
**Information technology — JPEG 2000
image coding system —**

**Part 8:
Secure JPEG 2000**

*Technologies de l'information — Système de codage d'images JPEG
2000 —*

Partie 8: JPEG 2000 sécurisé



Reference number
ISO/IEC 15444-8:2023(E)

Documents
Document Sharing Hub
© ISO/IEC 2023



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2023

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

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted.

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and <https://patents.iec.ch>. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by ITU-T (as ITU-T Rec T.807) and drafted in accordance with its editorial rules, in collaboration with Joint Technical Committee ISO/IEC JTC 1, *Information technology, Subcommittee SC 29, Coding of audio, picture, multimedia and hypermedia information*.

This second edition cancels and replaces the first edition (ISO/IEC 15444-8:2007), which has been technically revised. It also incorporates the Amendment ISO/IEC 15444-8:2007/Amd 1:2008.

A list of all parts in the ISO/IEC 15444 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

INTERNATIONAL STANDARD ISO/IEC 15444-8
RECOMMENDATION ITU-T T.807

Information technology – JPEG 2000 image coding system: Secure JPEG 2000

Summary

Rec. ITU-T T.807 | ISO/IEC 15444-8 provides a syntax that allows security services to be applied to JPEG 2000 coded image data. Security services include confidentiality, integrity verification, source authentication, conditional access, secure scalable streaming and secure transcoding. The syntax allows these security services to be applied to coded and uncoded image data in part or in its entirety. This maintains the inherent features of JPEG 2000 such as scalability and access to various spatial areas, resolution levels, colour components, and quality layers, while providing security services on these elements.

This second edition:

- 1) consolidates all solved/outstanding amendments and corrigenda published since the first edition;
- 2) removes the clause on Registration Authority (first edition's clause 7);
- 3) removes the annex of patent statements (first edition's Annex D).

This Recommendation | International Standard was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*, in collaboration with ITU-T SG16. The identical text is published in ISO/IEC as ISO/IEC 15444-8.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T T.807	2006-05-29	16	11.1002/1000/8830
1.1	ITU-T T.807 (2006) Amd. 1	2008-03-15	16	11.1002/1000/9364
2.0	ITU-T T.807 (V2)	2023-02-13	16	11.1002/1000/15208

Keywords

Image security, jpsec, JPEG 2000, security, still image.

* To access the Recommendation, type the URL <http://handle.itu.int/> in the address field of your web browser, followed by the Recommendation's unique ID. For example, <http://handle.itu.int/11.1002/1000/11830-01>

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2023

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

CONTENTS

	<i>Page</i>
1 Scope	1
2 Normative references	1
3 Definitions	1
4 Symbols and abbreviations	4
5 JPSEC syntax	5
5.1 JPSEC framework overview	5
5.2 JPSEC security services	6
5.3 Comments on design and implementation of secure JPSEC systems	7
5.4 Byte aligned segment (BAS)	8
5.5 Main security marker (SEC)	9
5.6 JPSEC tools	13
5.7 Zone of Influence (ZOI) syntax	16
5.8 Protection method template syntax (T)	25
5.9 Processing domain syntax (PD)	34
5.10 Granularity syntax (G)	35
5.11 Value list syntax (V)	36
5.12 Relationships among ZOI, Granularity (G) and Value List (VL)	37
5.13 In-codestream security marker (INSEC)	38
6 Normative-syntax usage examples (informative)	39
6.1 ZOI examples	39
6.2 Key information template examples	43
6.3 JPSEC normative tool examples	44
6.4 Distortion field examples	50
Annex A Guidelines and use cases	52
A.1 A class of JPSEC applications	52
Annex B Interoperability	58
B.1 Rec. ITU-T T.800 ISO/IEC 15444-1 – Core coding system	58
B.2 Rec. ITU-T T.808 ISO/IEC 15444-9 – JPIP	58
B.3 Rec. ITU-T T.810 ISO/IEC 15444-11 – JPWL	59
Annex C File format security	62
C.1 Scope	62
C.2 Introduction	62
C.3 Extension to ISO base media file format	64
C.4 Elementary stream and sample definitions	72
C.5 Protection at file format level	74
C.6 Examples (Informative)	75
C.7 Boxes defined in ISO/IEC 14496-12 (informative)	85
Annex D Technology examples	90
D.1 Introduction	90
D.2 A flexible access control scheme for JPEG 2000 codestreams	90
D.3 A unified authentication framework for JPEG 2000 images	92
D.4 A simple packet-based encryption method for JPEG 2000 codestreams	94
D.5 Encryption tool for JPEG 2000 access control	97
D.6 Key generation tool for JPEG 2000 access control	99
D.7 Wavelet and bitstream domain scrambling for conditional access control	102
D.8 Progressive access for JPEG 2000 codestream	104
D.9 Scalable authenticity of JPEG 2000 codestreams	106
D.10 JPEG 2000 data confidentiality and access control system based on data splitting and luring	108
D.11 Secure scalable streaming and secure transcoding	111
Bibliography	115

FIGURES

	<i>Page</i>
Figure 1 – Overview of the conceptual steps in JPSEC framework	5
Figure 2 – Byte aligned segment (BAS) structure	8
Figure 3 – Main security marker segment syntax	9
Figure 4 – Main security marker syntax when multiple marker segments are used	10
Figure 5 – Codestream security parameters (P_{SEC}) syntax	10
Figure 6 – TRLCp tag descriptor (P_{TRLCP}) syntax	11
Figure 7 – Use of multiple JPSEC tools	12
Figure 8 – JPSEC tool syntax ($Tool^{(i)}$)	13
Figure 9 – Parameters (P_{ID}) syntax for JPSEC normative tools ($t = 0$)	14
Figure 10 – ID_{RA} syntax	15
Figure 11 – Zone of Influence conceptual structure	17
Figure 12 – ZOI syntax	17
Figure 13 – Zone description class structure (DC_{zoi})	18
Figure 14 – Zone syntax consists of a description class and one or more parameter sets	18
Figure 15 – Distortion field syntax	20
Figure 16 – Distortion field syntax	21
Figure 17 – Relative importance field syntax	21
Figure 18 – Bit-rate field syntax	22
Figure 19 – ZOI example using image related descriptions	23
Figure 20 – ZOI example using image related and non-image related descriptions	23
Figure 21 – A second ZOI example using image related and non-image related descriptions	24
Figure 22 – ZOI description parameter syntax	24
Figure 23 – Decryption template syntax	26
Figure 24 – Block cipher template syntax	27
Figure 25 – Stream cipher template syntax	28
Figure 26 – Asymmetric cipher template syntax	29
Figure 27 – Authentication template syntax	29
Figure 28 – Hash-based authentication template	30
Figure 29 – Cipher-based authentication template syntax	31
Figure 30 – Digital signature template syntax	32
Figure 31 – Hash template syntax	33
Figure 32 – Key information template syntax	33
Figure 33 – ITU-T X.509 certificate syntax	34
Figure 34 – Processing domain syntax	34
Figure 35 – Granularity syntax	35
Figure 36 – Value list field syntax	37
Figure 37 – Granularity Level (GL) is resolution	37

Figure 38 – Granularity Level (GL) is layer	38
Figure 39 – In-codestream security marker syntax	38
Figure A.1 – Overview of a secure JPEG 2000 image distribution application	52
Figure A.2 – Legend description	53
Figure A.3 – Encryption procedure	54
Figure A.4 – Decryption procedure	54
Figure A.5 – Signature generation procedure	55
Figure A.6 – Authentication procedure	56
Figure A.7 – ICV generation procedure	56
Figure A.8 – Integrity check procedure	57
Figure B.1 – Typical JPWL and JPSEC combination	60
Figure C.1 – System diagram for time-sequenced scalable media	63
Figure C.2 – Self-contained ES and scalable composed ES	73
Figure C.3 – Self-contained ES and decodable composed ES.....	73
Figure C.4 – Relationship between iloc, iinf and ipro	74
Figure C.5 – An example sample description entry protected by authentication scheme followed by description scheme	75
Figure C.6 – Example 1: Item-based protection of JP2 file (authentication).....	76
Figure C.7 – Example 2: Item-based protection of a JPEG 2000 images (encryption)	77
Figure C.8 – Example 2: Secure transcoding to lower resolution (discarding resolution 2).....	77
Figure C.9 – Example 3: Item-based protection of a JPEG 2000 image (Authentication)	78
Figure C.10 – Example 3: Transcoding to resolution 1	79
Figure C.11 – Example 4: Sample-based protection of a time-sequenced JPEG 2000 pictures	80
Figure C.12 – Example 4: Secure transcoding to lower SNR quality (layer 1)	81
Figure C.13 – Example 5: Sample-based protection for video browsing or video summarization.....	82
Figure C.14 – Example 5: Transcoding to shorter time length (discarding the last 5000 pictures)	82
Figure C.15 – Example 6: Authentication transcoding, discarding received but unverifiable packets.....	83
Figure C.16 – Motion JPEG 2000 file with detailed box structure.....	83
Figure C.17 – Simplified Motion JPEG 2000 box structure showing references	84
Figure C.18 – Simplified Motion JPEG 2000 box structure showing references after length changing protection operations.....	84
Figure C.19 – JPM file with detailed box structure	85
Figure C.20 – Simplified JPM box structure showing references	85
Figure C.21 – JPM box structure showing references after length changing protection operations	85
Figure D.1 – SEC segment syntax	91
Figure D.2 – P _{ID} field syntax	91
Figure D.3 – TP _{ID} field syntax.....	91
Figure D.4 – AK _{info} field syntax	92
Figure D.5 – Image protection using unified authentication framework for JPEG 2000.....	93

	<i>Page</i>
Figure D.6 – Packet-based encryption principle.....	95
Figure D.7 – Overview of this technology	100
Figure D.8 – Block diagram for wavelet domain scrambling.....	102
Figure D.9 – Block diagram for bitstream domain scrambling	103
Figure D.10 – Non-normative protection tool syntax in the case of multiple keys	103
Figure D.11 – Syntax for AP: Wavelet domain scrambling (left), Bitstream domain scrambling (right)	104
Figure D.12 – Technical overview of this technology.....	105
Figure D.13 – Non-normative tool syntax	107
Figure D.14 – Security parameters TP _{ID} syntax	108
Figure D.15 – System overview	110
Figure D.16 – JPSEC enables end-to-end security and mid-network secure transcoding	112
Figure D.17 – An example of forming a JPSEC codestream.....	113

TABLES

	<i>Page</i>
Table 1 – Main security parameter values	10
Table 2 – Codestream security parameters (P _{SEC}) in first SEC marker segment	11
Table 3 – Semantics for F _{PSEC} values (FBAS).....	11
Table 4 – Parameter field for TRLC _P tag descriptor (P _{TRLC_P}).....	12
Table 5 – JPSEC tool parameter values.....	13
Table 6 – JPSEC normative tool Template ID values (ID _T)	14
Table 7 – JPSEC normative tool parameter values.....	15
Table 8 – Parameters values in ID _{RA} syntax	15
Table 9 – ID values for JPSEC non-normative tools (ID _{RA,id}).....	16
Table 10 – Zone of influence field (ZOI) parameter values	17
Table 11 – Zone parameter values.....	18
Table 12 – Description class indicator value	18
Table 13 – Image related description class	18
Table 14 – Non-image related description class	19
Table 15 – Distortion field parameter values.....	20
Table 16 – Distortion field parameter values.....	21
Table 17 – Relative importance field parameter values.....	21
Table 18 – Bit-rate field parameter values.....	22
Table 19 – Pzoi ⁱ parameter values	24
Table 20 – Mzoi parameter values.....	25
Table 21 – Template ID values (ID _T)	25
Table 22 – Decryption template parameter values	26
Table 23 – Marker emulation flag values (ME _{decry}).....	26

Table 24 – Cipher identifier values (CT_{decry})	26
Table 25 – Block cipher identifier values (CT_{decry})	26
Table 26 – Stream cipher identifier values (CT_{decry})	27
Table 27 – Asymmetric cipher identifier values (CT_{decry})	27
Table 28 – Block cipher template values	27
Table 29 – Block cipher mode values (M_{bc})	28
Table 30 – Padding mode for block cipher (P_{bc})	28
Table 31 – Stream cipher template values	28
Table 32 – Asymmetric cipher template values	29
Table 33 – Authentication template parameter values	29
Table 34 – Authentication methods (M_{auth})	29
Table 35 – Hash-based authentication template parameter values	30
Table 36 – Hash-based authentication method identifier (M_{HMAC})	30
Table 37 – Hash function identifier (H_{HMAC})	31
Table 38 – MAC template values	31
Table 39 – Cipher-based authentication method (C_{CMAC})	32
Table 40 – Digital signature template values	32
Table 41 – Digital signature methods (M_{DS})	32
Table 42 – Hash template parameter values	33
Table 43 – Key template values	33
Table 44 – Key information identifier values (KID_{KT})	34
Table 45 – ITU-T X.509 certificate values (KI_{KT} if $KID_{KT} = 2$)	34
Table 46 – Encoding rule values (ER_{KT})	34
Table 47 – Processing domain parameters	35
Table 48 – Processing Domain (PD) parameter values	35
Table 49 – Processing domain field (F_{PD}) parameter values in wavelet coefficient domain and quantized wavelet coefficient domain	35
Table 50 – Processing domain field (F_{PD}) parameter values in codestream domain	35
Table 51 – Granularity parameter values (G)	36
Table 52 – Processing order values (PO)	36
Table 53 – Granularity level values (GL)	36
Table 54 – Value list field (V) parameter values	37
Table 55 – In-codestream security parameter values (INSEC)	39
Table 56 – Relevance zone values (R)	39
Table 57 – ZOI in example 1	39
Table 58 – ZOI in example 2	40
Table 59 – ZOI in example 3	41
Table 60 – ZOI in example 4	41
Table 61 – ZOI in example 5	42

