ISO/IEC JTC 1/SC 29 N 3480

Date: 2004-11-12

ISO/IEC CD 15444-8

ISO/IEC JTC 1/SC 29/WG 1

Secretariat: JISC

Information technology — JPEG 2000 image coding system — Part 8: Secure JPEG 2000

Technologie de l'information — Système de codage d'image JPEG 2000 — Partie 8: JPSEG 2000 Sécurisé

Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard .

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Warning

This document is not an ISO International Standard. It is distribute d for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are awar e and to provide supporting documentation.

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. Na tional 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. In the field of information technology, ISO and IEC have established a joint tech nical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the e lements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 15444-8 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Audio, Picture, Multimedia and Hypermedia Information* .

This second/third/... edition cancels and replaces the first/second/... edition (), [clause(s) / subclause(s) / table(s) / figure(s) / annex(es)] of which [has / have] been technically revi sed.

ISO/IEC 15444 consists of the following parts, under the general title *Information technology — JPEG 2000 image coding system*:

Part 1: Core coding system

Part 2: Extensions

Part 3: Motion JPEG 2000

Part 4: Conformance testing

Part 5: Reference software

Part 6: Compound image file format

Part 8: Secure JPEG 2000

Part 9: Interactivity tools, APIs and protocols

Part 10: 3-D and floating point data

Part 11: Wireless

Part 12: ISO base media file format

Introduction

In the “Digital Age” the Internet provides many new opportunies for rightholders 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 answering that copyright, and the cultural and intellectual expression it fosters, remains well protected in the 21 century. The new Digital economy and the creative people in every country of the world depend on it. Also in Dec 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. [10]

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 must 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, authentification, and integrity checking.

Part 8 of JPEG 2000 standard intends to provide tools and solutions in terms of specifications that allow applications to generate, consume, and exchange Secure JPEG 2000 bitstreams. This is referred to as **JPSEC**.

Information technology — JPEG 2000 image coding system — Part 8: Secure JPEG 2000

1 Scope

This document specifies ISO/IEC International Standard| Recommendation 15444 -08 for securing JPEG 2000 bitstreams. It specifies the framework, concepts, methodology and criteria to be used to claim that a JPEG 2000 file or bitstream is secure according to ISO/IEC International Standard| Recommendation 15444 -08.

The scope of this document is to define:

1.) a normative codestream syntax containing information for interpreting a secure image data

2.) a normative process for registering JPSEC tools at a central registration authority delivering a unique identifier

3.) informative examples of JPSEC tools in typical use cases

4.) informative guidelines on how to implement the security services and related metadata.

The scope of this document is not to describe specific secure applications or to limit ISO/IEC International Standard| Recommendation 15444-08 to specific techniques but to enable future extensions as the development on secure data transfer goes on.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ITU–T Rec. T.800 | ISO/IEC 15444-1:2000, *Information technology — JPEG 2000 image coding system: Part 1: Core coding system*

ITU–T Rec. T.801 | ISO/IEC 15444-2, *Information technology — JPEG 2000 image coding system: Part 2: Extensions*

ISO/IEC 15444-3, *Information technology — JPEG 2000 image coding system -- Part 3: Motion JPEG 2000*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply. The definitions defined in ITU-T Rec. T.800 | ISO/IEC 15444–1:2000 Clause 3 apply to this International Standard.

3.1
Access control
The prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner [ISO/IEC 7499-2].

3.2
Authentication
The process of verifying an identity claimed by or for a system entity.

3.2.1
Source authentication
Source authentication is the verification that a source entity (say, user/party) is in fact the claimed source entity [ISO/IEC 9798-1: 1997, ISO/IEC 11770-2: 1996, ISO/IEC 11770-3: 1999, ISO/IEC FDIS 15946-3 (02/2001)].

3.2.2
Fragile/Semi-fragile image authentication
Fragile/semi-fragile image authentication includes both image source authentication and image data/image content integrity verification. The process should be able to detect any change in the signal and identify where it has taken place and possibly what the signal was before modification. It serves at proving the authenticity of a document. The difference between fragile and semi-fragile image authentication is that the former is to verify the image data integrity and the latter to verify the image content integrity.

3.3
Confidentiality
Confidentiality is the property that information is not made available or disclosed to unauthorized individuals, entities or processes [ISO/IEC PDTR 13335-1 (11/2001)].

3.4
Data splitting
Data splitting is a method to protect sensitive data from unauthorized access by encrypting the data and storing different portions of the file on different servers. 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 get access to each server, know what data to combine, and how to decrypt it.

3.5
Decryption
Inverse transformation of the encryption. An alternative term for decryption is deciphering.

3.6
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 [ISO/IEC 7499 -2].

3.7
Encryption
(Reversible) transformation of data by a cryptographic algorithm to produce ciphertext, i.e. to hide the information content of the data. An alternative term for an encryption algorithm is cipher.

3.8
Fingerprints
Fingerprints are characteristics of an object that tend to distinguish it from other similar objects. They enable the owner to trace authorized users distributing them illegally. In this respect, fingerprinting is usually discussed in the context of the traitor tracing problem.

3.9
Hash function: A function which maps strings of bits to fixed-length strings of bits, satisfying 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.

Computational feasibility depends on the user's specific security requirements and environment. [ISO/IEC 14888 -1: 1998].

3.10
Integrity
The property of being able to safeguard the accuracy and the completeness of assets. [ISO/IEC PDTR 13335 - 1 (11/2001)]

3.10.1
Image data integrity
Image data integrity denotes the property that data has not been altered or destroyed in an unauthorized manner. [ISO/IEC 9797-1: 1999]

3.10.2
Image content integrity
Image content integrity refers to the assurance the image content has not been modified by unauthorized parties in such a way that its perceptual meaning is changed. It allows the content-preserving operations to be performed on the image without triggering the integrity alarm.

3.11
JPSEC application
A JPSEC application is any software or hardware process that is capable of consuming JPSEC bitstreams by interpreting the JPSEC syntax in order to provide the specified security services. It makes use of one or several JPSEC tools.

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

3.12
JPSEC bitstream
A JPSEC bitstream contains JPEG 2000 coded data which has been modified to provide one or several security services. For instance the original JPEG 2000 coded data may be partially or totally scrambled or encrypted to provide confidentiality. Furthermore, a JPSEC bitstream includes signalling implemented by the SEC and INSEC marker segments.

3.12.1
JPSEC creator
The entity who creates a JPSEC bitstream from an image, a JPEG 2000 codestream, or a JPSEC codestream in order to provide some JPSEC services.

3.12.2
JPSEC consumer
The entity who receives a JPSEC bitstream and renders the JPSEC services based on the bitstream.

3.13
JPSEC service
A JPSEC service provides security for consumption of JPEG 2000 images. The service counters security attacks and makes use of one or several JPSEC tools.

3.14
JPSEC Registration Authority
JPSEC registration authority is in charge of delivering a u nique ID to reference a JPSEC tool and storing the parameter list of the JPSEC tool's description.

3.15
JPSEC tool
A JPSEC tool is a hardware or software process involving security techniques that implements a security service. In JPSEC there are two types of tools. JPSEC template tools are specified with a predefined tool template. JPSEC defines templates for decryption, authentication, and integrity. JPSEC registration authority tools are specified by an identification number given by the JPSEC registrati on authority.

3.16
JPSEC tool description
A JPSEC tool description consists of two parts: the parameter list and its values. In the case of JPSEC template tools, the parameter list is given by the standard. In the case of JPSEC non-normative tools the parameter list may be given by the registration authority. In both cases the parameter values are specified in the SEC and INSEC marker segments.

3.17
Key
A sequence of symbols that controls the operations of encipherment and decipherment [ISO/IEC 7499 -2].

3.17.1
Symmetric key
When both the originator and the recipient use the same secret key or two keys that can be easily computed from each other in a cryptographic system, the key or the pair of keys are referred to as symmetric key(s). [A. Menezes, P. van O oschot and S. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996.]

3.17.2
Asymmetric key
When a so-called public key is used in the originator, a private key is used in the recipient in a cryptographic system, and it is computationally infeasible to determine the private key from the public key, the pair of keys is referred to as asymmetric keys. [A. Menezes, P. van Ooschot and S. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996.]

3.17.2.1
Private key
That key of an entity's asymmetri c key pair which should only be used by that entity.

3.17.2.2
Public key
That key of an entity's asymmetric key pair which can be made public [ISO/IEC FDIS 9796 -2 (12/2001), ISO/IEC 11770-1: 1996].

3.18
Key generation
Key generating function: A function which takes as input a number of parameters, at least one of which shall be secret, and which gives as output keys appropriate for the intended algorithm and application. The function shall have the property that it shall be computationally infeasible to de duce the output without prior knowledge of the secret input [ISO/IEC 11770 -2: 1996].

3.19
Key management
The generation, storage, distribution, deletion, archiving and application of keys in accordance with a security policy [ISO/IEC 7499-2].

3.20
Marker emulation
A marker emulation occurs when the output of the encryption process results in cipher text that emulates a JPEG start code.

3.21
Message Authentication Code (MAC) algorithm
An algorithm for computing a function which maps strings of bits and a sec ret key to fixed-length strings of bits, satisfying the following two properties:

- for any key and any input string the function can be computed efficiently;
- for any fixed key, and given no prior knowledge of the key, it is computationally infeasible to compute the function value on any new input string, even given knowledge of the set of input strings and corresponding function values, where the value of the ith input string may have been chosen after observing the value of the first i -1 function values.

A MAC algorithm is sometimes called a cryptographic check function or cryptographic checksum function. Computational feasibility depends on the user's specific security requirements and environment. [ISO/IEC 9797-1: 1999]

3.21.1
Message Authentication Cod e (MAC)
The string of bits which is the output of a MAC algorithm.

3.22
Non-repudiation
The binding of an entity to a transaction in which it participates, so that the transaction cannot later be repudiated. 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.

3.23
Packet
In JPEG2000 Part 1, packet is defi ned as a part of the bit stream comprising a packet header and the compressed image data from one layer of the p recinct of one resolution of one tile-component. Note that this is different from the term “packet” used in data transmission through network.

3.24
Protection
Any process in order to protect content.

3.24.1
Protection method
A method used to create or c onsume protected content such as encryption, decryption, authentication, and integrity checking.

3.25
Security
All aspects related to defining, achieving, and maintaining confidentiality, integrity, availability, accountability, authenticity, and reliability. A product, system, or service is considered to be secure to the extent that its users can rely 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.

3.26
Signalling syntax
The signalling syntax specifies the format of the JPSEC bitstream. It contains all the required information for consuming secure JPEG 2000 images.

3.27
Transcoding
Transcoding is the operation of taking an input compressed bitstream and a dapting or converting it to produce an output compressed bitstream that has some desired property. For example, the output compressed bitstream may represent an image with a lower spatial resolution or lower bit rate than the input compressed bitstream.

3.27.1
Secure Transcoding
Secure transcoding is the operation of performing transcoding, or adaptation, of a protected input compressed content, without unprotecting the content. The term secure transcoding is used, as opposed to transcoding, to stress that the transcoding operation is performed without compromising security. Secure transcoding may also be referred to as performing transcoding in the encrypted domain

3.28
Watermarking
Watermark is what is imperceptibly added to the cover -signal in order to convey hidden data. Watermarking process imperceptibly inserts data representing some information into multimedia data in one of the following two ways: one is the lossy way which means the exact cover -signal will never be able to be recovered once the watermark is embedded. The other is the lossless way which means the exact cover -signal could be recovered after watermark extraction.

4 Symbols and abbreviated terms

IP	Intellectual Property related to technology
IPR	Intellectual Property Rights relate d to content
JPSEC	Secure JPEG 2000
PKI	Public Key Infrastructure
ZOI	Zone Of Influence
BAS	Byte Aligned Segment
LSB	Least Significant Bit
MSB	Most Significant Bit
RA	Registration Authority

5 JPSEC Syntax (normative)

5.1 JPSEC framework overview

Central to the JPSEC framework is the JPSEC bitstream (see Figure 1): this represents a secure JPEG 2000 image. The syntax assumes that the JPSEC bitstream contains JPEG 2000 coded data, but it may be partially scrambled or encrypted. In all cases it must follow the normative syntax defined in this IS/Recommendation.

To the JPSEC bitstream are associated a number of JPSEC security services including confidentiality of the image data, integrity of the image data, and authentication of image data origin.

The signalling syntax specifies :

- what security services are associated with the image data

- which JPSEC tools are required to render the corresponding services

- which parts of the image data are protected

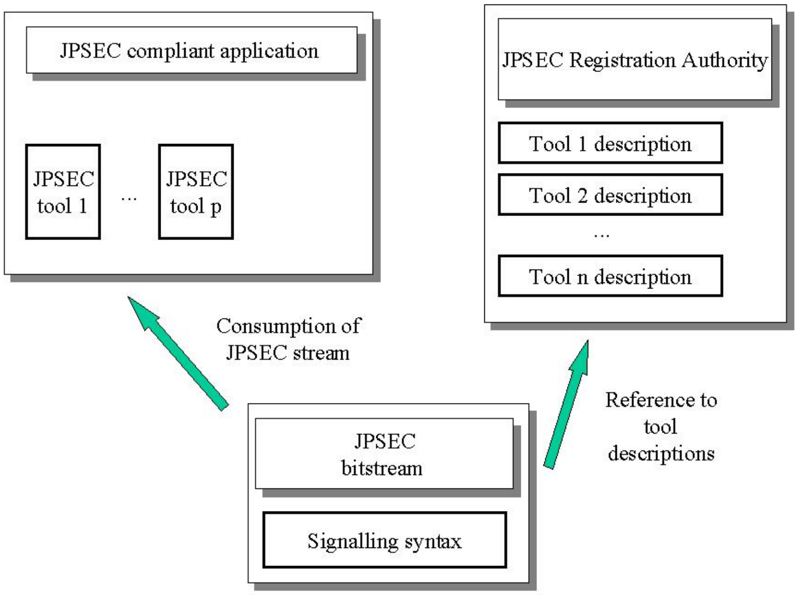


Figure 1. Overview of the JPSEC framework

The syntax of the JPSEC bitstream is normative. In this way, JPSEC applications are able to consume JPSEC bitstreams: The application interprets the bitstream, idenitifies and applies the signalled JPSEC tools and renders the corresponding security services.

From this it follows that the way in which the JPSEC bi tstream is generated is out of scope of this IS/R. Of course, JPSEC creation applications must generate JPSEC bitstreams that include the appropriate JPSEC signalling. JPSEC bitstreams can be created in a number of ways. A few illustrative examples are shown in Figure 2. JPSEC bitstreams can be created from an original image, from JPEG 2000 coded data, or from another JPSEC bitstream. In the first case, the encoding and protection operations are performed at the same time, so the JPSEC tool has access to the original image content. This may be important for JPSEC tools such as content authentication. In the second case, the JPSEC bit stream is created from JPEG 2000 coded data. This may occur when performing encryption on a database of JPEG 2000 images. Finally, as shown in Figure 3, the JPSEC bitstream may be created from another JPSEC bitstream. This may aris e 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 may be significant.

A JPSEC bitstream is created in view of its use by a JPSEC consumer application, so that the security services associated with the bitstream are applied correctly. For example, if the JPSEC bitstream is signed, the JPSEC application is able to verfiy the signature. I n the general case, there is no guarantee on the result of running a JPEG 2000 part 1 compliant decoder on JPSEC bitstreams: for example, the decoder could render a scrambled image, it could refuse to decode the image or it could in some cases crash. In th e special case where the JPSEC bitstream is also part 1 compliant, a JPEG 2000 decoder would be able to decode the image properly.

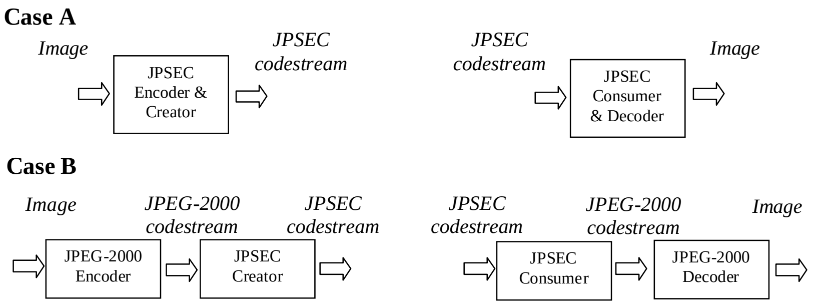


Figure 2 — Creation and consumption modes of JPSEC content.

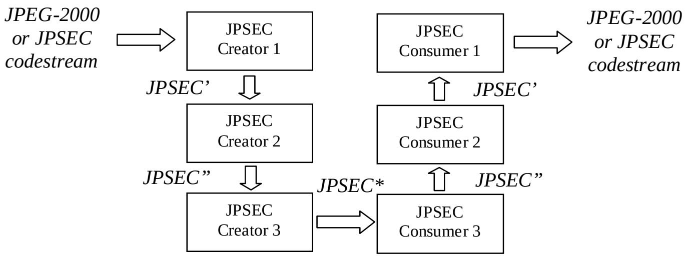


Figure 3 — Creation and consumption modes of JPSEC content.

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 -1 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, or registered by a JPSEC Registration Authority. JPSEC tools specified by the template s have unique processing behaviour and therefore do not require unique identification. Those specified by the registration authority are associated with a unique identification number provided by the common registry .

5.2 JPSEC security services

The objective in this section is to list and to explain the functionalities which are included in the scope of ISO/IEC International Standard| Recommendation 15444 -08.

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 (plain text) into a form (cipher text) that conceals the data's original meaning. By selective encryption we mea n that not the entire image and/or metadata but only parts of the image and/or meta data can be encrypted.

— Integrity verification

A JPSEC file can sup port means of detecting manipulations to the image and/or metadata and thereby verify their integrity. The re 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) verific ation.

Those fragile or semi fragile image integrity verification s might identify locations in the image data/image content where the integrity is put into question. Solutions may include

1. cryptographic methods such as Message Authentication Codes (MAC), digital signatures, cryptographic checksums or keyed hash

2. watermarking-based methods

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 fi le. 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 instan ce 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 mat ching 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 node can perform streaming and transcoding without requiring decryption or unprotecting the content. A n 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 Main security marker (SEC)

5.3.1 Security Marker Segments

In this section, we present a simple and flexible, yet powerful syntax for JPSEC signalling. 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 inform ation for securing JPEG 2000 images. To do so, it makes references to JPSEC template tools that are specified by the templates described in Section 5.6 or by JPSEC non-normative tools that may have been registered a priori with the JPSEC registration authority or defined privately, and it makes provisions for handling parameters related to these tools.

A JPSEC stream can be protected with one or more JPSEC tools. Each tool is a JPSEC template 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 must 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 may exceed the maximum marker segment size. When this occurs, additional SEC marker segments can be used for signalling.

Figure 4 shows the syntax of the SEC marker segment. The segment is signalled by the SEC marker 0x FF65. 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} is 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.

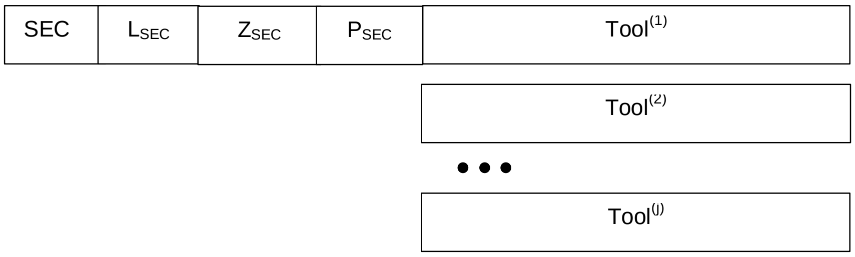


Figure 4 — Main security marker segment syntax

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 (not including the marker).

Z_{SEC}: Index of this marker segment relative to all other SEC marker segments present in the current header.

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 bitstream.

Table 1 — Main security parameter values

Parameter	Size (bits)	Values
SEC	16	0xFF65
L_{SEC}	16	$2 – (2^{16}-1)$
Z_{SEC}	8	0-255
P_{SEC}	0, if $Z_{SEC}>0$ 24 or 56, otherwise	If $Z_{SEC}=0$, See Table 2.
$Tool^{(i)}$	Variable	See Sec. 5.4.2 and Sec. 5.4.3

Figure 5 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 is 0 and increases 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.

[Editors note: Make provisions for one tool that spans multiple SEC marker segments.]

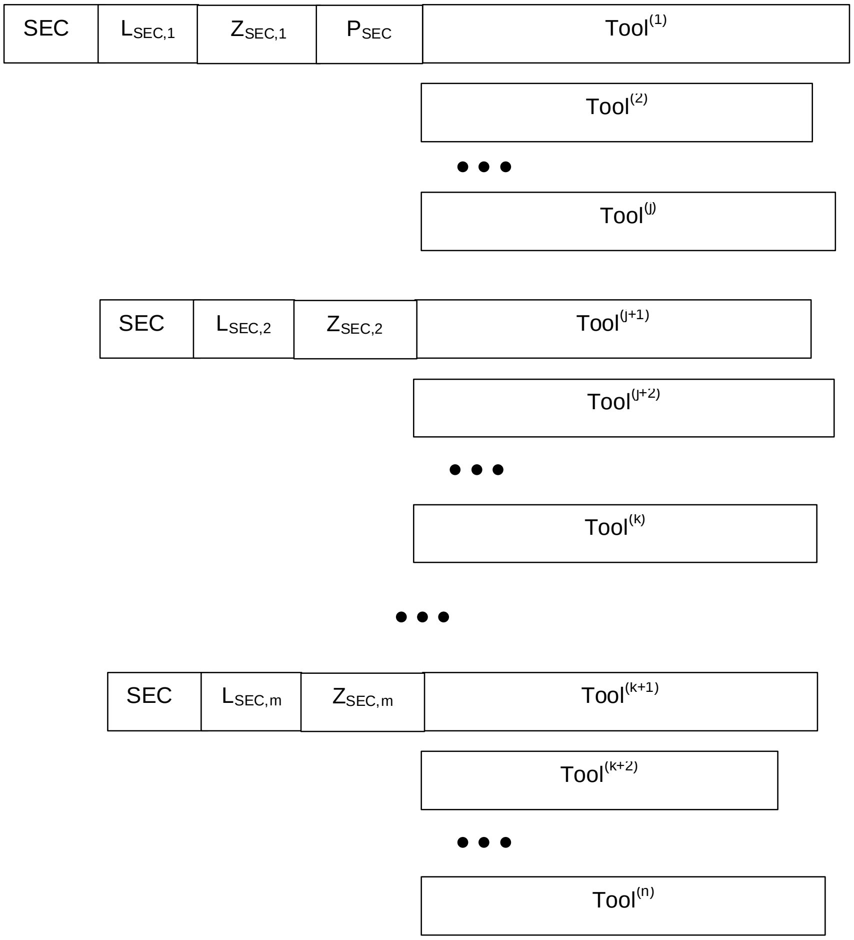


Figure 5 — Main security marker syntax when multiple marker segments are used.

P_{SEC} is a parameter field that describes security parameters for the entire codestream as opposed to for a particular tool. This is useful to flag events such as JPEG 2000 Part 1 compliance or the use of INSEC markers. The P_{SEC} parameters are:

F_{INSEC}: Flag to indicate if INSEC marker segment is used .

$F_{multiSEC}$: Flag to indicate if multiple SEC marker segments are used .

F_{J2K}: Flag to indicate that the codestream is decodable by JPEG 2000 Part 1 decoders .

F_{TRLCP}: Flag to indicate that TRLCP tag usage is defined in P_{SEC}.

N_{tools}: Number of JPSEC tools used in the codestream .

I_{max}: Maximum tool instance index value used in the codestream .

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_{INSEC}	1	0: if INSEC is not used 1: if INSEC is used
$F_{multiSEC}$	1	0: if one SEC marker segment is used 1: if multiple SEC m arker segments are used
F_{J2K}	2	1: if JPSEC stream is compliant with JPEG 2000, part 1 0: otherwise
F_{TRLCP}	1	1: if TRLCP tag usage is defined in P_{SEC} 0: otherwise
N_{tools}	7	$1 – (2^7-1)$
I_{max}	7	$0 - (2^7-2)$
padding	5	Reserved for ISO use
P_{TRLCP}	0, if F_{TRLCP}=0 32, if F_{TRLCP}=1	See Table 3

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 e ach 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:

b_T: Number of bits to represent tile i ndex in TRLCP tag.

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

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

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

b_P: Number of bits to represent precinct index i n TRLCP tag.

Table 3 — Parameter field for TRLCP tag descriptor (P_{TRLCP})

Parameter	Size (bits)	Values
b_T	8	$1 – (2^8-1)$
b_R	4	1 – 8
b_L	5	1 – 16
b_C	5	1 – 16
b_P	8	$1 – (2^8-1)$
Padding	2	0

The size of each resulting TRLCP tag is the smallest integer byte size that contains all the bits. The format of the TRLCP tag contains the bits for the tile index, the resolution 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 TRLCP 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 MSB's of the TRLCP tag if they exist.

5.3.2 Application of Multiple JPSEC Tools

In many applications it is necessary to apply multiple JPSEC tools to a single JPEG 2000 codestream. For example, both encryption and authentication may be applied to protect a JPEG 2000 image. The general situation of applying multiple JPSEC tools is illustrated in Figure 4, Figure 5, and Figure 6, where N tools are applied. The JPSEC consumer will read the N tools in order of placement in the JPSEC segment shown in Figure 4 or Figure 5, and apply them in that same order to perform the JPSEC decoding of the JPSEC file. Note that while the JPSEC consumer applies the JPSEC tools in order 1,2,…N, as read from the JPSEC segment, during creation of the JPSEC file these JPSEC tools were applied in the reverse order, i.e. N,N-1,…2,1, as illustrated in Figure 6. 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 protection tool in a JPSEC file 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 decoding 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 6, the state at the consumer after JPSEC decoding 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 may be desirable for a JPSEC consumer to decode the multiple JPSEC tools in a different manner than described above. For example, the JPSEC consumer may choose to decode the multiple tools in a different order, or to skip certain tools in the decoding. Furthermore, the JPSEC consumer may 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.

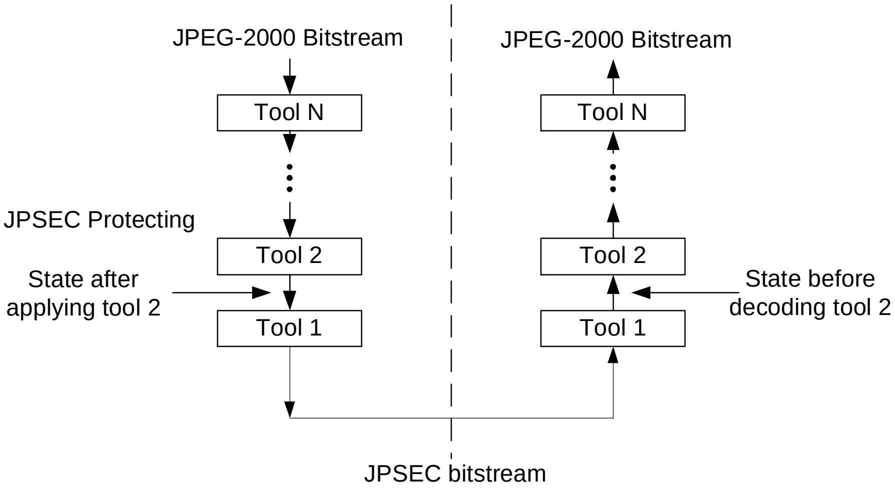


Figure 6 Use of multiple JPSEC tools.

5.4 JPSEC Tools

5.4.1 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 Section 5.6, and are also known as JPSEC template tools. JPSEC non-normative tools are specified by a JPSEC registration authority or by a particular JPSEC application based on their ID number, and are respectively referred to as JPSEC registration authority to ols or JPSEC application-defined tools. The syntax for JPSEC normative tools are discussed in Section 5.4.2. The syntax for JPSEC non-normative tools are discussed in Section 5.4.3.

The syntax for JPSEC tools is shown in Figure 7. 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 C BC mode with a specified set of initialization vectors and keys (how).

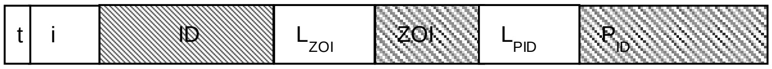


Figure 7 — JPSEC Tool syntax (Tool $^{(i)}$)

t: Tool type. 0 indicates a JPSEC normative tool. 1 indicates a JPSEC non-normative tool.

i: Tool instance index (can be used as a unique identifier)

ID: Identification value for protection tool *i*. For normative tools, the ID=ID_T is 8 bits and specifies the template type. For non-normative tools, the ID=ID_{RA} is 32 bits.

L_{ZOI}: Length of L_{ZOI} + ZOI in Bytes

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

L_{PID}: Length of L_{PID} + P_{ID} in Bytes

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

Table 4 — JPSEC Tool parameter values

Parameter	Size (bits)	Values
t	1	0 – 1
i	7	0 – (2^7-2) 2^7-1, reserved
ID	8, if t=0 32, if t=1	See Table 5 See Sec. 5.6
L_{ZOI}	16	0 – (2^{16}-1)
ZOI	Variable	See Sec. 5.5
L_{PID}	16	0 – (2^{16}-1)
P_{ID}	Variable	Defined by tool ID

The JPSEC non-normative 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 fiel d P_{ID}, which is a field to transmit one or more parameters for the JPSEC tool.

The one byte identifier at the beginning of the tool has the following structure: the most significant bit represents the tool type, t, where 0 specifies a JPSEC normative tool and 1 specifies a JPSEC non-normative tool, and the 7 least significant bits represent the instance index, i. The instance index shall be a unique identifier of the tool within the codestream, and thus should 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 contains the instance index of the tool to which it applies. It is recommended that the first tool applied at a JPSEC protector have 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 b its 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 integrity template. For JPSEC non-normative tools, the first bit indicates whether it is a JPSEC registration authority tool or a JPSEC user-defined tool. In either case, the ID number indicates the particular tool. A JPSEC registration authority can ensure that the valid ID numbers are unique. 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 protector, the P_{SEC} parameter field shown in Table 2 shall be updated. A JPSEC protector may refer to the I_{max} parameter given in the P_{SEC} parameter field when determining what instance index to assign to a JPSEC tool that it is applying to the codestream.

