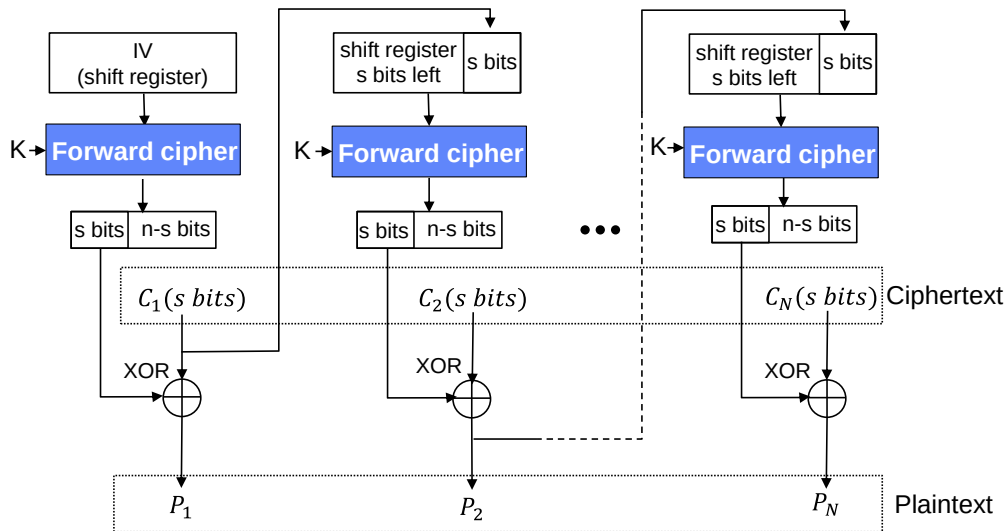


Figure 19 – Cipher feedback (CFB) mode decryption

When using Cipher-Feedback (CFB) mode the ciphertext will be the same size as the plaintext as padding is not required. The initialisation vectors (IV) shall be unique for each new encryption/decryption. The IV shall be specified as a dynamic parameter for the **ALGORITHM** information object while the number of s-bits shall be specified as a fixed parameter.

The formal definitions of symmetric-key algorithms using the **ALGORITHM** information object class for different block ciphers may be found in ITU-T X.510 | ISO/IEC 9594-11.

The formal definitions of cypher feedback mode symmetric-key algorithms using the **ALGORITHM** information object class for different block ciphers are listed in ITU-T X.510 | ISO/IEC 9594-11.

8.4.5 Output feedback cipher mode

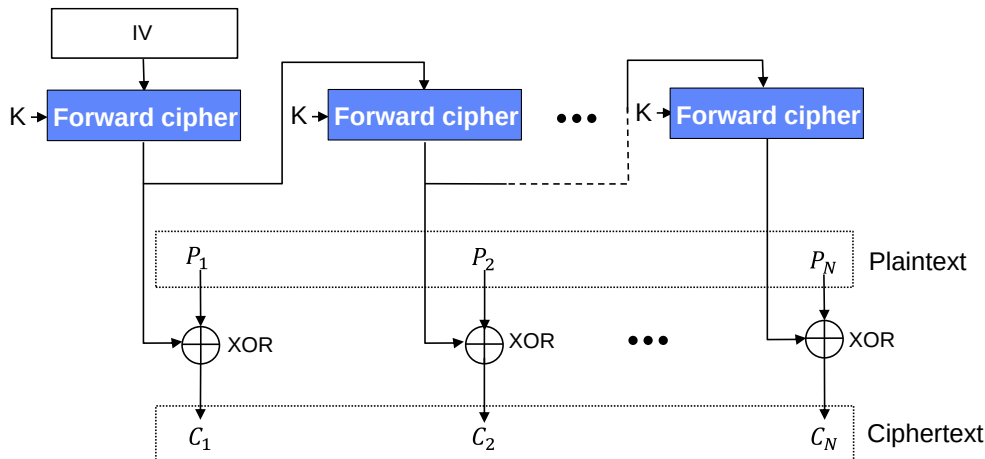


Figure 20 – Output feedback (OFB) mode encryption

As it is seen from both Figure 20 and 21, the forward cipher invoked on the input IV and the forward cipher in invoked again on the result. This repeated until there is an enciphered block for each block of plaintext or ciphertext. During encryption, each block of plaintext is XORed with the corresponding enciphered block to form a block of ciphertext. Likewise, during decryption, each block of cipher is XORed with the corresponding enciphered block to form a block of plaintext.

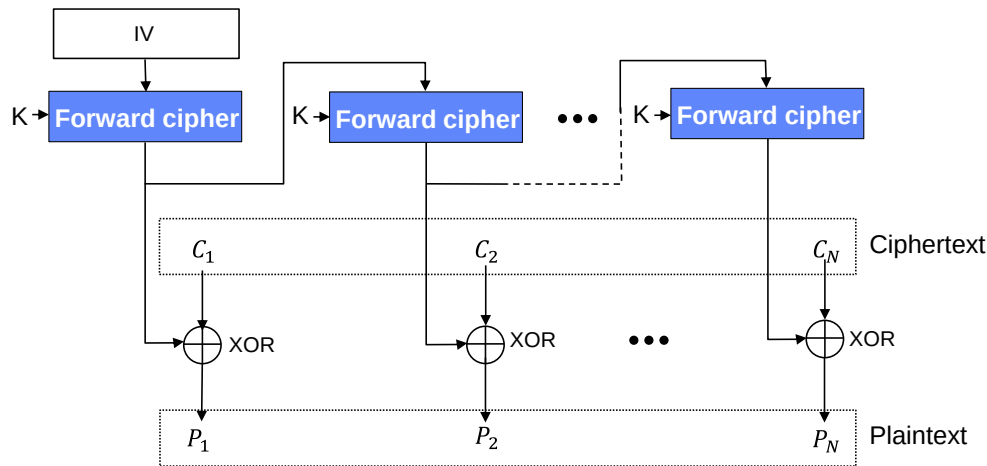


Figure 21 – Output feedback (OFB) mode decryption

It is possible in advance of an encryption or decryption in advance to apply the forward cipher to the IV and its derivatives,

The formal definitions of output feedback mode symmetric-key algorithms using the **ALGORITHM** information object class for different block ciphers are listed in ITU-T X.510 | ISO/IEC 9594-11.