	<i>Page</i>
Table 62 – ZOI in example 6	43
Table 63 – Key information in example 1	43
Table 64 – Key information in example 2	43
Table 65 – Key information in example 3	44
Table 66 – Key information in example 4	44
Table 67 – SEC marker segment for example 1	45
Table 68 – ZOI example	45
Table 69 – P _{ID} example	46
Table 70 – Decryption template example	47
Table 71 – Key template example	47
Table 72 – The SEC marker segment	48
Table 73 – ZOI signalling	48
Table 74 – P _{ID} signalling parameters	49
Table 75 – Associating distortion field to two data segments (extension of ZOI example 3 in 6.1.3)	50
Table 76 – Signalling a range of packets and associating distortions for each packet	51
Table C.1 – List of existing and new boxes	64
Table D.1 – Example parameters for this scheme	91
Table D.2 – P _{ID} parameters	91
Table D.3 – TP _{ID} parameters	92
Table D.4 – AK _{info} parameters	92
Table D.5 – Syntax for semi-fragile authentication	93
Table D.6 – Example of Zone of Influence, with spatial coordinates, resolutions and layers	95
Table D.7 – Decryption template description, in the case of AES-192/CBC	96
Table D.8 – Processing domain syntax	96
Table D.9 – Granularity and value list syntax	97
Table D.10 – Example parameters for this technology	98
Table D.11 – Example ZOI of this key generation tool	98
Table D.12 – P _{ID} for this technology	99
Table D.13 – Example of decryption template of this technology	99
Table D.14 – Recommended parameter in this technology	100
Table D.15 – Example ZOI of this key generation tool	101
Table D.16 – P _{ID} for this technology	101
Table D.17 – Example of decryption template of this technology	102
Table D.18 – Syntax and semantic for P _{ID}	103
Table D.19 – Syntax and semantic for AP	104
Table D.20 – Example parameters for this tool	105
Table D.21 – Example ZOI of this technology	106
Table D.22 – P _{ID} for this technology	106

Table D.23 – Example of decryption template of this technology 106

Table D.24 – Non-normative tool parameters 108

Table D.25 – Security parameters..... 108

Table D.26 – Parameter values for this tool 111

Table D.27 – Parameter values for template protection tool, processing domain and granularity 114

Table D.28 – Parameter values for authentication template protection tool..... 114

Introduction

In the "Digital Age", the Internet provides many new opportunities for right-holders regarding the electronic distribution of their work (books, videos, music, images, etc.).

At the same time, new information technology radically simplifies the access of content for the user. This goes hand in hand with the all-pervasive problem of pirated digital copies – with the same quality as the originals – and "file-sharing" in peer-to-peer networks, which gives rise to continued complaints about great losses by the content industry.

World Intellectual Property Organization (WIPO) and its Member countries (170) have an important role to play in assuring that copyright, and the cultural and intellectual expression it fosters, remains well protected in the 21st century. The new Digital economy and the creative people in every country of the world depend on it. Also in December 1996, WIPO Copyright Treaty (WCT) has been promulgated with two important articles (11 and 12) about technological measures and obligations concerning Right Management Information:

Article 11

Obligations concerning Technological Measures

Contracting Parties shall provide adequate legal protection and effective legal remedies against the circumvention of effective technological measures that are used by authors in connection with the exercise of their rights under this Treaty or the Berne Convention and that restrict acts, in respect of their works, which are not authorized by the authors concerned or permitted by law.

Article 12

Obligations concerning Rights Management Information

(1) Contracting Parties shall provide adequate and effective legal remedies against any person knowingly performing any of the following acts knowing, or with respect to civil remedies having reasonable grounds to know, that it will induce, enable, facilitate or conceal an infringement of any right covered by this Treaty or the Berne Convention:

(i) to remove or alter any electronic rights management information without authority;

(ii) to distribute, import for distribution, broadcast or communicate to the public, without authority, works or copies of works knowing that electronic rights management information has been removed or altered without authority.

(2) As used in this Article, "rights management information" means information which identifies the work, the author of the work, the owner of any right in the work, or information about the terms and conditions of use of the work, and any numbers or codes that represent such information, when any of these items of information is attached to a copy of a work or appears in connection with the communication of a work to the public.

This treaty provides a solid foundation to protect Intellectual Property. As of 2004, about 50 countries ratified this important treaty. Therefore, it is expected that tools and protective methods that are recommended in JPEG 2000 needs ensure the security of transaction, protection of content (IPR), and protection of technologies.

Security issues, such as authentication, data integrity, protection of copyright and Intellectual Property, privacy, conditional access, confidentiality, transaction tracing, to mention a few, are among important features in many imaging applications targeted by JPEG 2000.

The technological means of protecting digital content are described and can be achieved in many ways such as digital watermarking, digital signature, encryption, metadata, authentication, and integrity checking.

This Recommendation | International Standard intends to provide tools and solutions in terms of specifications that allow applications to generate, consume, and exchange Secure JPEG 2000 codestreams. This is referred to as **JPSEC**.

INTERNATIONAL STANDARD ITU-T RECOMMENDATION

Information technology – JPEG 2000 image coding system: Secure JPEG 2000

1 Scope

This Recommendation | International Standard specifies the framework, concepts, and methodology for securing JPEG 2000 codestreams. The scope of this Recommendation | International Standard is to define:

- 1) a normative codestream syntax containing information for interpreting secure image data;
- 2) informative examples of JPSEC tools in typical use cases;
- 3) informative guidelines on how to implement security services and related metadata.

The scope of this Recommendation | International Standard is not to describe specific secure imaging applications or to limit secure imaging to specific techniques, but to create a framework that enables future extensions as secure imaging techniques evolve.

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies. At the time of publication, the editions indicated in dated references 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.

- Recommendation ITU-T T.800 | ISO/IEC 15444-1, *Information technology – JPEG 2000 image coding system: Core coding system*.
- Recommendation ITU-T T.801 | ISO/IEC 15444-2, *Information technology – JPEG 2000 image coding system: Extensions*.
- Recommendation ITU-T T.808 | ISO/IEC 15444-9, *Information technology – JPEG 2000 image coding system: Interactivity tools, APIs and protocols*.
- Recommendation ITU-T T.814 | ISO/IEC 15444-15, *Information technology – JPEG 2000 image coding system: High-Throughput JPEG 2000*.
- Recommendation ITU-T X.509 | ISO/IEC 9594-8, *Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks*.
- ISO/IEC 14496-12, *Information technology – Coding of audio-visual objects – Part 12: ISO base media file format*.

3 Definitions

For the purposes of this document, the definitions contained in Rec. ITU-T T.800 | ISO/IEC 15444-1, Rec. ITU-T T.801 | ISO/IEC 15444-2, Rec. ITU-T T.808 | ISO/IEC 15444-9, Rec. ITU-T T.814 | ISO/IEC 15444-15, and ISO/IEC 14496-12, and the following apply.

3.1 access control: Prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner.

3.2 adaptive-format decoder: Process to decode a codestream which is not fully compliant with the normative part of the coding standard.

NOTE – It reconstructs the media (possibly with low quality or resolution) even if the codestream has missing packets or inconsistent packet headers. For example, an adaptive-format decoder is able to understand a simply transcoded codestream, such as the one that has its highest resolution packets removed.

3.3 adaptor/transcoder: process to transform media data to lower scalability level, like lower resolution or lower quality or bit-rate, by removing portions of the file.

NOTE – The adaptor/transcoder can transform media data based on the information specified in this Recommendation | International Standard. An adaptor/transcoder updates byte offset values in file format parameters that are impacted by the process.

3.3.1 authentication adaptor/transcoder: Process to remove data that is not verifiable with the available media and authentication data.

NOTE – For example, in a streaming system, some media packets can be lost during transmission. A file format receiver can reconstruct the received data to the best of its ability based on the available data. Then, an authentication adaptor/transcoder can determine which data can be verified, and then remove the packets that are not verified. The resulting file only contains the decodable, verified data.

3.3.2 JPEG 2000-aware adaptor/transcoder: Process that combines one or more scalable composed ESs to form a fully decodable media codestream.

3.3.3 simple adaptor/transcoder: Process to transform data based on information specified by this Recommendation | International Standard.

NOTE – A simple adaptor/transcoder might not be capable of generating media headers or modifying packet indices. It simply retrieves data pointed by pointer structure and removes the wrappers, and the resulting codestream can be decoded by adaptive-format decoder, which can cope with missing packets and inconsistent headers.

3.4 authentication: Process of verifying an identity claimed by or for a system entity.

3.4.1 source authentication: Verification that a source entity (say, user/party) is in fact the claimed source entity.

3.5 ByteData: Structure used to wrap a data segment which is physically located in a composed ES.

NOTE – It is contained inside a container structure.

3.6 confidentiality: Property that information is not made available or disclosed to unauthorized individuals, entities or processes.

3.7 container: Container structure is used to wrap a sample in a composed ES.

NOTE – It might contain any number of ByteData or pointer structures, but is not allowed to contain another container structure.

3.8 data splitting: Method to protect sensitive data from unauthorized access by encrypting the data and storing different portions of the file on different servers.

NOTE – When split data is accessed the parts are retrieved, combined and decrypted. An unauthorized person would need to know the locations of the servers containing the parts, be able to get access to each server, know what data to combine, and how to decrypt it.

3.9 decryption, deciphering: Inverse transformation of the encryption.

3.10 digital signature: Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery, e.g., by the recipient.

3.11 elementary stream (ES): Contains a sequence of samples, where each sample could be a video frame or a contiguous section of audio data.

NOTE – A sample in ES contains media data, ByteData structure, pointer structure, container structure, or any mixture of these.

3.11.1 self-contained ES: ES that contains only media data, whose format is not defined in this Recommendation | International Standard.

NOTE – The self-contained ES could be stored in MDAT box co-located with the file format specified in this Recommendation | International Standard, or be stored in a separate file whose format is not specified by this Recommendation | International Standard.

3.11.2 composed ES: ES that can contain a mixture of ByteData, pointer and container structures, that is, its samples are composed with data from other elementary streams.

NOTE – A composed ES can either copy (using ByteData structure) or reference (using pointer) data from other ESs.

3.11.3 scalable composed ES: ES made up of samples that might not be decodable by themselves.

NOTE – A scalable composed ES might need to be combined with other scalable composed ESs to form a fully decodable codestream. Scalable composed ES is designed to support scalability, i.e., to make media data "thinable". For example, for a Motion JPEG 2000 codestream where each picture has three layers, it can be divided into 3 scalable composed ESs: the first one consists of all layer 0 data, the second one consists of all layer 1 data and the third one consists of all layer 2 data.

3.11.4 decodable composed ES: ES made up of samples that are decodable by themselves.

NOTE – It is designed for simple adaptation where the adaptor just needs to retrieve data pointed by pointer structure and remove the wrapper to form a fully scalable codestream. For example, for a Motion JPEG 2000 codestream where each picture has three layers, it can form 3 decodable composed ESs: the first one consists of layer 0 data, the second one consists of layer 0 and layer 1 data and the third one consists of layer 0, 1 and 2 data.

3.12 encryption: Reversible transformation of data by a cryptographic algorithm to produce ciphertext, i.e., to hide the information content of the data.

NOTE – An alternative term for an encryption algorithm is cipher.

3.13 hash function: Function which maps strings of bits to fixed-length strings of bits.

NOTE 1 – This function satisfies the following two properties:

- For a given output, it is computationally infeasible to find an input which maps to this output.
- For a given input, it is computationally infeasible to find a second input which maps to the same output.

NOTE 2 – Computational feasibility depends on the user's specific security requirements and environment.

3.14 integrity: Property of being able to safeguard the accuracy and the completeness of assets.

3.14.1 image data integrity: Property that data has not been altered or destroyed in an unauthorized manner.

3.14.2 image content integrity: Assurance the image content has not been modified by unauthorized parties in such a way that its perceptual meaning is changed.

NOTE – It allows the content-preserving operations to be performed on the image without triggering the integrity alarm.