5.4.2 JPSEC normative tool

The JPSEC normative tool uses the JPSEC tool syntax described in Section 5.4.1 and shown in Figure 7, where the tool type t=0 and the size of the ID is 8 bits. JPSEC normative tools are based on t he protection method templates described in Section 5.6. 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 5.

Table 5 — JPSEC normative tool Template ID values (ID_T)

Values	Protection method template
0	Reserved
1	Decryption template.
2	Authentication template.
3	Integrity template.
	All other values are reserved for IS O use

In the case of JPSEC normative tools, the parameter field P_{ID} has the structure shown in Figure 8. P_{ID} consists of four main fields: the template T, its processing domain PD, its granularity G , and its value list V. The syntax for each of these fields is given in Sections 5.6, 5.7, 5.8, and 5.9, respectively. Together, these fields describe how the tool is applied. The template parameters T describe the particular protection method for the decryption template, authentication template, or integrity template specified by the normative tool ID. 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 may be needed by each protection method w ith 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. Fo r the integrity 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.

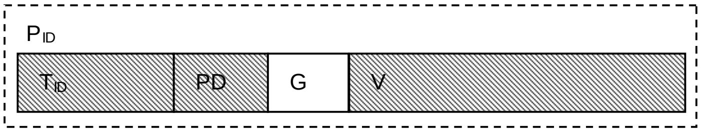


Figure 8 —Parameters (P_{ID}) syntax for JPSEC Normative Tools (t=0)

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.

Table 6 — JPSEC Template Tool parameter values

Parameter	Size (bits)	Values
T_{ID}	Variable	See Sec. 5.6.
PD	Variable	See Sec. 5.7.
G	24	See Sec. 5.8.
V	Variable	See Sec. 5.9.

5.4.3 JPSEC non-normative tool

In certain cases it may 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 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 uses the JPSEC tool syntax described in Section 5.4.1 and shown in Figure 7, where the tool type t=1 and the size of the ID_{RA} is 32 bits.

There are two classes of JPSEC non-normative tools:

1) JPSEC registration authority tools: JPSEC non-normative tools whose signalling is 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 are signalled using the 32-bit $\mathbf{ID_{RA}}$ identifier, where the identifiers whose first bit is a 0 are defined by a particular JPSEC application, and those whose first bit is a 1 are defined by a registration authority.

Note that the use of a registration authority to register each JPSEC non-normative tool supports the unique identification of each non-normative tool, thereby preventing collisions of IDs. However, without the use of a registration authority, ID collision can occur and should be carefully considered.

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 registration authority is used then the format is registered with the registration authority along with the ID. If a registration authority is not used and the tool is user-defined, 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 templates, processing domain, granularity, and value list fields described in Sections 5.6, 5.7, 5.8, and 5.9, 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.

5.5 Zone of Influence (ZOI) syntax

5.5.1 Introduction

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 bitstream segments or packet indices. In cases where image related parameters and non-image related 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 bitstream 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 9 illustrates a 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 International Standard defines two description classes: image related description class and non-image related description class. If multiple image related zones are used, then the ZOI is defined by their union. If a non-image related zone is also used, then the ZOI indicates that it corresponds to the image-related zone in the same ZOI. Each description class is associated with a set of parameters that specify the zone. 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.

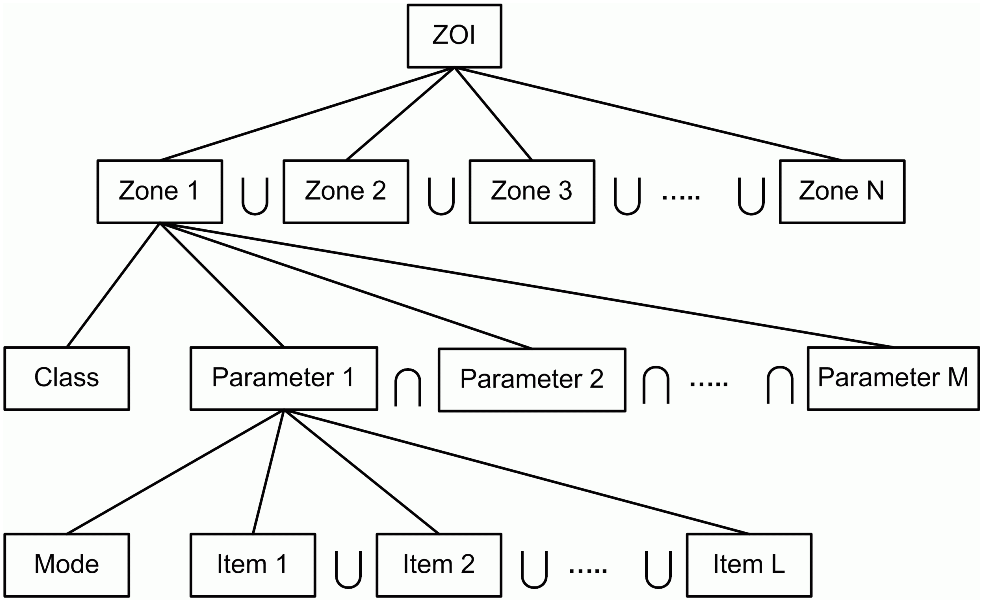

ZOI
Zone 1 ∪ Zone 2 ∪ Zone 3 ∪ ..... ∪ Zone N
Class
Parameter 1 ∩ Parameter 2 ∩ ..... ∩ Parameter M
Mode
Item 1 ∪ Item 2 ∪ ..... ∪ Item L

Figure 9 — Zone of Influence conceptual structure

5.5.2 Byte-aligned segment

In order to provide extensible signalling for classes and modes, this Internation al Standard uses a variable length data structure called a "byte -aligned segment" (BAS). As illustrated in Figure 10, the BAS is composed of a sequence of one or more BAS bytes. The most significant bit (MSB) of eac h 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 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 informa tion 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 .

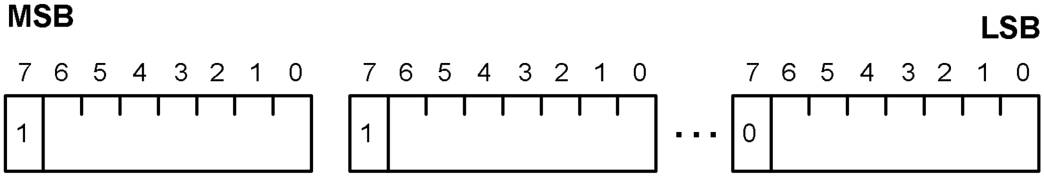


Figure 10 — Byte-aligned segment (BAS) structure

5.5.3 ZOI syntax

Figure 11 shows the ZOI syntax. The ZOI may contain one or more zones. It may also be empty, in which case NZzoi 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.

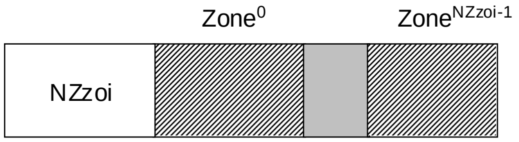


Figure 11 — ZOI syntax

NZzoi: Number of Zones.

Zone^k: Zone. Its structure is specified in 5.5.4

Table 7 — Zone of influence field (ZOI) parameter values

Parameter	Size (bits)	Values
NZzoi	8	0 — 254 255, reserved
Zone^k	Variable	See 5.5.4

5.5.4 Zone syntax

The Zone contains a zone description class field indicator followed by parameters of that class. The zone description class uses the BAS structure. As shown in Figure 12, the second most significant bit in each byte, labeled "x", flags the use of a specific description class. This International Standard defines two description classes: image related description class and a non-image related description class (See Table 9). Table 10 and Table 11 define the field indicator numbers for the image related description class and non-image related description class, respectively. By concatenating the six bits in each byte, labeled "y", which follows 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.

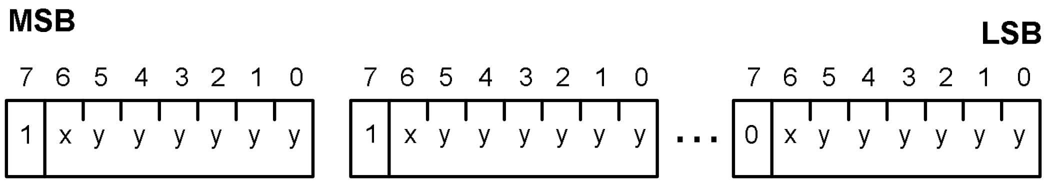


Figure 12 — Zone description class structure (DCzoi)

Figure 13 shows the Zone syntax.

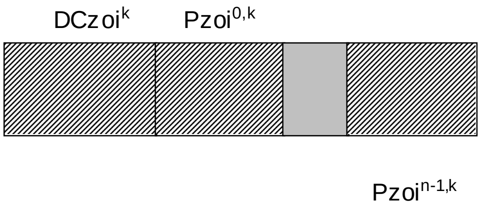


Figure 13 — Zone syntax consists of a description class and one or mor e parameter sets.

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

Pzoi^{i,k}: The Zone parameters according to the specified Zone description class (DCzoi^k). See 5.5.7.

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 a ppear in DCzoi^k.

Table 8 — Zone parameter values

Parameter	Size (bits)	Values
DCzoi^k	variable	Varies according to the value set in Table 9.
Pzoi^{i,k}	variable	See 5.5.7 for the syntax of this field

Table 9 — Description class indicator value

Value	Description class
0	Image related description class. The following bit numbers are defined in Table 10.
1	Non-image related description class. The following bit numbers are defined in Table 11.

Table 10 — Image related description class

Bit number	Semantics
1	Image region
2	Tile(s) as defined in JPEG 2000 Part 1
3	Resolution level(s) as defined in JPEG 2000 Part 1
4	Layer(s) as defined in JPEG 2000 Part 1
5	Component(s) as defined in JPEG 2000 Part 1
6	Precinct(s) as defined in JPEG 2000 Part 1
7	TRLCP tag(s)
8	Packet(s) as defined in JPEG 2000 Part 1

9	Sub-band(s) as defined in JPEG 2000 Part 1
10	Code-block(s) as defined in JPEG 2000 Part 1
11	ROI(s)
12	User-defined. The details shall be specified by other means. (e.g. JPSEC ID)
	All other values are reserv ed.

Packet indices are numbered sequentially within a tile, and therefore may not be unique across tiles. Furthermore, packet indices within a tile may roll over when their maximum value of 65536 is exceeded. For this reason, packet indexing is described in more detail. When the packet indices within a tile do not exceed 65536 packets, then the packet index described in Table 10 are defined by the packet index given by the SOP Nsop parameter as defined in Table A-40 in the JPEG-2000 part 1 standard. Note that when the maximum value does not exceed 65536, a single JPEG-2000 packet may be specified uniquely with a tile index and a packet index. When the packet indices exceed 65536 packets, then the JPEG -2000 part 1 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.

When the TRLCP tag is used, its format must 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 3. This defines the size of TRLCP tags in the ZOI.

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.

Table 11 — Non image related description class

Bit number	Semantics
1	Packet(s) as defined in JPEG 2000 Part 1
2	Byte range(s) (beginning first byte after the first SOD marker)
3	Byte range(s) (beginning at first byte after the first SEC marker)
4	Padded byte range (s)
5	TRLCP 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.

The non-image related description class may 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 Section 5.5.7) 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).

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 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. The format of the distortion field is described in Section 5.5.4.1.

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 may 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.

5.5.4.1 Distortion field and Relative importance field

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 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 14 and Table 12 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 10 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

$$D = M \text{ x } 10^{exp}$$

where M is given by

$$M = (m/16) \text{ x } 10$$

and M has a value in the range 0 <= M < 1. A distortion value of zero is represented by m = 0 and exp = 0, that is by the distortion field being zero. By allocating 4 bits for the mantissa m the accuracy is within ½ x (1/2^4) = 1/32 or about 3 %. With 4 bits for the exponent and using base-10 the dynamic range is from 0 to max, where max is given by m = 15 and exp = 15 which corresponds to a distortion of (15/16)x10 x 10^15 = 9.375 x 10^15.

exp	m

Figure 14 –Distortion field syntax

exp: Exponent of distortion field value (base 10)

m: Matissa of distortion field value

Table 12 —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.

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 a one-byte field which supports 256 possible rankings as shown in Figure 15 and Table 13. Increasing values correspond to increasing importance, in a similar manner to the distortion field.

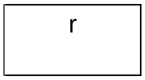


Figure 15 –Relative importance field syntax

r: Relative importance value

Table 13 —Relative importance field parameter values

Parameter	Size (bits)	Values
r	8	0 – 255

Since for both the distortion field and relative importance field the larger values correspond to greater importance, it is possible to make comparisons between two data units in the same manner irrespective of whether the distortion field specifies actual distortion or a relative importance.

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 may wish to 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 is a byte of all ones (0xFF). Note that this value is the highest possible distortion value for both the total squared error distortion field and for relative importance field.

5.5.5 Relationship between image related and non-image related description

As illustrated in Figure 9, the zone description class structure can have multiple fields set simultaneously, where N fields are image related descriptions (D_i^1, D_i^2, …, D_i^N) and M fields are non-image related descriptions (D_n^1, D_n^2, …, D_n^M). The semantics can be understood as $\{D_i^1 \bullet D_i^2 \bullet \ldots \bullet D_i^N\} = D_n^1 = D_n^2 = \ldots = D_n^M$, 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 the diagram below.

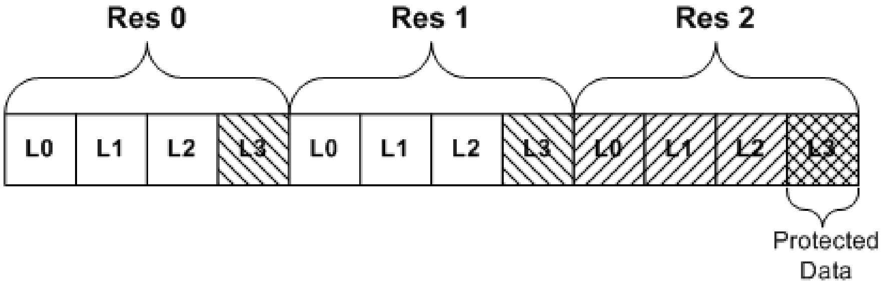


Figure 16 — ZOI example using image related descriptions

In the second example, the zone description has two image related description (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.

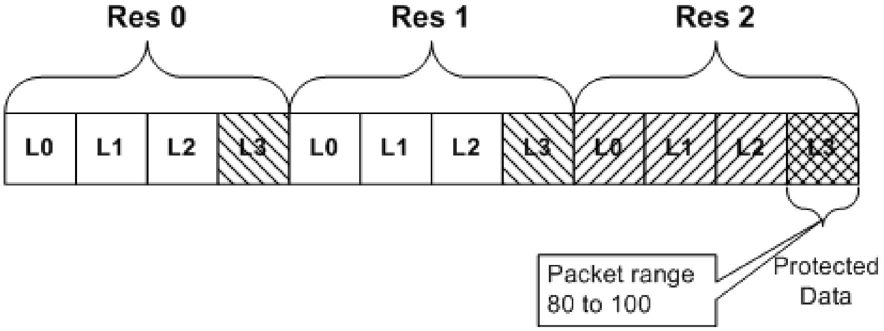


Figure 17 — ZOI example using image related and non-image related descriptions.

In the third example, the zone description has two image related description (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.

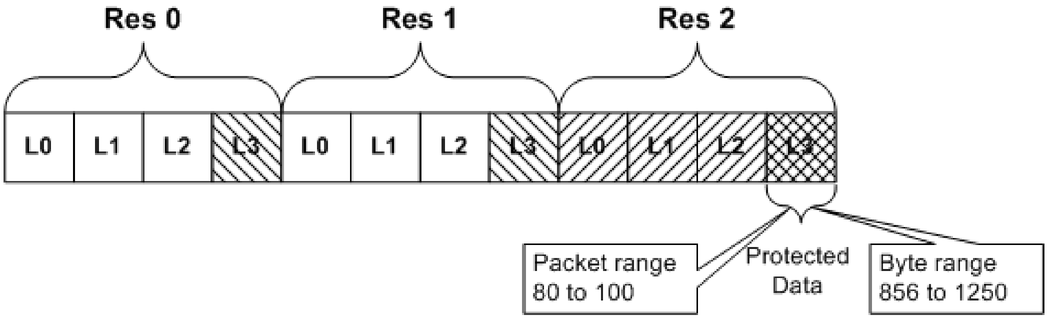


Figure 18 — A second ZOI example using image related and non -image related descriptions.

5.5.6 Protecting Any Data that Follows SEC Marker

The above discussion has largely focused on supporting protection services for the JPEG 2000 bitstream. 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 t his purpose.

Specifically, the byte range mode of the non -image related description class (Table 11) can be used to specify that a JPSEC tool should be applied to any data following the SEC marker. As described before, the first b yte of the SEC header is byte 1 for indexing the byte range. The data that follows the SEC marker and 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, m ay 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 must 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 decoder to first render the SEC segment data, which can then be used to process the remainder of the codestream.

5.5.7 Zone description parameter syntax (P_{zoi})

Figure 19 shows the ZOI description parameter syntax.

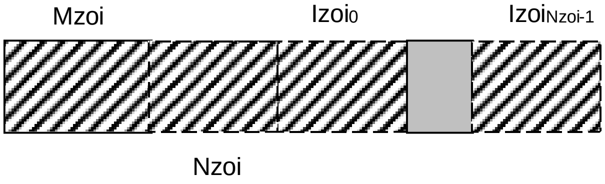


Figure 19 — ZOI description parameter syntax

Mzoi: ZOI description mode. This field uses the BAS structure.

Nzoi: Number of Izoi.

Izoi^i: Item.

Table 14 — Pzoi^i parameter values

Parameter	Size (bits)	Values
Mzoi	Variable	See Table 15
Nzoi	0 8	If bit number 2 of Mzoi is 0. 2 — 255
Izoi^i	Variable	Depends on the mode specified in Mzoi

Table 15 — Mzoi parameter values

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 s hall 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 JPEG 2000 Part 1..
	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 integers.
	01	Izoi^i uses 16 bit integers.
	10	Izoi^i uses 32 bit integer s.
	11	Izoi^i uses 64 bit integers.
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 1	Offset with lengths mode: Specifies the initial offset with lengths of contiguous bytes that follows. The existence of this flag shall override the modes specified in bit 3 and 4.
		All other values are reserved.

When TRLCP tags are used, their size is defined by P_{TRLCP} as specified in Table 3. In this case, bits 5 and 6 of the M_{ZOI} parameter are overridden.

5.6 Protection method template syntax (T)

5.6.1 General

Protection method templates contain parameters for specific JPSEC tools described in Section 5.4.1. For example, they are used in JPSEC normative tools described in Section 5.4.2. Also, they can be used in JPSEC non-normative tools described in Section 5.4.3. There are three types of protection method templates: decryption template, authentication template, and integrity template. The template used by a JSPEC normative tool is specified by its ID as shown in Table 5 and again here in Table 16 with references to the appropriate sections where they are defined.

As described in Section 5.4.2, 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 16 — Template ID values (ID_T)

Values	Protection method template
0	Reserved
1	Decryption template. See Sec 5.6.2.
2	Authentication template. See Sec 5.6.3.
3	Integrity template. See Sec 5.6.4.
	All other values are reserved for ISO use

5.6.2 Decryption template (T = T_{dcry}, if t=0 and ID=1)

The decryption template is used to communicate to the decryptor, how to decrypt the received bitstream. Table 17 shows the sizes and values of the symbols and parameters for the decryption template.

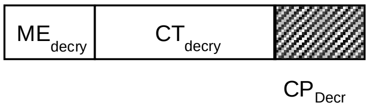


Figure 20 — Decryption template syntax

ME_{decry}: False marker emulation flag indicates whether a false marker emulation has occurred in the encrypted data. A false marker emulation may adversely affect compliance with JPEG 2000 Part 1 decoders.

CT_{decry}: Cipher type identification.

CP_{decry}: Cipher parameter.

Table 17 — Decryption template parameter values

Parameter	Size (bits)	Values
ME_{decry}	8	Table 18
CT_{decry}	16	Table 19
CP_{decry}	Variable	If CT_{decry} < 0x6000, see Sec 5.6.2.1. If 0x6000<=CT_{decry} <0xC000, see Sec 5.6.2.2. If CT_{decry} >=0xC000, see Sec 5.6.2.3.

Table 18 — Marker emulation flag values

Values	Method type
1	Marker emulation has not occurred.
0	Otherwise
	All other values are reserved for ISO use

The default value of the marker emulation flag is 0. This flag may be set to 1 to indicate that the JPSEC encrypted data does not contain a false marker emulation. A JPSEC creator may choose to leave this flag at its default value of 0.

Table 19 — Cipher identifier values

Values	Cipher type
0 – 0x5FFF	Block cipher (see Table 20)
0x6000 - 0xBFFF	Stream cipher (see Table 21)
0xC000 – 0xFFFF	Asymmetric cipher (see Table 22)

Table 20 — Block cipher identifier values

Values	Cipher type
0	NULL (no encryption)
1	IDEA (ISO/IEC 18033)
2	DES (ISO/IEC 18033)
3	AES (FIPS-197)
4	3DES
	All other values are reserved for ISO use

Table 21 — Stream cipher identifier values

Values	Cipher type
0x6000	RC4 (ISO/IEC 18033)
	All other values are reserved for ISO use

Table 22 — Asymmetric cipher identifier values

Values	Cipher type
0xC000	RSA (ISO/IEC 18033)
	All other values are reserved for ISO use

5.6.2.1 Block cipher template (CP_{decry} for block ciphers)

The block cipher template is used to communicate to the block decryptor, how to decrypt the received bitstream. Figure 21 shows the block cipher mode, padding mode, block size and key information .

Some block cipher modes can use initialization vectors. For these modes, the tool's initialization vectors are specified using the tool's granularity field (G) described in Section 5.8 and Value list field (V) described in Section 0. Specifically, initialization vectors are only used for modes with ID M_{bs}>0x80, for instance CBC, CFB, OFB, CTR. In the CTR case, it is not really an IV but a *counter*. The size of the initialization vector specified in the Value list V shall be set to the block size SIZ_{bs}.

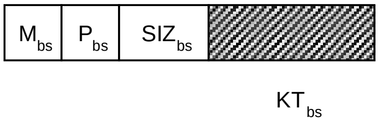


Figure 21 — Block cipher template syntax

M_{bs}: Block cipher mode. The first bit indicates the use of initialization vectors with this tool. If M_{bs} < 0x8, IVs are not used, otherwise one or more IV values are required for the mode.

P_{bs}: Padding mode.

SIZ_{bs}: Size of block in Bytes.

KT_{bs}: Key template (see section 5.6.5). It holds information on the keys used by the stream cipher.

Table 23 — Block cipher template values

Parameter	Size (bits)	Values
M_{bs}	6	Table 24 and Table 25
P_{bs}	2	Table 26
SIZ_{bs}	8	1-256
KT_{bs}	Variable	See Sec. 5.6.5

Table 24 — Block cipher mode values

Values	Cipher type
00 0000 – 01 1111	Modes which are used without IV (See Table 25)
10 0000 – 11 1111	Modes which are used with an IV (See Table 25)

Table 25 — Block cipher mode values

Values	Mode type
0	Reserved
x0 zzzz	Bits are not padded
x1 zzzz	Bits are padded
0y 0001	ECB (ISO/IEC 10116)
1y 0000	CBC (ISO/IEC 10116)
1y 0001	CFB (ISO/IEC 10116)
1y 0010	OFB (ISO/IEC 10116)
1y 0011	CTR (NIST SP 800-38A)
	All other values are reserved for ISO use

Table 26 --- Padding mode for block cipher

Values	Padding type
00	Ciphertext stealing (RFC 2040)
01	PKCS#7-padding (PKCS#7)
10	0x80*padding
11	All other values are reserved for ISO use

5.6.2.2 Stream cipher template (CP_{decry} for stream ciphers)

The stream cipher template is used to communicate to the stream decryptor, how to decrypt the received bitstream. Table 27 shows the values of the stream cipher template.

The stream cipher's initialization vectors are specified using the tool's granularity field (G) described in Section 5.8 and Value list field (V) described in Section 5.9. The size of the initialization vector specified in the Value list V shall be set to the key size defined in the key information template KT_{sc}.

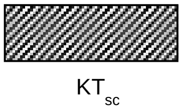


Figure 22 — Stream cipher template syntax

KT_{sc}: Key information template (see section 5.6.5). It holds information on the keys used by the stream cipher.

Table 27 — Stream cipher template values

Parameter	Size (bits)	Values
KT_{sc}	Variable	See Sec 5.6.5

5.6.2.3 Asymmetric cipher template (CP_{decry} for asymmetric ciphers)

The asymmetric cipher template is used to communicate to the asymmetric cipher decryptor, how to decrypt the received bitstream. Table 28 shows the values of the asymmetric cipher template.

For tools that use the asymmetric cipher template, the tool's granularity field (G) specifies the granularity with which the cipher is applied. However, the Value list field (V) is not used to represent any values. Thus, the number of elements in the Value list field shall be set to 0.

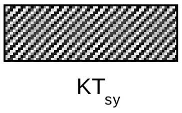


Figure 23 — Asymmetric cipher template syntax

KT_{sy}: Key information template (see section 5.6.5). It holds information on the keys used by the asymmetric cipher.

Table 28 — Asymmetric cipher template values

Parameter	Size (bits)	Values
KT_{sy}	Variable	See Sec. 5.6.5

5.6.3 Authentication template ($T = T_{auth}$, if t=0 and ID=2)

The authentication template is used to communicate to the verifier, how to verify the authenticity of received bitstream. There are three general classes of authentication methods: hash-based authentication, cipher-based authentication, and digital signatures. Both hash-based and cipher-based authentication methods are also generally referred to as message authentication codes (MACs), and their computed values which are used for authentication are generally referred as MAC values. The authentication template is shown in Figure 24, and Table 29 shows the sizes and values of the symbols and parameters for the authentication template.

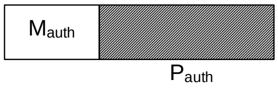


Figure 24 — Authentication template syntax

M_{auth}: Authentication method.

P_{auth}: Authentication parameters.

Table 29 — Authentication template parameter values

Parameter	Size (bits)	Values
M_{auth}	8	Table 30
P_{auth}	Variable	If M_{auth} =0, see Sec 5.6.3.1, If M_{auth} =1, see Sec 5.6.3.2, If M_{auth} =2, see Sec 5.6.3.3.

Table 30 Authentication Methods

Values	Method
0	Hash-based MAC
1	Cipher-based MAC
2	Digital Signature
	All other values are reserved for ISO use

5.6.3.1 Hash-based Authentication (P_{auth} for hash-based MAC)

The hash-based authentication MAC is used to communicate to the verifier how to verify the authenticity of the received bitstream. Figure 25 shows the hash-based authentication template and Table 31 shows the parameter values.

The MAC values are specified using the tool's granularity field (G) described in Section 5.8 and Value list field (V) described in Section 5.9. The size of the MAC value specified in the Value list V shall be set to the MAC size defined by SIZ_{MAC}.

M_{ID}	H_{ID}	KT	SIZ_{MAC}

Figure 25 Hash-based Authentication Template

M_{ID}: Hash-based authentication method identifier

H_{ID}: Hash identifier

KT: Key template.

SIZ_{MAC}: Size of MAC. (bits)

Table 31 Hash-based authentication template parameter values

Parameter	Size (bits)	Values
M_{ID}	8	Table 32
H_{ID}	8	Table 33
KT	variable	See Sec. 5.6.5
SIZ_{MAC}	16	0-65535

Table 32 Hash-based authentication method identifier

Values	Hash-based authentication method
0	Reserved
1	HMAC (RFC 2104)
	All other values are reserved for ISO use

Table 33 — Hash function identifier

Values	Hash function
0	Reserved
1	SHA-1 (ISO/IEC 10118)
2	RIPEMD-128 (ISO/IEC 10118)
3	RIPEMD-160 (ISO/IEC 10118)
4	MASH-1 (ISO/IEC 10118)
5	RIPEMD-2 (ISO/IEC 10118)
6	MD5 (RFC1321)
7	SHA-256 (FIPS 180-2)
8	SHA-384 (FIPS 180-2)
	All other values are reserved for ISO use

Note that if the SIZ_{MAC} is less than the nominal size of the hash, then it is the truncated version corresponding to the first SIZ_{MAC} bits of the hash.

5.6.3.2 Cipher-based Authentication Template (P_{auth} for cipher-based MAC)

The cipher-based authentication MAC is used to communicate to the verifier, how to verify the authenticity of the received bitstream. Figure 26 is its template and Table 34 shows the key size and keyed hash. An example cipher based authentication scheme is CBC-MAC. In these block-cipher techniques for authentication, the initialization vector is one blocksize in l ength and of value 0. The blocksize is the default for the block cipher. Note that if the SIZ_{MAC} is less than the nominal size of the cipher-based authentication MAC, then it is the truncated version corresponding to the first SIZ_{MAC} bits of the MAC.

Note that, if the number of data bits is not a multiple of (64) <a cipher block size>, then the final input block will be a partial block of data, left justified, with zeroes appended to form a full (64-bit) <cipher> block.

The MAC values are specified using the tool's granularity field (G) described in Section 5.8 and Value list field (V) described in Section 5.9. The size of the MAC value specified in the Value list V shall be set to the MAC size defined by SIZ_{MAC}.

[NOTE: This is a description which is generalized from FIPS PUB 113. This description should be widely reviewed by security experts. In the previous sentence, ( ) is original description and [ ] is modified description.]