8.4.6 Counter cipher mode

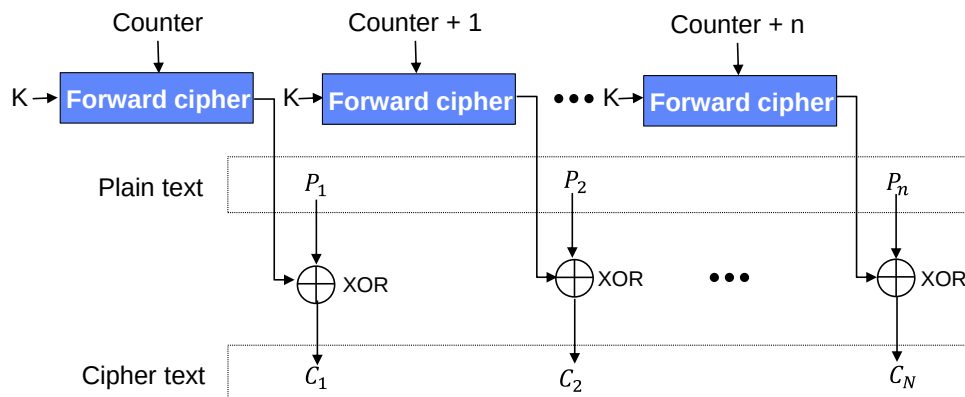


Figure 22 – Encryption in counter mode

In counter mode, sequence numbers, called counters, are used together with a block of plain text, one at a time as. An initial counter value is used for the first block of plain text. This counter is the size of half the block size. It is concatenate with a nonce also having a size of half the block. The result is encrypted, and the result is XORed with the first plain text block to create the first ciphertext block. The following blocks are treated in the same way, except that the counter is increased with a fixed value, e.g., 1, for each block processed. The same nonce is used for the entire process and for decryption. However, it is important that a new nonce is generated for each message to be encrypted.

This mode has the advantage that a multi-processor can encrypt/decrypt the blocks in parallel. It has the disadvantage that it requires that both parties in a communication agree on the initial value for counter and the increment to be used.

No padding is needed. If the last block is less than 128 bits, the last, unused part of the encrypted counter is discarded.

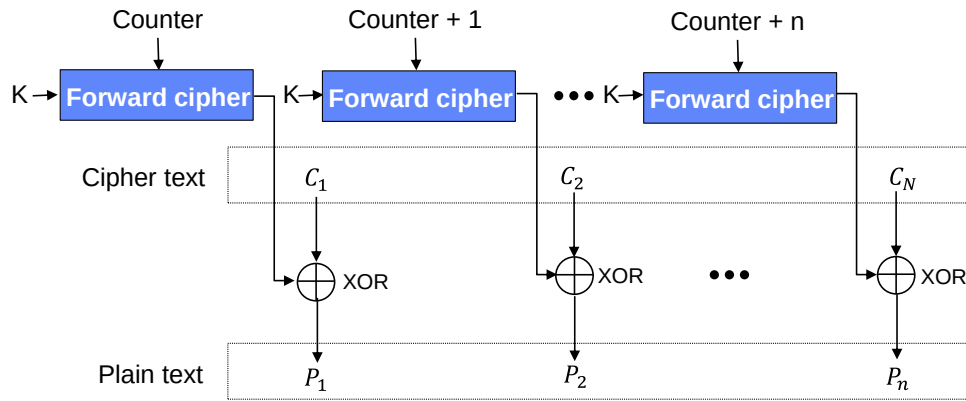


Figure 23 – Decryption in counter mode

The formal definitions of counter mode symmetric-key algorithms using the **ALGORITHM** information object class for different block ciphers are listed in ITU-T X.510 | ISO/IEC 9594-11.

8.3.7 Padding

Some of the cipher operation modes above requires padding.

The length of the padding is given by:

$$x \equiv l \pmod{b} \sim x \equiv l \pmod{128}$$

except if x is '0', the value of x is replaced by b (which is in same equivalence class as '0'). This means that a message is always padded, even when the message size is a multiple of the block size.

PKCS#7 as defined in IETF RFC 2315 specifies a the most used padding technique. According to this technique, all the octets in the padding have the same value and the value is the length (in hex) of the padding, i.e., the padding is:

$$\begin{aligned} &01 \text{ if } x = 1 \\ &02 \text{ if } x = 2 \\ &\dots \\ &80 \text{ if } x = 128 \end{aligned}$$

9 Hash algorithms

9.1 General

A hash algorithm is the specification on how to take an arbitrary message (bitstring) 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:

- 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 11 or as part of integrity check value (ICV) as discussed in clauses 14.2 and 14.4. Hash algorithms are also used for ensuring blockchain integrity (ensuring immutability).
- For generating fingerprints to generate a small representation of a larger data item, like a public-key certificate.
- Key derivation functions: This is a function that derives one or more symmetric keys from a shared secret value generated in some way.
- For random number generation.

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