International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

T.807

(05/2006)

SERIES T: TERMINALS FOR TELEMATIC SERVICES

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

ITU-T Recommendation T.807

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

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

Summary

The purpose of this Recommendation | International Standard is to provide a syntax that allows security services to be applied to JPEG 2000 coded image data. Security services include confidentiality, integrity verification, source authentication, conditional access, and secure scalable streaming and secure transcoding. The syntax allows these security services to be applied to coded and uncoded image data in part or in its entirety. This maintains the inherent features of JPEG 2000 such as scalability and access to various spatial areas, resolution levels, colour components, and quality layers, while providing security services on these elements.

Source

ITU-T Recommendation T.807 was approved on 29 May 2006 by ITU-T Study Group 16 (2005-2008) under the ITU-T Recommendation A.8 procedure. An identical text is also published as ISO/IEC 15444-8.

FOREWORD

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

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

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

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

NOTE

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

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

INTELLECTUAL PROPERTY RIGHTS

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

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

Introduction

In the "Digital Age", the Internet provides many new opportunities 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 assuring that copyright, and the cultural and intellectual expression it fosters, remains well protected in the 21st century. The new Digital economy and the creative people in every country of the world depend on it. Also in December 1996, WIPO Copyright Treaty (WCT) has been promulgated with two important articles (11 and 12) about technological measures and obligations concerning Right Management Information:

Article 11
Obligations concerning
Technological Measures

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

Article 12
Obligations concerning Rights
Management Information

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

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

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

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

This treaty provides a solid foundation to protect Intellectual Property. As of 2004, about 50 countries ratified this important treaty. Therefore, it is expected that tools and protective methods that are recommended in JPEG 2000 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, authentication, and integrity checking.

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

INTERNATIONAL STANDARD
ITU-T RECOMMENDATION

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

1 Scope

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

1) a normative codestream syntax containing information for interpreting secure image data;
2) a normative process for registering JPSEC tools with a registration authority delivering a unique identifier;
3) informative examples of JPSEC tools in typical use cases;
4) informative guidelines on how to implement security services and related metadata.

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

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations'.

– ITU-T Recommendation T.800 (2002) | ISO/IEC 15444-1:2004, *Information technology – JPEG 2000 image coding system: Core coding system*.
– ITU-T Recommendation T.801 (2002) | ISO/IEC 15444-2:2004, *Information technology – JPEG 2000 image coding system: Extensions*.

3 Terms and definitions

For the purposes of this Recommendation | International Standard, the following definitions apply. The definitions defined in ITU-T Rec. T.800 | ISO/IEC 15444-1 clause 3 apply to this Recommendation | International Standard.

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

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

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

3.2.2 **fragile/semi-fragile image authentication**: Process for both image source authentication and image data/image content integrity verification that 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.

NOTE – 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**: Property that information is not made available or disclosed to unauthorized individuals, entities or processes.

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

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

3.5 decryption, deciphering: Inverse transformation of the encryption.

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.

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

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

3.8 fingerprints: Characteristics of an object that tend to distinguish it from other similar objects to enable the owner to trace authorized users distributing them illegally.

NOTE – In this respect, fingerprinting is usually discussed in the context of the traitor tracing problem.

3.9 hash function: Function which maps strings of bits to fixed-length strings of bits, satisfying the following two properties:

NOTE – 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.

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

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

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

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

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

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

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

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

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

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

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

3.14 JPSEC registration authority: Entity in charge of delivering a unique ID to reference a JPSEC tool and storing the parameter list of the JPSEC tool's description.

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

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

3.15.2 JPSEC non-normative tool: JPSEC tool that is specified by an identification number given by the JPSEC registration authority or by a user-defined application.

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

3.15.4 JPSEC registration authority tool: JPSEC non-normative tool that is defined by the JPSEC registration authority.

3.16 JPSEC tool description: A description of the parameters used by the JPSEC tool.

NOTE – However, JPSEC tool description does not describe the algorithm or method used. A JPSEC tool description consists of two parts: the parameter list and its values. In the case of JPSEC normative tools, the parameter list is given by the standard. In the case of JPSEC non-normative tools, the parameter list 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: Sequence of symbols that controls the operations of encipherment and decipherment.

3.17.1 symmetric keys: Pair of keys for which 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.

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

3.17.2.1 private key: Key of an entity's asymmetric key pair which should not be disclosed.

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

3.18 key generation, key generating function: 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.

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

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

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

3.21 message authentication code algorithm, cryptographic check function, cryptographic checksum function: Algorithm for computing a function which maps strings of bits and a secret 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.

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

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

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

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

3.23 packet: A part of the JPEG 2000 Part 1 bit stream comprising a packet header and the compressed image data from one layer of the precinct of one resolution of one tile-component.

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

3.24 protection: Process to secure content.