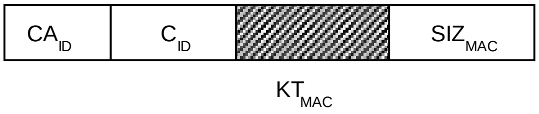


Figure 26 — Cipher-based authentication template syntax

CA_{ID}: Cipher-based authentication method

C_{ID}: Block-cipher identifier value

KT_{MAC}: Key template

SIZ_{MAC}: Size of MAC. (bits)

Table 34 — MAC template values

Parameter	Size (bits)	Values
CA_{ID}	8	Table 35
C_{ID}	8	Table 20
KT_{MAC}	Variable	See Sec. 5.6.5
SIZ_{MAC}	16	0-65535

Table 35 Cipher-based Authentication Method

Values	Method
0	CBC-MAC (FIPS PUB 113)
1	
	All other values are reserved for ISO use

5.6.3.3 Digital Signature template (P_{auth} for digital signatures)

The digital signature is used to communicate to the verifier, how to verify the authenticity of received bitstream, as well as verifying the identity of the sender for both identity and non-repudiation purposes. Figure 27 defines its template and Table 36 lists the values.

The digital signatures are specified using the tool's granularity field (G) described in Section 5.8 and Value list field (V) described in Section 5.9. The size of the digital signatures value specified in the Value list V shall be set to accommodate the size defined by SIZ_{ds}. Because the Value list size is represented by bytes rather than bits, its size should be the minimum numer of bytes that can accommodate SIZ_{ds}. Each value should be represented with the least significant bits, and the extra MSB bits shall be set to 0.

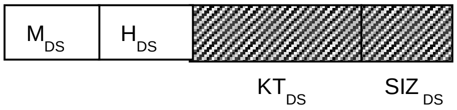


Figure 27 — Digital Signature template syntax

M_{DS}: Digital Signature Method

H_{DS}: Hash function

KT_{DS}: Key template (see section 5.6.5). It holds all the information related to the public key or the certificate required to verify the digital signature.

SIZ_{DS}: Size of digital signature. (bits)

Table 36 — Digital Signature template values

Parameter	Size (bits)	Values
M_{DS}	8	Table 37
H_{DS}	8	Table 33
KT_{DS}	Variable	See Sec. 5.6.5
SIZ_{DS}	16	0-65535

Table 37 Digital Signature Methods

Values	Method
1	RSA (ISO/IEC 9796)
2	Rabin (ISO/IEC 9796)
3	DSA (FIPS 186)
4	ECDSA (FIPS 186-2)
	All other values are reserved for ISO use

5.6.4 Integrity template (T = T_{inte}, if t=0 and ID=3)

The integrity template is used to communicate to the verifier, how to verify the integrity of received content bitstream. Table 38 shows the sizes and values of the symbols and parameters for integrity template.

Note that in contrast to the authentication template discussed in Section 5.6.3 which involves the use of a secret key, this integrity check template provides a key-less integrity check. As a result, this integrity check can be used to detect an accidental error or accidental change to the data ; however, it does not prevent malicious alteration of the data. In order to prevent malicious alteration of the data an authentication template must be used, since the secret key used by the authentication template prevents the data from being altered without being discovered.

The hash values are specified using the tool's granularity field (G) described in Section 5.8 and Value list field (V) described in Section 5.9. The size of the hash value specified in the Value list V shall be set to the hash value size defined by SIZ_{inte}.

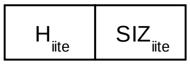


Figure 28 — Integrity template syntax

H_{inte}: Hash function identifier.

SIZ_{inte}: Size of hash value (bytes).

Table 38 — Integrity template parameter values

Parameter	Size (bits)	Values
H_{inte}	8	Table 33
SIZ_{inte}	8	1-256

5.6.5 Key Information template (KT)

The key information template is used to communicate key information. Figure 29 defines its template and Table 36 lists the values.

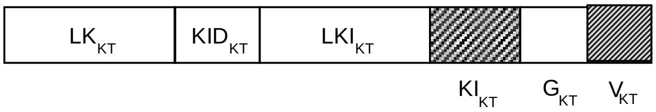


Figure 29 — Key information template syntax

LK_{KT}: Length of key in bits.

KID_{KT}: Key information identifier. It indicates the meaning of **KI_{KT}**. In the decryption template this value should be set to 3 (URI to retrieve the secret key) or 4 (the wrapped key). In the case of digital signature, the value of this field is free.

LKI_{KT}: Length of key information in Bytes.

KI_{KT}: Key information. The verifier can obtain a public key directly if KID=1, or from a certificate if KID=2. Otherwise it can obtain a location information from an URI, if KID=3. Furthermore, it can obtain a secret key by decrypting a wrapped key if KID=4.

G_{KT}: Granularity field to represent the granularity with which the key information changes.

V_{KT}: Value list field to represent the changing list of key information .

Note that in the case of a secret key (decryption template), the public key and certificate have no meanings: the key template should hold some information on the location of the key (e.g. URI).

The key information can be represented with a single value or updated with multiple values using the tool's granularity field (G) described in Section 5.8 and Value list field (V) described in Section 5.9. It has a similar usage in that if the value of N_V is zero, then the field is terminated and the key value given in KI_{KT} is used. If N_V is non-zero, then the key values and their granularities are specified by this granularity -value field.

Table 39 — Key template values

Parameter	Size (bits)	Values
LK_{KT}	16	1-65535
KID_{KT}	8	Table 40
LKI_{KT}	16	1-65535
KI_{KT}	Variable	Table 41: if $KID_{KT} = 2$ Table 43: if $KID_{KT} = 4$ Open : Otherwise
G_{KT}	24	See Sec 5.8
V_{KT}	Variable	See Sec 0

Table 40 — Key information identifier values (KID_{KT})

Values	Key information identifier
0	Reserved
1	Public key
2	X.509 Certificate (ISO/IEC 9594-8)
3	URI for certificate or secret key
4	Wrapped key
	All other values are reserved for ISO use

5.6.5.1 X.509 Certificate template

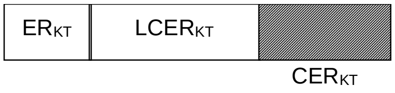


Figure 30 — X.509 Certificate syntax

ER_{KT}: Encoding rule for X.509 Certificate

$LCER_{KT}$: Length of X.509 Certificate (CER_{KT}) in bytes.

CER_{KT}: X.509 Certificate.

Table 41 — X.509 Certificate values (KI_{KT} if KI_{KT}=2)

Parameter	Size (bits)	Values
ER_{KT}	8	0-255 (see Table 42)
$LCER_{KT}$	16	1-65535
CER_{KT}	Variable	-

Table 42 — Encoding rule values (ER_{KT})

Values	Encoding rule identifier
0	Reserved
1	DER (RFC3217)
2	BER (RFC3394)
	All other values are reserved for ISO use

5.6.5.2 Wrapped key template

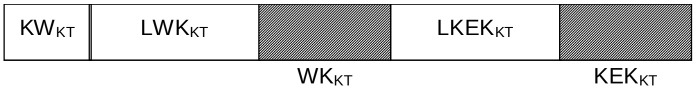


Figure 31 — Wrapped key syntax

KW_{KT}: ID for key wapping algorithm.

LWK_{KT}: Length of wapped key (WK_{KT}) in bytes.

WK_{KT}: Wapped key (i.e. key encrypted by KEK_{KT}). Encrypted codestream can be decrypted using the WK_{KT} decrypted with KEK_{KT}.

$LKEK_{KT}$: Length of key encrypting key (KEK_{KT}) in bytes.

KEK_{KT}: Key information for encrypting key. (e.g. URI)

Table 43 — Wrapped key values (KI_{KT} if KI_{KT}=4)

Parameter	Size (bits)	Values
KW_{KT}	8	0-255 (see Table 44)
LWK_{KT}	16	1-65535
WK_{KT}	Variable	-
$LKEK_{KT}$	16	1-65535
KEK_{KT}	Variable	-

Table 44 — Key wrapping algorithm identifier values (KW_{KT})

Values	Key wrapping alrogithm identifier
0	Reserved
1	Triple-DES and RC2 Key Wrapping (RFC3217)
2	AES Key Wrapping (RFC3394)
	All other values are reserved for ISO use

5.7 Processing Domain syntax (PD)

Processing Domain (PD) is used to indicate at which domain each protection method is used. There are three possible PDs: wavelet coefficient domain, quantized wavelet coefficient domain, and bitstream domain.

PD uses the BAS structure. As shown in Figure 32, the second and third most significant bits in the first byte, labeled "x", flags the specific PD, as shown in Table 46. By concatenating the remaining bits except for the most significant bit in each byte, labeled "y", are used to signal additional information for the specified PD.

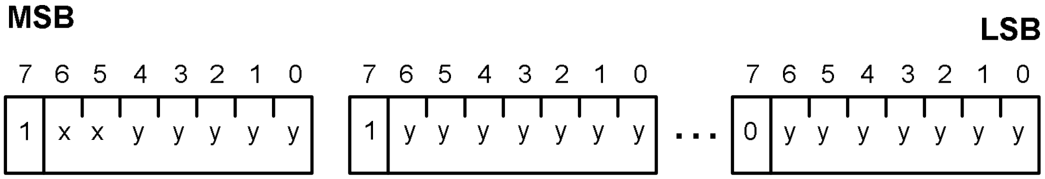


Figure 32 — Processing Domain structure

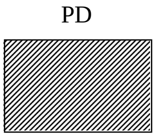


Figure 33 — Processing Domain syntax

PD: Processing Domain. This field uses the BAS structure.

Table 45 — PD parameter value

Parameter	Size (bits)	Values
PD	variable	See Table 46

Table 46 — Processing Domain (PD) parameter values

Values	Processing Domain
MSB LSB	
00	Wavelet coefficient domain. Protection method is applied on wavelet coefficients.
01	Quantized wavelet coefficient domain: Protection method applied on quantized wavelet coefficient
10	Codestream domain: Protection method is applied on codestream generated from arithmetic coder.
11	Reserved by ISO

Table 47 — Processing domains for BAS

Bit number	Semantics
1	Sign bit plane
2	Most significant bitplane
3	Packet bodies only
	All other values are reserved.

The sign bit plane and most significant bitplane flags only apply when the processing domain is set to wavelet coefficient domain or the quantized wavelet coefficient domain.

The packet bodies only flag only applies when the processing domain is set to the codestream domain. If the processing is in the codestream domain and all the BAS parameters are set to zero, then the default behaviour is to encrypt packet headers and packet bodies.

5.8 Granularity syntax (G)

Granularity is used to indicate the unit of protection for each protection method. Table 50 defines possible granularities. Figure 34 shows the Granularity syntax.

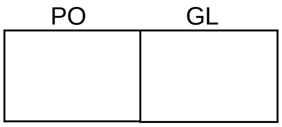


Figure 34 — Granularity syntax

PO: Processing order

GL: Granularity level

Table 48 — Granularity parameter values (G)

Parameter	Size (bits)	Values
PO	16	See Table 49
GL	8	See Table 50

Table 49 — Processing order values (PO)

Values MSB LSB	Granularity
000 001 010 011 100 0	Processing order is tile -resolution-layer-component-precinct
000 011 100 001 010 0	Processing order is tile -component-precinct-resolution-layer
000 010 001 011 100 0	Processing order is tile -layer-resolution-component-precinct
000 100 011 001 010 0	Processing order is tile -precinct-component- resolution-layer
000 001 100 011 100 0	Processing order is tile -resolution-precinct-component-layer
	All other values are reserved

Table 50 — Granularity parameter (GL)

Values MSB LSB	Granularity
0000 1111	Unit of protection is the entire codestream
0000 0000	Unit of protection is tile.
0000 0001	Unit of protection is tile -part.
0000 0010	Unit of protection is component.
0000 0011	Unit of protection is resolution level.
0000 0100	Unit of protection is layer.
0000 0101	Unit of protection is precinct.
0000 0110	Unit of protection is packet .
0000 0111	Unit of protection is sub -band.
0000 1000	Unit of protection is code -block.
0000 1001	Unit of protection is Zone identified in ZOI
	All other values are reserved.

5.9 Value List syntax (V)

The Value list field is used to specify values that c hange as the tool is applied and the granularity with which it changes. This is used to signal changing values such as keys, initialization vectors, MAC values, digital signatures, and hash values. The Value List field first specifies the number of values in the list and the size of each value. It then lists the values themselves.

As discussed in Section 5.4.2, for JPSEC normative tools the Value list field represents a different parameter for each template. For the decryption template, it represents the initialization vectors IV_{bs} or IV_{sc} depending on whether a block cipher or stream cipher are used. For the authentication template it represents the MAC value VAL_{MAC} for hash-based and cipher-based authentication. For the digital signature template, it represents the digital signature SIG_{DS}. For the integrity template it represents the hash value HV_{inte}. Some usages of the templates do not require values to be specified, e.g., not all decryption modes use initialization vectors. In these cases, the Value list field should set N_v and S_v equal to zero so that the value list VL has no elements. If only a single value needs to be specified, e.g., if a single key is used throughout the image, then N_v will be set to one so that a single value is contained in the value list.

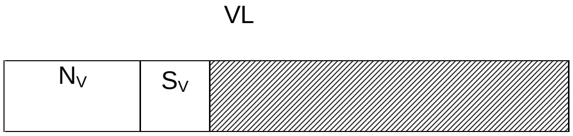


Figure 35 –Value list field syntax

N_V: Number of values in the value list V, If N_V=0, then the field terminates

S_V: Size of each value in V in bytes

VL: List of values

Table 51 —Value list field (V) parameter values

Parameter	Size (bits)	Values
N_V	16	$0 – (2^{16}-1)$
S_V	8	$1 – (2^{8}-1)$
VL	0, if N_V=0 $N_V * S_V$	N/A Determined by template

5.10 In-codestream security marker (INSEC)

The in-codestream security marker (INSEC) provides an additional means to transmit security information. It is optional and is used in conjunction with the SEC security marker. Specifically, it is used in conjunction with a JPSEC non-normative tool.

More precisely, the SEC marker is present in the Main Header and gives overall information about the JPSEC tools applied to protect the image. The INSEC marker is present in the bitstream data itself and gives additional or alternative parameters for the JPSEC non-normative tool identified by the tool instance index parameter. Therefore, the tool instance index in the INSEC marker shall correspond to one of the tool instance index in the Main Header.

The INSEC marker segment can be placed in the bitstream data. It uses the fact that the arithmetic decoder in JPEG 2000 stops reading bytes from the bitstream when it encounters a termination marker (i.e. two bytes with a value greater than 0xFF8F).

The information carried in the INSEC marker segment is relevant for the preceding or following secured codeblock(s), until another INSEC marker is found.

Note that inclusion of INSEC markers results in a file that may not comply with JPEG 2000 Part 1. Note that some decoders may have difficulty handling a marker in the middle of a packet. Insertion anywhere inside a packet will invalidate the length of the packet as indicated in the packet header. Also, there may be issues with encryption and INSEC markers due to a) lack of marker emulation restrictions on the encryption, and/or b) inability to locate the marker itself in the presence of encryption.

[Note: This section is impacted by definitions of Part 1 compliance and Decoder behaviour. Part 1 defines that the arithmetic decoder shall terminate when facing an unexected marker (FF90 -FFFF). However, Part 1 does not specify a Part 1 compliant decoder's behaviour after such an event. This section should be edited after considering Part 1 compliance and decoder behaviour issues. When addressing these issues, consider the following points: Note that a carefully designed error-resilient decoder should resynchronize and continue decoding without crashing. This discussion may be moved to a section addressing Part 1 compliance and decoder behaviour issues.]

The syntax of the INSEC marker is defined in Figure 36.

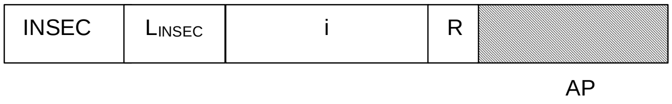


Figure 36 — In-codestream security syntax

INSEC: Marker code. Table 52 shows the sizes and values of the symbols and parameters for in-codestream security marker segment.

L_{INSEC}: Length of marker segment in bytes (not including the marker). Note that the INSEC marker segment should be byte-aligned.

i: Tool instance index corresponding to one of the tool instance index parameters in the SEC marker segment and therefore identifying the instance of the JPSEC tool this INSEC marker is referring to.

R: Relevance zone for the INSEC information.

AP: Additional or alternative parameters for protection method. The encoder should always make sure that the encoder does not emulate a marker in thi s parameter.

Table 52 — In-codestream security parameter values (INSEC)

Parameter	Size (bits)	Values
INSEC	16	0xFF94
L_{INSEC}	16	$2 – (2^{16}-1)$
i	7	$0 – (2^{7}-1)$
R	1	see Table 53
AP	Variable	Defined by registr ation authority or application.

Table 53 — Relevance zone values (R)

Values	Relevance zone
0	preceding code-blocks
1	following code-blocks

Because INSEC is used in conjunction with JPSEC non-normative tools, the format of the additional or alternative parameters is defined by the tool itself which is identified by the tool ID. Specifically, JPSEC non-normative tools are defined by a registration authority or by private JPSEC applications. Thus, the definition of these tools should include the INSEC usage if it is allowed.

6 Normative-Syntax Usage Examples (informative)

6.1 ZOI examples

This section contains examples that show of how the Zone of Influence syntax can be used.

In the examples that follow, the superscripts used in Pzoi, Mzoi, and Izoi correspond to the index of the image-related and non-image-related items signalled by the BAS structure in DCzoi in the order they appear within the DCzoi.

6.1.1 Example 1

This section shows the example that resolution levels more than 3 in the image region whose upper-left corner is (100, 120) and lower-right (180, 210) are influenced. In this example, 9 bytes are necessary.

Table 54 — ZOI in example 1

Parameter			Value (in order)	Size (bits)	Derived meaning
NZzoi			1	8	Number of Zones is one.
Zone^0	DCzoi		0	1	The byte aligned-segment does not follow.
			0	1	Image related description class.
			101000	6	Image regions and resolution levels are specified in order.
	Pzoi^1	Mzoi^1	0	1	The byte aligned-segment does not follow.
			0	1	The specified zones are influenced by the JPSEC tool.
			0	1	Single item is specified.
			00	2	Rectangle mode.
			00	2	Izoi uses 8 bits integer.
			1	1	Izoi is described in two dimensions.
		Izoi^1	0110 0100	8	Xul is 100.
			0111 1000	8	Yul is 120.
			1011 0100	8	Xlr is 180.
			1101 0010	8	Ylr is 210.
	Pzoi^3	Mzoi^3	0	1	The byte aligned-segment does not follow.
			1	1	The specified zones are not influenced by the JPSEC tool.
			0	1	Single item is specified.
			11	2	Max mode.
			00	2	Izoi uses 8 bits integer.
			0	1	Izoi is described in one dimension.

		Izoi^3	0000 0010	8	Resolution levels ≤ 2 are specified. (i.e. Resolution levels > 3 are specified with Max mode and complement switch.).

6.1.2 Example 2

This section shows the example that the code-blocks whose upper-left corner's index 5 and lower-right corner's index is 10 in the subband 1, in the resolution level 0 are influenced. In this example, 10 bytes are necessary.

Table 55 ZOI in example 2

Parameter			Value (in order)	Size (bits)	Derived meaning
NZzoi			1	8	Number of Zones is one.
Zone^0	DCzoi^1		1	1	The byte aligned-segment follows.
			0	1	Image related description class.
			001000	6	Resolution levels are specified.
	DCzoi^2		0	1	The byte aligned-segment does not follow.
			0	1	Image related description class.
			011000	6	Sub-bands and code-blocks are specified.
	Pzoi^3	Mzoi^3	0	1	The byte aligned-segment does not follow.
			0	1	The specified zones are influenced by the JPSEC tool.
			0	1	Single item is specified.
			10	2	Index mode.
			00	2	Izoi uses 8 bits integer.
			0	1	Izoi is described in one dimension.
		Izoi^3	0000 0000	8	Resolution level index is 0.
	Pzoi^8	Mzoi^8	0	1	The byte aligned-segment does not follow.
			0	1	The specified zones are influenced by the JPSEC tool.
			0	1	Single item is specified.
			10	2	Index mode.
			00	2	Izoi uses 8 bits integer.
			0	1	Izoi is described in one dimensions.
		Izoi^8	0000 0001	8	Subband 1 is specified.
	Pzoi^9	Mzoi^9	0	1	The byte aligned-segment does not follow.
			0	1	The specified zones are influenced by the JPSEC tool.
			0	1	Single item is specified.
			00	2	Rectangle mode.
			00	2	Izoi uses 8 bits integer.
			0	1	Izoi is described in one dimensions.
		Izoi^9	0001 0010	8	Code-block index for the upper-left corner is 18.
			0010 1101	8	Code-block index for the lower-right corner is 45.

6.1.3 Example 3

This section shows the example that the data segments from bytes 10 to 100 and from bytes 10000 to 12000 are influenced. In this example, 12 bytes are necessary.

Table 56 — ZOI in example 3

Parameter			Value (in order)	Size (bits)	Derived meaning
NZzoi			1	8	Number of Zones is one.
Zone^0	DCzoi		0	1	The byte aligned-segment does not follow.
			1	1	Non-image related description class.
			010000	6	Byte ranges after the SOD marker are specified.
	Pzoi^2	Mzoi^2	0	1	The byte aligned-segment does not follow.
			0	1	The specified zones are influenced by the JPSEC tool.
			1	1	Multiple items are specified.
			01	2	Range mode.
			01	2	Izoi uses 16 bits integer.
			0	1	Izoi is described in one dimensions.
		Nzoi^2	0000 0010	8	Number of data segments is 2.
		Izoi^{21}	0000 0000 0000 1010	16	Starting byte location is 10th (bytes).
			0000 0000 0110 0100	16	Ending byte location is 100th (bytes).
		Izoi^{21}	0010 0111 0001 0000	16	Starting byte location is 10000th (bytes).
			0010 1110 1110 0000	16	Ending byte location is 12000th (bytes).

6.1.4 Example 4

This section shows the example that resolution level 0 is influenced and that the byte segment 10 through 100 correspond to the data for resolution level 0 . In this example, 10 bytes are necessary.

Table 57 — ZOI in example 4

Parameter		Value (in order)	Size (bits)	Derived meaning
NZzoi		1	8	Number of Zones is one.
Zone^0	DC$_{zoi}^{1}$	1	1	The byte aligned-segment follows.
		0	1	Image related description class.
		001000	6	Resolution levels are specified in order.
	DC$_{zoi}^{2}$	0	1	The byte aligned-segment does not follow.

			1	1	Non-image related description class.
			010000	6	Byte ranges are specified.
	Pzoi^1	Mzoi^1	0	1	The byte aligned-segment does not follow.
			0	1	The specified zones are influenced by the JPSEC tool.
			0	1	Single item is specified.
			10	2	Index mode.
			00	2	Izoi uses 8 bits integer.
			0	1	Izoi is described in one dimensions.
		Izoi^1	0000 0000	8	Resolution level is 0 .
	Pzoi^2	Mzoi^2	0	1	The byte aligned-segment does not follow.
			0	1	The specified zones are influenced by the JPSEC tool.
			0	1	Single items specified.
			01	2	Range mode.
			01	2	Izoi uses 16 bits integer.
			0	1	Izoi is described in one dimensions.
		Izoi^1	0000 0000 0000 1010	16	Starting byte location is 10th (bytes).
			0000 0000 0110 0100	16	Ending byte location is 100th (bytes).

6.1.5 Example 5

This section shows the example that resolution levels more than 3 in the tiles whose upper-left tile index is 0 and lower-right tile index is 5, and layers equal to or less than 5 in the tiles whose upper -left tile index is 10 and lower-right tile index is 15 are influenced. In this example, 13 bytes are necessary.

Table 58 — ZOI in example 5

Parameter			Value	Size	Derived meaning
			(in order)	(bits)	
NZzoi			2	8	Number of Zones is two.
Zone^0	DCzoi		0	1	The byte aligned-segment does not follow.
			0	1	Image related description class.
			011000	6	Tiles and resolution levels are specified in order.
	Pzoi^2	Mzoi^2	0	1	The byte aligned-segment does not follow.
			0	1	The specified zones are influenced by the JPSEC tool.
			0	1	Single item is specified.
			00	2	Rectangle mode.
			00	2	Izoi uses 8 bits integer.
			0	1	Izoi is described in one dimension.
		Izoi^2	0000 0000	8	Upper-left tile index is 0.
			0000 0101	8	Lower-right tile index is 5.
	Pzoi^3	Mzoi^3	0	1	The byte aligned-segment does not follow.

			1	1	The specified zones are not influenced by the JPSEC tool.
			0	1	Single item is specified.
			11	2	Max mode.
			00	2	Izoi uses 8 bits integer.
			0	1	Izoi is described in one dimension.
		Izoi^3	0000 0010	8	Resolution levels ≤ 2 are specified. (i.e. Resolution levels > 3 are specified with Max mode and complement switch.).
Zone^1	DCzoi		0	1	The byte aligned-segment does not follow.
			0	1	Image related description class.
			010100	6	Tiles and layers are specified in order.
	Pzoi^2	Mzoi^2	0	1	The byte aligned-segment does not follow.
			0	1	The specified zones are influenced by the JPSEC tool.
			0	1	Single item is specified.
			00	2	Rectangle mode.
			00	2	Izoi uses 8 bits integer.
			0	1	Izoi is described in one dimension.
		Izoi^2	0000 1010	8	Upper-left tile index is 10.
			0000 1111	8	Lower-right tile index is 15.
	Pzoi^4	Mzoi^4	0	1	The byte aligned-segment does not follow.
			0	1	The specified zones are influenced by the JPSEC tool.
			0	1	Single item is specified.
			11	2	Max mode.
			00	2	Izoi uses 8 bits integer.
			0	1	Izoi is described in one dimension.
		Izoi^4	0000 0101	8	layers ≤ 5 are specified with Max mode.

6.1.6 Example 6

This section shows the example that the header segment from byte 10 to 100 is influenced. In this example, 12 bytes are necessary.

Table 59 — ZOI in example 6

Parameter			Value (in order)	Size (bits)	Derived meaning
NZzoi			1	8	Number of Zones is one.
Zone^0	DCzoi		0	1	The byte aligned-segment does not follow.
			1	1	Non-image related description class.
			001000	6	Byte ranges after the SEC marker are specified.
	Pzoi^3	Mzoi^3	0	1	The byte aligned-segment does not follow.
			0	1	The specified zones are influenced by the JPSEC tool.

			0	1	Single item is specified.
			01	2	Range mode.
			01	2	Izoi uses 16 bits integer.
			0	1	Izoi is described in one dimension.
		Nzoi[3]	0000 0001	8	Number of data segments is 1.
		Izoi[3]	0000 0000 0000 1010	16	Starting byte location is 10th (bytes).
			0000 0000 0110 0100	16	Ending byte location is 100th (bytes).

6.2 Key information template examples

6.2.1 Example 1

Table 60 shows the example that a single secret key (128 bits) is used to decrypt a codestream, where the secret key is identified using URI and retrieved from the key server based on UR I in the decryption stage.

Table 60 — Key information in example 1

Parameter		Size (bits)	Value	Derived meaning
LK_{KT}		16	128	Length of key is 128 bits.
KID_{KT}		8	3	URI for secret key is identified.
LKI_{KT}		16	19	Length of KI_{KT} is 16 bytes.
KI_{KT}		152	https://server/file	Secret key can be retviered from https://server/file
G_{KT}	PO	16	000 001 010 011 100 0	Processing order is tile-resolution-layer-component-precinct
	GL	8	0000 1111	Unit of protection is entire codestream.
V_{KT}	N_V	16	0	Number of values in the value li st V is 0.
	S_V	8	0	

6.2.2 Example 2

Table 61 shows that the example that a X.509 certificate is used to authenticate a codestream, where the X.509 certificate is embedded into KI_{KT} with encoding method DER.

Table 61 — Key information in example 2

Parameter		Size (bits)	Value	Derived meaning
LK_{KT}		16	1024	Length of key is 1024 bits.
KID_{KT}		8	2	X.509 Certificate is identified.
LKI_{KT}		16	Variable	Length of KI_{KT} is
KT_{KT}	ER_{KT}	8	1	X.509 Certificate is encoded with encoding method DER.
	$LCER_{KT}$	16	Varialbe	Length of CER_{KT} is

	CER_{KT}	Variable	*Certificate value*	Certificate with 1024 bit public key is embedded.
G_{KT}	PO	16	000 001 010 011 100 0	Processing order is tile-resolution-layer-component-precinct
	GL	8	0000 1111	Unit of protection is entire codestream.
V_{KT}	N_V	16	0	Number of values in the value list V is 0.
	S_V	8	0	

6.2.3 Example 3

Table 62 shows that a single public key is used to authenticate a codestream, where the public key is embedded into KI_{KT}.

Table 62 — Key information in example 3

Parameter		Size (bits)	Value	Derived meaning
LK_{KT}		16	1024	Length of key is 1024 bits.
KID_{KT}		8	1	Public key is identified.
LKI_{KT}		16	2048	Length of KI_{KT} is 2048 bits.
KI_{KT}		2048	*Public key value*	Public key is embedded.
G_{KT}	PO	16	000 001 010 011 100 0	Processing order is tile-resolution-layer-component-precinct
	GL	8	0000 1111	Unit of protection is entire codestream.
V_{KT}	N_V	16	0	Number of values in the value list V is 0.
	S_V	8	0	

6.2.4 Example 4

Table 63 shows that multiple secret keys are used to decrypt a codestream, where different secret keys are used for different layers.

Table 63 — Key information in example 4

Parameter		Size (bits)	Value	Derived meaning
LK_{KT}		16	128	Length of key is 128 bits.
KID_{KT}		8	3	URI for secret key is identified.
LKI_{KT}		16	19	Length of KI_{KT} is 20 bytes.
KI_{KT}		128	https://server/1	Secret key for the 1^{st} layer can be retviered from https://server/1.
G_{KT}	PO	16	000 001 010 011 100 0	Processing order is tile-resolution-layer-component-precinct
	GL	8	0000 0100	Unit of protection is layer.
V_{KT}	N_V	16	3	Number of values in the value list V is 3.
	S_V	8	19	Length of each Vn is 19 bytes.

	V1	128	https://server/2	Secret key for the 2nd layer can be retrieved from https://server/2.
	V2	128	https://server/3	Secret key for the 3rd layer can be retrieved from https://server/3.
	V3	128	https://server/4	Secret key for the 4th layer can be retrieved from https://server/4.