3.15 JPSEC application: Software or hardware process that is capable of consuming JPSEC codestreams by interpreting the JPSEC syntax in order to provide the specified security services.

NOTE – A JPSEC application makes use of one or several JPSEC tools.

EXAMPLE – A JPSEC application would be able to read encrypted JPSEC codestreams, decrypt them when provided with the appropriate key and render the JPEG 2000 original clear-text image data.

3.16 JPSEC codestream: Sequence of bits resulting from coding and securing an image using JPEG 2000 coding and JPSEC security tools.

3.16.1 JPSEC creator: Entity who creates a JPSEC codestream from an image, a JPEG 2000 codestream, or another JPSEC codestream in order to provide some JPSEC services.

3.16.2 JPSEC consumer: Entity who receives a JPSEC codestream and renders a JPSEC service based on the codestream.

3.17 JPSEC service: Service that provides security for consumption of JPEG 2000 images. The service counters security attacks and makes use of one or several JPSEC tools.

3.18 JPSEC tool: Hardware or software process that uses security techniques to implement a security service.

3.18.1 JPSEC normative tool: JPSEC tool that uses predefined tool templates for decryption, authentication, or hashing specified by the normative part of this Recommendation | International Standard.

3.18.2 JPSEC non-normative tool: JPSEC tool that is specified by an identification number given by a user-defined application.

NOTE – Examples of JPSEC non-normative tools are describe in Annex B.

3.18.3 JPSEC RA tool: JPSEC non-normative tool that definition is registered by JPSEC registration authority (RA). (Deprecated)

3.18.4 JPSEC user-defined tool: JPSEC non-normative tool that is defined by a user-defined application.

3.19 JPSEC tool description: Description of the parameters used by the JPSEC tool.

NOTE – JPSEC tool description does not describe the algorithm or method used. A JPSEC tool description consists of two parts: the parameter list and its values. In the case of JPSEC normative tools, the parameter list is given by the standard. In the case of JPSEC non-normative tools, the parameter list can be given by a user-defined application. In both cases, the parameter values are specified in the SEC and INSEC marker segments.

3.20 key: Sequence of symbols that controls the operations of encipherment and decipherment.

3.20.1 asymmetric key pair: Pair of related keys where the private key defines the private transformation, and the public key defines the public transformation.

3.20.1.1 private key: Key of an entity's asymmetric key pair which is intended to remain private.

3.20.1.2 public key: Key of an entity's asymmetric key pair which can be made public.

3.21 key generation, key generating function: Function which takes as input a number of parameters, at least one of which is secret, and which gives as output keys appropriate for the intended algorithm and application.

NOTE – The function has the property that it is computationally infeasible to deduce the output without prior knowledge of the secret input.

3.22 key management: Generation, storage, distribution, deletion, archiving and application of keys in accordance with a security policy.

3.23 marker emulation: Cipher text resulting from the encryption process that contains a JPEG start code.

3.24 message authentication code (MAC): String of bits which is the output of a MAC algorithm.

3.25 non-repudiation: Binding of an entity to a transaction in which it participates, so that the transaction cannot later be repudiated (denied).

NOTE – That is, the receiver of a transaction is able to demonstrate to a neutral third party that the claimed sender did indeed send the transaction.

3.26 normal decoder: Standard decoder is a process to decode a codestream that is fully compliant with the normative part of coding standard. Its behaviour is not defined if it tries to decode a non-compliant codestream.

3.27 packet: Part of the JPEG 2000 Core coding system bit stream comprising a packet header and the compressed image data from one layer of the precinct of one resolution of one tile-component.

NOTE – This is different from the term "packet" used in data transmission through network.

3.28 pointer: Pointer structure is used to reference a data segment in another ES. It is contained inside a container structure.

3.29 protection: Process to secure content.

3.29.1 protection template: Template or list of parameter fields necessary for the operation of a protection method.

3.29.2 protection method: Method used to create or consume protected content such as encryption, decryption, authentication, and integrity checking.

3.30 security: All aspects related to defining, achieving, and maintaining confidentiality, integrity, availability, accountability, authenticity, and reliability.

NOTE – A product, system, or service is considered to be secure to the extent that its users can rely on that it functions (or will function) in the intended way. This is usually considered in the context of an assessment of actual or perceived threats.

3.31 signalling syntax: Specification of the format of the JPSEC codestream that contains all the required information for consuming secure JPEG 2000 images.

3.32 transcoding: Operation of taking an input compressed codestream and adapting or converting it to produce an output compressed codestream that has some desired property.

EXAMPLE – The output compressed codestream can represent an image with a lower spatial resolution or lower bit rate than the input compressed codestream.

3.32.1 secure transcoding: Operation of performing transcoding, or adaptation, of a protected input compressed content, without unprotecting the content.

NOTE – The term secure transcoding is used, as opposed to transcoding, to stress that the transcoding operation is performed without compromising security. Secure transcoding can also be referred to as performing transcoding in the encrypted domain.

3.33 watermark: Signal imperceptibly added to the cover-signal in order to convey hidden data.

3.33.1 watermarking: Process that imperceptibly inserts data representing some information into multimedia.

NOTE – The multimedia data in one of the following two ways:

- The lossy way which means the exact cover-signal will never be able to be recovered once the watermark is embedded.
- The lossless way which means the exact cover-signal could be recovered after watermark extraction.

3.34 4CC Code: 32-bit identifier, normally 4 printable characters.

NOTE – This 4CC code can be used to indicate the file type, the type of file format box, type of a file format track, type of a file format sample description and type of file format track reference.

4 Symbols and abbreviations

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

BAS	Byte Aligned Segment
FBAS	Field Byte Aligned Segment
G	Granularity
GL	Granularity Level
ICV	Integrity Check Value
INSEC	In-codestream security marker
IP	Intellectual Property related to technology
IPR	Intellectual Property Rights related to content

JPSEC	Secure JPEG 2000
KT	Key Template
LSB	Least Significant Bit
MAC	Message Authentication Code
MSB	Most Significant Bit
PD	Processing Domain
PKI	Public Key Infrastructure
PO	Processing Order
RBAS	Range Byte Aligned Segment
SEC	Security marker
T	Template
V	Values
VL	Value List
ZOI	Zone of Influence

5 JPSEC syntax

5.1 JPSEC framework overview

JPSEC defines a framework for the securing of JPEG 2000 coded data. The core of this Recommendation | International Standard is the specification of the syntax of the secure JPEG 2000 image, the *JPSEC codestream*. The syntax is targeted toward JPEG 2000 coded data and allows for protection of the entire codestream or of parts of the codestream. In all cases the protected data (i.e., the JPSEC codestream) shall follow the normative syntax defined in this Recommendation | International Standard.

To the JPSEC codestream are associated a number of *JPSEC security services* including confidentiality and authentication of origin and of content.

The *signalling syntax* specifies:

- what security services are associated with the image data;
- which *JPSEC tools* are required to deliver the corresponding services;
- how the JPSEC tools are applied;
- which parts of the image data are protected.

Case A: Image



Case B: JPEG 2000 codestream



Case C: JPSEC codestream



T.807(23)_F01

Figure 1 – Overview of the conceptual steps in JPSEC framework

The syntax of the JPSEC codestream is normative. The purpose is to allow JPSEC applications to consume JPSEC codestreams in an interoperable way (see Figure 1). The JPSEC consumer application interprets the JPSEC codestream, identifies and applies the signalled JPSEC tools, delivers the corresponding security services and then passes on the output JPEG 2000 codestream or image for subsequent processing, for example by an image viewer.

As shown in case C of Figure , the JPSEC codestream can be created from another JPSEC codestream. This can arise when multiple JPSEC tools are applied to the same content, but at different times or by different entities. When this occurs, the ordering in which the JPSEC tools are applied during the creation and consumption operations can be significant.

The signalling syntax identifies tools that are used by a JPSEC consumer. Tools are either defined by the normative part of the standard, or by private tools. The normatively defined tools support confidentiality (through encryption tools), and authentication of the source and of the content. They allow for the highest type of interoperability since independent implementations of the consuming process are able to process the same JPSEC codestream and render the corresponding services with the same behaviour.

The way in which the JPSEC codestream is created is out of scope of this Recommendation | International Standard. To be compliant, JPSEC creators shall generate JPSEC codestreams that include the appropriate JPSEC signalling. JPSEC codestreams can be created in a number of ways. For example, a JPSEC tool can be applied to image pixels or it can be applied on wavelet coefficients, or on quantized coefficients, or on packets.

A consumer can implement one or more JPSEC tools. For example, it could be capable of performing decryption using AES block cipher in ECB mode and signature verification using SHA-128 hash and an RSA public key. With these capabilities, it would be capable of performing the security services of confidentiality and authentication.

In the JPSEC framework, JPSEC tools are specified by templates, defined privately. JPSEC tools specified by the templates have unique processing behaviour and therefore do not require unique identification. Those tools are associated with a unique identification number provided by the common registry.

The guideline and use cases of JPSEC are described in Annex A, and the file format security is described in Annex D. The interoperability of JPEG 2000 family, e.g., JPX, JPIP and JPWL, are described in Annex C.

5.2 JPSEC security services

The objective in this clause is to list and to explain the functionalities that are included in the scope of this Recommendation | International Standard.

JPSEC tools are used to implement security functions. JPSEC is an open framework, which means that it is extensible in the future. Currently it focuses on the following aspects:

- Confidentiality via encryption and selective encryption

A JPSEC file can support a transformation of the (image and/or metadata) data (plaintext) into a form (cipher text) that conceals the data's original meaning. Selective encryption means that not the entire image and/or metadata but only parts of the image and/or metadata can be encrypted.

- Integrity verification

A JPSEC file can support means of detecting manipulations to the image and/or metadata and thereby verify their integrity. There are two classes of integrity verification:

- 1) Image data integrity verification where even only one bit of image data in error results in verification failure (i.e., the verification returns "no integrity"). This verification is also often referred to as fragile image (integrity) verification.
- 2) Image content integrity verification where even some incidental alteration of image data results in verification success as long as the alteration does not change image content from the human visual system point of view; in other words, the image perceptual meaning does not change. This verification is also often referred to as semi-fragile image (integrity) verification.

Those fragile or semi-fragile image integrity verifications might identify locations in the image data/image content where the integrity is put into question. Solutions can include:

- 1) Cryptographic methods such as Message Authentication Codes (MAC), digital signatures, cryptographic checksums or keyed hash.
- 2) Watermarking-based methods. This Recommendation | International Standard does not define normative template for watermarking technology, although it supports non-normative tools using watermarking technology.
- 3) Combination of the above two types of methods.

- Source authentication

A JPSEC file can support a verification of the identity of a user/party which generated the JPSEC file. This can comprise methods of e.g., digital signatures or message authentication code (MAC).

- Conditional access

A JPSEC file can support a mechanism and policy to grant or restrict access to image data or portions of those. This could allow for instance to view a low resolution (preview) of an image without being able to visualize a higher resolution.

- Registered Content identification

A JPSEC file can be registered at a Content Registration Authority. It can support a method of matching the (claimed) image data/image content to the registered image data/image content. For example, such a method could be: Reading a file identifier (Licence Plate) which was placed inside the metadata, checking the coherence between this Licence Plate and the information that has been uploaded when the registration process was done. The Licence Plate might contain enough information to be able to request information from the Content Registration Authority where the file was registered and verify that the file corresponds to the identifier.

- Secure scalable streaming and secure transcoding

A JPSEC file or sequence of packets can support methods such that the same or different nodes can perform streaming and transcoding without requiring decryption or unprotecting the content. An example is the case where protected JPEG 2000 content is streamed to a mid-network node or proxy that in turn transcodes the protected JPEG 2000 content in a manner that preserves end-to-end security.

5.3 Comments on design and implementation of secure JPSEC systems

This Recommendation | International Standard supports a rich and flexible set of security services. For example, the encryption primitives can be applied in a variety of different ways to achieve different goals, ranging from encryption of the entire JPEG 2000 codestream to selective encryption of only a small portion of the codestream. However, it is important to stress that significant care needs to be taken when implementing any security system, including one based on JPSEC.

It is strongly recommended that the designers of any security system carefully consider the recommended guidelines (Annex A) for the security primitives that are being employed. For most of the security primitives signalled using JPSEC, the associated ISO/IEC standards provide important guidance on their correct use. For example, for encryption using a block cipher and an associated block cipher mode (Table 29), guidelines for block cipher mode choice and operation are given in ISO/IEC 10116.

In addition, in many security applications authentication is the most important security service. Even when confidentiality is the targeted security service, it should be augmented by authentication to prevent various forms of attacks. Specifically, even in many imaging applications where the primary goal is confidentiality, it is recommended that authentication also be employed.