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

3.24.2 protection method: Method used to create or consume 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.

NOTE – 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: Specification of the format of the JPSEC codestream that contains all the required information for consuming secure JPEG 2000 images.

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

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

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

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

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

3.28.1 watermarking: Process that imperceptibly inserts data representing some information into multimedia data in one of the following two ways:

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

4 Symbols and abbreviations

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

BAS	Byte Aligned Segment
FBAS	Field Byte Aligned Segment
G	Granularity
GL	Granularity Level
INSEC	In-codestream security marker
IP	Intellectual Property related to technology
IPR	Intellectual Property Rights related to content
JPSEC	Secure JPEG 2000
KT	Key Template
LSB	Least Significant Bit
MAC	Message Authentication Code
MSB	Most Significant Bit
PD	Processing Domain
PKI	Public Key Infrastructure
PO	Processing Order
RA	Registration Authority
RBAS	Range Byte Aligned Segment
SEC	Security marker
T	Template
V	Values
VL	Value List
ZOI	Zone of Influence

5 JPSEC syntax (normative)

5.1 JPSEC framework overview

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

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

The *signalling syntax* specifies:

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

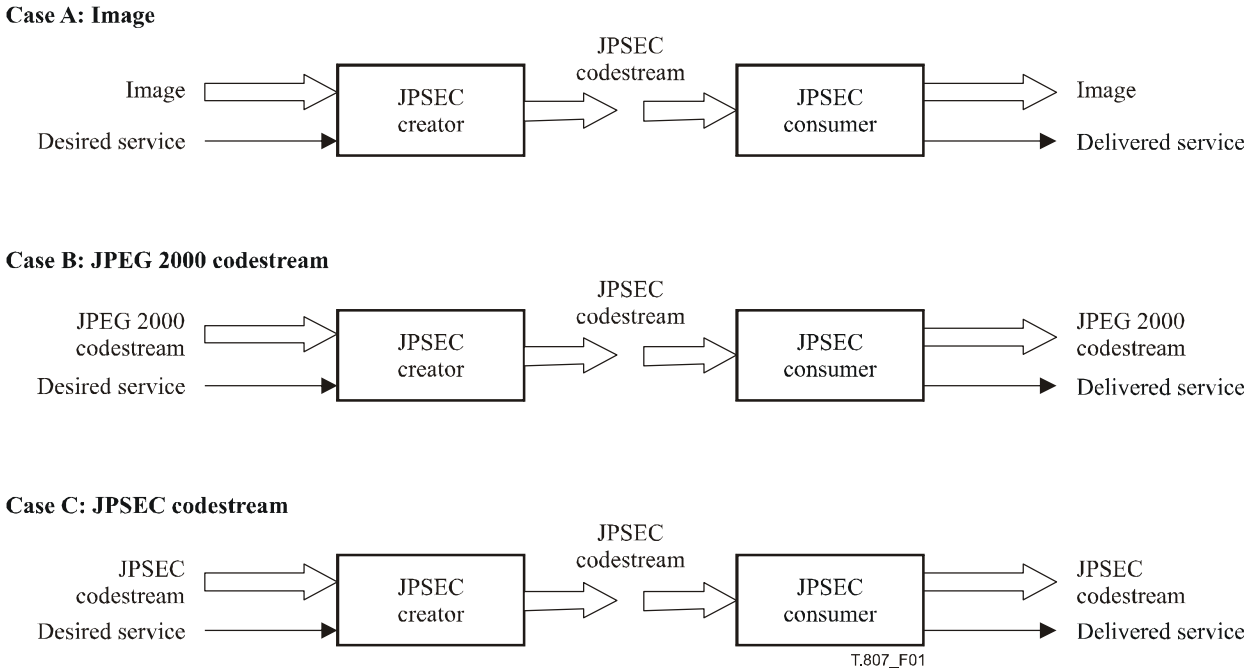


Figure 1 – Overview of the conceptual steps in JPSEC framework

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

As shown in case C of Figure 1, the JPSEC codestream may be created from another JPSEC codestream. This may arise when multiple JPSEC tools are applied to the same content, but at different times or by different entities. When this occurs, the ordering in which the JPSEC tools are applied during the creation and consumption operations may be significant.

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

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

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

In the JPSEC framework, JPSEC tools are specified by templates, defined privately, or registered by a *JPSEC Registration Authority*. JPSEC tools specified by the templates 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 subclause is to list and to explain the functionalities which are included in the scope of this Recommendation | International Standard.

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

– *Confidentiality via encryption and selective encryption*

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

– *Integrity verification*

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

1) Image data integrity verification where even only one bit of image data in error results in verification failure (i.e., the verification returns "no integrity"). This verification is also often referred to as fragile image (integrity) verification.