6.3 JPSEC normative tool examples

The following examples describe how the ZOI and key templates can be used to perform basic security services such as encryption and authentication on a JPEG -2000 coded image.

6.3.1 Example 1

An image is coded with JPEG-2000 and has three resolutions. In this example the first resolution is not encrypted in order to provide preview capabilitiy, and the second and third resolutions are encrypted with keys k1 and k2, respectively. The input image in this case is coded in RLCP progression order, and has 1 tile, 3 resolutions, 3 layers, Nc components, and Np precincts (the number of components and precints is not significant in this specific example). Encryption is performed using AES in CBC mode without padding (using cipher-text stealing), using key k0 to encrypt resolution 1 and using key k2 to encrypt resolution 2, and resolution 0 is left unencrypted.

JPSEC signals how a JPSEC consumer should decrypt the JPSEC codestream. First, the tool template ID for the decryption template is signaled. Two ZOI's are specified for resolution 1 and its corresponding byte range B0-B1, and for resolution 2 and its corresponding byte range B2-B3. The decryption template parameters identify that AES encryption is applied without padding (using cipher-text stealing). The keying information and the fact that different keys are applied to different resolutions are signaled with the key information parameters. Specifically the key granularity is specified as resolution so each resolution has a different key, where the processing order is signaled as TRLCP. The key information for each resolution is contained in the value list of keys. The encryption is performed on the codestream, encrypting both the packet headers and packet bodies. The encryption granularity is resolution, where the processing is performed in TRLCP ordering which is the same ordering as the original codestream. Since the two resolutions are encrypted separately, two initialization vectors (IVs) are required and these are contained in the value list.

Note that a packet's cipher text results are specified by the processing order and therefore are independent of input codestream's progression order, however, the placement of the encrypted packets in the output codestream follow the ordering of the input codestream packets.

Table 64 — SEC marker segment for example 1

Parameter		Size (bits)	Values	Meaning
SEC		16	0xFF65	SEC marker.
L_{SEC}		16	138	Length of SEC marker segment is 138 bytes.
Z_{SEC}		8	1	Index of this SEC marker segment.
P_{SEC}	F_{INSEC}	1	0	INSEC is not used.
	$F_{multiSEC}$	1	0	One SEC marker segment is used.
	F_{J2K}	2	00	Not compliant with JPEG 2000 part 1
	F_{TRLCP}	1	0	TRLCP tag usage is not defined in P_{SEC}
	N_{tools}	7	0000001	Number of security tool is one.
	I_{max}	7	0000000	Maximum tool instance index is zero.

	padding	5	00000	Padding bits
t		1	1	JPSEC normative tool
i		7	0	Tool instance index
ID		8	1	Decryption template
L_{zoi}		16	12	Length of ZOI is 12 bytes.
ZOI		96	See Table 65	Zone of Influence for this tool.
L_{PID}		16	98	Length of P_{ID} is 98 bytes.
P_{ID}		784	See Table 65	Parameters for this technology

Table 65 — ZOI example

Parameter			Value (in order)	Size (bits)	Derived meaning
NZzoi			1	8	Number of Zone s is one.
Zone^0	DC_{ZOI}^1		1	1	The byte aligned -segment follows
			0	1	Image related description class
			001000	6	Resolution is specified
	DCzoi^2		0	1	The byte aligned -segment does not follow.
			1	1	Non-image related description class.
			010000	6	Byte ranges after the SOD marker are specified.
	Pzoi^{0,1}	Mzoi^1	1	0	The byte aligned -segment does not follow.
			1	0	The specified zones are influenced by the JPSEC tool.
			1	0	Single item is specified.
			2	01_b	Index mode.
			2	00_b	Izoi uses 8 bits integer.
			1	0	Izoi is described in one dimensions.
		I_{ZOI}	8	1	Resolution 1 is specified
	Pzoi^{0,2}	Mzoi^2	0	1	The byte aligned -segment does not follow.
			0	1	The specified zones are influenced by the JPSEC tool.
			1	1	Single items are specified.
			01	2	Range mode.
			01	2	Izoi uses 16 bits integer.
			0	1	Izoi is described in one dimensions.
		Izoi^{21}	0000 0000 0110 0100	16	Starting byte location is 100th (bytes). (B0)
			0000 0000 1001 0110	16	Ending byte location is 150th (bytes). (B1)
Zone^1	DC_{ZOI}^1		1	1	The byte aligned -segment follows
			0	1	Image related description class
			001000	6	Resolution is specified
	DCzoi^2		0	1	The byte aligned -segment does not follow.

			1	1	Non-image related description class.
			010000	6	Byte ranges after the SOD marker are specified.
	Pzoi^{0,1}	Mzoi^{1}	1	0	The byte aligned-segment does not follow.
			1	0	The specified zones are influenced by the JPSEC tool.
			1	0	Single item is specified.
			2	01_b	Index mode.
			2	00_b	Izoi uses 8 bits integer.
			1	0	Izoi is described in one dimensions.
		I_{ZOI}	8	2	Resolution 2 is specified
	Pzoi^{0,2}	Mzoi^{2}	0	1	The byte aligned-segment does not follow.
			0	1	The specified zones are influenced by the JPSEC tool.
			1	1	Single items are specified.
			01	2	Range mode.
			01	2	Izoi uses 16 bits integer.
			0	1	Izoi is described in one dimensions.
		Izoi^{22}	0000 0001 0010 1100	16	Starting byte location is 300th (bytes). (B2)
			0000 0001 0101 1110	16	Ending byte location is 350th (bytes). (B3)

Table 66 — P_{ID} example

Parameter		Size (bits)	Values	Meaning
T_{ID}		456	See Table 67	Decryption templates
PD		8	0	The byte aligned-segment does not follow.
			10	Codestream domain
			00000	Packet header is encrypted.
G	PO	16	000 001 010 011 100 0	TRLCP
	GL	8	0000 0011	Unit of protection is resolution level
V	N_V	32	2	Number of values in the value list V is 2
	S_V	8	128	Length of each Vn is 128 bits.
	V1	128	*IV0*	Initialization vector value for R0
	V2	128	*IV1*	Initialization vector value for R1

Table 67 — Decryption template example

Parameter		Size	Value (in order)	Derived meaning
ME_{decry}		8	0	Marker emulation has been occurred.
CT_{decry}		16	3	Block cipher (AES)
CP_{decry}	M_{bs}	6	100000	CBC mode. Bits are not padded.

	P_{bs}	2	00	Ciphertext stealing
	SIZ_{bs}	8	128	Block size (128 bits)
	KT_{bs}	416	See Table 68	Key template

Table 68 — Key template example

Parameter		Size (bits)	Value	Derived meaning
LK_{KT}		16	128	Length of key is 128 bits.
KID_{KT}		8	3	URI for secret key
LKI_{KT}		16	16	Length of KI_{KT} is 19 bytes.
KI_{KT}		152	https://server/key1	Secret key for the 1st layer can be retviered from https://server/key1.
G_{KT}	PO	16	000 001 010 011 100 0	Processing order is tile-resolution-layer-component-precinct
	GL	8	0000 0011	Unit of protection is resolution level.
V_{KT}	N_V	32	1	Number of values in the value list V is 1.
	S_V	8	16	Length of each Vn is 19 bytes.
	V	152	https://server/key2	Secret key for the 2nd resolution level can be retrieved from https://server/key2.

6.3.2 Example 2

In this case authentication is applied to the same JPEG-2000 coded image as above. In this example all three resolutions and three layers per resolution are authenticated, where the authentication of each resolution uses a different key. Since there are three resolutions there are three keys, and since there are three layers per resolution there will be three MAC values per resolution . Thus, there will be a total of nine MAC values for the entire JPSEC image. Specifically,

- Resolution 0 has MAC values M0, M1, M2 (one for each layer) using key0
- Resolution 1 has MAC values M3, M4, M5 (one for each layer) using key1
- Resolution 2 has MAC values M6, M7, M8 (one for each layer) using key2

This example illustrates how authentication can be signaled as well as the flexibility provided by the ZOI and granularity tools. As in the prior example, the input image is coded in RLCP progression order, and has 1 tile, 3 resolutions, 3 layers, Nc components, and Np precincts (the number of components and precints is not important in this specific example). Authentication is performed using HMAC with SHA-1.

JPSEC signals how a JPSEC consumer can verify or authenticate the JPSEC protected content. First, the tool template ID for the authentication template is signaled. Then the ZOI is used to signal that there are three resolutions and the associated byte ranges for each resolution. The authentication template parameters signal that HMAC is applied using SHA-1. The key information template provides information about the keys including that the key granularity is resolution and suppling the information for each of the three keys in the value list for the keys. The processing domain for authentication is specified as the codestream including packet headers. The tool granularity for authentication is specified as the layer, therefore there are 3 MACs for each resolution, for a total of nine MAC values. The value list contains the nine MAC values. The processing order for the above was idenifided as TRLCP, which is the same as the original codestream order.

Note that the use of processing order in the granularity field ensures that the same MAC values would result independent of the codestream's progression order.

Note that while this example demonstrated the use of MACs, the same approach can be used to signal the use of multiple digital signatures.

Table 69. The SEC marker segment

Parameter		Size (bits)	Value (in order)	Derived meaning
SEC		16	0xFF65	SEC Marker
L_{SEC}		16	68 + SZ_KT*3 + SZ_MAC*9	Length of the SEC marker segment, where SZ_KT is the size of key used and SZ_MAC is the size of generated MAC
Z_{SEC}		8	0	The first marker segment that appears in the codestream
P_{SEC}	F_{INSEC}	1	0	INSEC marker segment is not
	$F_{multiSEC}$	1	0	Only one SEC marker segment in this codestream
	F_{J2K}	2	1	The JPSEC stream is compliant with JPEG 2000 part 1
	F_{TRLCP}	1	0	The TRLCP tag is not used
	N_{tools}	7	1	Only one tool is used in this codestream
	I_{max}	7	0	The maximum tool instance index is 0
	Padding	5	0	Padding with 0
	P_{TRLCP}	0	N.A	P_{TRLCP} is not present since TRLCP is not used.
Tool^0	T	1	0	JPSEC Normative tool
	I	7	0	Tool instance index
	ID	8	2	This normative tool uses a authentication template
	L_{ZOI}	16	22	Length of ZOI and L_{ZOI} in bytes
	ZOI	Variable	Table 70	The covered zone of the image
	L_{PID}	16	38 + SZ_KT*3 + SZ_MAC*9	Length of L_{PID} and P_{ID} field, where SZ_KT is the size of key used and SZ_MAC is the size of generated MAC
	P_{ID}	Variable	Table 71	Parameters for JPSEC tool

Table 70. ZOI signalling

Parameter		Size (bits)	Value (in order)	Derived meaning
NZzoi		8	1	Number of Zones is 1
Zone^0	DC_{zoi}^1	1	1	The byte aligned-segment follows.
		1	0	Image related description class.
		6	001000_b	Resolution levels are specified in order.
	DC_{zoi}^2	1	0	The byte aligned-segment does not follow.
		1	1	Non-image related description class.
		6	010000_b	Byte ranges are specified.

	Pzoi^{0,1}	Mzoi^{1}	1	0	The byte aligned -segment does not follow.
			1	0	The specified zones are influenced by the JPSEC tool.
			1	0	Single item is specified.
			2	01_b	Range mode.
			2	00_b	Izoi uses 8 bits integer.
			1	0	Izoi is described in one dimensions.
		Izoi^{1}	8	0	The beginning of the range is 0 .
			8	2	The end of the range is 2.
	Pzoi^{0,2}	Mzoi^{2}	1	0	The byte aligned -segment does not follow.
			1	0	The specified zones are influenced by the JPSEC tool.
			1	1	Multiple items specified.
			2	01_b	Range mode.
			2	01_b	Izoi uses 16 bits integer.
			1	0	Izoi is described in one dimensions.
		N$_{ZOI}$	8	3	Number of I$_{ZOI}$ is 3
		Izoi^{1}	16	*B0*	Starting byte location is B0th (bytes).
			16	*B1*	Ending byte location is B1th (bytes).
		I$_{ZOI}^{2}$	16	*B2*	Starting byte location is B2th (bytes).
			16	*B3*	Ending byte location is B3th (bytes).
		I$_{ZOI}^{3}$	16	*B4*	Starting byte location is B4th (bytes).
			16	*B5*	Ending byte location is B5th (bytes).

Table 71. P$_{ID}$ signalling parameters

Parameter					Size (bits)	Value (in order)	Derived meaning
T$_{auth}$	M$_{auth}$				8	0	Authentication methods: Hash -based authentication
	P$_{auth}$	M$_{ID}$			8	1	HMAC is used for authentication
		H$_{ID}$			8	1	SHA-1 is used for hashing
		KT	LK$_{KT}$		16	*SZ_KT*	Length of the key in bits
			KID$_{KT}$		8	3	KI$_{KT}$ contains the URI for the private key
			LKI$_{KT}$		16	9	Length of KI$_{KT}$ is 9 bytes
			KI$_{KT}$		variable	http://KI	URI where private key can be retrieved.
			G$_{KT}$	PO	16	0000010100 111000_b	The order is tile – resolution – layer – component – precinct
				GL	8	00000011_b	Key granularity is resolution
			V$_{KT}$	N$_{V}$	32	3	There are 3 keys in the list
				S$_{V}$	8	*SZ_KT*	Size of each key is *SZ_KT*
				VL	*SZ_KT*	*Key0*	The first key is *key0,* for resolution 0.
					SZ_KT	*Key1*	The second key is *key1*, for resolution 1

					SZ_KT	*Key2*	The third key is *key2*, for resolution 2
		SIZ$_{MAC}$			16	*SZ_MAC*	Size of MAC is *SZ_MAC*
PD					1	0	The byte-aligned segment does not follow
					2	10$_b$	Processed on codestream domain.
					2	00$_b$	Reserved
					1	0	Both header and body are protected
					2	0	Reserved
G	PO				16	0000010100111000$_b$	The order is tile – resolution – layer – component – precinct
	GL				8	00000100$_b$	Tool granularity is layer
V	N$_V$				32	9	There are 9 MACs (3 MACs per resolution)
	S$_V$				8	*SZ_MAC*	Size of each MAC is *SZ_MAC*
	VL				*SZ_MAC*	*M0*	The first MAC is *M0*
					SZ_MAC	*M1*	The second MAC is *M1*
					SZ_MAC	*M2*	The third MAC is *M2*
					SZ_MAC	*M3*	The fourth MAC is *M3*
					SZ_MAC	*M4*	The fifth MAC is *M4*
					SZ_MAC	*M5*	The sixth MAC is *M5*
					SZ_MAC	*M6*	The seventh MAC is *M6*
					SZ_MAC	*M7*	The eighth MAC is *M7*
					SZ_MAC	*M8*	The ninth MAC is *M8*

6.4 Distortion field examples

This section provides a few simple examples on the use of the distortion field.

6.4.1 Example 1

This example builds on the ZOI example 3 in Section 6.1.3 to show how distortion values can be associated with the two data segments signalled by the ZOI in that example. As review, example 3 in Section 6.1.3 signalled two data segments: (1) bytes 10 to 100 and (2) bytes 10000 to 12000. To associate distortion fields to these two data segments requires two steps. First, the distortion field is signalled in DCzoi. Second, the distortion values are signalled using Pzoi2. Therefore the only changes to the ZOI example 3 in Section 6.1.3 are to set the distortion field bit in DCzoi, and to add Pzoi2 (the final 9 lines in the following table).

Table 72 — Associating distortion field to two data segments (extension of ZOI example 3 Sec. 6.1.3)

Parameter		Value (in order)	Size (bits)	Derived meaning
NZzoi		1	8	Number of Zones is one.
Zone^0	DCzoi	0	1	The byte aligned-segment does not follow.
		1	1	Non-image related description class.
		010001	6	Byte ranges after the SOD marker are specified and associated distortion fields are specified .

	Pzoi^2	Mzoi^2	0	1	The byte aligned-segment does not follow.
			0	1	The specified zones are influenced by the JPSEC tool.
			1	1	Multiple items are specified.
			01	2	Range mode.
			01	2	Izoi uses 16 bits integer.
			0	1	Izoi is described in one dimensions.
		Nzoi^2	0000 0010	8	Number of data segments is 2.
		Izoi^{2,1}	0000 0000 0000 1010	16	Starting byte location is 10th (bytes).
			0000 0000 0110 0100	16	Ending byte location is 100th (bytes).
		Izoi^{2,2}	0010 0111 0001 0000	16	Starting byte location is 10000th (bytes).
			0010 1110 1110 0000	16	Ending byte location is 12000th (bytes).
	Pzoi^6	Mzoi^6	0	1	The byte aligned-segment does not follow
			0	1	The specified zones are influenced by the JPSEC tool.
			1	1	Multiple items are specified.
			10	2	Index mode.
			00	2	Izoi uses 8 bits to represent each distortion value .
			0	1	Izoi is described in one dimensions.
		Nzoi^6	0000 0010	8	Number of data segments is 2.
		Izoi^{6,1}	*D1 value*	8	Distortion value for the first segment
		Izoi^{6,2}	*D2 value*	8	Distortion value for the second segment

6.4.2 Example 2

This example describes how distortion values can be associated with JPEG-2000 packets. The DCzoi specifies a range of 4 packets and the distortion field is also signalled. Pzoi1 gives the range of packets and Pzoi2 describes the distortion associated with each of these packet. Notice that since Pzoi1 specifies a range of length 4, and Pzoi2 specifies 4 values, each item in the range is associated with one value, e.g. each packet is associated with one distortion.

Table 73 — Signalling a range of packets and associating distortions for each packet

Parameter			Value (in order)	Size (bits)	Derived meaning
NZzoi			1	8	Number of Zones is one.
Zone^0	DCzoi		0	1	The byte aligned-segment does not follow.
			1	1	Non-image related description class.
			100001	6	Packets are specified and associated distortion fields are specified.
	Pzoi^1	Mzoi^1	0	1	The byte aligned-segment does not follow.

			0	1	The specified zones are influenced by the JPSEC tool.
			0	1	Single item is specified.
			01	2	Range mode.
			00	2	Izoi uses 8 bit integers.
			0	1	Izoi is described in one dimension.
		Nzoi^{1}	0000 0001	8	Number of data segments is 1.
		Izoi^{11}	0000 0000	8	Starting packet is number 0.
			0000 0011	8	Ending packet is number 3.
	Pzoi^{6}	Mzoi^{6}	0	1	The byte aligned-segment does not follow
			0	1	The specified zones are influenced by the JPSEC tool.
			1	1	Multiple items are specified.
			10	2	Index mode.
			00	2	Izoi uses 8 bits to represent each distortion value .
			0	1	Izoi is described in one dimensions.
		Nzoi^{6}	0000 0100	8	Number of data segments is 4.
		Izoi^{6,1}	*D1 value*	8	Distortion value for the first packet
		Izoi^{6,2}	*D2 value*	8	Distortion value for the second packet
		Izoi^{6,3}	*D3 value*	8	Distortion value for the third packet
		Izoi^{6,4}	*D4 value*	8	Distortion value for the fourth packet

7 Metadata for JPSEC

This section considers the case when JPSEC content consists of jp2 files. The file contains meta data boxes and the codestream box. For the codestream, the signalling follows the rules defined above. This section defines which metadata must be added to a JPSEC file. It identifies the JPSEC metadata to be inserted in the file and specifies in which box it is to be added.

7.1 JPSEC metadata

The format of JPSEC metadata is XML. Its required fields include:

- The tool identifier or list of tool identifiers.
- A link to the REGAUT, under the form of an URL where to get the complete information available to the public about the tool(s) in use.
- The time stamp of tool application must be added. The JPSEC XML must be created at the time when the JPSEC codestream is created to comply with the WIPO rules. So the time stamp is that of the creation time, expressed in GMT to avoid any problem with priority. The application must follow this recommendation to create the time stamp.

Furthermore, its optional fields include:

- All of REGAUT input form representing all that is available to the public about the JPSEC tool can be made available inside the file, allowing for off-line processing of such information: a signature, some parameters etc…
- Operational conditions for the use of the JPSEC tools, for example the conditions and cost for decrypting.
- Technical details about the technology can be added to help understanding the goals of the JPSEC application.

- Other information from the SEC marker such as the list of JPSEC tool identifiers that are required for further processing, public parameters used in the applied tools or zone of influence of the tools.
- As a result of the application of the JPSEC tools, values of digital signature, watermark payload, etc can be included in the file metadata : this could help checking the authenticity without the need to hook to a remote location. Some applications would even not work if the signature check gives a negative response.

7.2 Location of JPSEC metadata

IPR box: When the JPSEC tool is devoted to IPR protection, the IPR flag must be raised in the image header box and the IPR box must contain the JPSEC metadata.

History box: In case there is no IPR box, but there is a History box, the JPSEC metadata must be inserted in the History box.

Other: If there is neither an IPR box nor a history box, a specific XML JPSEC box must be created to contain the JPSEC metadata.

8 Registration authority

8.1 General introduction

This description makes reference to JPSEC but can be extended to any other JPEG 2000 existing or new work item.

As stated below, JPSEC addresses a number of techniques from which use cases can be defined and applications developed. In order to have a reference site where information about the different JPSEC items is available, it has been decided to create a registration authority (RA). By JPSEC items we understand methods used for protection of content, integrity, authentication, access control and any other security related process.

The JPSEC RA should allow for registration, identification and dissemination of essential information about the JPSEC item it is meant for. A central authority is the proposed solution to make sure that each of the items has a unique identifier linking to the registered descriptive information.

Typically, the proposed solution is based on a unique Registration Authority, but it may be necessary, in particular for linguistic and/or character set usage purposes, to create a number of intermediate sites whose role would be to interface the unique Registration Authority with local language user interface. However, an automatic translation will have to make it possible to submit the registration with the same conditions as if the submission was directly addressing the unique RA.

8.2 Security tool registration authority

A number of security methods are defined in JPSEC specifications. As this list does not contain all possible security tools, there is a facility for registering new ones. As an example, here is the current list of item types:

- Digital signature
- Watermarking
- Encryption, including scrambling
- Key generation and management
- Identification and registration
- Others (e.g. data luring and splitting)

A specific JPSEC item number is reserved for provisional use with extensions to be defined later. Temporary numbers to be defined can be allocated to items undergoing the registration process if and when applicable. These values may be used until assignment of a permanent JPSEC item identification number from the JPSEC Authority. The last possible number can also be reserved for future use and shall indicate that the JPSEC item number follows in a special documentary field.

8.3 Purpose of JPSEC tool registration

The registration mechanism has been defined to allow for well defined methods used in various JPSEC operations; it does not allow for registration of application specific methods and/or technologies. Its purpose is also to allow for addition of other standardized JPSEC methods to the initial list above. On top of all JPSEC items types, the addition of new items will be strictly controlled. Applicants may suggest standardized technologies that they wish to be included in the J PSEC reference list. Note that the JPSEC item use is mentioned with a JPSEC marker present in the codestream. When an application finds an unknown JPSEC ID, it can hook to the JPSEC RA and get all registered information about the item. Ideally, when applic able it can also download the tools to exploit the particular function of the JPSEC tool. In any case, the registration process will create the link to the technology provider for further information. In some use cases, this link will allow for automatic operation.

8.3.1 Criteria

A proposed new JPSEC item must meet the following criteria:

§ Unique - it must not duplicate a JPSEC item already defined.
§ Correct submission - the syntactically and technically correct submission along with all appropriate explanations of purpose must be submitted. The proposed JPSEC item must be classified in one of the existing categories or identified as "others"
§ Utility - the new JPSEC item should demonstrate utility to the user and give examples of use cases when relevant.

Given these criteria are met, the new JPSEC item will be allocated an identification (ID) number and considered as referenced. The ID number can then be used for signaling in the JPEG 2000 codestream.

8.3.2 Contents of the submission

The submission should conform to a nor mative form and make any possible reference to existing standards document describing similar processes.

The submission may also include an informative section that describes the reason for inclusion of this new JPSEC item. It should also explain and demo nstrate proper usage of this new JPSEC item in the JPEG 2000 environment.

8.3.3 Normative description

The normative form has three different parts :

- Identification of the registrant, technology provider, owner or right holder
- Identification of the new JPSEC ite m
- Automatically generated registration data.

8.3.4 Example of new JPSEC item registration documents

XYZ watermarking algorithm

Identifier : 275 (hex 0113)

Part 1 : identification of the registrant

Name (mandatory)

Company, postal address (optional)

Email link (mandatory)

Occupation, title, … (optional)

Part 2 : identification of the item

Registrant identification (mandatory)

Category of item (from list or “others”, mandatory)

IPR status, e.g. owner, right holder, mandated dealer… (mandatory)

Title or item name (optional)

Short technical description (optional)

Operational example use case description (optional)

Parameters table, number and specification, including possible values (optional)

Restrictions of use (optional)

Guidelines for optimum usage (optional)

Downloads when relevant (optional)

Additional comments, motivation, references… (optional)

Part 3 : automatically generated registration data

Identifier

Time stamp

Version when relevant

Last modification when relevant

Part 4 : additional registration information

Special access control conditions

Validity (temporary or final version)

Update and erase procedure (profiles)

Email confirmation

8.4 Operating conditions for the Registration Authority - Submission, review and appeal process

This section provides requirements for the submission and approval of new or updated JPSEC item as defined previously. All the registration processes have a common set of requirements for the registrants as well as requirements and processes based on the category of submission, as described in detail below.

8.4.1 Submission process

The registration submission process conforms to the specifications described above in this document and is graphically shown in Figure 37 below, showing the process flow for submission of a new item to the JPSEC Authority. This process flow is focused on access based on an Internet based submission and approval system, but could be applied to other (non-electronic) processes going for example through the Authority Administrator assistance.

The registrant (individual or institution) shall submit a request for registration of a new item to the JPSEC Authority, possibly via Internet access. The first operation is the identification of the registrant, which takes the steps shown on the block diagram: either the registrant is unknown to the JPSEC Authority, or he has previously registered an item and is listed in the JPSEC Authority registers. If it is the first time the registrant provided a submission to the Authority, he must complete a genera l identification form and get a personal password for access controlled operations.

In order to provide some level of security to the registration process, a password is set by the registrant and must be used every time the registrant contacts the JPSEC Authority for an active operation (this excludes consultation of the contents but concerns registration and update of registered material). Only the registrant will be allowed modification of the password and all the registered information regarding his personal identification (name, company, etc.). The registrant may also request, under certain conditions to be defined by the Authority and registration submission form, that part of the input information remains undisclosed.

It is possible to modify the submitted content. For example, the registrant may want to correct the information in the submission form. However, it is highly desirable that the modified item is backward compatible with the original submission. A modification can be made directly by the registrant under his own responsibility. Alternatively, a modification can be made by a third party; in this case the modified submission must be approved by either the registrant or the administrator of the registration authority. [NOTE: third party modification to be further discussed]

8.4.2 Review process

Following validation of the individual's right to access the Authority's Registration System, the actual registration of the desired item begins. This is done according to the conformity of the submission t o the specific contents. The JPSEC Authority verifies (manually or automatically) the conformity of the request to the specifications, and notifies the registrant of a positive or negative response to the registration request. The Authority must respond within one month to the registrant's request. If no response is received within one month, the probationary period shall begin.

The response of the JPSEC Authority shall be positive if the item has not been registered (i.e. is new and unique), and if the contents of the submission conforms to the applicable forms. This positive response, however, is not final. It does allow the submitter to implement the proposed item for a probationary "validation" period to be defined by the Authority. This probationary p eriod may last for three to six months, for example. It is during this time that the content of the submission is made accessible to the public for comments. This will be done via the direct link available from the J PEG authoritative web site, and will be coordinated and validated by the Expert technical advisory group.

During the validation period, the registration is temporary, and the JPSEC Authority may require more information from the registrant regarding the submission. During this period, omission s or errors in the submission, lack of implementation support, or related problems, may cause the rejection of the submission. However, if after the probationary period no inconsistencies or concerns are raised, the submission shall be considered accepted and formally approved by Authority and will be accordingly disseminated on the public register. Any individual reviewing the public register shall be clearly informed if a submission has successfully completed its probationary period based on a "status" field or similar notification method.

The response of the JPSEC Authority may also be negative if the submission is not acceptable as received, due to errors, inconsistencies or technical concerns with the proposed content. Examples of why the Authority would reject a submitted item include:

- If an approved, registered item already exists that contains the identical contents of the submission.
- The Authority considers that there is not enough originality in the proposed item which could easily be implemented with an existing, approved item.
- If the submission contains errors or is not compliant with the specifications or standard it is based on.

8.4.3 Notification and appeal process

A negative response may be appealed if the registrant believes that there was an er ror made in the rejection, or when further information is required to clarify issues or concerns. If the registrant requires additional review beyond the Authority's process, he may submit his case for review by the WG1 at the next appropriate WG1 group meeting. He may then be required to provide additional information at the request of the experts, who, under the authority of WG1, will provide a final, definitive response of acceptance or rejection. A refused item may still be used by the registrant, bu t it will not be allowed to claim standards compliance and hence, may become de facto a proprietary item. In order to have a rejected item reviewed by the WG1, the registrants must re-submit the proposal through their National Body, specifying why the sub mission requires consideration by WG1.