Key management is outside the scope of JPSEC. However, its criticality needs to be stressed. Of paramount importance in any cryptographic system is the management of the cryptographic keys that control the operations. If these keys are compromised, then the security of the whole system is compromised and in such a way that the compromise is not necessarily detected. It is therefore imperative that the keys are generated, distributed, stored and destroyed at a security level that is at least equal to that of the data that it is protecting. Furthermore, since the chances a key is compromised increase over time, it is also imperative that keys only be used for a fixed key lifetime. For more information on the use and management of cryptographic keys, see ISO/IEC 11770.

As with all security systems, the use of cryptographic operations are completely opaque to the user. That is, the user should not be able to discover any information about the cryptographic operations except for the output. For example, the user should not be able to access information about why a cryptographic operation failed to produce an output. Similarly, a user should not be able to find out any extra information even if he/she resorts to measuring "side channels" such as timing and/or power analysis. In short, the user should not be able to notice any difference in any of the applications outputs, regardless of what the application is currently doing, for if this is not the case the resulting leakage of information can potentially compromise the security of the system.

To summarize, it is strongly recommended that the designer of any security system, including one based on JPSEC, pay special attention to the details of the system design to ensure a secure system.

5.4 Byte aligned segment (BAS)

5.4.1 Byte aligned segment

In order to provide extensible signalling for classes and modes, this Recommendation | International Standard uses a variable length data structure called a "byte aligned segment" (BAS). Parameter fields with an extensible number of fields are represented with the Field BAS (FBAS) structure. Parameter values with large ranges are represented extensively with the Range BAS (RBAS) structure.

As illustrated in Figure 2, the BAS is composed of a sequence of one or more BAS bytes. The most significant bit (MSB) of each BAS byte indicates the existence of a following BAS byte. Specifically, if MSB = 1 then a subsequent BAS byte follows, while if MSB = 0 then a subsequent BAS byte does not exist, and the BAS structure is terminated. The remaining least significant bits of each BAS byte are concatenated to form a list of bits which are used in different ways for different BAS parameters. Often, they are used in conjunction with a parameter list that has a number of elements, and each BAS bit is set to 1 or 0 to flag information about its corresponding element. This flexible structure was chosen because of its extensibility for future evolutions of the standard, since it allows new parameters to be signalled in an extensible way.

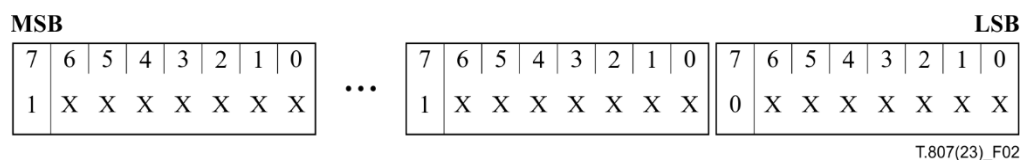


Figure 2 – Byte aligned segment (BAS) structure

5.4.2 Field BAS (FBAS)

A Field BAS (FBAS) is a type of BAS where the remaining bits of the BAS bytes are used to set fields to 1 or 0. An example of FBAS usage is the description class of the zone of influence (DCzoi), where multiple image descriptions such as tile index, resolution level, and colour component can be specified. If this is done, the three BAS bits corresponding to tile, resolution, and colour would be flagged to 1.

For example, to represent a Field BAS with 9 fields, f1 through f9, at most two BAS bytes would be needed. If the two bytes were byte "a" and byte "b", and the most significant bit of each byte were a0 and b0, then the FBAS would look like:

a0 a1 a2 a3 a4 a5 a6 a7 | b0 b1 b2 b3 b4 b5 b6 b7

a0 and b0 are the indicator bits. Field f1 through f7 are represented in bits a1 through a7, and field f8 is in bit b1 and field f9 is in bit b2. The remaining bits b3 through b7 are reserved and set to 0.

a0 f1 f2 f3 f4 f5 f6 f7 | b0 f8 f9 0 0 0 0

When used in a JPSEC stream, the FBAS in this example can be represented with one or two bytes, depending on the actual values of the field. This stems from the fact that the default value of the fields is 0. Thus, if fields f8 and f9 are not set (i.e., their value is 0), then the second byte of the BAS is not needed, and a0 is set to 0. On the other hand, if field 8 or field 9 is set, then two bytes are needed. In this case, a0 is set to 1 and b0 is set to 0.

Notice that the field bits are "left aligned". This allows us to add more fields over time in a compatible manner.

5.4.3 Range BAS (RBAS)

The Range BAS (RBAS) is used to extend the range or the number of bits used to represent a value. There are two types of RBAS, RBAS-8 and RBAS-16.

The RBAS-8 contains one or more RBAS bytes that contain the bits of the value. As in the FBAS, the first bit of each byte indicates whether another RBAS byte follows.

Unlike the FBAS, the RBAS is "right aligned". Thus, if a value has 9 significant bits v1 through v9, where v1 is the most significant bit, then it would be represented with two BAS bytes:

a0 a1 a2 a3 a4 a5 a6 a7 | b0 b1 b2 b3 b4 b5 b6 b7

as follows:

$$1\ 0\ 0\ 0\ 0\ 0\ v1\ v2\ |\ 0\ v3\ v4\ v5\ v6\ v7\ v8\ v9$$

If the value was small such that bits $v1$ and $v2$ were zero, then the two-byte representation above could be used with $v1$ and $v2$ set to zero, or a one-byte RBAS could be used as follows:

$$0\ v3\ v4\ v5\ v6\ v7\ v8\ v9$$

The RBAS-16 can be used to represent values that are typically more than 7 bits but less than 15. In this case, the first RBAS chunk is two bytes where the first bit is the indicator and then next 15 bits are value bits, then the remaining bytes extended one byte at a time using the typical BAS structure where the first bit of each byte is the indicator of following BAS bytes.

$$a0\ a1\ a2\ a3\ a4\ a5\ a6\ a7\ |\ b0\ b1\ b2\ b3\ b4\ b5\ b6\ b7\ |\ c0\ c1\ c2\ c3\ c4\ c5\ c6\ c7$$

If a parameter value had 22 bits, then it could be represented with the three-byte RBAS-16 structure shown below, where $a0$ and $c0$ are indicator bits to specify whether a BAS byte follows. Any remaining BAS bytes are traditional one-byte BAS segments.

$$a0\ v1\ v2\ v3\ v4\ v5\ v6\ v7\ |\ v8\ v9\ v10\ v11\ v12\ v13\ v14\ v15\ |\ c0\ v16\ v17\ v18\ v19\ v20\ v21\ v22$$

Thus, the indicator bits would be set as follows:

$$1\ v1\ v2\ v3\ v4\ v5\ v6\ v7\ |\ v8\ v9\ v10\ v11\ v12\ v13\ v14\ v15\ |\ 0\ v16\ v17\ v18\ v19\ v20\ v21\ v22$$

For both the RBAS-8 and RBAS-16, the value bits are also "right aligned".

Note that when writing JPSEC creators and consumers, it is important to pay attention to the big endian/little endian representations.

5.5 Main security marker (SEC)

5.5.1 Security marker segments

In this clause, a simple and flexible, yet powerful syntax for JPSEC signalling is presented. SEC marker segments are defined for this purpose and are located in the main header. The SEC marker segment syntax allows for the description of all required information for securing JPEG 2000 images. To do so, it makes references to JPSEC normative tools that are specified by the templates described in 5.8 or by JPSEC non-normative tools that can be defined *a priori* privately, and it makes provisions for handling parameters related to these tools.

A JPSEC codestream can be protected with one or more JPSEC tools. Each tool is a JPSEC normative tool or a JPSEC non-normative tool. The parameters for these tools are signalled in one or more SEC marker segments located in the main header of the codestream after the SIZ marker segment. When multiple SEC marker segments are used, they are concatenated and shall appear consecutively in the main header. In most cases, all the JPSEC parameters can be signalled in one SEC marker segment. However, in some cases the length of the signalling can exceed the maximum marker segment size. When this occurs, additional SEC marker segments can be used for signalling.

Figure 3 shows the syntax of the SEC marker segment. The segment is signalled by the SEC marker $0xFF65$. L_{SEC} is the length of the SEC marker segment, including the 2 bytes for L_{SEC} , but not the two bytes for the SEC marker itself. Z_{SEC} is a SEC marker segment index. Z_{SEC} shall be set to 0 for the first marker segment that appears in the codestream. P_{SEC} is a parameter field that describes the security parameters relevant to the entire codestream and only exists in the first SEC marker segment, i.e., if $Z_{SEC} = 0$. The syntax supports the use of several JPSEC tools that are signalled in one or more marker segments. If more than one JPSEC tool is used, then a JPSEC consumer shall process the tools in the order in which they appear in the codestream.

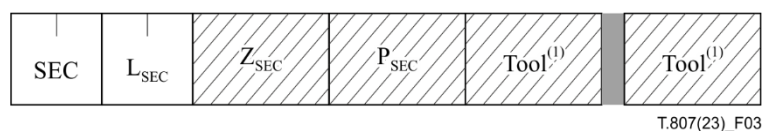


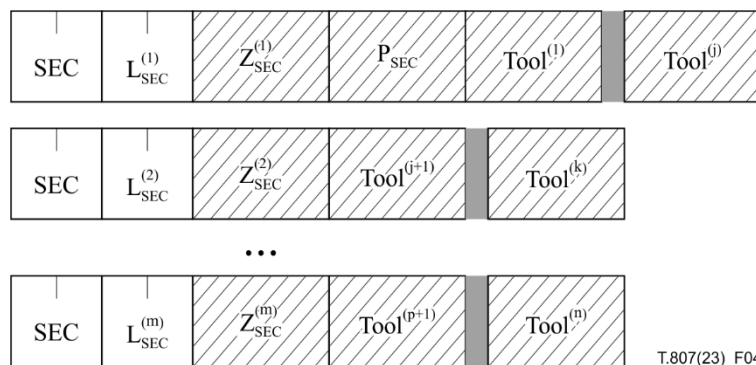
Figure 3 – Main security marker segment syntax

- SEC:** Marker code. Table 1 shows the sizes and values of the symbols and parameters for the main security marker segment.
- L_{SEC}:** Length of marker segment in bytes (including L_{SEC} itself, but excluding the marker).
- Z_{SEC}:** Index of this marker segment relative to all other SEC marker segments present in the current header. This field uses the RBAS structure.
- P_{SEC}:** Parameter field for codestream security parameters. This field is only present in the first SEC marker segment, i.e., when Z_{SEC} is 0.
- Tool⁽ⁱ⁾:** Parameters for JPSEC tool *i*. If multiple JPSEC tools are signalled, then a JPSEC consumer shall process each tool in the order of appearance in the JPSEC codestream.

Table 1 – Main security parameter values

Parameter	Size (bits)	Values
SEC	16	0xFF65
L _{SEC}	16	2 ... (2 ¹⁶ – 1)
Z _{SEC}	8 + 8 * n (RBAS)	0 ... 2 ^{7+7*n}
P _{SEC}	0, if Z _{SEC} > 0 Variable, otherwise	If Z _{SEC} = 0, see Table 2
Tool ⁽ⁱ⁾	Variable	See 5.6.2 and 5.6.3

Figure 4 shows the syntax of the security parameters in the main header when multiple SEC marker segments are used. In this case, the JPSEC tool parameters are in different SEC marker segments. Each marker segment begins with the SEC marker, 0xFF65, and is followed by the length and index of the marker segment. The index of the first marker segment shall be set to 0 and shall increase by one for each marker segment in the order it appears. Only the first marker segment contains the security parameters for the codestream, P_{SEC}. All the marker segments contain the parameters for one or more JPSEC tools.

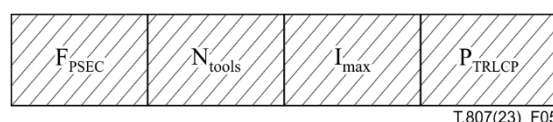


T.807(23)_F04

Figure 4 – Main security marker syntax when multiple marker segments are used

If needed, a JPSEC tool description can span multiple SEC marker segments, e.g., this can occur if the length exceeds the maximum SEC marker size. Since the length of the tool description is completely specified, the JPSEC creator simply splits the tool across SEC marker segments. The decoder should then concatenate all segments, minus the SEC marker and the L_{SEC} and Z_{SEC} values, and then interpret the tools accordingly.

P_{SEC} is a parameter field that describes security parameters for the entire codestream as opposed to for a particular tool. This is used to indicate events such as JPEG 2000 Core coding system compliance or the use of INSEC markers. The P_{SEC} parameters are shown in Figure 5.



T.807(23)_F05

Figure 5 – Codestream security parameters (P_{SEC}) syntax

F_{PSEC}: Flag to indicate if INSEC marker segment is used, if multiple SEC marker segments are used, if the original JPEG 2000 Core coding system codestream data was modified, and if TRLCP tag usage is defined. FBAS structure is used by this field.

N_{tools}: Number of JPSEC tools used in the codestream. This field uses the RBAS structure.