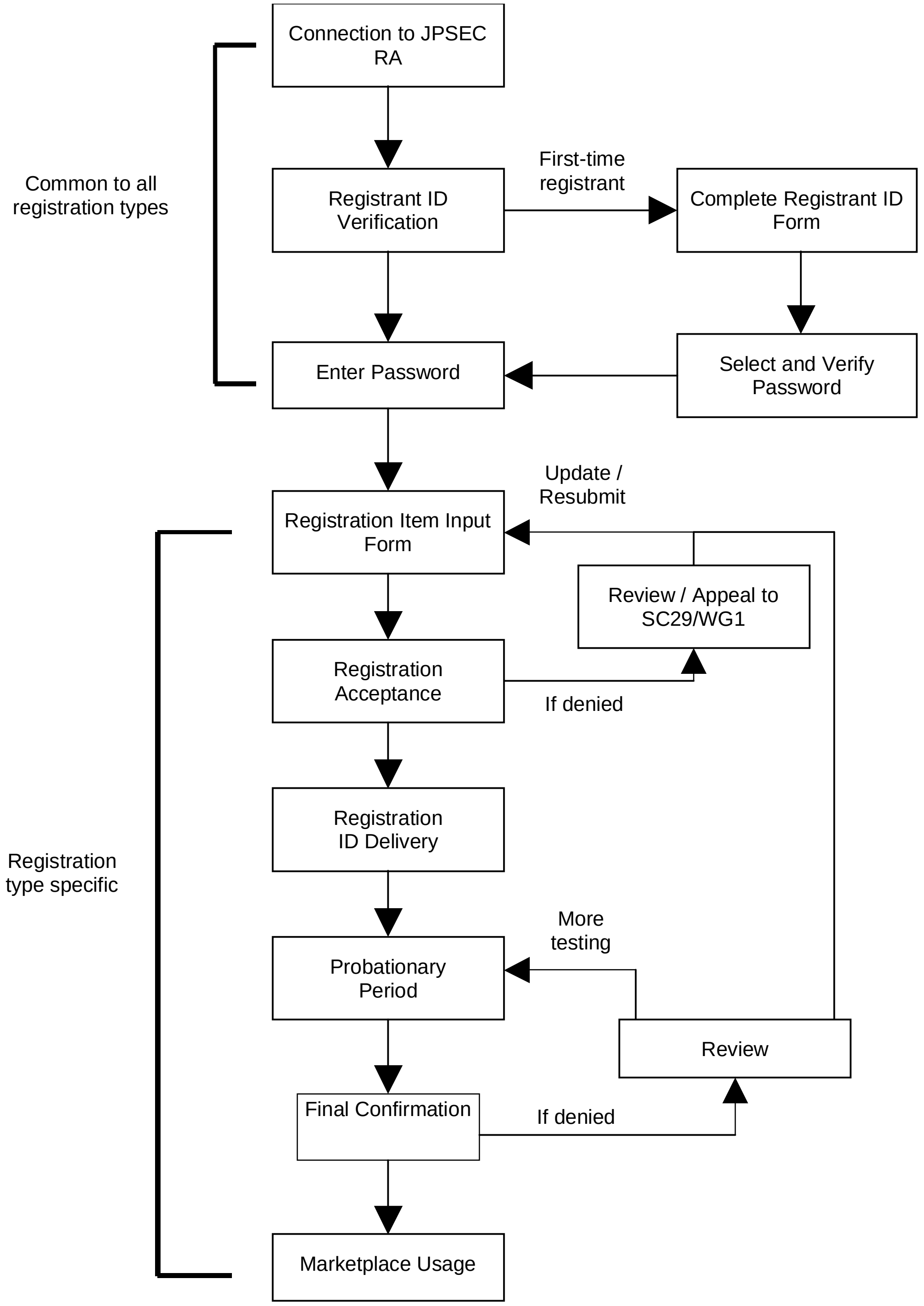


Figure 37 — JPSEC RA Registration Process

Annex A
(informative)

Guidelines and use cases

A.1 JPSEC walkthrough

This section clause presents the logical steps that are involved in the implementing processing of JPSEC servicesprotected content. The following walkthrough focuses on JPSEC tools.

The following four steps are usually required for processing JPSEC bitstream.

Step 1. JPSEC bitstream content delivery

A JPSEC enabled application requests JPSEC protected content from a server. How this is done is out of scope of this IS/Rdocument. But in order to process properly the content the application should pass the content over to a JPSEC decoder which will read the JPSEC signalling information from the JPSEC protected content.

Step 2. Identification of JPSEC tools method description access

Upon reception of the content the JPSEC decoder reads from the SEC marker in the main header of the code-stream which tools methods are required for the processing of the data. It can thus determine which tools methods must be applied to the content, and in which order.

Step 3. Localization of JPSEC tools method availability

The JPSEC tools methods can be implemented in software or hardware. In some applications, the JPSEC tools methods will be embedded in the JPSEC enabled application: new tools methods would only be provided by new releases of the application. Other applications would allow download of missing protection toolsmethods. How this is done is out of scope of this IS/R document. Refer to figure below for local and remote implementation of JPSEC tools.

Step 4. Processing of JPSEC bitstreme the content

If a decoder implements the JPSEC tools methods, it proceeds with applying them to the content. If it does not implement the toolsmethods, it can inform the application and the end-user, and refer to the registration authority for more information.

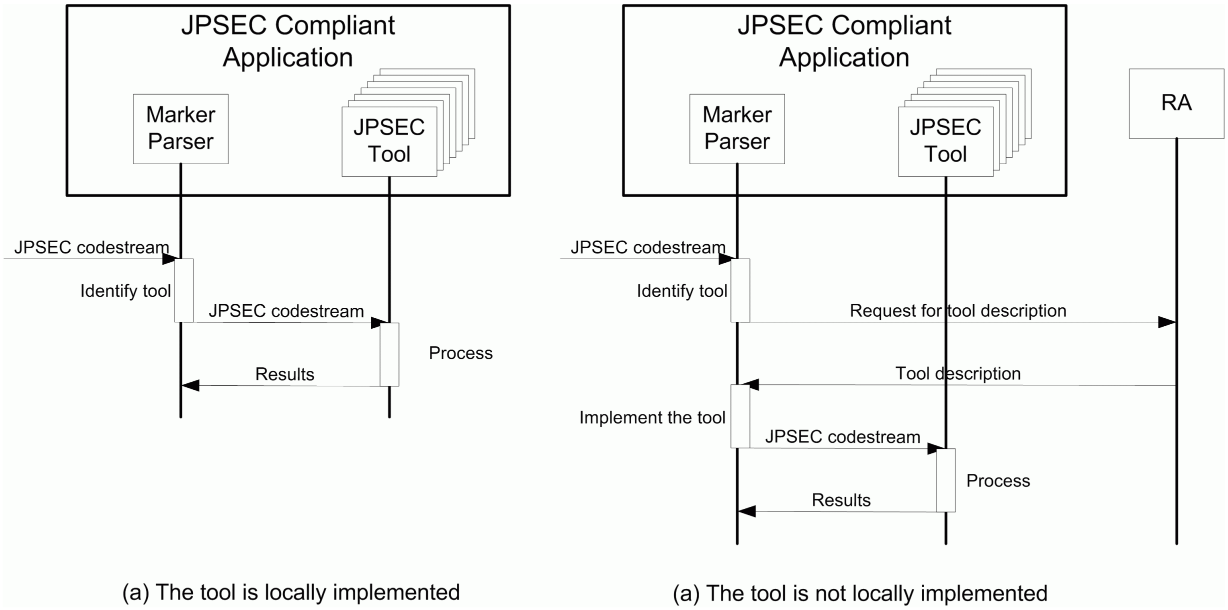


Figure A.1 — Local and remote implementation of JPSEC tools

A.2 A class of JPSEC applications

A.2.1 Introduction

This clause gives a conceptual description of how a class of JPSEC applications may be implemented. This class of application exemplifies scenarios of secure JPEG 2000 image distribution. The following subclauses describe an overview of a conceptual JPSEC application including JPSEC entities and information that are communicated between them. This description is conceptual and neither intends to define a concrete implementation nor specify requirements for an implementation ; specific applications may or may not include entities identified in the following description.

A.2.2 Overview of a secure JPEG 2000 image distribution

Figure A.2 shows an overview of the class of JPSEC applications of secure JPEG 2000 image distribution. In these applications, the JPSEC application may be required to provide various security services for JPEG 2000, for example, confidentiality of image exchange, authentication of ima ge origin, and integrity of image content.

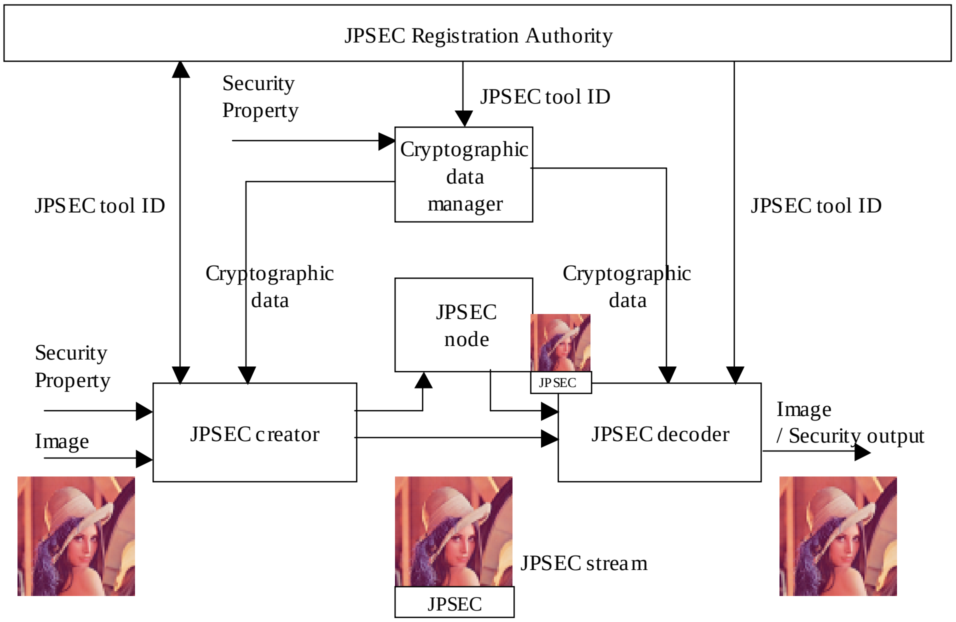


Figure A.2 — Overview of a secure JPEG 2000 image distribution application

In the secure JPEG 2000 image distribution application, one can identify the following steps:

Step 1: A JPSEC bitstream is created by a JPSEC creator.

Step 2: The JPSEC bitstream is distributed through some JPSEC node or nodes

Step 3: The JPSEC bitstream received and consumed rendered by a JPSEC decoder.

Step 1: JPSEC bitstream creation:

The creator is in charge of creating the secure JPEG 2000 bitstream. This bitstream may be created from bitmap data or from JPEG 2000 compressed data. A JPSEC creator applies various security techniques, such as encryption, signature generation, and ICV (Integrity Check Value) generation to a given image data.

To secure the image data the creator defines which Security Parameter Property is associated to the image. A "Security Parameter Property" includes the following attributes:

- Zone of Influence (coverage area of each protection method)
- Processing Domain (domain to be processed by each protection method)
- Granularity (unit of each protection method)
- JPSEC tool identification (applied cryptographic algorithm and related parameters)

Step 2: JPSEC bitstream delivery:

A JPSEC bitstream can be transferred to a JPSEC consumer either directly via a network or media (such as a CD-ROM). It can also be transferred through a JPSEC node which can apply various types of additional processing, such as a transcoding, to the JPSEC bitstream.

When required by the JPSEC security tool methods in the Security Property Parameter of the JPSEC bitstream (e.g. for encryption, or for authentication), the JPSEC creator must distribute to the JPSEC consumer the corresponding cryptographic data through an independent ('secret') channel. This data, such as key or digital signature, can be managed either manually or automatically by a cryptographic data manager.

Step 3: JPSEC bitstream consumption rendering:

A JPSEC bitstream is subject to a JPSEC consumer process according to the applied Security Parameter property: this implies applying the appropriate security techniques, such as decryption, authentication and integrity check. Further, for each JPSEC tool security method a JPSEC creator and JPSEC consumer may use various types of cryptographic data.

As an output of the JPSEC consumer, a decrypted image data and/or security output, such as a verification result, is produced.

A JPSEC creator, JPSEC consumer and cryptographic data manager may reference the JPSEC Registration Authority to obtain necessary processing instructions of a specific JPSEC tool ID.

The following subclauses provide further detail of a conceptual JPSEC entity according to a JPSEC service. Figure A.3 shows the legend description to be used.

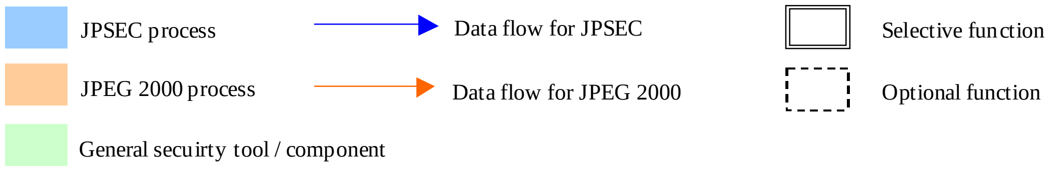


Figure A.3 — Legend description

JPSEC process : A process that uses tools defined in this International Standard.

— **JPEG 2000 process**: A process defined in ISO/IEC 15444-1 (JPEG 2000 Part-1).

— **General security tool / component**: A well-known security process where an identifier is listed in this International Standard but the tool description is defined elsewhere.

— **Data flow for JPSEC**: A data flow that communicates information defined in this International Standard. A dashed line indicates optional.

— **Data flow for JPEG 2000**: A data flow defined in the ISO/IEC 15444-1 (JPEG 2000 Part-1).

— **Selective function**: A function which has several alternate functions that can be selected by an application.

— **Optional function**: A function which can be optionally applied in a JPSEC application.

A.2.3 Encryption end description procedure

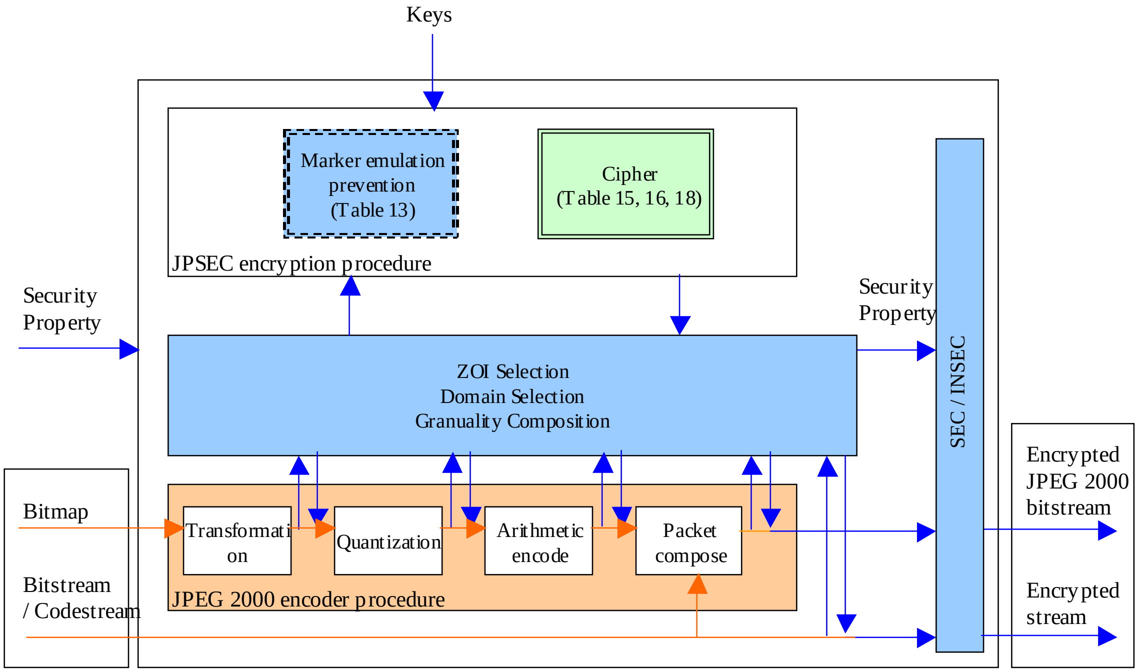


Figure A.4 — Encryption procedure

Figure A.4 shows the overview of an example encryption procedure for a JPSEC creator. This procedure includes the following processes:

— extract data according to the specified Processing Domain

— select a portion of extracted data according to the specified Zone of Influence (i.e. a partial encryption)

— encrypt the selected data using the specified security technique. Further, it is possible to encrypt data in a unit based on the Granularity. In this case, different keys can be used for different units.

— replace the encrypted data with plain data

— (optional) apply a marker emulation prevention mechanism

— compose the Security Parameter operty in the SEC and/or INSEC marker segment

Note that in JPSEC, the encryption procedure can pro duce two types of a secure image data: an encrypted JPEG 2000 bitstream and an encrypted stream. An "encrypted JPEG 2000 bitstream" is compliant to JPEG 2000 part-1, therefore a part-1 compliant decoder is guaranteed to completely decode the bitstream with out decrypting the data. In this case, a noisy or a restricted image will be displayed. On the other hand, an "encrypted stream" is not necessarily compliant to JPEG 2000 part -1; the image data is intended to be passed on to a part-1 compliant decoder afte r appropriate decryption.

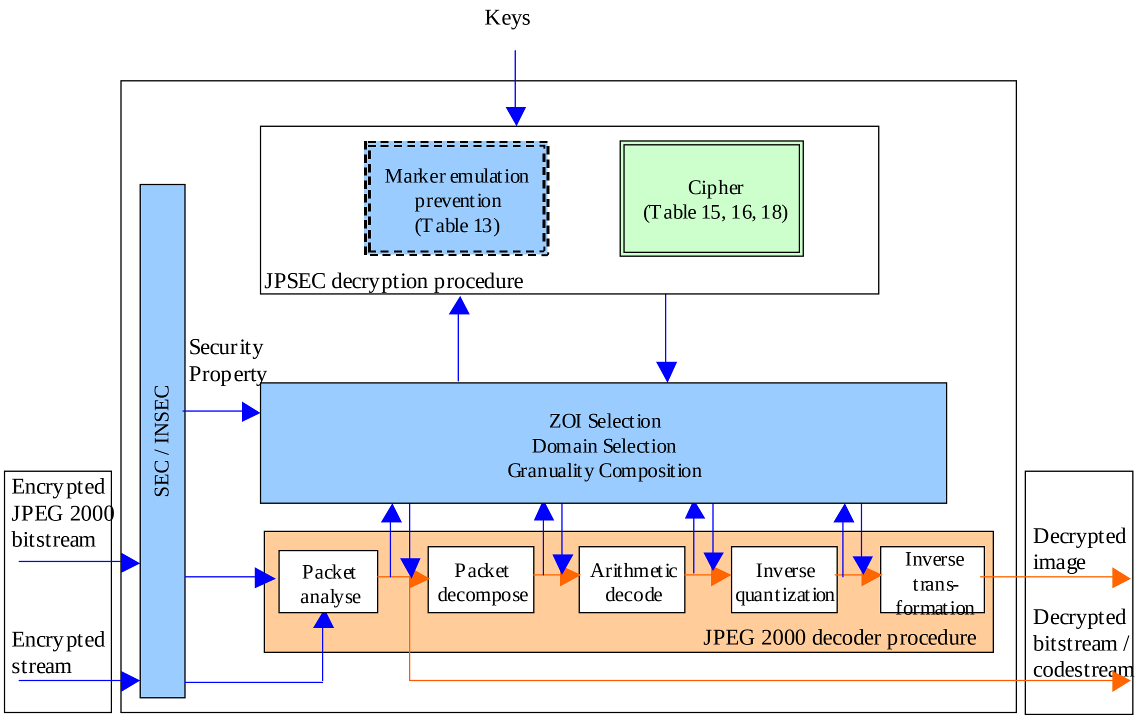


Figure A.5 — Decryption procedure

Figure A.5 shows the overview of an example decryption procedure for a JPSEC consumer. This procedure includes the following processes:

- parse the Security Parameter property in the SEC and/or INSEC marker segment
- extract data according to the signaled Processing Domain
- select a portion of extracted data according to keys to be retained (i.e. a partial decryption)
- decrypt the selected data using a signaled security technique. Further, it is possible to decrypt data in a unit based on the Granularity.
- replace encrypted data with decrypted data
- apply a marker emulation prevention mechanism, if applied at the encryption process

A.2.4 Signature generation and authentication procedure

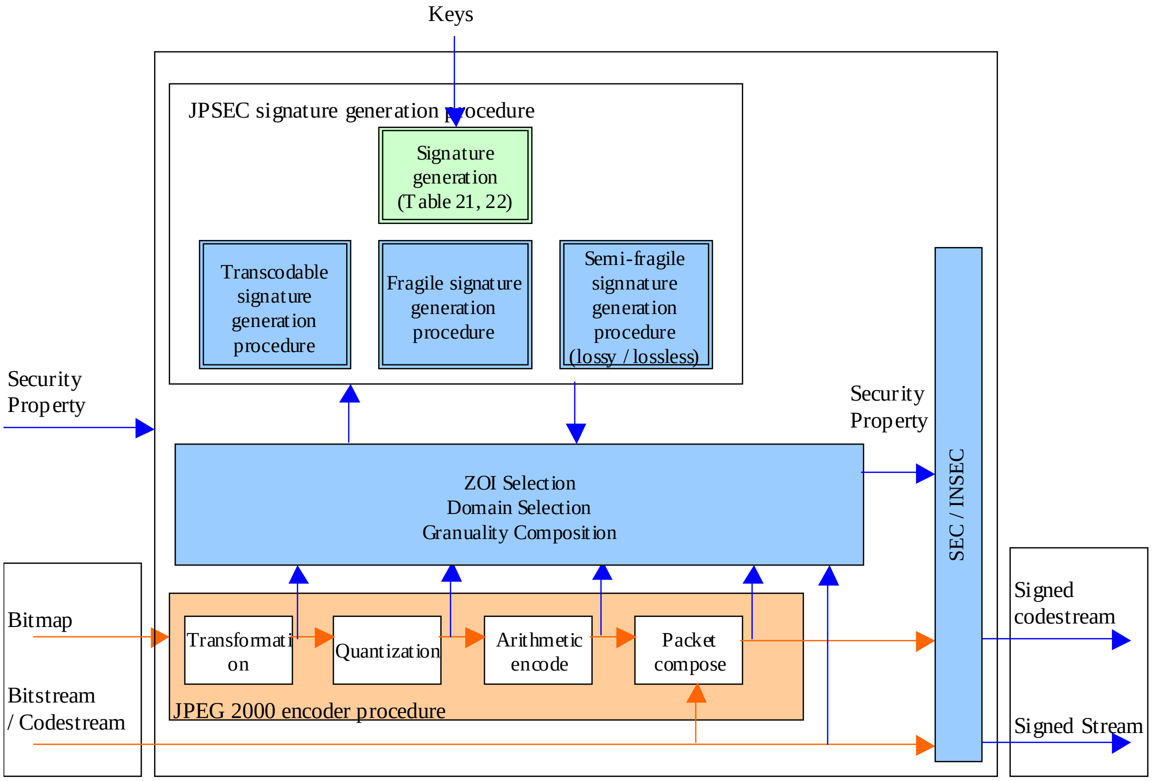

Keys
JPSEC signature generation procedure
Signature generation (Table 21, 22)
Transcodable signature generation procedure
Fragile signature generation procedure
Semi-fragile signnature generation procedure (lossy / lossless)
Security Property
Security Property
ZOI Selection
Domain Selection
Granuality Composition
SEC / INSEC
Bitmap
Bitstream / Codestream
Transformation
Quantization
Arithmetic encode
Packet compose
JPEG 2000 encoder procedure
Signed codestream
Signed Stream

Figure A.6 — Signature generation procedure

Figure A.6 shows the overview of an example signature generation procedure for a JPSEC creator. This procedure includes the following processes:

— extract data according to the specified Processing Domain

— select a portion of extracted data according to the specified Zone of Influence (i.e. partial signature)

— calculate digital signatures corresponding to selected data using the specified security technique. Further, it is possible to generate digital signatures in a unit based on the Granularity.

— compose the Security Parameter property, including the calculated digital signatures, in the SEC and/or INSEC marker segment.

Note that in JPSEC, three modes of authentication are defined: "fragile mode", "semi -fragile mode (lossy / lossless)", and "transcodable mode". A "fragile mode" authentication can detect any one -bit modification for a codestream, where a "semi-fragile" mode authentication can detect any intentional detection but survive incidental distortion up to a pre-determined extent. Further, a "transcodable mode" authentication can verify a codestream transcoded by a publisher.

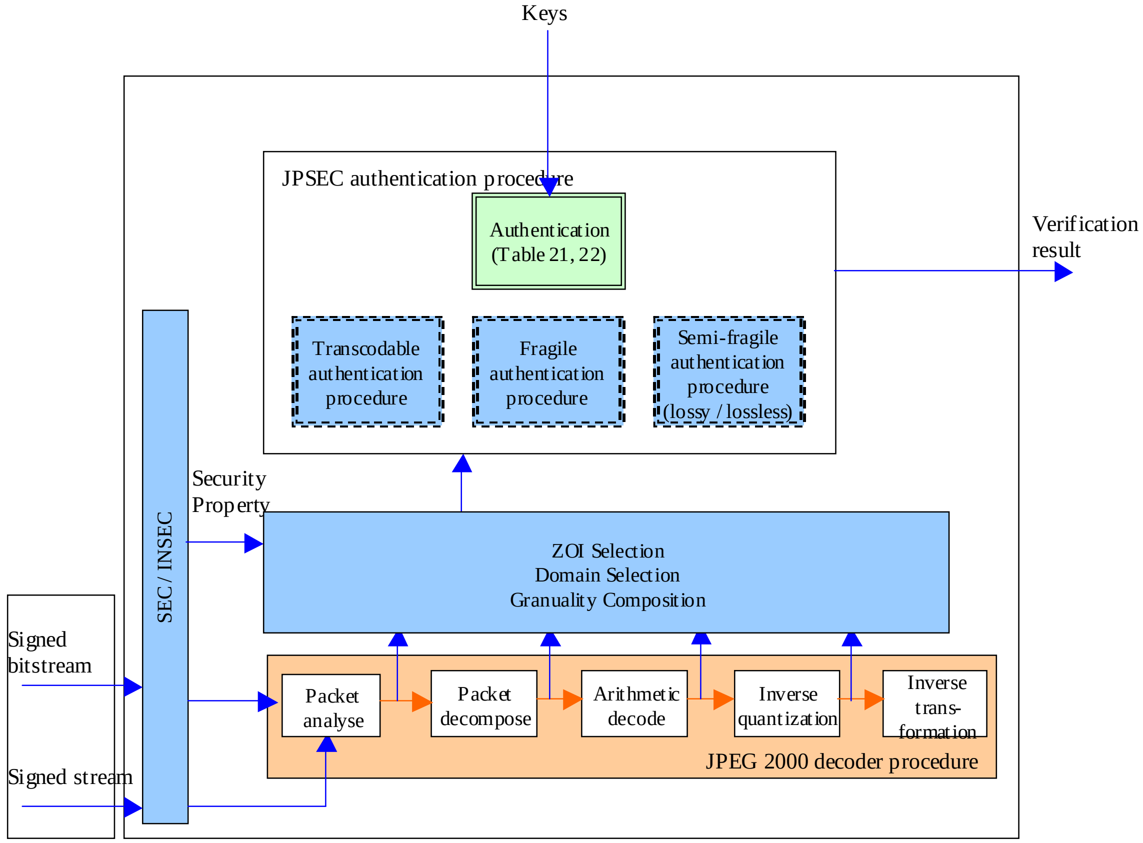


Figure A.7 — Authentication procedure

Figure A.7 shows the overview of an example authentication procedure for a JPSEC consumer. This procedure includes the following processes:

— extract data in a signalled Processing Domain

— select a portion of extracted data according to the signalled Zone of Influence

— verify the selected data using the signalled security tech nique. Further, it is possible to verify the selected data in a unit based on the Granularity.

A.2.5 ICV (Integrity Check Value) generation and integrity check procedure

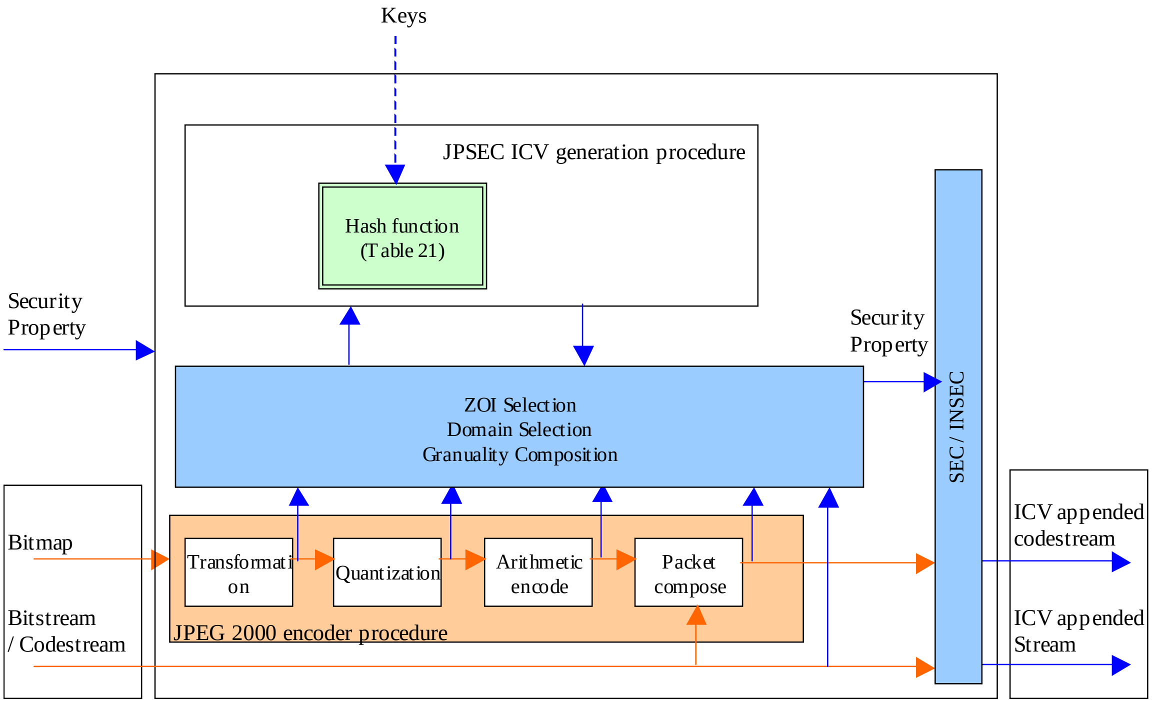


Figure A.8 — ICV (Integrity Check Value) generation procedure

Figure A.8 shows the overview of an example ICV generation procedure for a JPSEC creator. This procedure includes the following processes:

— extract data in a specified Processing Domain

— select a portion of extracted data according to the specified Zone of Influence

— calculate ICVs corresponding to selected data using the specified security technique. Further, it is possible to generate ICVs in a unit based on the Granularity.

— compose the Security Parameter property, including the calculated ICVs, in a SEC and/or INSEC marker segment.

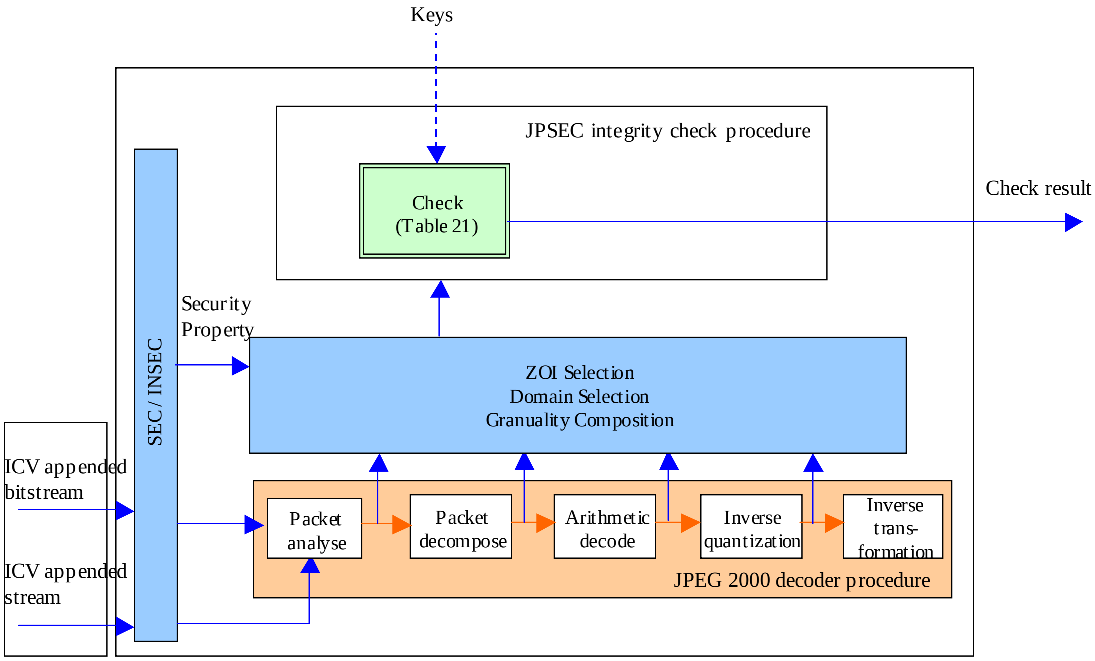


Figure A.9 — Integrity check procedure

Figure A.9 shows the overview of an example integrity check procedure for a JPSEC consumer. This procedure includes the following processes:

- extract data according to the signalled Processing Domain
- select a portion of extracted data according to a signalled Zone of Influence
- verify selected data using a signalled security technique. Further, it is possible to verify the selected data in a unit based on the Granularity.

A.2.6 Key generation procedure

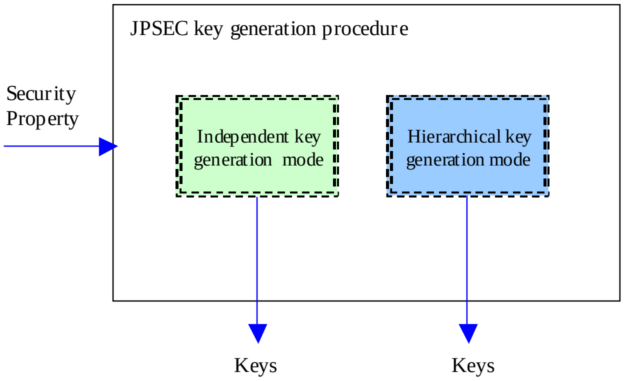


Figure A.10 — Key generation procedure

Figure A.10 shows the overview of an example key generation procedure for a cryptographic data manager. Keys for each security service can be generated manually by a user, or automatically using a key generation tool. For automatic key generation, two types of key generation modes are defined: "independent key generation mode" and "hierarchical key generation mode." In the "independent key generation mode", keys are independently generated using a PRN generator, for example. Therefore, a key is independent of another key. On the other hand, in the "hierarchical key generation mode", keys are hie rarchically generated using a hash function, for example, according to the Security Parameter property (for instance, a structure of a codestream).

Annex B
(informative)

Technology Examples

The JPSEC syntax allows normative and non-normative security tools to be applied to JPEG 2000 images. This section describes ten informative technology examples that demonstrate different usages of JPSEC. These examples are purely informative and not endorsed by the JPSEC standard. However, they are provided to demonstrate the flexibility of the standard.

The technology examples include:

A Flexible Access Control Scheme for JPEG 2000 Codestreams

A Unified Authentication Framework for JPEG 2000 images

A Simple Packet-based Encryption Method for JPEG 2000 bitstreams

Encryption tool for JPEG 2000 access control

Key generation tool for JPEG 2000 access control

Wavelet and Bitstream Domain Scrambling for Conditional Access Control

Progressive Access for JPEG 2000 codestream

Scalable Authenticity of JPEG 2000 Code-streams

JPEG 2000 Data Confidentiality and Access Control System Based On Data Splitting and Luring

Secure Scalable Streaming and Secure Transcoding

B.1 A Flexible Access Control Scheme for JPEG 2000 Codestreams

B.1.1 Security Service

An access control scheme allows for rendering JPEG 2000 code-streams according to any combination of resolutions, quality layers, tiles and precincts.

B.1.2 Typical Application

It provides protection of content delivery via variable media, e.g., Internet, digit al cable TV, satellite broadcast and CD-ROM. Generally, the technology is viable to the applications where a codestream is encrypted only once on the publisher side but the protected codestream is decrypted many ways according to the different privilege on the user sides.

B.1.3 Motivation

In the Super-distribution model, the publisher distributes the protected content freely and the content keys securely. A user who desires to access portions of a code -stream sends his/her request to the key server. The key server, in turn, responses with appropriate decryption keys according to user's privilege. The user can obtain the required sub-images only.

B.1.4 Technical Overview

A protected JPEG 2000 codestream is produced by encrypting each packet by the publisher. The core of the technology is how to manage a key tree which is constructed in any order of tiles, components, resolutions, layers, precincts, and even code-blocks. To describe the technology easily, assume that the key tree order is RLCP (resolution-layer-component-precinct) and each resolution has the same number of precincts. In the following, given a one way hash function h(.), consider a JPEG 2000 image codestream with n_T tiles, n_C components, n_L layers, n_R resolution per tile-component, n_P precincts per resolution. With a master key K for a JPEG 2000 codestream. Construct a key tree as follows.

1. Generate key $k^t = h(K|\text{“}T\text{”}|t)$, for each tile $t =$ 0,1, ..., n_T–1, where "|" is the concatenation, and *“T”* denotes the ASCII code of the letter *T*.
2. Generate key $k^r = h(k^{r+1})$, for each $r = n_R$–2, ..., 1, 0, where $k^{n_R-1} = h(k^t|\text{“}R\text{”})$ and “*R*” denotes the ASCII code of the letter *R*.
3. Compute key $k^{rl} = h(k^{r(l+1)})$, for each $r = n_R$–1, ..., 1, 0, $l = n_L$–2, ..., 1, 0, where $k^{r(n_L-1)} = h(k^r|\text{“}L\text{”})$ and where “*L*” denotes the ASCII code of the lette r *L*.
4. Calculate key $k^{rlc} = h(k^{rl}|\text{“}C\text{”}|c)$, for each $r = n_R$–1, ..., 1, 0, $l = n_L$–1, ..., 1, 0, c = 0, 1, ..., n_C–1, where “*C*” denotes the ASCII code of the letter *C* and *c* denotes the index of this component.
5. Produce keys $k^{rlcp} = h(k^{rlc}|\text{“}P\text{”}|p)$, for each $r = n_R$–1, ..., 1, 0, $l = n_L$–1, ..., 1, 0, c = 0, 1, ..., n_C–1, p = 0, 1, ..., n_R–1, where “*P*” denotes the ASCII code of the letter *P* and *p* denotes the index of this precinct.

The protected codestream is generated by encrypting each packet body with its cor responding key (a leaf of the key tree).

To render a sub-image from a protected codestream, a user obtains the corresponding access keys (e.g., granted from a key server). These access keys are able to exactly recover the key tree leaves corresponding to the packets of the requested sub-image. The process of key reconstruction is similar to that of key tree generation. The leaves are used to decrypt the corresponding packets.

B.1.5 Bitstream Syntax

Table B.1 illustrates the structure of SEC segment. The *ZOI* field signals the granted parameters, P_{ID} field signals the protection method parameters for this access control scheme. The PM_{ID} field is always set to 1 to notify that the decryption template is used. The TP_{ID} field signals the additional parameters for th is access control scheme. *KTO* is the key tree generation order. The field L_{aki} indicates the length of the access key information.

Table B.1 Recommended parameter for thi s scheme

t	i	ID	L_{ZOI}	ZOI	L_{PID}	P_{ID}

Parameter	Size (bits)	Values	Meaning
t	1	1	Registration authority protection tool
i	7	*Instance value*	Tool instance identifier
ID	32	*Tool ID value*	Registered ID
L_{ZOI}	16	$[2, 2^{16}-1]$	Length of L_{ZOI} + ZOI.
ZOI	Variable	See Sec 5.5	Zone of influence for this scheme.
L_{PID}	16	$[2, 2^{16}-1]$	Length of L_{PID} + P_{ID}
P_{ID}	Variable	See Table B.2	Parameters for this scheme

Table B.2 P_{ID}

PM_{ID}=1	T_{dcry}	TP_{ID}

Parameter	Size (bits)	Values	Meaning
ID_T=1	8	Always set to 1	Tag for Decryption template
T_{dcry}	Variable	*Decryption template values*	Decryption template
TP_{ID}	Variable	See Table B.3	Additional information fo r this scheme

Table B.3 TP_{ID}

KTO	L_{aki}	AK_{Info}

Parameter	Size (bits)	Values	Meaning
KTO	8	$0 – (2^8-1)$	Key Tree order. It may be different from the code-stream progressio n order, Tentatively, 0x00: LRCP 0x01: RLCP 0x02: RPCL 0x03: PCRL 0x04: CPRL others: Reserved
L_{aki}	16	$0 – (2^{16}-1)$	Length of access key information, if L_{aki} = 0, no AK_{Info} filed is presented.
AK_{Info}	Variable	See Table B.4	Information on th e access key (e.g. length of key, number of keys,)

Table B.4 AK_{info}

L_{uk}	UK	E_{ak}	N_{ak}	AK

Parameter	Size (bits)	Values	Meaning
L_{uk}	16	$0 – (2^{16}-1)$	Length of the user key
UK	L_{uk}	NaN	User key information
E_{ak}	16	See Table 19	Cipher used to encrypt the access keys
N_{ak}	16	$0 – (2^{16}-1)$	Number of access keys
AK	N_{ak} * K_{bs}	NaN	Access keys

B.1.6 Conclusion

This technology enables a publisher to protect a JPEG 2000 codestream with a master key. The protected codestream is permitted to be delivered to any number of users, but the keys for packets are kept secret. The key server generates different access keys for the users according to their priorities. The users generate the granted packets keys from their access keys and get different granted images. That is to say, the technology has the property called "*encrypt once, access many ways*".

[Note: Appropriate reference will be added when document repository is created.

The complete version of this example can be found as ISO/IEC JTC1/SC29/WG1 N3311.]

B.2 A Unified Authentication Framework for JPEG 2000 images

B.2.1 Operational Description

This JPSEC tool provides the following JPSEC services: image data/content integr ity verification and source authentication, i.e., fragile/semi-fragile authentication for JPEG 2000 images based on digital signature schemes.

As this tool supports both fragile and semi-fragile authentications, it can be used in different application scenarios, including image distribution, image streaming, medical and military imaging, law enforcement, E-commerce and E-government.

In pervasive environment, images might experience various kinds of incidental distortions like transcoding and format conversion. Traditional cryptography-based authentication techniques protect JPEG 2000 images at data integrity level and cannot survive these types of content-preserving distortions. Therefore semi-fragile authentication techniques are required to protect JPEG 2000 images at image content level. This tool unifies both image data and image content authentication and proposes a new concept called lowest authentication bit rate (LABR). That is, if the image is transcoded to a bit rate that is not less then the LABR, it will be rendered as authentic, otherwise, unauthentic. The authentication may be fragile or semi -fragile authentication. In semi-fragile authentication, the tool is able to identify the place where alteration has taken place when the image is deemed unauthentic.

B.2.2 Technical Overview

To provide fragile and semi-fragile authentication, a group of techniques has been applied in this JPSEC informative tool. They include feature selection, digital signature, lossy and lossless data hiding, and ECC (error correction codes). According to the LABR specified by users, the corresponding features are selected based on an analysis applied to JPEG 2000 structure, and digital signature is then generated. For semi-fragile authentication, ECC is utilized to enhance the robustness level. The parity check bits (PCB) are embedded into the image as watermark so as to identify locations of the attack. Data embedding may be conducted in two different ways: lossy and lossless. With lossy data hiding, the original image cannot be recovered after data hiding. With lossless data hiding, on the other hand, the image is modified in a reversible way, i.e., the original image can be recovered if the marked image has not been altered. The lossless semi-fragile authentication is useful for JPEG 2000 since the standard supports lossy-to-lossless compression. In particular, it is useful to medical imaging and remote-imaging applications, where lossless is an essential requirements.

Similar to image compression bit rate, which is used to control and characterize the compression strength, The LABR (lowest authentication bit rate) parameter is used to quantitatively control the protection strength. For example, when a JPEG 2000 image is protected with a LABR of 2 bpp (bits per pixel), any transcoded version of the image will be rendered as authentic by the proposed system as long as the bit rate after transcoding is greater than or equal to 2 bpp.

The block diagram below illustrates how the tool can be used to protect images.

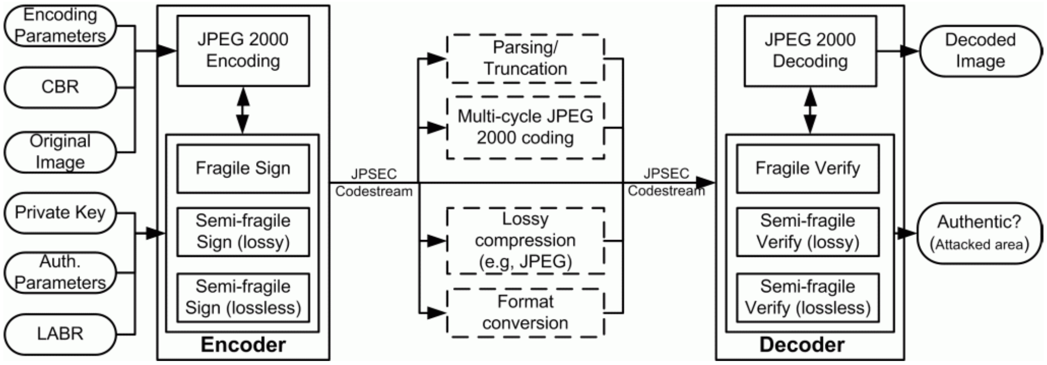


Figure B.1. Image protection using unified authentication framework for JPEG 2000

This tool can use different signaling syntaxes depending on the chosen authentication method. For fragile authentication, it uses the JPSEC template tool syntax, as defined in Section 5.6.3. For semi-fragile authentication, it uses the JPSEC non-normative tool syntax, as illustrated in the table below. In addition, the F_{INSEC} should be set to 0 as INSEC marker is not used by this tool, and F_{J2K} should be set to 1 because the resulted bitstream of this JPSEC tool is still compliant with JPEG 2000 part 1.

Table B.5 Syntax for semi-fragile authentication

Parameter				Size (bits)	Value	Derived meaning
t				1	1	Non-normative tool syntax is used
i				7	0 – (2^7-1)	Tool instance index
ID				32	0 – (2^{32}-1)	ID number to be assigned by RA
L_{ZOI}				16	0 – (2^{16}-1)	Length of ZOI
ZOI				Variable	*ZOI values*	The covered zone in the image protected by the tool.
L_{PID}				16	0 – (2^{16}-1)	Length of P_{ID} and L_{PID} in bytes
P_{ID}	ID_T			8	2	Authentication Template is used, as defined in Table 16
	T_{auth}	M_{auth}		8	2	Digital Signature Method is used, as defined in Table 30
		P_{auth}	M_{DS}	8	See Table 37	Digital signature algorithm used, such as DSA or RSA
			H_{DS}	8	See Table 33	Hashing function used
			KT_{DS}	Variable	*Key template values*	The public key is stored in KT_{DS}. This tool uses one public key only.
			SIZ_{DS}	16	0 – (2^{16}-1)	Size of the digital signature in bytes
	PD			1	0	The BAS structure is terminated.
				2	1	Processed in quantized wavelet coefficients domain
				5	xxxxx	Reserved
	G	PO		16	*Processing order values*	Processing order
		GL		8	0000 1111	Granularity level: generate one signature for the entire bitstream

	V	N_V	16	1	Number of digital signatures in the list is 1.
		S_V	8	1 – (2^8-1)	Size of the digital signature in bytes
		VL	1* S_V	*Digital signature value*	The digital signatures generated by the tool.
	LABR	$LABR_{int}$	8	0 – (2^8-1)	The integer part of LABR
		$LABR_{fra}$	8	0 – (2^8-1)	The fractional part of LABR
	Threshold		8	[0,2^8-1]	The threshold value. (Valid only for lossless authentication)
	Shuffle		8	[0,2^8-1]	The number of shuffling in order to embed watermark bits (Valid only for lossless authentication)

The unique ID of this tool is to be assigned by the registration authority. The tool description can be downloaded from the Registration Authority (RA) using the assigned ID.

B.2.3 Conclusions

In summary, this tool has achieved the following special features:

- Authentication for JPEG 2000 images at either image data level or image content level by integrating fragile and semi-fragile authentication in one framework. Furthermore, the semi-fragile authentication includes both lossy and lossless modes.
- Robustness against various incidental distortions like introduced by transcoding, format conversion, lossy compression and multi-cycle of JPEG 2000 coding. Therefore, this tool can be used to protect JPEG 2000 images in pervasive environment.
- Scalable protection of the JPEG 2000 images. Specifically, this tool is able to protect any tile, component, resolution, layer, precinct, or codeblock.
- Compatibility with state-of-arts information security framework called Public Key Infrastructure which is the basis of existing international standards like X.509.
- Quantitative protection strength controlled by a single parameter called LABR, which brings much convenience to end users.
- Capability to locate the possibly attacked image areas if the image is deemed unauthentic. It could help to visually convince the users.
- Support for lossy-to-lossless protection, corresponding to lossy-to-lossless compression of JPEG 2000 coding standards. Thus, the tool has much broader applications, including medical imaging and remote imaging applications.

[Note: Appropriate reference will be added when document repository is created.

More detailed technical information and implementation information can be found in WG1N2946, WG1N3074, WG1N3107, WG1N3308, or at http://web.njit.edu/~shi/JPEG2000.]

B.3 A Simple Packet-based Encryption Method for JPEG 2000 bitstreams

B.3.1 Operational description

This clause presents a selective encryption technique for JPEG 2000 images. It is based on a packet level encryption and on standard robust cipher algorithms.

The security service addressed by the technique is co nfidentiality of JPEG 2000 images, obtained through ciphering of the codestream. Consequently, IPR protection as well as privacy protection can be achieved using this technique.

The approach supports transcoding, scalability, and other content processing f unctionality without having to access the cryptographic key and to perform decryption and re -encryption. It does not interfere with the coding and decoding processes and have very limited adverse impact on the compression efficiency and no adverse impact on error resilience. Such an approach allows a maximum flexibility to implement scenarios and applications with various levels of security.

The technique may be used by content producers to limit the access to the image content or by content providers to insure confidential delivery of the content to the end users.

B.3.2 Technical overview

The technique consists of encrypting the codestream after compression of the image, as shown in Figure B.2.

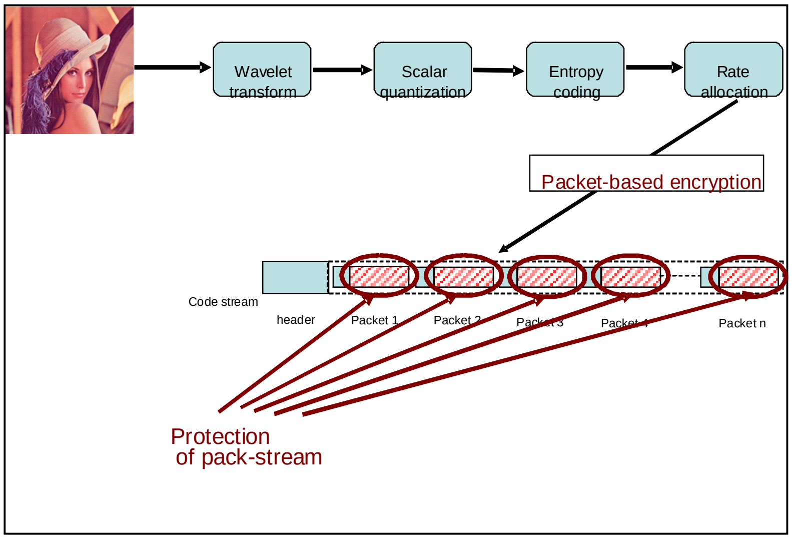


Figure B.2 — Packet-based Encryption Principle

This JPSEC tool can take several image -related parameters as an input: resolution levels, quality layers, components, precincts or tiles. Only the packet payloads correspondi ng to these input parameters are then processed. Thus the protected codestream keeps a regular JPEG 2000 structure. Once the codestream has been ciphered, the SEC marker segment is added to the main header to allow any JPSEC decoder to correctly decrypt the image later on.

This method uses well known standard underlying algorithms to selectively encrypt the packets: DES or AES methods, associated with standard modes described in [1] such as ECB, CBC, CFB, OFB and CTR. Any other block cipher algorithms could of course be used: DES and AES are given here as examples of standard ciphers.

B.3.2.1 Signalling example

The technique can be signalled with the template based syntax of the normative clause. Below is an example of signalling for this techniwe, which specifies one zone for the ZOI, but of course there could be more, following the same syntax as Zone^{0}.

Parameter			Value (in order)	Size (bits)	Derived meaning
NZzoi			1	8	Number of Zones is one.
Zone^{0}	DCzoi		0	1	The byte aligned segment does not follow
			0	1	Image related description class.
			101100	6	Image regions, resolution levels, quality layers and components are specified in order.
	Pzoi^{1}	Mzoi^{1}	0	1	The byte aligned segment does not follow.
			0	1	The specified zones are influenced by the protection method.
			0	1	Single item is specified.
			00	2	Rectangle mode.
			00	2	Izoi uses 8 bits integer.
			1	1	Izoi is described in two dimensions.
		Izoi^{1}	0110 0100	8	Xul is 100.
			0111 1000	8	Yul is 120.
			1011 0100	8	Xlr is 180.
			1101 0010	8	Ylr is 210.
	Pzoi^{3}	Mzoi^{3}	0	1	The byte aligned segment does not follow.
			1	1	The specified zones are not influenced by the protection method.
			0	1	Single item is specified.
			11	2	Max mode.
			00	2	Izoi uses 8 bits integer.
			0	1	Izoi is described in one dimension.
		Izoi^{3}	0000 0010	8	Resolution levels $\leq$ 2 are specified. (i.e. Resolution levels > 3 are specified with Max mode and complement switch.).
	Pzoi^{4}	Mzoi^{4}	0	1	The byte aligned segment does not follow
			0	1	The specified zones are influenced by the protection method.
			0	1	Single item is specified.
			11	2	Max mode.
			00	2	Izoi uses 8 bits integer.
			0	1	Izoi is described in one dimension.
		Izoi^{4}	0000 0101	8	layers $\leq$ 5 are specified with Max mode.

Table B.6 Example of Zone of Influence, with spatial coordinates, resolutions and layers

Parameter					Value (in order)	Size (bits)	Derived meaning
P_{PM}	ME_{decry}				0000 0000	8	NULL: no marker emulation prevention method
	CT_{decry}				0x0003	16	Cipher identifier: AES (block cipher)
	CP_{decry}	M_{bs}			10 0000	6	Cipher mode: CBC
		P_{bs}			01	2	Padding mode (PKCS#7 -padding)
		SIZ_{bs}			0001 0000	8	Size of block: 16 bytes (128 bits)
		KT_{bs}	LK_{KT}		0x00C0	16	Size of key: 192 bits
			KID_{KT}		0000 0011	8	Key information is a URI
			LKI_{KT}		0x0021 (=33)	16	Length of the URI: 33 bytes
			KI_{KT}		https://server/path/secretkey.pem	264	This URI is an https URL; it has to be understood by the application using JPSEC. The effective retrieval of the key is beyond the standard.
			G_{KT}	PO	000 001 010 011 100 00	16	Processing order is TRLCP
				GV	0000 1001	0	Granularity of key is Zone in ZOI
			V_{KT}	Nv	0	16	Single key value in KI_{KT}; Values not specified in V_{KT}

Table B.7 Decryption Template Description, in the case of AES-192/CBC

Parameter	Value (in order)	Size (bits)	Derived meaning
PD	0 10 00100	8	codestream domain: packet bodies only are encrypted

Table B.8 Processing Domain syntax

Parameter		Value (in order)	Size (bits)	Derived meaning
G	PO	000 001 010 011 100 00	16	Processing order is TRLCP
	GV	0000 0110	8	Unit of protection is packet
V	N_V	1	16	Number of IV values specified
	Sv	16	8	IV size in bytes
	VL	*Value*	128	IV value

Table B.9 Granularity and Value List Syntax

B.3.3 Conclusion

The technique presented in this clause demonstrates selective encryption for JPEG 2000 images. It is based on a packet level encryption and on standard robust cipher algorithms. It can be signalled using the templates defined in the normative clause of this ISO/IEC International Standard| Recommendation and supports various levels of complexity.

Morris Dworkin, *Recommendation for Block Cipher Modes of Operation*, *Methods and Techniques*, NIST Special Publication 800-38A

B.4 Encryption tool for JPEG 2000 access control

B.4.1 Security services addressed

This technology provides an encryption tool which can prevent marker emulation in an encrypted codestream.

B.4.2 Typical applications

This technology allows selective and full encryption of JPEC 2000 codestreams. Such selective encryption methods can be used to display only an approved image, such as a thumnail, a low qu ality image, and a partially scrambled image.

B.4.3 Potential users, implementation model and motivations

Basically, this technology is based on a packet-based encryption for a JPEG 2000 codestream with well-known cipher algorithm. Specifically, this technology prevents marker emulation in the encrypted codestream. Therefore, even if the resulting encrypted codestream is input to JPEG 2000 part-1 compatible decoder, the decoder is unlikely to crash and can play the protected image correctly.

B.4.4 Technical overview

(1) Encryption

***Step 1*.** 2 (bytes) code is temporarily encrypted using well -known cipher algorithm.

***Step 2*.** If the temporarily encrypted code or its relating code is more than 0xFF8F, th en the 2 (bytes) code is not encrypted.

Otherwise, the temporarily encrypted code is output as the encrypted code.

***Step 3*.** Moving on the next 2 (bytes) code, and Step 1 and Step 2 are continued.

All 2 (bytes) code in the plain text shall be less than 0xFF9 0 according to part-1 specification. Further, if the temporarily encrypted code or its relating code is more than 0xFF8F, then the 2 (bytes) code is not encrypted. As a result, all 2 (bytes) code in the cipher text is less than 0xFF90.

(2) Decryption

***Step 1*.** 2 (bytes) code is temporarily decrypted using the same cipher algorithm as the encryption.

***Step 2*.** If the temporarily decrypted code or its relating code is more than 0xFF8F, then the 2 (bytes) code is not decrypted. Otherwise, the temporarily decrypted code is output as the decrypted code.

***Step 3*.** Moving on the next 2 (bytes) code, and Step 1 and Step 2 are continued.

All 2 (bytes) code in the original plain text before encryption shall be less than 0xFF90. So it is possible to make a decision that the 2 (bytes) code is not encrypted if the temporal decrypted code or its relating code is more than 0xFF8F.

B.4.5 Signaling method

Table B.10 shows recommended parameters in this technology. Any parameters for this technology shall be signalled according to the syntax which is identified in JPSEC. Especially, this technology should use “decryption” template, “packet” granularity, and “bitstream” processing domain with the appropriate ZOI.

Table B.10 — Recommended parameter in this technology

Parameter	Values	Size (bits)	Meaning
SEC	0xFF65	16	SEC marker.
L_{SEC}	Variable	16	Length of SEC marker segment.

Z_{SEC}		1 (example)	8	Index of this SEC marker segment.
P_{SEC}	F_{INSEC}	1	1 (example)	INSEC is used.
	$F_{multiSEC}$	1	0	One SEC marker segment is used.
	F_{J2K}	2	01	Compliant with JPEG 2000 part 1
	F_{TRLCP}	1	0	TRLCP tag usage is not defined.
	N_{tools}	7	0000001	Number of security tool is one.
	I_{max}	7	0000000	Maximum tool instance index is zero.
	padding	5	00000	Padding bits
t		1	1	RA protection JPSEC non -normative tool
i		0000000 (example)	7	Tool instance index
ID		0 (temporally)	32	Registered ID
L_{zoi}		9	16	Length of ZOI is 9 bytes.
ZOI		See Table B.11 (example)	variable	Zone of Influence for this tool.
L_{PID}		Variable	16	Length of L + T + PD + G
P_{ID}		See Table B.12 (example)	variable	Parameters for this technology

Table B.11 — Example ZOI of this key generation tool

Parameter			Value (in order)	Size (bits)	Derived meaning
NDzoi			1	8	Number of Zone is one.
Zone^0	Dczoi		0	1	The byte aligned -segment does not follow.
			0	1	Image related desription class.
			101000	6	Image regions and resolution levels are specified in order.
	Pzoi^1	Mzoi^1	0	1	The byte aligned -segment does not follow.
			0	1	The specified zones are influenced by the protection method.
			0	1	Single item is specified.
			00	2	Rectangle mode.
			00	2	Izoi uses 8 bits integer.
			1	1	Izoi is described in two dimensions.
		Izoi^1	0110 0100	8	Xul is 100.
			0111 1000	8	Yul is 120.
			1011 0100	8	Xlr is 180.
			1101 0010	8	Ylr is 210.
	Pzoi^3	Mzoi^3	1	1	The specified zones are not influenced by the protection method.
			0	1	Single item is specified.
			11	2	Max mode.
			00	2	Izoi uses 8 bits integer.
			0	1	Izoi is described in one dimension.
			0	1	The byte aligned -segment does not follow.