I_{max}: Maximum tool instance index value used in the codestream. This field uses the RBAS structure.

P_{TRLCP}: Parameter field to define the format of TRLCP tag. This field exists if $F_{TRLCP} = 1$.

Table 2 – Codestream security parameters (P_{SEC}) in first SEC marker segment

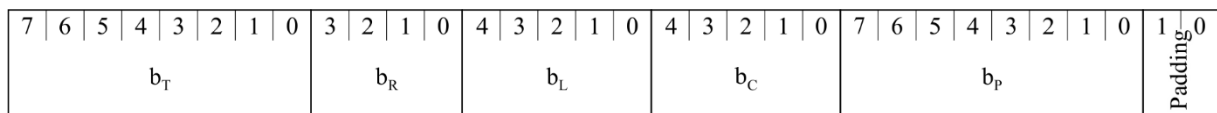
Parameter	Size (bits)	Values
F _{PSEC}	Variable (FBAS)	See Table 3
N _{tools}	8 + n * 8 (RBAS)	1 ... 2 ^{7+7*n}
I _{max}	8 + n * 8 (RBAS)	0 ... 2 ^{7+7*n}
P _{TRLCP}	0, if F _{TRLCP} = 0 32, if F _{TRLCP} = 1	See Table 4

F_{PSEC} is an FBAS structure used to indicate a number of parameter flags about the JPSEC codestream. The fields represented by F_{PSEC} are shown in Table 3. F_{INSEC} shall be set to 1 if INSEC markers are used in the JPSEC codestream. F_{multiSEC} shall be set to 1 if multiple SEC marker segments are used in the JPSEC codestream. F_{mod} shall be set to 1 if the original JPEG 2000 data was modified in the JPSEC codestream. Note that if INSEC markers are used, then the original JPEG 2000 data is modified and thus F_{INSEC} and F_{mod} shall be set to 1. F_{TRLCP} shall be set to 1 if the TRLCP tag usage is defined in P_{SEC}. If it is defined, then the TRLCP tag descriptor, P_{TRLCP}, is specified in the P_{SEC} parameter field. The TRLCP tag usage shall be specified if any tool within the JPSEC codestream uses TRLCP tags.

Table 3 – Semantics for F_{PSEC} values (FBAS)

BAS field	BAS bit number	Value (bits)	Semantics
F _{INSEC}	1	0	INSEC is not used
		1	INSEC is used
F _{multiSEC}	2	0	One SEC marker segment is used
		1	Multiple SEC marker segments are used
F _{mod}	3	1	Original JPEG 2000 data was modified
		0	Otherwise
F _{TRLCP}	4	0	TRLCP tag usage is not defined in P _{SEC}
		1	TRLCP tag usage is defined in P _{SEC}

JPSEC defines a structure called a TRLCP tag that can be used to uniquely identify a JPEG 2000 packet. A JPEG 2000 packet can be uniquely specified by its tile index, resolution level index, layer index, component index, and precinct index. A TRLCP tag is defined as a data unit with a fixed number of bits used to specify each of these index values. The number of bits for each index is set in P_{SEC}. P_{TRLCP} is a parameter field that describes the format of the TRLCP tag as it shall be used in the JPSEC tools. This field only exists if F_{TRLCP} = 1. P_{TRLCP} consists of the following variables in Figure 6.



T.807(23)_F06

Figure 6 – TRLCP tag descriptor (P_{TRLCP}) syntax

b_T: Number of bits to represent tile index is b_T + 1 in TRLCP tag.

b_R: Number of bits to represent resolution level index is b_R + 1 in TRLCP tag.

b_L: Number of bits to represent layer index is b_L + 1 in TRLCP tag.

b_C: Number of bits to represent component index is b_C + 1 in TRLCP tag.

b_P: Number of bits to represent precinct index is b_P + 1 in TRLCP tag.

Table 4 – Parameter field for TRLCF tag descriptor (P_{TRLCF})

Parameter	Size (bits)	Values
b_T	8	0 ... $(2^8 - 1)$
b_R	4	0 ... 15
b_L	5	0 ... 31
b_C	5	0 ... 31
b_P	8	0 ... $(2^8 - 1)$
Padding	2	0

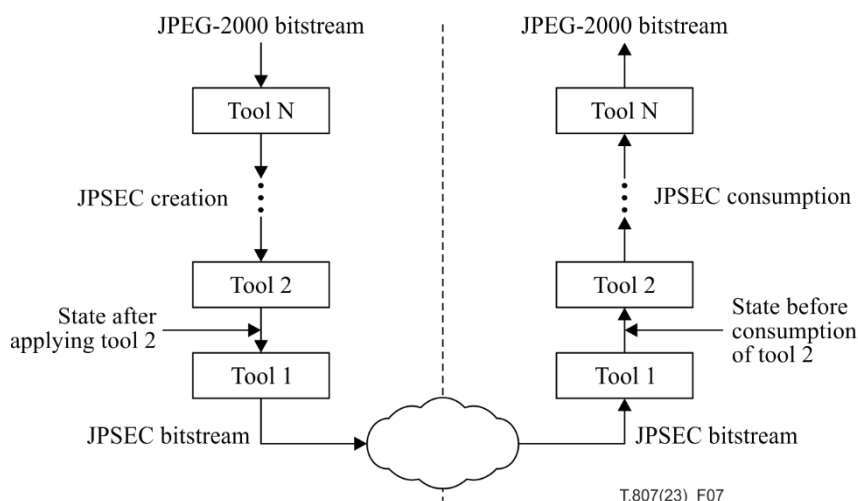
The size of each resulting TRLCF tag is the smallest integer byte size that contains all the bits. The format of the TRLCF tag contains the bits for the tile index, the resolution level index, the layer index, the component index, and the precinct index in that order. If extra bits are needed to fill the integer byte size requirement, then the TRLCF tag will be placed in the least significant bits possible, and the extra bits are set to 0. Note that these extra bits will be the MSBs of the TRLCF tag if they exist.

5.5.2 Application of multiple JPSEC tools

In many applications multiple JPSEC tools apply to a single JPEG 2000 codestream. For example, both encryption and authentication can be applied to protect a JPEG 2000 image. The general situation of applying multiple JPSEC tools is illustrated in Figures 3, 4 and 7, where N tools are applied. The JPSEC consumer will read the N tools in order of placement in the SEC marker segment shown in Figure 3 or Figure 4, and apply them in that same order to perform the JPSEC consumption of the JPSEC codestream. Note that while the JPSEC consumer applies the JPSEC tools in order 1, 2, ..., N , as read from the SEC marker segment, during creation of the JPSEC codestream these JPSEC tools were applied in the reverse order, i.e., $N, N-1, \dots, 2, 1$, as illustrated in Figure 7. Note that the numbering of the tools in the figure was chosen to highlight that the JPSEC consumer applies the JPSEC tools in the reverse order from the JPSEC creator. However, any numbering of the JPSEC tools is acceptable, as long as each JPSEC tool in a JPSEC codestream is given a unique number for identification purposes.

Generally speaking, JPSEC tools are created and consumed in reverse order of one another. For example, if the JPSEC creator applies N JPSEC tools, then the JPSEC consumer typically applies the same N JPSEC tools but in the reverse order. Correct JPSEC consumption of multiple JPSEC tools can be guaranteed by sequential consumption of the N tools in the correct order and by requiring any intermediate stage at the consumer to match the corresponding state at the creator. For example, in Figure 7, the state at the consumer after JPSEC consumption of tool 1 should be equal to the state after applying tool 2 during the JPSEC creation process. As a specific example of the state, the byte ranges should be consistent, therefore any bytes added when applying tool 1 should be removed when removing tool 1 at the JPSEC consumer.

In certain applications, it can be desirable for a JPSEC consumer to consume the multiple JPSEC tools in a different manner than described above. For example, the JPSEC consumer can choose to consume the multiple tools in a different order, or to skip certain tools in the consumption. Furthermore, the JPSEC consumer can prefer to apply certain JPSEC tools, but not remove them, e.g., to check a digital signature but not remove it. Careful consideration should be given in these cases to ensure that the out-of-order or skipped processing does not lead to incorrect or unintended consequences. This behaviour is not recommended unless the JPSEC application is fully aware of the potential ramifications.

**Figure 7 – Use of multiple JPSEC tools**

5.6 JPSEC tools

5.6.1 JPSEC tool syntax

As mentioned earlier, there are two types of JPSEC tools. JPSEC normative tools are specified with the protection method templates described in 5.8, and are also known as JPSEC normative tools. JPSEC non-normative tools are specified by a particular JPSEC application based on their ID number, and are respectively referred to as JPSEC user-defined tools. The syntax for JPSEC normative tools are discussed in 5.6.2. The syntax for JPSEC non-normative tools are discussed in 5.6.3.

The syntax for JPSEC tools is shown in Figure 8. The JPSEC tool syntax has three main parts that describe:

- 1) what tool is applied with its identification;
- 2) where the tool is applied with a zone of influence structure; and
- 3) how the tool is applied with a more detailed parameter field.

For example, using this syntax, a JPSEC tool syntax could specify that a decryption tool should be used (what) on the lowest resolution component located in a particular byte range (where) using AES decryption in CBC mode with a specified set of initialization vectors and keys (how).

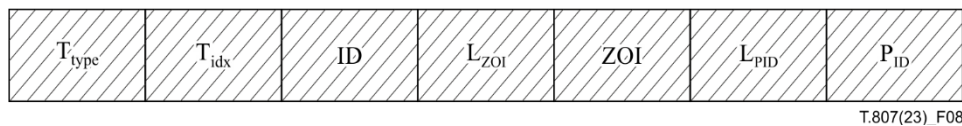


Figure 8 – JPSEC tool syntax (Tool⁽ⁱ⁾)

T_{type}: Tool type. The value of 0 for the first BAS bit indicates a JPSEC normative tool. The value of 1 for the first BAS bit indicates a JPSEC non-normative tool. This field uses the FBAS structure.

T_{idx}: Tool instance index (can be used as a unique identifier). This field uses the RBAS structure.

ID: Identification value for JPSEC tool *i*. For JPSEC normative tools, the ID = ID_T is 8 bits and specifies the template type. For JPSEC non-normative tools, the ID = ID_{RA} is defined by Figure 10 and Table 8.

L_{ZOI}: Length of ZOI in Bytes (excluding L_{ZOI}). This field uses the RBAS structure.

ZOI: Zone of influence for JPSEC tool *i*.

L_{PID}: Length of P_{ID} in Bytes (excluding L_{PID}). This field uses the RBAS structure.

P_{ID}: Parameters for JPSEC tool *i*.

Table 5 – JPSEC tool parameter values

Parameter	Size (bits)	Values
T _{type}	8 + 8 * n (FBAS)	x0xx xxxx b, x1xx xxxx b
T _{idx}	8 + 8 * n (RBAS)	0 ... (2 ^{7+7*n} - 2) (2 ^{7+7*n} - 1), reserved
ID	8, if T _{type} = 0 Variable, if T _{type} = 1	See Table 6 See Figure 10 and Table 8
L _{ZOI}	16 + 8 * n (RBAS)	0 ... 2 ^{15+7*n}
ZOI	Variable	See 5.7
L _{PID}	16 + 8 * n (RBAS)	0 ... 2 ^{15+7*n}
P _{ID}	Variable	Table 7, if T _{type} = 0

Each JPSEC tool has the following syntax. The initial one-byte identifies if the tool is a JPSEC normative tool or JPSEC non-normative tool and assigns an instance identifier to the tool. This is followed by the tool identifier **ID**. This is followed by L_{ZOI}, which indicates the length of the subsequent zone of influence field ZOI, and the zone of influence itself, which describes where in the data stream the JPSEC tool is applied. This is followed by L_{PID}, which indicates the length of the following parameter field P_{ID}, which is a field to transmit one or more parameters for the JPSEC tool.

The first byte of the tool uses a one-byte FBAS structure whose first BAS bit represents the tool type, T_{type}, where 0 specifies a JPSEC normative tool and 1 specifies a JPSEC non-normative tool. This is followed by the instance index, T_{idx}, which is represented using the RBAS structure. The instance index shall be a unique identifier of the tool within the

codestream, and thus shall not be repeated by any other tool in the codestream, even if it is in a different SEC marker segment. The instance index is especially critical (and necessary) when INSEC markers are used, because each INSEC marker segment contains the instance index of the tool to which it applies. It is recommended that the first tool applied at a JPSEC creator has an instance index of 1, and that each additional tool be indexed sequentially as it is applied at the protector.

In addition, each JPSEC tool has an ID number which is 8 bits for JPSEC normative tools and 32 bits for JPSEC non-normative tools. For JPSEC normative tools, the ID number describes which protection method template is used, i.e., it specifies the decryption template, authentication template, or hash template. For JPSEC non-normative tools, the first bit indicates whether it is a JPSEC user-defined tool. In either case, the ID number indicates the particular tool. However, a JPSEC application that uses user-defined ID numbers runs the risk of choosing an ID number that is also used by another JPSEC application, so this should be used cautiously.