		Izoi^{3}	0000 0010	8	Resolution levels > 3 are specified.

Table B.12 — PID for this technology

Parameter		Values	Size (bits)	Meaning
T		See Table B.13	Variable	Decryption templates
PD		0	1	Subsequent BAS byte does not exist .
		10	2	Codestream domain
		00000	5	(Not used)
G	PO	000 001 010 011 100 0 0	16 (example)	Processing order is tile-resolution-layer-component-precinct.
	GL	0000 0110	8	Unit of protection is packet
Skip		0 (example)	8	*Skip* parameter for this tool.s

Table B.13 — Example of decryption template of this technology

Parameter		Value (in order)	Size (bits)	Derived meaning
ME$_{decry}$		1	8	Marker emulation has not occurred.
CT$_{decry}$		3	16	Block cipher (AES)
CP$_{decry}$	M$_{bs}$	10 0010	6	OFB mode is used. (Bits are not padded.)
	SIZ$_{bs}$	128	16	Block size (128 bits)
	KT$_{bs}$	*Key template values*	Variable	Key template
	IVsc	*Inivial vector value*	128	Initial vector value

B.4.6 Conclusion

This section described an encryption technology for a JPEG 2000 codestream. The significant advantage of this technology is to prevent a marker emulation in the encrypted codestream, namely part -1 compatible.

[Note: Appropriate reference will be added when document repository is created.

Further detailed description is available in ISO/IEC JTC 1/SC 29/WG1N 3205.]

[Note: This technology is being evaluated by the contributors for proof of reversibility.]

B.5 Key generation tool for JPEG 2000 access control

B.5.1 Security services addressed

This technology provides an image related access control for JPEG 2000 according to a hierarchical structure in a JPEG 2000.

B.5.2 Typical applications

A typical application of this technolo gy is secure image distribution where only an authorized user can play the accepted image. For example, a thumnail is free to display, but a large resolution image can be decoded by only the user who owns the key.

B.5.3 Potential users, implementation model and motivations

This technology supports to generate keys to be used in a secure JPEG 2000 image distribution. This technology is based on an image related access control, such as image region, resolution and image quality. The principle of this technology is to generate encryption and decryption keys hierarchically using a cryptographic one-way hash function such as a hash function.

B.5.4 Technical overview

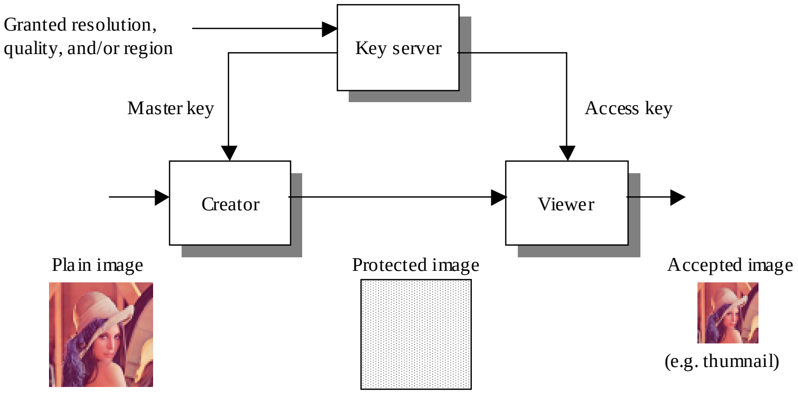


Figure B.3 — Overview of this technology

In the encryption stage, a key server generates a master key. Then, a creator encrypts an image using packet keys which are generated from the master key. In the decryption stage, a key server generates an access key according to granted resolution, quality, and/or re gion. Then, a viewer decrypts the encrypted image using packet keys which are generated from the access key. Note that these keys are sequentially generated based on secure hash chain.

Specifically, this technology uses the following access control policy: "if a user can access to a resolution level / layer, then that user can also access to the lower resolution levels / layers ". On the other hand, even if a user can access to a tile, that user can not access to the other tiles at all. "

The significant advantage of this technology is that the number of keys needed to pass on from a key server to a viewer is much less than the conventional case. This means that this technology allows for smaller overhead in terms of storage usage.

B.5.5 Signaling method

Table B.14 shows recommended parameters in this technology. Any parameters shall be signalled according to the syntax which is identified in JPSEC. Especially, this tool should use "decryption" template, "packet" granularity, and "bitstream" processing domain with th e appropriate ZOI.

Table B.14 — Recommended parameter in this technology

Parameter		Size (bits)	Values	Meaning
SEC		16	0xFF65	SEC marker.
L_{SEC}		16	0 – 255	Length of SEC marker segment.
Z_{SEC}		8	1 (example)	Index of this SEC marker se gment.
P_{SEC}	F_{INSEC}	1	1 (example)	INSEC is used.
	$F_{multiSEC}$	1	0	One SEC marker segment is used.
	F_{J2K}	2	01	Compliant with JPEG 2000 part 1
	F_{TRLCP}	1	0	TRLCP tag usage is not defined.
	N_{tools}	7	0000001	Number of security tool is one.
	I_{max}	7	0000000	Maximum tool instance index is zero.
	padding	5	00000	Padding bits
t		1	1	JPSEC non-normative tool
i		7	0 (example)	Instance index for this tool
ID		32	5 (temporally)	Registered ID for this tool
L_{zoi}		16	Variable	Length of ZOI for this tool
ZOI		variable	*ZOI value*	Zone of Influence for this tool.
L_{PID}		16	Variable	Length of L + T + PD + G
P_{ID}		variable	See Table B.16	Parameters for this technology

Table B.15 — Example ZOI of this key generation tool

Parameter			Value (in order)	Size (bits)	Derived meaning
NDzoi			1	8	Number of Zone is one.
Zone^0	Dczoi		0	1	Image related description class.
			101000	6	Image regions and resolution levels are specified in order.
			0	1	The byte aligned -segment does not follow.
	Pzoi^1	Mzoi^1	0	1	The byte aligned-segment does not follow.
			0	1	The specified zones are influenced by the protection method.
			0	1	Single item is specified.
			00	2	Rectangle mode.
			00	2	Izoi uses 8 bits integer.
			1	1	Izoi is described in two dimensions.
		Izoi^1	0110 0100	8	Xul is 100.

			0111 1000	8	Yul is 120.
			1011 0100	8	Xlr is 180.
			1101 0010	8	Ylr is 210.
	Pzoi[3]	Mzoi[3]	1	0	The byte aligned-segment does not follow.
			1	1	The specified zones are not influenced by the protection method.
			0	1	Single item is specified.
			11	2	Max mode.
			00	2	Izoi uses 8 bits integer.
			0	1	Izoi is described in one dimension.
		Izoi[3]	0000 0010	8	Resolution levels > 3 are specified.

Table B.16 — PID for this technology

Parameter		Size (bits)	Values	Meaning
T		variable	See Table B.17	Decryption templates
PD		0	1	Subsequent BAS byte does not exist .
		10	2	Codestream domain
		00000	5	(Not used)
G	PO	000 001 010 011 100 0	16 (example)	Processing order is tile-resolution-layer-component-precinct.
	GL	0000 0110	8	Unit of protection is packet
H		16	See Table 38 in section 5.5.3.1	Hash function for this key generation tool
L_k		8	0 - 255	Length of access key information
AK_{info}		Variable	*Access key value*	Access key information (this information is encrypted using KT_{bs} in T.)

Table B.17 — Example of decryption template of this technology

Parameter		Value (in order)	Size (bits)	Derived meaning
ME_{decry}		1	8	Marker emulation has not occurred.
CT_{decry}		3	16	Block cipher (AES)
CP_{decry}	M_{bs}	10 0010	6	OFB mode is used. (Bits are not padded.)
	SIZ_{bs}	128	16	Block size (128 bits)
	KT_{bs}	See section 5.6.5	Variable	Key template
	IVsc	*Inivial vector value*	128	Initial vector value

B.5.6 Conclusion

This section described an image related access control technology for a JPEG 2000 codestream. The significant advantage of this technology is that the number of keys to be managed and to be accessed is much less than the conventional case.

[Note: Appropriate reference will be added when document repository is created.

Further detailed description is available in ISO/IEC JTC 1/SC 29/WG1N 3206.]

B.6 Wavelet and Bitstream Domain Scrambling for Conditional Access Control

B.6.1 Summary

Access control to an image is an important function ality in secure imaging. Often it is desirable to give access to a small resolution thumbnail or a low quality image, while access to higher resolutions or qualities are subject to authorization. In this clause, a technique for conditional access control i s presented. The method was initially presented in [1]. Basically, it adds pseudo -random noise to the image. Authorized users know the pseudo-random sequence and thus can remove this noise. In opposite, unauthorized users only have access to severly distorted images. The system is composed of three main components: scrambling, pseudo -random number generator and encryption algorithm. In order to fully exploit and retain the properties of JPEG 2000, the scrambling is selectively applied on the code-blocks composing the code stream. Consequently, the distortion level introduced in specific parts of the image can be controlled. This enables access control by resolution, quality or regions of interest in an image.

B.6.2 Technical Overview

The system is composed of thre e main components:

- Scrambling: two approaches are supported. The scrambling is either performed on the quantized wavelet coefficients, or directly on the bitstream. In the first case, the signs of the coefficients in each code-block are inverted pseudo-randomly. In the second case, the bits of the bitstream are pseudo - randomly inverted.

- Pseudo-random number generator (PRNG): the PRNG is used to drive the scrambling. It is based on a seed value. In a preferred embodiment of the technique, the SHA1PRNG algori thm [2] with a 64-bit seed is used for the pseudo-random number generator (PRNG). Note that other PRNG algorithms could be used as well.

- Encryption algorithm: to communicate the seeds to authorized users, they are encrypted and inserted in the bitstream. . In a preferred embodiment of the technique, the RSA algorithm is used for encryption [3]. Other encryption algorithms could be used as well. The length of the key can be selected at the time the image is protected.

Two block diagrams below correspond to t he two cases of wavelet and bitstream domain scrambling.

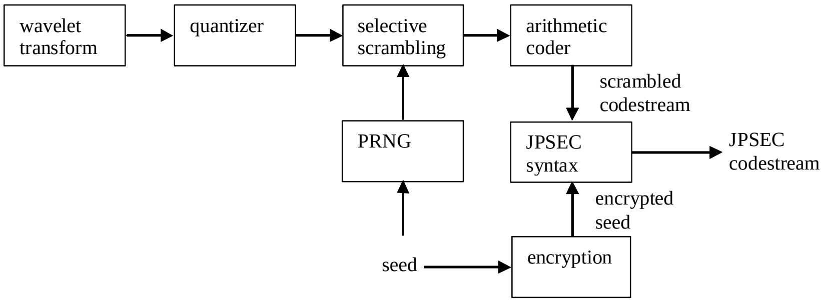


Figure B.4 — Block Diagram for wavelet domain scrambling

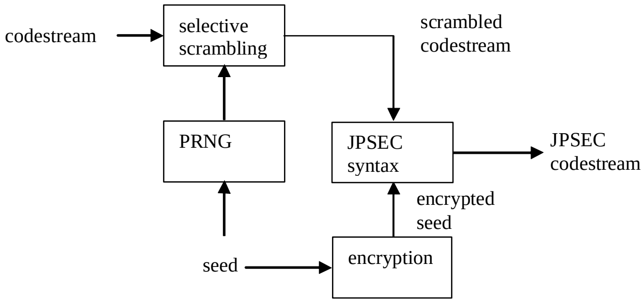


Figure B.5 — Block Diagram for bitstream domain scrambling

In order to improve the security of the system, the seed can be changed from one code -block to another. Also, several levels of access can be defined, using different encryption keys. Th e syntax given below is very flexible and supports the usage of multiple seeds and multiple keys.

B.6.3 Bitstream syntax

In this example, both the SEC and INSEC marker segments are used. The bitstream syntax is defined below. The SEC marker segment is using the Non-Normative Protection Tool syntax. The INSEC marker segment is used to signal which codeblocks are scrambled and which seeds are used.

B.6.3.1 Syntax for SEC marker segment

The Non-Normative Protection Tool syntax is used. As the Zone of Influence is not needed in this example, the syntax of ZOI$^{(i)}$ is reduced to NZ$_{ZOI}$=0. In the case of multiple keys, several instances of the tool are used in the SEC marker segment. More specifically, several instances i=0, 1, 2, … with the same ID are present, each one correspon ding to a different key identification KeyID$^{(i)}$. This is illustrated below.

t	i=0	ID	$L_{ZOI}^{(0)}$	ZOI$^{(0)}$	$L_{PID}^{(0)}$	$N_S^{(0)}$	KeyID$^{(0)}$	Data
t	i=1	ID	$L_{ZOI}^{(1)}$	ZOI$^{(1)}$	$L_{PID}^{(1)}$	$N_S^{(1)}$	KeyID$^{(1)}$	Data
t	i=2	ID	$L_{ZOI}^{(2)}$	ZOI$^{(2)}$	$L_{PID}^{(2)}$	$N_S^{(2)}$	KeyID$^{(2)}$	Data

Figure B.6 — Non-Normative Protection Tool syntax in the case of multiple keys

With the following semant ic for P$_{ID}$:

Table B.18 — Syntax and semantic for P$_{ID}$

Parameters	Size (in bits)	Meaning
N_s	16	The number of seeds used by this instance
KeyID	32	Identification of the key to be used for decrypting
Data	Variable	The encrypted seeds

B.6.3.2 Syntax for INSEC marker segment

To include the information of which seed is used to protect which codeblocks, the in -codestream security marker (INSEC) is also used. In this example, it is adde d before the secured codeblock(s), to indicate which seed has been used to protect this/these codeblock(s). Instead of indicating the seed itself, the marker contains an index which refers to the seeds in the main header SEC marker segment. As in this example the INSEC information applies to the following code -blocks, R is always equal to 1. The syntax of AP is different in the case of wavelet scrambling and bitstream scrambling:

S_{idx}

Off	S_{idx}

Figure B.7 — Syntax for AP: Wavelet domain scrambling (left), Bitstream domain scrambling (right)

With the following semantic:

Table B.19 — Syntax and semantic for AP

Parameters	Size (in bits)	Meaning
Off	16	The offset in the code-block bitstream of the first scrambled byte
S_{idx}	16	The seed index for the code-block

In the case of multiple keys, the combination of the tool instance i and the seed index S_{idx} uniquely identifies which seed/key this INSEC marker segment is referring to.

B.6.4 Conclusions

In this clause, a security tool was presented for conditional access control to JPEG 2000 images. The technique introduces a pseudo -random noise to selected parts of the bitstream. Per consequent, the decoded image appears much distorted for an unauthorized decoder which does not know how to remove this noise.

The security of the technique depends on the security of the specific algorithms for pseud o-random number generator and encryption of the seed, in our preferred embodiment SHA1PRNG and RSA respectively. SHA1PRNG is a secure PRNG, as no knowledge of the sequence can be deduced by knowing some of the numbers in the sequence. In this example, the PRNG seed is 64 bits which should make a brute force attack unfeasible. The seeds are encrypted with RSA using a user defined key length. RSA is regarded as a secure algorithm, provided a sufficient key length is used.

Note: Appropriate reference will be added when document repository is created.

[1] R. Grosbois, P. Gerbelot, T. Ebrahimi, "Authentication and access control in the JPEG 2000 compressed domain", In Proc. of the SPIE 46th Annual Meeting, Applications of Digital Image Processing XXIV, San Dieg o, July 29th -August 3rd, 2001.

[2] http://java.sun.com/j2se/1.4.2/docs/guide/security/CryptoSpec.html , Java Cryptography Architecture API Specification and reference.

[3] R.L. Rivest, A. Shamir, and L.M. Adleman, “A method for obtaining digital signatures and public -key cryptosystems”, Communications of the ACM (2) 21, 1978, Page(s): 120 -126.

B.7 Progressive Access for JPEG 2000 codestream

B.7.1 Security services addressed

This method provides a non-image related access control for JPEG 2000 according to a progression order in a codestream.

B.7.2 Typical applications

A typical application of this technology is secure image distribution where only an authorized user can play the accepted image. ExpeciallySpeciffically, this technology is suitable for access control according to a progression order in a codestream.

B.7.3 Potential users, implementation model and motivations

The challenge in the design of the access control scheme is to strike a de licate balance among security, efficiency and flexibility. This access control technique for a JPEG 2000 codestream constructs a hash chain to generate the keys for each packet so as to encrypt packets in the code -stream. Therefore, only users with the right security clearance can decrypt the packets corresponding to the granted image in the codestream.

B.7.4 Technical overview

In the encryption stage, a key server generates a master key. Then, a creator encrypts a codestream using packet keys which are generated from the master key. In the decryption stage, a key server generates an access key according to granted packet. Then, a viewer decrypts the encrypted codestream using packet keys which are generated from the access key.

Specifically, this technology uses the following access control policy: “if a user can access to a packet, then that user can also access to the antecedent packets in a codestream ”. Therefore, we call such kind of access control “Progressive Access”.

The significant advantage of this technology is that the number of keys needed to pass on from a key server to a viewer is much less than the conventional case. This means that this technology allows for smaller overhead in terms of storage usage.

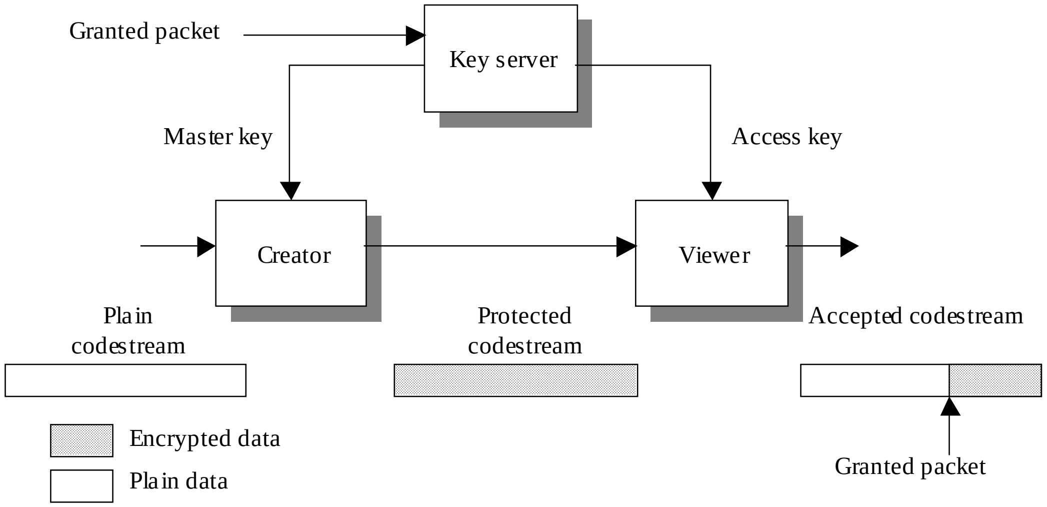


Figure B.8 — Technical overview of this technology

B.7.5 Signaling method

Table B.20 shows recommended parameters in this technology. Any parameters shall be signaled according to the syntax which is identified in JPSEC. Especially, this technology sho uld use “decryption” template, “packet” granularity, and “bitstream” processing domain with the appropriate ZOI.

Table B.20 — Recommended parameter in this tool

Parameter		Size (bits)	Values	Meaning
SEC		16	0xFF65	SEC marker.
L_{SEC}		16	Variable0 – 255	Length of SEC marker segment.
Z_{SEC}		8	1 (example)	Index of this SEC marker segment.
P_{SEC}	F_{INSEC}	1	1 (example)	INSEC is used.
	$F_{multiSEC}$	1	0	One SEC marker segment is used.
	F_{J2K}	2	01	Compliant with JPEG 2000 part 1
	F_{TRLCP}	1	0	TRLCP tag usage is not defined.
	N_{tools}	7	0000001	Number of security tool is one.
	I_{max}	7	0000000	Maximum tool instance index is zero.
	padding	5	00000	Padding bits
t		1	1	RA protection tool
i		7	0 (example)	Instance index
ID		32	7 (temporarily)	Registered ID
L_{zoi}		16	Variable	Length of ZOI
ZOI		variable	See Table B.21 (example)	Zone of Influence for this tool.
L_{PID}		16	Variable	Length of L + T + PD + G
P_{ID}		variable	See Table B.22 (example)	Parameters for this tool

Table B.21 — Example ZOI of this technology

Parameter			Value (in order)	Size (bits)	Derived meaning
NDzoi			1	8	Number of Zone is one.
Zone^0	DCzoi		1	1	Non-image related description class.
			000100	6	Packet are specified.
			0	1	The byte aligned-segment does not follow.
	Pzoi^4	Mzoi^4	01	11	The byte aligned-segment does not follow.The specified zones are not influenced by the protection method.
			11	11	The specified zones are not influenced by the protection method.Multiple items are specified.
			101	12	Multiple items are specified.Max mode.
			0101	22	Max mode.Izoi uses 8 bits integer.
			010	21	Izoi uses 8 bits integer.Izoi is described in one dimensions.
			00	11	Izoi is described in one dimensions.The byte aligned-segment does not follow.

		Izoi^{11}	0000 1010	8	Packet index > 10 is identified

Table B.22 — P_{ID} for this technology

Parameter		Size (bits)	Values	Meaning
T		variable	See Table B.23	Decryption templates
PD		0	1	Subsequent BAS byte does not exist .
		10	2	Codestream domain
		00000	5	(Not used)
G	PO	000 001 010 011 100 0	16 (example)	Processing order is tile-resolution-layer-component-precinct.
	GL	0000 0110	8	Unit of protection is packet
H		16	See Table 38 in section 5.5.3.1	Hash function for this key generation tool
L_k		8	0 - 255	Length of access key information
AK_{info}		vVariable	*Access key value*	Access key information (this information is encrypted using KT_{bs} in T.)

Table B.23 — Example of decryption template of this technology

Parameter		Value (in order)	Size (bits)	Derived meaning
ME_{decry}		1	8	Marker emulation has not occurred.
CT_{decry}		3	16	Block cipher (AES)
CP_{decry}	M_{bs}	10 0010	6	OFB mode is used. (Bits are not padded.)
	SIZ_{bs}	128	16	Block size (128 bits)
	KT_{bs}	*Key template values*	Variable	Key template
	IVsc	*Inivial vector value*	128	Initial vector value

B.7.6 Conclusion

This section described an access control technology for a JPEG 2000 codestream. The significant advantage of this technology is that the number of keys to be managed and to be accessed is much less than the conventional case. This technology provides a flexible and efficient JPEG 2000 access control according to a progression over order in a codestream.

[Note: Appropriate reference will be added when document repository is created.

Further detailed description is available in ISO/IEC JTC 1/SC 29/WG1N 3204.]

B.8 Scalable Authenticity of JPEG 2000 Code-streams

B.8.1 Security Service

This clause provides a flexible authentication mechanism for JPEG 2000 code-streams. It allows users to verify the authenticity and integrity of different sub -images with a single digital signature.

B.8.2 Typical Application

In critical application fields such as government, finance, healthcare and law, clients normally demand authenticity of the received content. Accordingl y, a scalable security mechanism for authenticating document is required in content dissemination.

B.8.3 Motivation

In the third party publishing applications, an image producer generates a code -stream and its signature. The producer then delivers the code-stream and the signature to a third party publisher. The users may ask the publisher for a transcoded code-stream due to resource limitation (e.g. bandwidth, computation). The publisher will delivers to the user the sub -image data as well as its authenticity pr oof.

B.8.4 Technical Overview

The scheme provides a flexible authentication mechanism for JPEG 2000 code-streams. It includes three modules: Signing, Transcoding and Verifying. The basic technology is the Merkle tree which organizes the JPEG 2000 packets.

B.8.4.1 Signing Module

The signing module generates a signature on an input JPEG 2000 code-stream according to preferred digital signature scheme. The protected code -stream is produced by inserting a SEC marker into the original code - stream. Specifically, the producer

- Read a JPEG 2000 code-stream.
- Construct a hash tree so as to produce the *root* value. The value of each leaf node is the hash of a packet. The value of each internal node is the hash of its child nodes. The tree structure is similar to the progress order of codestream.
- Sign the *root* value of the hash tree with a private key based on a signature algorithm .
- Create the SEC parameters. Insert the parameters into the SEC segment so as to produce an authentic code-stream.

B.8.4.2 Transcoding module

It generates Subsidiary Integrity Tokens (SITs) and a transcoded code-stream based on the requested resolution, layer, component and region. The SEC of the new code -stream includes the SITs and some other parameters. Specifically, the publisher and /or proxy

- Read the discarded packets which are not included in the transcoded code -stream.
- Construct the hash sub-trees with the discarded packets.
- Insert root values of the sub-trees into SEC segment.

The transcoded code-stream includes the updated SEC segment and code-stream excluding the discarded packets.

B.8.4.3 Verifying Module

The verifying module checks the authenticity of the protected code-stream. According to preferred digital signature scheme, the verifier obtains the public key, then

- Read the received code-stream
- Construct the hash tree with the received packets and code-stream headers from bottom to top. If some packets are discarded, replace the sub-tree with the corresponding SIT. Thus the *root′* value is constructed.
- Check the *root′* value against the signature in the SEC segment based on the specific signature system. If match, the code-stream is accepted, otherwise, rejected the received packets.

B.8.5 Bitstream syntax

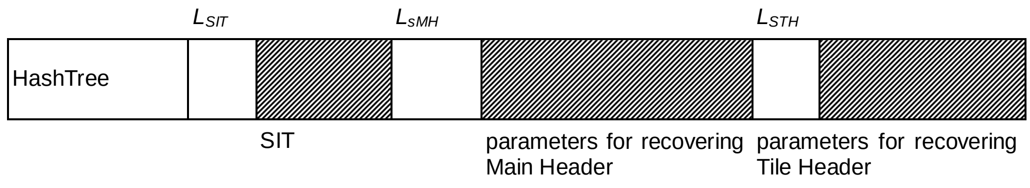
The SEC structure is shown in Table B.24. It includes the SEC marker, tool ID, and ZOI, the authentication template, and the security parameters for verification. The security parameters include data for recovering the codestream headers.

Table B.24 : Non-normative tool syntax

t	i	ID	L_{ZOI}	ZoI_{ID}	L_{ID}	PM_{ID}	T	TP_{ID}

Mnemonic	Size (bits)	Values	Semantic
t	1	1	Registration authority protection tool
i	7	*Instance value*	Tool instance identifier
ID	32	*ID value*	Registered ID
L_{ZOI}	16	$[0,2^{16}\text{-}1]$	Length of parameters for ZOI
ZoI_{ID}	Variable	*ZOI values*	Zone parameters
L_{ID}	16	$[19,2^{16}\text{-}1]$	Length of parameters
ID_T	8	2	Authentication template class id
T	Variable	*Authentication template values*	Authentication/MAC Template
TP_{ID}	Variable	See Table B.25	Security Parameters

Table B.25: Security parameters


L_{SIT}
L_{sMH}
L_{STH}
HashTree
SIT
parameters for recovering Main Header
parameters for recovering Tile Header

Parameter	Size (bits)	Values	Meaning
HashTree	8	$0 - (2^8\text{-}1)$	Hash Tree order. It may be different from the

			code-stream progression order. Tentatively, 0x00: LRCP 0x01: RLCP 0x02: RPCL 0x03: PCRL 0x04: CPRL others: Reserved
L_{SIT}	16	0 – (2^{16}-1)	Number of SITs
SIT	Variable: L_{hash}*L_{SIT}	NaN	Subsidiary Integrity Token
L_{SMH}	16	0 – (2^{16}-1)	length for SMH
SMH	Variable		Parameters for recovering main header
L_{STH}	16	0 – (2^{16}-1)	length for STH
STH	Variable		Parameters for recovering tile header

[*] For Keyed-MAC authentication, the (verification) key should be delivered separately.

[**] NaN: Not a Number.

[***] L_{hash} is the size of hash value, e.g., 160 for SHA1.

B.8.6 Conclusion

The technology provides a flexible authentication mechanism for JPEG 2000 code-stream. It has the property of "sign once, verify many ways". Concretely, after an original JPEG 2000 code-stream is signed once, various code-streams transcoded from the original code -stream can be verified with trust on the producer only. This property perfectly matches the functionality of "compress once, decompress many ways". It is in contrast with the traditional image authentication method which allows one signature to authenticate only one image.

[Note: Appropriate reference will be added when document repository is created.

The complete version of the example can be found as ISO/IEC JTC1/SC29/WG1 N3310.]

B.9 JPEG 2000 Data Confidentiality and Access Control System Based On Data Splitting and Luring

The described system in this clause is based on the sp litting, through a process called *Data_Splitting and Luring*, of an original JPEG 2000 file into two new files called respectively the *Lured_jp2file*, which conveys a protected content, and the *Control_File*, which conveys necessary information to access to t he protected content. Only a real time combination of those two files, through the *Live_Composing* process, allows for the rebuilding of the original JPEG 2000 file. The Live_Composing is managed by access control rules and rights management. The described system provides a high level of robustness and flexibility in JPEG 2000 data confidentiality and access control and is based on low time consuming and low cost computational operations.

B.9.1 Operational Description

B.9.1.1 Security services addressed

- **Confidentiality**: a Lured_jp2file conveys a protected content. By decoding only a single Lured_jp2file, the rendered content is visually scrambled, hence preventing an access to the original content. Only the real time recovering of the data stored in the Control_File throug h the Live_Composing Process enables an access to the original content.

- **Access Control**: this system can be used to perform access control on image content: several users sharing the same Lured_jp2file but owning different access rights won't be allowed t o access to the same parts of the content.

Note about **IPR protection**: by linking the content access with authentication and rights management, efficient control and tracking of the broadcasting and usage of a protected content can be ensured according to the content owner's will and perogative, possibly by combining this system with watermarking or fingerprinting.

B.9.1.2 Typical applications

One of the key idea of the system described is the splitting of the initial JPEG 2000 into two files, the first (Lured_jp2file) conveying only 99% of the original data and 1% of dummy data called lures and can be freely distributed, broadcasted, exchanged or copied through any classical networks or media, and the second one (Control_File) conveying 1% of the original data pl us some information which both are absolutely required to access to the protected content conveyed in the Lured_jp2file.

The other key idea is to link the access to the protected content conveyed in the Lured_jp2file with an identification and rights man agement steps whose results will trigger the streaming of the needed information used to recover in real time only an unscrambled content.

Finally, an efficient usage tracking and reporting is enabled through the statistics collected from the secured control_files server logfiles.

B.9.1.3 Potential users, implementation model and motivations

The potential users of the described system are the content creators, owners and providers as the system ensures that once the content is protected and conveyed in a Lured_jp2f ile, only authenticated and allowed users will be able to access to the original content. It is important to highlight that only 99% of the original content is provided freely, whereas the 1% needed to access to the original content will be distributed onl y after authentication and rights management protocols are passed.

B.9.2 Technical Overview

Figure B.9 shows an overview of the system.

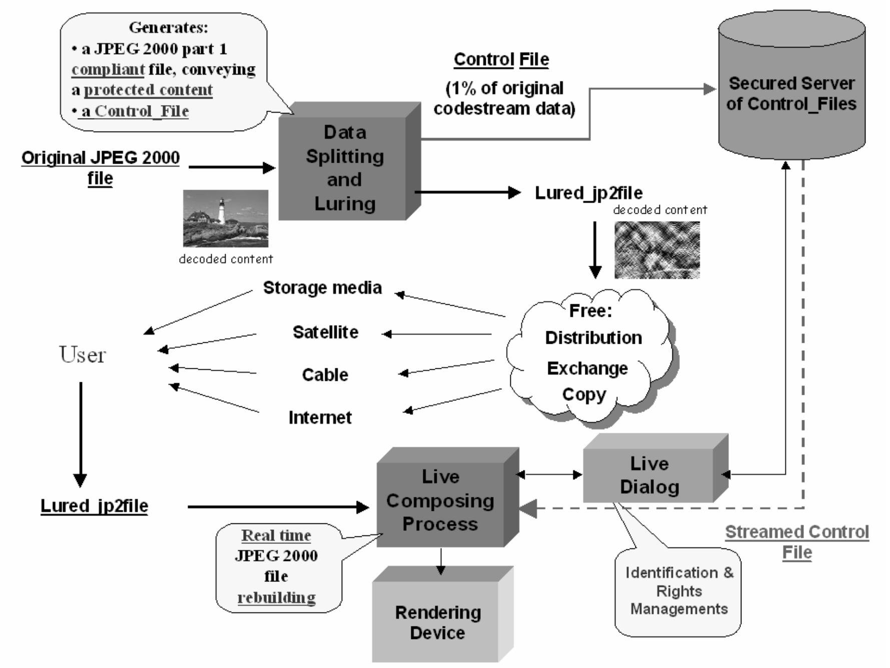


Figure B.9 System Overview

- An input JPEG 2000 file is split into two new files through an operation called *Data Splitting and Luring*. Two new files are then generated: a ***Lured_jp2file***, conveying a protected content (JPSEC content) and a ***Control_File***.

Through the Data Splitting and Luring process, some portions of the original JPEG 2000 file are extracted and replaced by *lures*. A Lured_jp2file conveys about 99 % of the original content whereas the last 1% are dummy data called lures, i.e. data without any a priori known link with the original data. Unlike to classical encryption, the luring process is not key based. A Lured_jp2file can be freely distributed, exchanged or copied by any user. The Control_File contains the 1% of original data extracted from the original file. It is stored in a *Secured Server of Control_Files*.

When the Lured_jp2file is decoded by any JPEG 2000 Part -1 compliant decoder, the content appears visually scrambled. The only way to access to the original content is to recover the extracted original data thanks to the Control_File. The *Live_Composing* device connects to the Secure Server of Control_Files through the *Live_Dialog* protocol and an identification and rights management protocol occurs:

- if the user owns the rights or agrees with the content access conditions (e.g. payment or subscription), the extracted data are retrieved from the Control_File and the original JPEG 2000 file is recovered in real time. However, according to user's rights, the rebuilding of the original JPEG 2000 may be partial (for instance to only allow an access to a particular tile and/or color component and/or resolution and/or precinct and/or quality layers) or full.
- if the user does not own the rights or does not accept the conditions, only scrambled content is displayed.

Main features of the described system are:

- <u>Splitting of the original</u> JPEG 2000 file into two files, the first one conveying a protected JPEG 2000 content with only 99% of the original data plus 1% of dummy data called lures (Lured_jp2file), the second one storing some original information data (1%) needed to rebuild the original JPEG 2000 content,
- Content <u>visual scrambling,</u>
- JPEG 2000 Part-1 <u>compliance and file size preserving</u>,
- <u>Low bit-rate and low computational</u> cost protection system.

The described system can be used with any environment and/or operating system. No particular hardware or software requirements are needed.

The Luring process will insert the following SEC marker in the Lured_jp2file:

Parameter					Value (in order)	Size (bits)	Derived meaning
SEC					0xFF65	16	SEC marker
L_{SEC}					0xXXXX	16	Length of the SEC marker segment
Z_{SEC}					1-255	8	Index of the marker segment.
P_{SEC}	F_{INSEC}				0	1	INSEC is not used
(if Z_{SEC} = 1)	$F_{multiSEC}$				0	1	one SEC marker segment is used
	F_{J2K}				1	2	JPSEC stream compliant with JPEG 2000 part 1
	F_{TRLCP}				0	1	TRLCP tag usage is not defined in this field.
	N_{tools}				1	7	one security tool is used in the codestream
	I_{max}				1	7	maximum tool instance index value used
	padding				0	5	padding
Tool$^{(0)}$	t				1	1	Non-normative protection tool
	i				0	7	Tool instance index
	ID				2^{31} + ID	32	RA is used to deliver the ID number.
	L_{ZOI}				*Length value*	16	Length of L_{ZOI} + ZOI
	ZOI	NZ_{ZOI}			0-254	8	Number of Zones
		Zone^{0}	DC_{ZOI}		0	1	The byte aligned segment does not follow.
					1	1	Non-image related description class.
					000010	6	Packet indexes are specified
			Pzoi^{0,0}	Mzoi	0	1	The byte aligned segment does not follow.
					0	1	The specified zones are influenced by the protection method.
					1	1	Multiple items are specified
					10	2	Index mode.
					xx	2	Izoi uses 8 or 16 or 32 bit integer.
					0	1	Izoi is described in one dimension.
				Nzoi	variable	8	2-255 (number of packets indexes)
				Izoi^{i}	variable	xxx Nzoi	Packet index
	L_{PID}				0 – (2^{16}-1)	16	Length of L_{PID} + P_{ID} in Bytes
	P_{ID}				Variable	Variable	Control_File ID, URL of Control_File server, etc.); full syntax provided by the RA

The tools needed to perform the Data Splitting and Luring and/or the Live Composing processes will possibly be provided via a connection to the Registration Authority and a download from it.

1. **N2984**, “Medialiving as a new method to protect JPEG 2000 data”, Jérôme Caporossi, Medialive, JPEG Meeting, Strasbourg, July 2003,
2. **N3080**, “Medialiving Experiment : a contribution to JPSEC Core Experiments”, Jérôme Caporossi, Medialive, September 2003
3. **N3207**, “Protection of JPEG 2000 data: the Medialiving technology”, Jérôme Caporossi, Medialive, March 2004
4. **N3208**, “The Medialiving Core Experiment v2.0”, Jérôme Caporossi, Medialive, March 2004

B.10 Secure Scalable Streaming and Secure Transcoding

B.10.1 Summary and Motivation

This section describes a method for providing the protection services of confidentiality and authentication of JPEG 2000 codestreams in a manner that 1) allows a (potentially untrusted) entity to securely transcode or adapt JPSEC protected streams wit hout requiring the entity to unprotect or decrypt the content, and 2) allows a client to validate that the transcoding operation was performed in a valid and permissible manner .

Transcoding is often required to adapt JPEG 2000 coded content for clients wit h diverse device capabilities (e.g., small display sizes or low -bit-rate network connections) and for time -varying network conditions. JPEG 2000 is especially well-suited for transcoding applications because of its inherent scalability properties. However, if one is not careful when protecting JPEG 2000 codestreams, the scalability property can be lost. For example, this occurs when the entire codestream is encrypted as a single file. In this case, the only way to transcode the protected codestream is to first decrypt it and then transcode or adapt the decrypted stream. Since the transcoder must decrypt the content, this breaks the end-to-end security of the system.

JPSEC was designed to enable secure transcoding of JPSEC -protected content, where secure t ranscoding is defined as *transcoding without unprotecting (decrypting) the content*. This is achieved with secure scalable streaming, which carefully combines scalable coding , encryption, and signalling in a manner that allows low-complexity, secure transc oding by a (potentially untrusted) server or mid-network node or proxy. This enables JPSEC to achieve the seemingly conflicting properties of mid -network transcoding and end-to-end security.

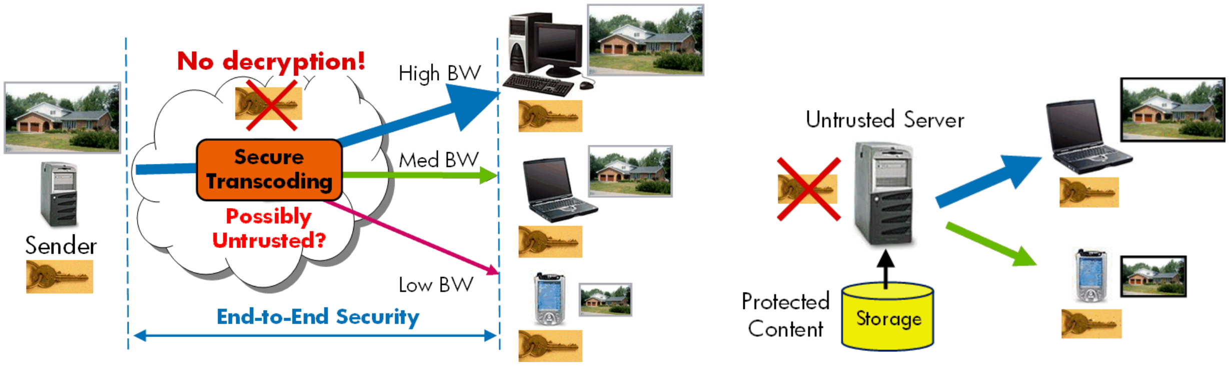


Figure B.10: JPSEC achieves end-to-end security and mid-network secure transcoding. The media is encrypted at the sender and decrypted only at the receiver, and remains encrypted at all points in-between. Left: A mid-network node securely transcodes protected content for each JPSEC client. Right: An untrusted server securely transcodes and streams JPSEC content without unprotecting it.

B.10.2 Operational description and two example usages

In the first example, the original JPEG 2000 codestream is in RLCP ordering and the goal is to protect this stream with encryption and authentication while enabling secure transcoding by resolution on the protected codestream. Since the original JPEG 2000 codestream used an RLCP ordering, each resolu tion component is represented by a contiguous data segment. Encryption can be performed on each of the three contiguous data segments. The JPSEC header then specifies three zones of influence describing the resolution component, bitstream segment, and en cryption template used for each segment. Authentication is also performed on each of the three data segments, either before or after encryption depending on the desired functionality. This is also specified in the SEC header using the authentication temp late.

In order to perform secure transcoding on the JPSEC codestream, a transcoder simply reads and parses the SEC header, identifies the locations of the resolution segments, and then retains or removes the appropriate data segments/resolutions. Notice that this transcoding operation corresponds to a simple parsing operation and that it does not require unprotecting the data. Authentication is performed by authenticating the received transcoded data with the MAC value s that are placed in the SEC header during the JPSEC protection process.

In the second example, the desired goal is once again to protect the codestream while allowing transcoding by resolution, however this example is slightly more complex in that the original JPEG 2000 codestream is in PCRL rather than RLCP ordering, so the data segments corresponding to the three resolution components are not contiguous in the original codestream. JPSEC allows the desired goal of secure transcoding or scaling by resolution to be achieved in a number of w ays. One method is to encrypt by individual packets while leaving packet headers unencrypted. This retains the highest level of scalability in the stream but also requires the most complex secure transcoding operation because the transcoder must parse th e JPSEC stream at the packet level. The other extreme that results in the simplest secure transcoding operation is to reorder the data such that the resolution components are once again in contiguous segments whose offsets are signaled in the SEC header. This can be achieved in a JPEG 2000 compliant manner by reordering the JPEG 2000 packets from PCRL to RLCP ordering and signaling the new progression order in the COD marker segment or with the progression order change (POC) marker segment. The resulting data reordering and protection transformation is shown in Figure B.11. Once again, the main SEC header contains ZOI parameters that describe the corresponding image -related and bitstream-related parameters associated with each data segment, but this time on the reordered JPEG 2000 codestream.

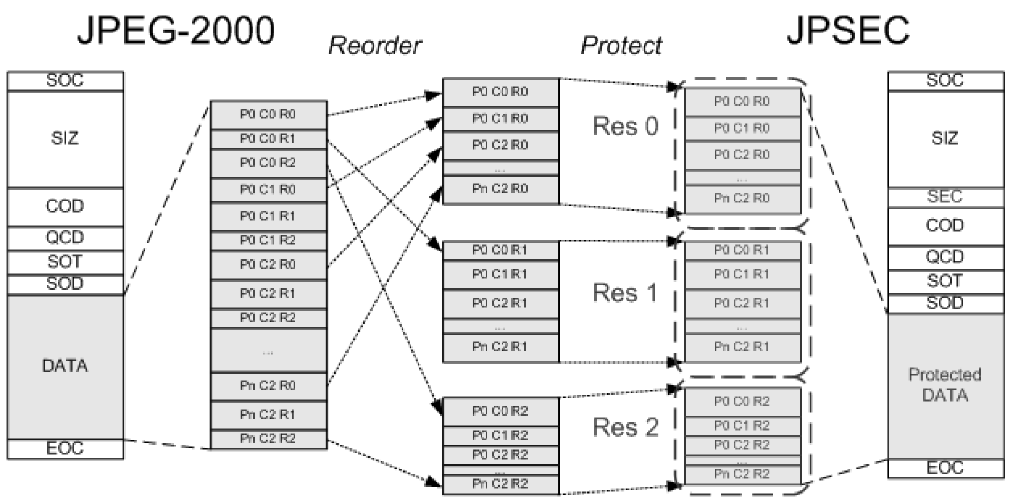


Figure B.11: A second example of forming a JPSEC codestream.

B.10.3 Bitstream syntax

The JPSEC syntax can be used to create a secure scalable st reaming and secure transcoding system with the template protection tool. Specifically, the Zone of Influence (ZOI) can be used with the decryption template, processing domain, and granularity to fully define the decryption process that an allowed JPSEC consumer should use to decrypt the stream. Furthermore, the ZOI parameters signal information that transcoding nodes can use to perform secure transcoding.

The ZOI specifies three zones, one for each resolution, and the byte ranges associated with the encry pted bits for each zone. The signaling syntax for the decryption protection template, processing domain, and granularity are shown in Table B.26. The decryption method is signaled with the decryption protection template. In this case, it specifies AES encryption in CTR mode , and the block size and key length. The processing domain and granularity further specify how the decryption is performed. It signals that the processing domain is the bitstream itself, and that packet header s and packet bodies are encrypted. Different decryption methods can be specified by changing the processing domain and granularity. For example, the granularity of the encryption can be on individual packets or only on the packet bodies. Furthermore, the authentication method is specified with the same ZOI as above, but with the following authentication protection template. The syntax for the authentication template is shown in Table B.27 for using HMAC with SHA-1 for authentication. Of course, other JPSEC ciphers and MACs may also be used. In addition, the proposed solution may be used with other digital signature, access control, and key management tools. Furthermore, a distortion can be associated with each packet (or other zone of data) using the distortion field (Sec. 5.5.4.1) to enable rate-distortion (R-D) optimized secure streaming and secure transcoding [1,2,3].

Table B.26 — Parameter values for template protection to ol, processing domain, and granularity

Parameter			Value (in order)	Size (bits)	Derived meaning
T_{decry}	ME_{decry}		0	8	Marker emulation flag is NULL .
	CT_{decry}		3	16	AES encryption
	CP_{decry}	M_{bs}	10 0011	6	CTR and no padding.
		P_{bs}	0	2	Padding is not used for CT R mode.
		SIZ_{bs}	128	8	Block size is 128 .
		KT_{bs}	*Key value*	Variable	Key information template
PD			0	1	Byte aligned segment (BAS) does not follow .
			10	2	Processing domain is codestream domain .
			00000	5	Packet headers & bodies are protected.
G	PO		0000010100111000	16	Processing order is TRLCP .
	GL		0000 1001	8	Unit is zones specified by ZOI .
V	N_V		1	16	IV is not specified
	Sv		16	8	IV size in bytes
	VL		*Nonce value*	128	The IV is a counter for CTR mode

Table B.27 — Parameter values for authentication template protection tool

Parameter			Value (in order)	Size (bits)	Derived meaning
T_{auth}	M_{auth}		0	8	Hash-based MAC.
	P_{auth}	M_{ID}	1	8	HMAC.
		H_{ID}	1	8	Hash ID is SHA-1.
		KT_{MAC}	*Key value*	variable	See key template.
		SIZ_{MAC}	80	16	MAC size is 80 bits (truncated from 160).

B.10.4 Conclusions

This section describes secure scalable streaming and secure transcoding with JPSEC, which enables the two seemingly conflicting properties of end-to-end security with secure transcoding at mid-network nodes. This allows the JPSEC codestream to be transcoded *without requiring decryption*. Furthermore, this method provides authentication that the transcoding was performed only in a valid and permissible manner, and no unintentional or malicious modification from an error or attacker has occurred. This allows a (potentially untrusted) server or mid-network node such as a proxy to perform secure transcoding while allowing a JPSEC consumer to authenticate that the received content was transcoded in a valid and permissibl e manner.

[1] "Secure Scalable Video Streaming for Wireless Networks", S. Wee, J. Apostolopoulos, IEEE Inter. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), March 2001. Also available at: www.hpl.hp.com/personal/John_Apostolopoulos/papers/SecureScalableStreaming_ICASSP01.pdf

[2] "Secure Scalable Streaming Enabling Transcoding Without Decryption", S. Wee, J. Apostolop oulos, IEEE Inter. Conf. on Image Processing (ICIP), Sept 2001. http://lib.hpl.hp.com/techpubs/2001/HPL-2001-320.html

[3] "Secure Scalable Streaming and Secure Transcoding with JPEG 2000", S. Wee, J. Apostolopoulos, IEEE Inter. Conf. on Image Processing (ICIP), Sept 2003. http://lib.hpl.hp.com/techpubs/2003/HPL-2003-117.html

Annex C
(informative)

Basic Use Cases

C.1 Introduction

This clause provides example JPSEC use cases. It is an informative part and is provided as a help to readers and implementers. It only addresses simple cases. More complex examples are provided in Anne x B. The following examples are provided:

- Encryption of a JPEG 2000 codestream

- Signature of a JPEG 2000 codestream

C.2 Encryption of the JPEG 2000 codestream

This example uses only templates defined in this ISO/IEC International Standard| Recommendation.

The following example defines:

- A single Zone of influence: the entire codestream (from SOD to the end)

- The processing domain is the codestream domain and is only applied to the packet bodies: protection is applied after packets are composed. In another instance, it is applied across packet headers and bodies.

- The granularity level is packet in one case and resolution in another case.

The encryption is carried out with AES/192 in CBC mode with padding. There is a single key that can be retrieved from a server through a URI.

C.3 Signature of the JPEG 2000 codestream

This example uses only templates defined in this ISO/IEC International Standard| Recommendation.

The following example defines:

- A single Zone of influence: the entire codestream (from SOD to th e end)

- The processing domain is the codestream domain and is only applied to the packet bodies: protection is applied after packets are composed. In another instan ce, it is applied across packet headers and bodies.

- The granularity level is packet in on e case and resolution in another case.

The signature is carried out with SHA-1. There is a single key that can be retrieved from a server through a URI.

[Note: Details of these example use cases are being added. Note: Add example of header protection. Note: Add examples differentiating authentication template and integrity template. Note: Incorporate use of part-1 compliance in signature example.]

Annex D
(informative)

Interoperability

D.1 Part 1

A number of protection methods can be applied to a JPEG-2000 codestream to create JPSEC codestreams that are still strictly compliant with JPEG-2000 part 1. We use the term "Part 1 compliance" to refer to JPSEC codestreams that have a strictly defined behaviour for JPEG-2000 part 1 decoders including those that are not aware of JPSEC.

A JPEG-2000 part 1 decoder will skip marker segments that it does not recognize. A JPSEC tool such as the JPSEC template tool for authentication inserts message authentication code values that a re computed from the JPEG-2000 data into the SEC marker segment along with the parameters that describe the particular authentication methods that can be used by a JPSEC consumer. These parameters and values tell a JPSEC consumer how to verify that the received JPSEC codestream is authentic. Notice that the JPSEC authentication tool does not manipulate the JPEG-2000 data. Thus, a JPEG-2000 part 1 decoder that receives this JPSEC bitstream will begin decoding the JPSEC stream, it will then skip the SEC ma rker segment, and continue to decode the JPSEC stream as if it were a JPEG-2000 part 1 stream. The JPSEC template tool for integrity shares these characteristics and thus also results in a part 1 compliant codestream.

JPSEC allows encryption and decryption to be performed on JPEG-2000 and JPSEC codestreams. When encryption is used, the JPEG-2000 data is of course changed. Strictly speaking, part 1 compliance is not possible with encrypted streams since it will most likely causes a JPEG-2000 part 1 decoder to see illegal values. One possible way of overcoming or at least mitigating this problem is to use the error resilience capabilities of JPEG-2000. With error resilience, it may be possible to have encrypted JPSEC bitstreams that have a defined behaviour for JPEG-2000 part 1 decoders.

JPSEC has a P_{sec} parameter field that contains security parameters for the entire codestream. This includes a flag F_{J2K} that may be set to 1 indicate that a JPSEC codestream is decodable by JPEG 2000 Part 1 decoders. A JPSEC creator may set this parameter as it applies JPSEC tools to the JPEG-2000 codestream. It was mentioned that a JPSEC creator can accept a protected JPSEC codestream as input. If a JPSEC creator receives an input JPSEC codestream has the flag F_{J2K} flag set to indicate part 1 compliance and then applies a JPSEC tool that loses the part 1 compliance, it must set the F_{J2K} flag to 0.

For JPSEC streams that are not part 1 compliant, it is recommended to use a file extension of .jp2s to indicate that a JPEG-2000 part 1 decoder may not be able to decode the protected codestream.

D.2 Part 2

JPEG 2000 Part 2 Amendment 2 on the extended capabilities marker segment (CAP) can be used to indicate that JPSEC is used. Specifically, Part 2 uses R_{siz} parameter to indicate the presence of a CAP marker segment, which contains a C_{cap} parameter that can be used to signal which JPEG-2000 parts are used in the codestream. One can specify that JPEG 2000 part 8 (JPSEC) is used by setting an appropriate bit in C_{cap}.

Thus, a JPSEC creator may set the R_{siz} parameter to indicate the presence of a CAP marker segment. It may insert or edit the CAP marker segment to set the C_{cap} parameter to indicate that Part 8 is used.

D.3 JPIP

D.3.1 General Relationship between JPIP and JPSEC

JPIP specifies a protocol consisting of a structured series of interactions between a client and a server by means of which image file metadata, structure, and partial or whole image code streams may be exchanged in a communications efficient manner.

JPIP can be tailored via the various extensions to the JPEG 2000 file format, as defined in ITU –T Rec. T.801 | ISO/IEC 15444–2, ISO/IEC 15444–3 and ISO/IEC 15444–6. However, to achieve a simple level of interactivity that allows portions of a single JPEG 2000 file or codestream to be transferred, these other capabilities are not mandated.

Provisions have been included for the extension of the JPIP protocol to support the current JPEG 2000 Standards ISO/IEC 15444–3, Motion JPEG 2000, and ISO/IEC 15444–6, Compound Documents, and the future parts of JPEG 2000 (currently JP3D, JPSEC, and JPWL).

JPSEC provides security services for JPEG 2000 images. The JPSEC syntax supports two types of markers: SEC and INSEC. One or more SEC markers appear in the main header of JPSEC bit stream. In other words, JPSEC consumes a JPEG 2000 bitstream, modifies the JPEG 2000 main header to form a new JPSEC "main header", and modifies the corresponding JPEG 2000 data stream to form a new protected data stream if applicable. INSEC markers may optionally appear in the "data" portion of the data stream. It specifies some "smaller size" or "local area" parameters compared to SEC marker and can be used to complement the SEC marker.

It is observed that JPIP is just beyond the transport layer, while JPSEC is at the application layer. From this point of view, JPIP provides a transport service to JPSEC. That is, the JPIP offers efficient tools to deliver image information, including main header (all of the markers) and code streams, between servers and clients. This section considers how JPIP can be used to transport JPSEC content.

D.3.2 Specific Issues on interactivity between JPIP and JPSEC

This section describes the issues that a JPIP sender and receiver must consider to transport JPSEC content.

In A.3.5 "Main header data-bin" of JPIP FCD 2.0, both JPP- and JPT-stream media types use the main header data-bin. This data-bin consists of a concatenated list of all markers and marker segments in the main header, starting from the SOC marker. It contains no SOT, SOD or EOC markers. However, the main header of JPEG 2000 does not include a SEC marker and its segment. As a result, A.3.5 of JPIP FCD 2.0 does not specify support for the SEC marker segment specified in JPSEC. Thus, a JPIP sender and receiver must be modified to recognize the SEC marker segment(s) that appear in the main header of a JPSEC codestream.

A.3.2 "Precinct data-bins" of JPIP FCD 2.0 describes its support to the precinct data. However, A.3.2 of JPIP FCD 2.0 does not specify if it supports INSEC marker and its segment specified in JPSEC. Thus, a JPIP sender and receiver must be modified to recognize the INSEC marker segment that may appear in the data portion of a JPSEC codestream.

In A.3.3 "Tile header data-bins" of JPIP FCD 2.0, the tile header data-bins appear only within the JPP-stream media type. For data-bins belonging to this class, the in-class identifier holds the index (starting from 0) of the tile to which the data-bin refers. This data-bin consists of markers and marker segments for tile n. It shall not contain an SOT marker segment. Inclusion of SOD markers is optional. This data bin may be formed from a legal codestream, by concatenating all marker segments except SOT and POC in all tile -part headers for tile n.

In A.3.4 "Tile data-bins" of JPIP FCD 2.0, the tile data-bins shall be used only with the JPT-stream media type. For data-bins belonging to this class, the in-class identifier is the index (starting from 0) of the tile to which the data-bin belongs. Each tile data-bin corresponds to the string of bytes formed by concatenating all tile-parts belonging to the tile, in order, complete with their SOT, SOD and all other relevant marker segments.

As mentioned above, A.3.4 and A.3.5 of JPIP FCD 2.0 describe the support to the tile -part header and tile-part data. However, A.3.4 and A.3.5 of JPIP FCD 2.0 do not specify if they support of SEC marker segments and INSEC marker segments. Thus, a JPIP sender and receiver must be modified to recognize and transport these marker segments along with the protected data.

D.3.3 Summary

Generally speaking, JPSEC makes itself suitable to be transported by JPIP. INSEC marker is used in the code stream to describe some "small" specific data part that is protected by security tool/tools. It makes JPSEC more flexible. To make INSEC more robust, the service layer (currently we mean JPIP) should provide the good Quality of Service or protection on the INSEC marker and its segment. In order to achieve this goal, JPIP and JPSEC need to work out some issues and make sure the inter activity between JPIP and JPSEC.

D.4 JPWL

Annex E
(informative)

Patent statements

The International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this part of ISO/IEC 15444 may involve the use of patents.

ISO and IEC take no position concerning the evidence, validity and scope of these patent rights.

The holders of these patent rights have assured the I SO and IEC that they are willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statements of the holders of these patents right are registered with ISO and IEC. Information may be obtained from the companies listed below.

Attention is drawn to the possibility that some of the elements of this part of ISO/IEC 15444 may be the subject of patent rights other than those identified in this annex. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Table E.1 — List of statements

Number	Submitting entity	Terms and conditions	Relevant clause(s)
1	[Company name]	reasonable and non-discriminatory terms and conditions	B.x.x

Bibliography

[1] ISO/IEC 7499-2, *[Title 1] — [Sub-title]: Part 2: title*.

[2] ISO/IEC FDIS 9796-2: 2001, *[Title 1] — [Sub-title]: Part 2: title.*

[3] ISO/IEC 9797-1:1999, *[Title 1] — [Sub-title]: Part 1: title*

[4] ISO/IEC 9798-1:1997, *[Title 1] — [Sub-title]: Part 1: title*

[5] ISO/IEC 11770-1:1996, *[Title 1] — [Sub-title]: Part 1: title*

[6] ISO/IEC 11770-2:1996, *[Title 1] — [Sub-title]: Part 2: title*

[7] ISO/IEC 11770-3:1996, *[Title 1] — [Sub-title]: Part 3: title*

[8] ISO/IEC PDTR 13335-1:2001, *[Title 1] — [Sub-title]: Part 1: title*

[9] ISO/IEC TR 13335-4:2001, *[Title 1] — [Sub-title]: Part 4: title*

[10] ISO/IEC 14888-1:1998, *[Title 1] — [Sub-title]: Part 1: title*

[11] ISO/IEC FDIS 15946-3:2001, *[Title 1] — [Sub-title]: Part 3: title*