When each JPSEC tool is applied at the JPSEC creator, the P_{SEC} parameter field shown in Table 2 shall be updated. For example, the P_{SEC} parameter field contains the I_{max} parameter that specifies the maximum instance index used for the tools in the JPSEC codestream. When a new tool is applied, it shall be assigned a unique instance index. A JPSEC protector can refer to the I_{max} parameter given in the P_{SEC} parameter field to determine the instance index to assign to a JPSEC tool, for example, it can choose a value that is one greater than the current I_{max} value, and it should then increment the value of I_{max} by 1 accordingly.

5.6.2 JPSEC normative tool

The JPSEC normative tool uses the JPSEC tool syntax described in 5.6.1 and shown in Figure 8, where the tool type $T_{type} = 0$ and the size of the ID is 8 bits. JPSEC normative tools are based on the protection method templates described in 5.8. There are three types of protection method templates; the type used by the tool is specified by the tool identifier $ID = ID_T$ using the values shown in Table 6.

Table 6 – JPSEC normative tool Template ID values (ID_T)

Values	Protection method template
0	Reserved
1	Decryption template
2	Authentication template
3	Hash template
4	NULL tool
All other values are reserved for ITU-T ISO use	

In the case of JPSEC normative tools, the parameter field P_{ID} has the structure shown in Figure 9. P_{ID} consists of four main fields: the protection method template T , its processing domain PD , its granularity G , and its value list V . The syntax for each of these fields is given in 5.8, 5.9, 5.10 and 5.11, respectively. Together, these fields describe how the tool is applied. The protection method template T describes the particular protection method for the decryption template, authentication template, or hash template specified by the normative tool ID. It can also specify the NULL tool, in which case no template is used, but other functionalities can still be used. For example, the zone of influence can be specified to represent image regions and their corresponding byte ranges. The processing domain PD describes the domain in which the protection method is applied. The granularity G describes the granularity with which the protection method is applied. The value list V contains a list of values that can be needed by each protection method with finer granularity. For the decryption template, the value list can be used to specify a finer grain set of initialization values that are to be used. For the authentication template, the value list contains a set of MAC values or digital signatures. For the hash template, the value list contains a set of hash values. In all cases, the value list contains a granularity of values specified by the granularity field G .

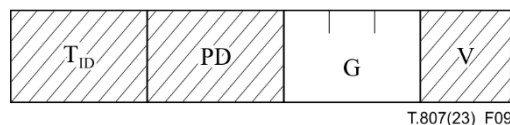


Figure 9 – Parameters (P_{ID}) syntax for JPSEC normative tools ($t = 0$)

T_{ID} : Template parameters for JPSEC normative tool with template identifier ID_T .

PD : Processing domain for JPSEC normative tool.

G : Granularity for JPSEC normative tool.

V: Value list for JPSEC normative tool, e.g., initialization vectors, MAC values, digital signatures, or hash values depending on template ID.

Note that the template parameters depend on the template ID. However, the processing domain, granularity, and value list are independent of the template ID.

Table 7 – JPSEC normative tool parameter values

Parameter	Size (bits)	Values
T_{ID}	0, if $ID_T = 4$ Variable, otherwise	N/A See 5.8
PD	Variable	See 5.9
G	24	See 5.10
V	Variable	See 5.11

5.6.3 JPSEC non-normative tool

The mechanism to register JPSEC user defined tool to JPSEC RA tool described in this clause is deprecated because the registration authority defined in the previous edition, ISO/IEC 15444-8:2007, is cancelled.

In certain cases, it can be useful for a JPSEC application to have the ability to apply a tool that extends beyond the JPSEC normative tools. This capability is supported by using a JPSEC user defined tool which was called as JPSEC non-normative tool. This enables one to use many elements of JPSEC normative tools, including the ZOI and the JPSEC templates, but adds the flexibility of using the parameters in a different manner associated with a tool ID value.

The JPSEC non-normative tool used the JPSEC tool syntax described in 5.6.1 and shown in Figure 8, where the tool type $T_{type} = 1$ and the identifier ID_{RA} consists of a name space and an ID number, as defined by Figure 10 and Table 8.

There were two classes of JPSEC non-normative tools:

- 1) JPSEC registration authority tools (Deprecated): JPSEC non-normative tools whose signalling was specified with a registration authority.
- 2) JPSEC user-defined tools: JPSEC non-normative tools whose signalling is specified by a JPSEC application.

These two classes of JPSEC non-normative tools were signalled using the 32-bit $ID_{RA,id}$ identifier shown in Table 9, where the identifiers whose first bit is a 0 were defined by a registration authority, and those whose first bit is a 1 are defined by a particular JPSEC application.

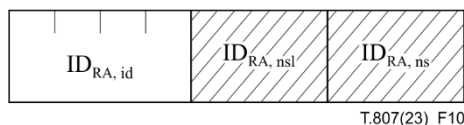


Figure 10 – ID_{RA} syntax

$ID_{RA,id}$: Tool identifier for user-defined tool

$ID_{RA,nsl}$: Length of the field $ID_{RA,ns}$ in bytes. This field uses RBAS.

$ID_{RA,ns}$: A string containing the name space of the specified user-defined tool

Table 8 – Parameters values in ID_{RA} syntax

Parameter	Size (bits)	Values
$ID_{RA,id}$	32	See Table 9
$ID_{RA,nsl}$	$8 + 8 * n$ (RBAS)	$0 \dots (2^{7+7*n} - 1)$
$ID_{RA,ns}$	Variable	A string containing namespace

Table 9 – ID values for JPSEC non-normative tools (ID_{RA,id})

ID _{RA,id}	Meaning
0x00 00 00 00 ... 0x7F FF FF FF	Previously managed by JPSEC RA (Registration Authority) tool. Values were to be managed by JPSEC registration authority (Deprecated). Reserved for ITU-T ISO use.
0x80 00 00 00 ... 0xEF FF FF FF	JPSEC user-defined tool. Values can be defined by a particular JPSEC application.
0xF0 00 00 00 ... 0xFF FF FF FF	Reserved for ITU-T ISO use.

For RA tools, the field ID_{RA,ns} was the name space of the Registration Authority (RA) with which this tool is registered. As each RA had a unique name space, the ID_{RA,id} and ID_{RA,ns} were used together to identify an RA tool. For user-defined tools, the field ID_{RA,ns} is chosen by the developers. In order to limit the risk of ID collisions, it is recommended that the developers seek uniqueness when choosing their name space, for example, by choosing the domain name of their organization or company. However, note that for user-defined tools, there is no way to guarantee that uniqueness of the name space, so ID collision can occur and should be carefully considered when using user-defined tools.

The P_{ID} field is used to transmit one or more parameters for the JPSEC non-normative tool *i*. The format of the P_{ID} field is not fully given in the scope of JPSEC. If a user-defined is used, then only the length of this field is specified, and it is up to the users to appropriately use this field.

However, JPSEC does allow the syntactical structures defined for JPSEC normative tools to be used in the P_{ID} field for JPSEC non-normative tools. For example, a JPSEC non-normative tool can use the protection method templates, processing domain, granularity, and value list fields described in 5.8, 5.9, 5.10 and 5.11, respectively.

This syntax is very flexible and can accommodate a wide variety of security techniques, such as image data integrity, access control and rights protection methods. Hence, it offers a rich set of functionalities while being simple and concise.

5.7 Zone of Influence (ZOI) syntax

5.7.1 Introduction

The Zone of Influence (ZOI) can be used to describe the coverage area of a JPSEC tool. The data within the coverage area (specified by the ZOI) is referred to as the influenced data. JPSEC normative tools shall use the ZOI to describe their coverage area. JPSEC non-normative tools can use the ZOI to describe their coverage area or they can use an alternative method. If an alternative method is used, then the ZOI length is 0, i.e., it does not exist.

The Zone of Influence (ZOI) describes the coverage area of each JPSEC tool. This coverage area can be described by image-related parameters, e.g., by resolution or image area; or by non-image related parameters, e.g., by codestream segments or packet indices. In cases where image related parameters and non-image related parameters are used together, the ZOI describes the correspondence between these areas. For example, the ZOI can be used to indicate that the resolutions and image area specified by the image related parameters correspond to the codestream segments specified by the non-image related parameters. This allows the ZOI to be used as metadata that signals where certain parts of the image are located in the JPSEC codestream.

Figure 11 illustrates the conceptual structure of the ZOI. The ZOI contains one or more zones. When multiple zones are used within a single ZOI, the ZOI is defined by their union. This indicates that the JPSEC tool should be applied to all the zones. Each zone in a ZOI is described by three fundamental units: description class, parameter mode and parameter items (values). This Recommendation | International Standard defines two description classes: image related description class and non-image related description class. These parameters can be specified using a number of modes, for example, by a single value, multiple listed values, or by a range. The parameter values or items are then listed in accordance with the mode.

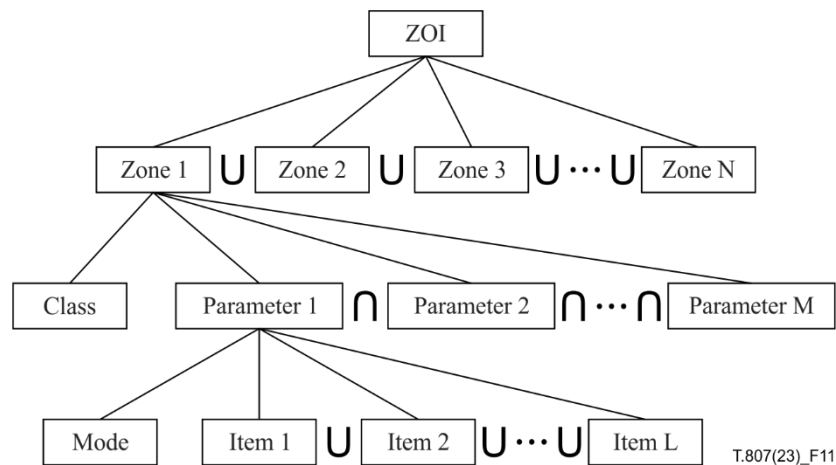


Figure 11 – Zone of Influence conceptual structure

5.7.2 ZOI syntax

Figure 12 shows the ZOI syntax. The ZOI can contain one or more zones. It can also be empty, in which case NZ_{zoi} shall be 0. When this occurs, the influence of the tool is specified by other means, such as by the INSEC marker or by parameters defined by a JPSEC non-normative protection tool.

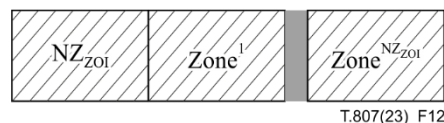


Figure 12 – ZOI syntax

NZ_{zoi} : Number of Zones. This field uses the RBAS structure.

$Zone^k$: Zone. Its structure is specified in 5.7.3.

Table 10 – Zone of influence field (ZOI) parameter values

Parameter	Size (bits)	Values
NZ_{zoi}	$8 + 8 * n$ (RBAS)	$0 \dots (2^{7+7*n} - 2)$ $(2^{7+7*n} - 2)$, reserved
$Zone^k$	Variable	See 5.7.3

5.7.3 Zone syntax

The Zone contains a zone description class field indicator followed by parameters of that class. The zone description class uses the FBAS structure. As shown in Figure 13, the second most significant bit in each byte, labelled "x", flags the use of a specific description class. This Recommendation | International Standard defines two description classes: image related description class and a non-image related description class (see Table 12). Tables 13 and 14 define the field indicator numbers for the image related description class and non-image related description class, respectively. The concatenation of the six bits labelled "y", in each byte that follow the description class flag, indicates the use of a specific description within a given description class. A bit value of "1" at a bit number in each class indicates that the corresponding parameter field exists. The number of parameters shall be the same as the number of zone description class field indicators set to '1', and shall appear in order which the class field indicator is signalled. The zone description class has variable number of bytes; when the MSB equals 1, then another zone description class byte follows. The MSB of the last description class byte equals 0. If both the image related and non-image related description classes are used, then the image related description class bytes shall precede the non-image related description class byte. When a number of items are represented using this structure, the first item in the list shall correspond to the most significant available bit of the first byte.

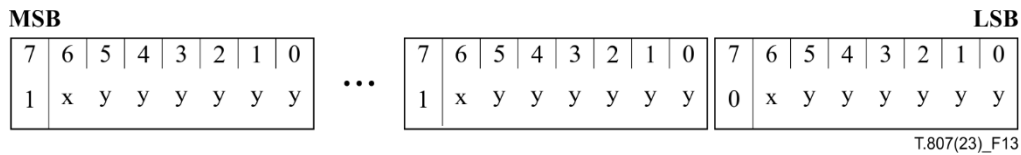


Figure 13 – Zone description class structure (DCzoi)

Figure 14 shows the Zone syntax.

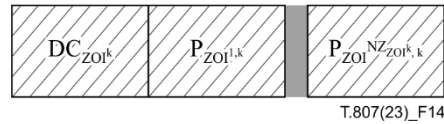


Figure 14 – Zone syntax consists of a description class and one or more parameter sets

DCzoi^k: kth Zone description class. This field uses the FBAS structure.

Pzoi^{i,k}: The Zone parameters according to the specified Zone description class (DCzoi^k). See 5.7.6.

DCzoi^k specifies the number n of zone description class fields that exist, based on the number of bits that are set to one. For each zone description class field, there exists one Pzoi^{i,k} zone parameter field. These fields appear sequentially in the same order that the flags appear in DCzoi^k.ZA

Table 11 – Zone parameter values

Parameter	Size (bits)	Values
DCzoi ^k	Variable (FBAS)	Varies according to the value set in Table 12
Pzoi ^{i,k}	Variable	See 5.7.6 for the syntax of this field

Table 12 – Description class indicator value

Value	Description class
0	Image related description class. The following bit numbers are defined in Table 13
1	Non-image related description class. The following bit numbers are defined in Table 14

Table 13 – Image related description class

Bit number	Semantics
1	Image region
2	Tile(s) as defined in the JPEG 2000 Core coding system
3	Resolution level(s) as defined in the JPEG 2000 Core coding system
4	Layer(s) as defined in the JPEG 2000 Core coding system
5	Component(s) as defined in the JPEG 2000 Core coding system
6	Precinct(s) as defined in the JPEG 2000 Core coding system
7	TRLCP (Tile-Resolution-Layer-Component-Precinct) tag(s)
8	Packet(s) as defined in the JPEG 2000 Core coding system
9	Sub-band(s) as defined in the JPEG 2000 Core coding system
10	Code-block(s) as defined in the JPEG 2000 Core coding system
11	ROI(s)
12	Bit-rate
13	User-defined. The details shall be specified by other means. (e.g., JPSEC ID)
	All other values are reserved

Table 14 – Non-image related description class

Bit number	Semantics
1	Packet(s) as defined in the JPEG 2000 Core coding system
2	(Padded) Byte range(s) (beginning at first byte after the first SOD marker)
3	(Padded) Byte range(s) (beginning at first byte after the first SEC marker)
4	Unpadded byte range(s) when padding is used
5	TRLCP (Tile-Resolution-Layer-Component-Precinct) tag(s)
6	Distortion value(s)
7	Relative importance(s)
8	User-defined. The details shall be specified by other means. (e.g., JPSEC ID)
	All other values are reserved

Packet indices are numbered sequentially within a tile, and therefore they are not necessarily unique across tiles. Furthermore, packet indices within a tile can roll over when their maximum value of 65535 is exceeded. For this reason, packet indexing is described in more detail. When the packet indices within a tile do not exceed 65535 packets, then the packet index described in Table 13 is defined by the packet index given by the SOP Nsop parameter as defined in Table A.40 in the JPEG 2000 Core coding system standard. Note that when the maximum value does not exceed 65536, a single JPEG 2000 packet can be specified uniquely with a tile index and a packet index. When the packet indices exceed 65535 packets, then the JPEG 2000 Core coding system packet index is defined to roll over to 0. In this case, the packet index does not uniquely identify a packet and shall not be used. In this case, it is recommended to use the TRLCP tag instead. Please note that security services which require unique packet indices are vulnerable if the packet index rolls over and repeats.

When the TRLCP tag is used, its format shall be defined in the P_{SEC} parameter field shown in Table 2. Specifically, the TRLCP tag format is specified by the P_{TRLCP} parameter field in Table 4. This defines the size of TRLCP tags in the ZOI.

The non-image related description class can also have multiple fields set simultaneously. When this occurs, the modes for the various parameter fields shall have the same number of items (one exception to this rule is described below), and these items shall correspond with one another in a one-to-one manner in the same order. For example, if the zone uses byte ranges and packet ranges, each should have the same number of range items where the first byte range corresponds to the first packet range, and so on.

There is one exception to the above rule on requiring the same number of items for each field. This occurs when one of the fields f_1 contains 1 item which specifies a range of items (as described by the range mode in 5.7.6) where this range contains N elements and when another field f_2 is specified by a list of N items. In this case, the field f_1 , which contains only 1 item (the range) is interpreted as a list of N items. These N items specified by the range in f_1 shall correspond one-to-one with the N items listed in f_2 . Therefore, a range of items can be associated to either a single item or to multiple items (one for each item in the range).

The bytes are indexed either from the first byte after the first SOD marker or from the first byte after the first SEC marker. In either case, this first byte is labelled as byte 0.

The distortion fields (both the distortion and relative importance fields) provide the capability to signal the importance of areas specified by the ZOI. The distortion parameter specifies the distortion-reducing contribution of the specified data segment, be it for a set of packets or a byte range or for the specified image-related area. The distortion is expressed in terms of the total squared error, using either a one-byte or a two-byte description signalled in the M_{zoi} . The relative distortion parameter can be used to specify the relative importance of specified data segments, using either one-byte, two-byte, or four-byte values signalled in the M_{zoi} . Additional details and the formats of these fields is described in 5.7.3.2.

The TRLCP tag specifies a protected packet's tile, resolution, layer, component, and precinct in the codestream. This tag is used in the ZOI to specify these parameters because this information can be difficult to infer in a protected codestream.

Note that when only image-related descriptions are used, the field can be terminated. Thus, one does not need to represent non-image related descriptions if they are not used.

5.7.3.1 Byte range fields

The non-image-related description class allows the ZOI to be described in byte ranges. In general, the 2nd and 3rd elements of Table 14 should be used to represent the byte ranges for most tools such as authentication and encryption/decryption without padding. However, some protection methods, such as encryption/decryption with padding, change the length of the data. When this occurs, both the padded byte range and the unpadded or original byte range are specified. In this case, the padded byte range is specified by the 2nd or 3rd element of Table 14 according to the needs of the protection tool. (Note that these two elements cannot be used together.) In addition, the unpadded byte range is

specified by the 4th element of Table 14. The unpadding byte range should be specified with the same description mode as the padded byte range and have same number of items. These items should correspond to each other in a one-to-one manner in the same order.

5.7.3.2 Distortion field and relative importance field

The distortion and relative importance fields provide the capability to signal the importance of areas specified by the ZOI.

The distortion field is used to associate a distortion with an area specified by the ZOI. The distortion value specifies the total squared error (or sum of squared error) distortion that would result if the associated area is not available for decoding. Total squared error distortion is a basic distortion metric used in image and video processing, and it is used to derive the common mean-squared-error (MSE) distortion and peak-signal-to-noise (PSNR) ratio. The distortion field is expressed using a one-byte or a two-byte description, where these one-byte and two-byte descriptions are described below, and the choice of one-byte or two-byte description is signalled by the Mzoi parameter value which specifies the length of this field. The relative importance field can be used to describe the relative importance among different areas specified by associated ZOIs, without necessarily being tied to a specific distortion metric. The length of the relative importance field is also signalled by the Mzoi. These fields are discussed in more detail in the following.

5.7.3.2.1 One-byte distortion field

The total squared error distortion is expressed using a one-byte distortion field with a pseudo floating-point type representation. The 8 bits available in the distortion field are allocated as shown in Figure 15 and Table 15 to provide an appropriate trade-off between accuracy and dynamic range. Note that a sign bit is unnecessary since distortion is non-negative. To cover a sufficient dynamic range, base 16 is used and 4 bits are used for the exponent (exp). The mantissa (m) is expressed using 4 bits. Therefore, the total distortion value D is given by:

$$D = m \times 16^{\text{exp}}$$

where m has a value in the range $0 \leq m \leq 15$ and exp has a value in the range $0 \leq \text{exp} \leq 15$. A distortion value of zero is represented by $m = 0$ and $\text{exp} = 0$, that is by the distortion field being zero. By allocating 4 bits for the mantissa m the accuracy is within $\frac{1}{2} \times (1/2^4) = 1/32$ or about 3%. With 4 bits for the exponent and using base-16 the dynamic range is from 0 to max, where max is given by $m = 15$ and $\text{exp} = 15$ which corresponds to a distortion of $15 \times 16^{15} = 1.7 \times 10^{19}$.

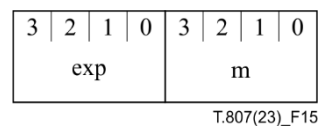


Figure 15 – Distortion field syntax

exp: Exponent of distortion field value (base 16)

m: Mantissa of distortion field value

Table 15 – Distortion field parameter values

Parameter	Size (bits)	Values
exp	4	0 ... 15
m	4	0 ... 15

Note that with this format for the distortion, a comparison between two distortions to determine which is larger can be simply achieved by comparing the two distortion values as unsigned char. Specifically, to perform this comparison there is no need to convert from the pseudo floating-point format to the actual total distortion in order to determine which of two distortion values is larger or smaller. This property can simplify the processing in various applications.

5.7.3.2.2 Two-byte distortion field

In the two-byte format as shown in Figure 16, distortion values shall be expressed as a two-byte number in pseudo floating-point format. The pseudo floating-point format for distortion is defined as follows. This format is used in D.1.1.1 (Equation D.3) of Rec. ITU-T T.800 | ISO/IEC 15444-1 to express the quantization step size for JPEG 2000. Each 16-bit number contains the exponent (5 bits) and mantissa (11 bits) of the metric value. In particular, the floating-point value V of the metric is given by the following formula:

$$V = 2^{\varepsilon-15} \left(1 + \frac{\mu}{2^{11}} \right) \quad \text{if } \varepsilon \neq 0$$

$$V = 0 \quad \text{if } \varepsilon = 0$$

where ε is the unsigned integer obtained from the first five most significant bits of the parameter, and μ the unsigned integer obtained from the remaining 11 bits. The special case of $V = \infty$ correspond to $\mu = 0$ and $\varepsilon = 31$. Note that values that would underflow the representation are set to zero.

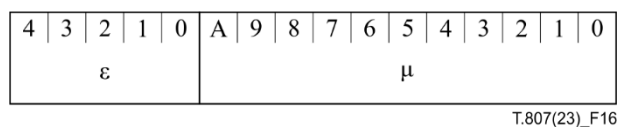


Figure 16 – Distortion field syntax

ε : Exponent of two-byte distortion field value.

μ : Mantissa of two-byte distortion field value.

Table 16 – Distortion field parameter values

Parameter	Size (bits)	Values
ε	5	0 ... 31
μ	11	0 ... ($2^{11} - 1$)

The algorithm to compute ε and μ is not defined as a mandatory part of this Recommendation | International standard. A possible technique performs the following steps (an example of conversion of the number 12.25 is provided). If $V = 0$, set $\varepsilon = \mu = 0$. Otherwise:

- convert V to a binary number ($12.25_{10} = 1100.01_2$);
- normalize the number; this means there should be a 1 digit to the left of the binary point and multiplication by the appropriate power of two to represent the original value. The normalized form of 1100.01 is 1.10001×2^3 ;
- the exponent is the power of 2, presented in excess notation. The exponent bias is 15; hence for this example the exponent is represented as 18_{10} (10010₂);
- the mantissa represents the significant bits, *except for the bit to the left of the binary point*, which is always one and therefore does not need to be stored; zeros are possibly appended so as to obtain 11 bits. For this example, the mantissa is 10001000000.

5.7.3.2.3 Relative importance field

The relative importance field r can be used to describe the relative importance among different coding units, without necessarily being tied to a specific distortion metric. This enables one to describe the relative importance or prioritization among coding units without explicitly describing how much more important one is from another. This relative importance of the associated data is specified by an n -byte field which supports 2^{8n} possible rankings as shown in Figure 17 and Table 16, where the number of bytes n for this field is specified by $Mzoi$. For example, by using a one-byte relative importance field a total of 256 possible rankings are supported. Increasing values correspond to increasing importance, in a similar manner to the distortion field.

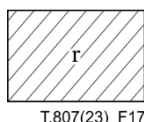


Figure 17 – Relative importance field syntax

r : Relative importance value

Table 17 – Relative importance field parameter values

Parameter	Size (bits)	Values
r	$8 * n$	0 ... ($2^{8n} - 1$)

5.7.3.2.4 Additional comments on distortion field and relative importance field

Since for both the one-byte distortion field and one-byte relative importance field the larger values correspond to greater importance, it is possible to make comparisons for these two data units in the same manner irrespective of whether the distortion field specifies actual distortion or a relative importance. This can simplify applications.

Headers can be specified using the distortion or relative importance fields. The loss of various types of data, such as the main and tile-part headers or the SEC header, prevent the decoding of the related image data. The JPSEC creator can assign distortion to this data using either:

- 1) the highest distortion value (specified next) to signal the header or critical data; or
- 2) to describe the total distortion that would be created if the image or portion of the image is undecodable.

The creator then has some flexibility in how to signal the headers.

The highest distortion value for the one-byte fields is a byte of all ones (0xFF). Note that this value is the highest possible distortion value for both the one-byte total squared error distortion field and for the one-byte relative importance field. The highest distortion value for the two-byte distortion field is the two bytes of all ones (0xFFFF). The highest importance for the relative importance field of length n-bytes is an n-byte value of all ones.

5.7.3.2.5 Joint use of distortion field and relative importance field

The distortion field and relative importance field can be used simultaneously to describe the area specified by the ZOI. In this case both fields specify squared-error distortion, however the distortion field specifies the incremental reduction in distortion while the relative importance field specifies the total distortion. Specifically, the distortion field specifies the incremental reduction in distortion that the ZOI would produce if decoded. This assumes that all information required to decode the ZOI is available, and focuses on the incremental reduction in distortion produced by the ZOI. The relative importance field specifies the total distortion that would be incurred if the ZOI is not available, i.e., it specifies the total distortion that would result if the given ZOI is unavailable for decoding by accounting not only for the value of the ZOI itself (as expressed by the distortion field) but also accounting for the distortion produced because other parts of the compressed bitstream which depend on the ZOI are undecodable. The total distortion associated with different ZOIs provides a useful metric for the relative importance of the different ZOIs. When both fields are used they will use the same mathematical expression for distortion, as signalled by the distortion field.

5.7.3.3 Bit-rate field

The Bit-rate field is used to specify the protected zone in wavelet coefficient domain. It identifies the most significant bit-planes whose compressed bit-rate is specified by this field. The MSBs are selected using the rate-distortion optimization process specified in the JPEG 2000 Core coding system. For example, if the Bit-rate value is 2.5, the protected zone includes the MSBs of all wavelet coefficients whose compressed bit-rate is 2.5 bit per pixel. The syntax of Bit-rate field is shown in Figure 18 and Table 18. The specified bit-rate is given by:

$$R = I_R + F_R/16$$

For example, a bit-rate of zero is represented by $I_R = 0$ and $F_R = 0$; a bit-rate value of 2.5 is represented by $I_R = 2$ and $F_R = 8$.

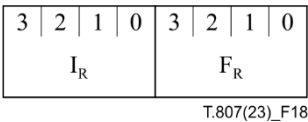


Figure 18 – Bit-rate field syntax

I_R : The integer part of the specified bit-rate.
 F_R : The fractional part of the specified bit-rate.

Table 18 – Bit-rate field parameter values

Parameter	Size (bits)	Values
I_R	4	0 ... 15
F_R	4	0 ... 15

5.7.4 Relationship between multiple parameters

5.7.4.1 Global

When the image-related description class has multiple fields set simultaneously, the resulting zone shall be the intersection of these fields. For example, a zone could specify the lowest resolution level in the 2nd tile. The union of fields can be specified by using multiple zones in the ZOI.

The non-image related description class can also have multiple fields set simultaneously. When this occurs, the modes for the various parameter fields shall have the same number of items (one exception to this rule is described below), and these items shall correspond with one another in a one-to-one manner. For example, if the zone uses byte ranges and packet ranges, each should have the same number of range items where the first byte range corresponds to the first packet range, and so on.

There is one exception to the above rule on requiring the same number of items for each field. This occurs when one of the fields $f1$ contains 1 item which specifies a range of items (as described by the range mode in 5.7.6) where this range contains N elements and when another field $f2$ is specified by a list of N items. In this case, the field $f1$, which contains only 1 item (the range) is interpreted as a list of N items. These N items specified by the range in $f1$ shall correspond one-to-one with the N items listed in $f2$. Therefore, a range of items can be associated to either a single item or to multiple items (one for each item in the range).

5.7.4.2 Examples

As previously illustrated in Figure 11, the zone description class structure can have multiple fields set simultaneously, where N fields are image related descriptions ($D_i^1, D_i^2, \dots, D_i^N$) and M fields are non-image related descriptions ($D_n^1, D_n^2, \dots, D_n^M$). The semantics can be understood as $\{D_i^1 \cap D_i^2 \cap \dots \cap D_i^N\} = D_n^1 = D_n^2 = \dots = D_n^M$, that is, the intersection of the N image related descriptions is corresponding to each of the M non-image related descriptions, and in addition, the M non-image related descriptions are mutually corresponding to each other. This relationship is further illustrated with three examples below.

In the first example, the zone description has two image related descriptions: one for resolution 2 and the other for layer 3. In this case, the influenced data is the intersection of resolution 2 and layer 3, as illustrated in Figure 19.

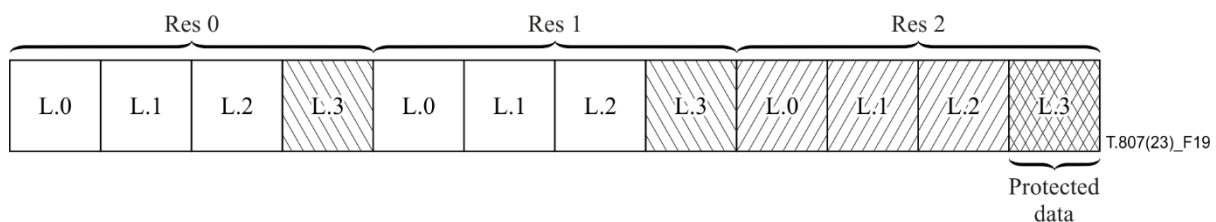


Figure 19 – ZOI example using image related descriptions

In the second example as illustrated in Figure 20, the zone description has two image related descriptions (which are resolution 2 and layer 3) and one non-image related description (which is packet range 80-100). In this case, the influenced data is the intersection of resolution 2 and layer 3. Furthermore, this indicates that the influenced data is contained in packets ranging from 80 to 100.

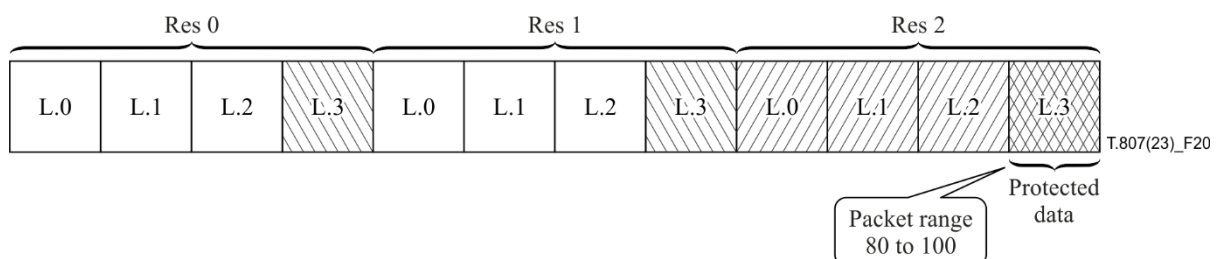


Figure 20 – ZOI example using image related and non-image related descriptions

In the third example as illustrated in Figure 21, the zone description has two image related descriptions (which are resolution 2 and layer 3) and two non-image related descriptions (which are packet range 80-100 and byte range 856-1250). Once again, the influenced data is the intersection of resolution 2 and layer 3, and the influenced data is

contained in packets ranging from 80 to 100. Furthermore, these packets and influenced area are located in the byte range 856-1250.

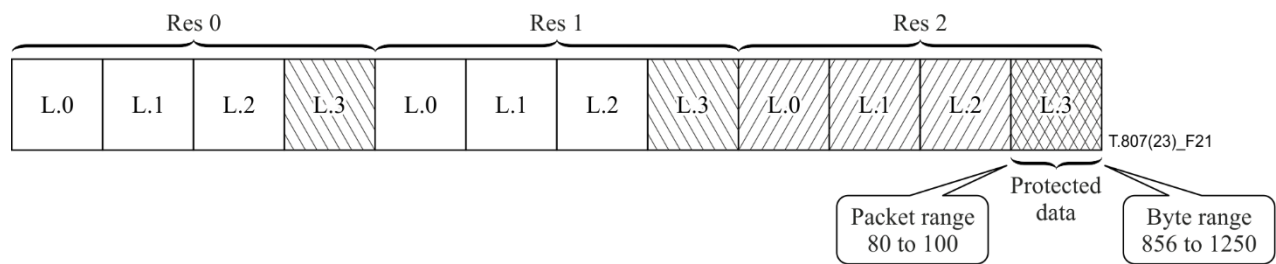


Figure 21 – A second ZOI example using image related and non-image related descriptions

5.7.5 Protecting any data that follows the SEC marker

The above discussion has largely focused on supporting protection services for the JPEG 2000 codestream. However, many elements of the main header, including JPSEC signalling, should also be protected, and the ZOI and protection methods can also be used for this purpose.

Specifically, the byte range mode of the non-image related description class can be used to specify that a JPSEC tool should be applied to any data following the SEC marker. As described before, the first byte of the SEC header is byte 1 for indexing the byte range. The data that follows the SEC marker and that can be protected includes the SEC segment and most of the main header. Note that all of the JPEG 2000 main header, except for the SIZ marker segment, can be moved after the SEC marker and hence can be protected using the above approach. If the JPEG 2000 SIZ marker segment is to be protected, it can be done at a higher level, e.g., file format layer.

The JPSEC tools for protecting the SEC segment should generally be the first tools in the SEC segment. This enables the consumer to first render the SEC segment data, which can then be used to process the remainder of the codestream.

5.7.6 Zone description parameter syntax (Pzoi)

Figure 22 shows the ZOI description parameter syntax.

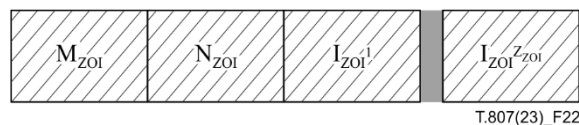


Figure 22 – ZOI description parameter syntax

- Mzoi:** ZOI description mode. This field uses the FBAS structure.
- Nzoi:** Number of Izoi. This field uses RBAS structure.
- Izoiⁱ:** Item.

Table 19 – Pzoiⁱ parameter values

Parameter	Size (bits)	Values
Mzoi	Variable (FBAS)	See Table 20
Nzoi	0 8 + 8 * n (RBAS)	If bit number 2 of Mzoi is 0. 2 ... (2 ^{7+7*n} - 1)
Izoi ⁱ	Variable	Depends on the mode specified in Mzoi

Table 20 – Mzoi parameter values

FBAS bit number	Values (bits)	Semantics
1	0	The specified zones are influenced by the JPSEC tool
	1	The complement of specified zones are influenced
2	0	Single item is specified
	1	Multiple items are specified
3, 4	00	Rectangle mode. A rectangle region where the first value pair specifies the upper-left corner and the second value pair specifies the lower-right corner such that both corners are inclusive. For each corner, the first value shall be the horizontal position and the second value shall be the vertical position. The indexing shall begin at 0, and shall use the reference grid defined in the JPEG 2000 Core coding system.
	01	Range mode. A range of values where the first value specifies the start index and the second value specifies the last index, both inclusive.
	10	Index mode. Specifies single value(s).
	11	Max mode. Specifies the maximum value.
5, 6	00	Izoi ⁱ uses 8-bit integer
	01	Izoi ⁱ uses 16-bit integer
	10	Izoi ⁱ uses 32-bit integer
	11	Izoi ⁱ uses 64-bit integer
7, 8	00	Izoi ⁱ is described in one dimension
	10	Izoi ⁱ is described in two dimensions
	01	Izoi ⁱ is described in three dimensions
9	0	Offset with lengths mode is not used
	1	Offset with lengths mode is used: Specifies the initial offset with lengths of contiguous bytes that follows. The existence of this flag shall override the modes specified in bits 3 and 4.
		All other values are reserved

When TRLCp tags are used, their size is defined by P_{TRLCp} as specified in Table 4. In this case, bits 5 and 6 of the M_{ZOI} parameter are overridden.

The Offset with lengths mode can be used to efficiently represent a series of consecutive segments, for example, a series of consecutive byte ranges. The first value specifies the initial offset, the following values specify the lengths of each consecutive segment. If this field is used to represent n segments, then N_{ZOI} should be set to $n + 1$.

5.8 Protection method template syntax (T)

5.8.1 General

Protection method templates contain parameters for specific JPSEC tools described in 5.6.1. For example, they are used in JPSEC normative tools described in 5.6.2. Also, they can be used in JPSEC non-normative tools described in 5.6.3. There are three types of protection method templates: decryption template, authentication template, and hash template. The template used by a JPSEC normative tool is specified by its ID as shown in Table 6 and again here in Table 21 with references to the appropriate clauses where they are defined.

As described in 5.6.2, the protection method template T together with a JPSEC tool's processing domain PD , granularity G , and value list V describe how a JPSEC tool is applied.

Table 21 – Template ID values (ID_T)

Values	Protection method template
0	Reserved
1	Decryption template. See 5.8.2.
2	Authentication template. See 5.8.3.
3	Hash template. See 5.8.4.
4	NULL tool
	All other values are reserved for ITU-T ISO use