2) Image content integrity verification where even some incidental alteration of image data results in verification success as long as the alteration does not change image content from the human visual system point of view; in other words, the image perceptual meaning does not change. This verification is also often referred to as semi-fragile image (integrity) verification.

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

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

2) Watermarking-based methods. This Recommendation | International Standard does not define normative template for watermarking technology, although it supports non-normative tools using watermarking technology.

3) Combination of the above two types of methods.

– *Source authentication*

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

– *Conditional access*

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

– *Registered Content identification*

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

– *Secure Scalable Streaming and Secure Transcoding*

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

5.3 Comments on design and implementation of secure JPSEC systems

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

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

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

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

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

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

5.4 Byte aligned segment (BAS)

5.4.1 Byte aligned segment

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

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

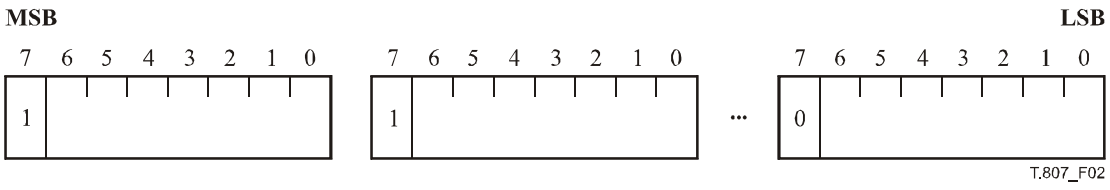


Figure 2 – Byte aligned segment (BAS) structure

5.4.2 Field BAS (FBAS)

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

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

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

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

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

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

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

5.4.3 Range BAS (RBAS)

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

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

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

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

as follows:

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

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

0 v3 v4 v5 v6 v7 v8 v9

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

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

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

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

Thus, the indicator bits would be set as follows:

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

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

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

5.5 Main security marker (SEC)

5.5.1 Security marker segments

In this subclause, 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 information for securing JPEG 2000 images. To do so, it makes references to JPSEC normative tools that are specified by the templates described in 5.8 or by JPSEC non-normative tools that 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 codestream can be protected with one or more JPSEC tools. Each tool is a JPSEC normative tool or a JPSEC non-normative tool. The parameters for these tools are signalled in one or more SEC marker segments located in the main header of the codestream after the SIZ marker segment. When multiple SEC marker segments are used, they are concatenated and 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 3 shows the syntax of the SEC marker segment. The segment is signalled by the SEC marker 0xFF65. L_{SEC} is the length of the SEC marker segment, including the 2 bytes for L_{SEC}, but not the two bytes for the SEC marker itself. Z_{SEC} is a SEC marker segment index. Z_{SEC} shall be set to 0 for the first marker segment that appears in the codestream. P_{SEC} is a parameter field that describes the security parameters relevant to the entire codestream and only exists in the first SEC marker segment, i.e., if $Z_{SEC} = 0$. The syntax supports the use of several JPSEC tools that are signalled in one or more marker segments. If more than one JPSEC tool is used, then a JPSEC consumer shall process the tools in the order in which they appear in the codestream.

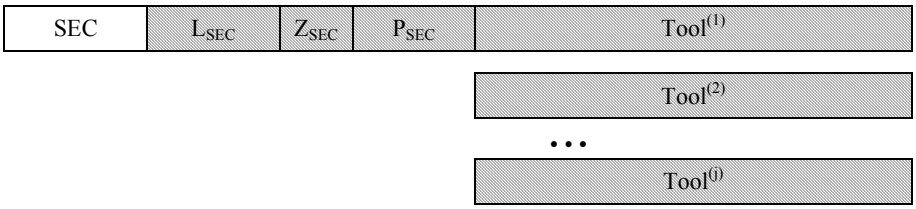


Figure 3 – Main security marker segment syntax

SEC: Marker code. Table 1 shows the sizes and values of the symbols and parameters for the main security marker segment.

L_{SEC}: Length of marker segment in bytes (including L_{SEC} itself, but excluding the marker).

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

P_{SEC}: Parameter field for codestream security parameters. This field is only present in the first SEC marker segment, i.e., when Z_{SEC} is 0.

$Tool^{(i)}$: Parameters for JPSEC tool *i*. If multiple JPSEC tools are signalled, then a JPSEC consumer shall process each tool in the order of appearance in the JPSEC codestream.

Table 1 – Main security parameter values

Parameter	Size (bits)	Values
SEC	16	0xFF65
L_{SEC}	16	2 ... $(2^{16} - 1)$
Z_{SEC}	8 + 8 * n (RBAS)	0 ... 2^{7+7*n}
P_{SEC}	0, if $Z_{SEC} > 0$ Variable, otherwise	 If $Z_{SEC} = 0$, see Table 2
$Tool^{(i)}$	Variable	See 5.6.2 and 5.6.3

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

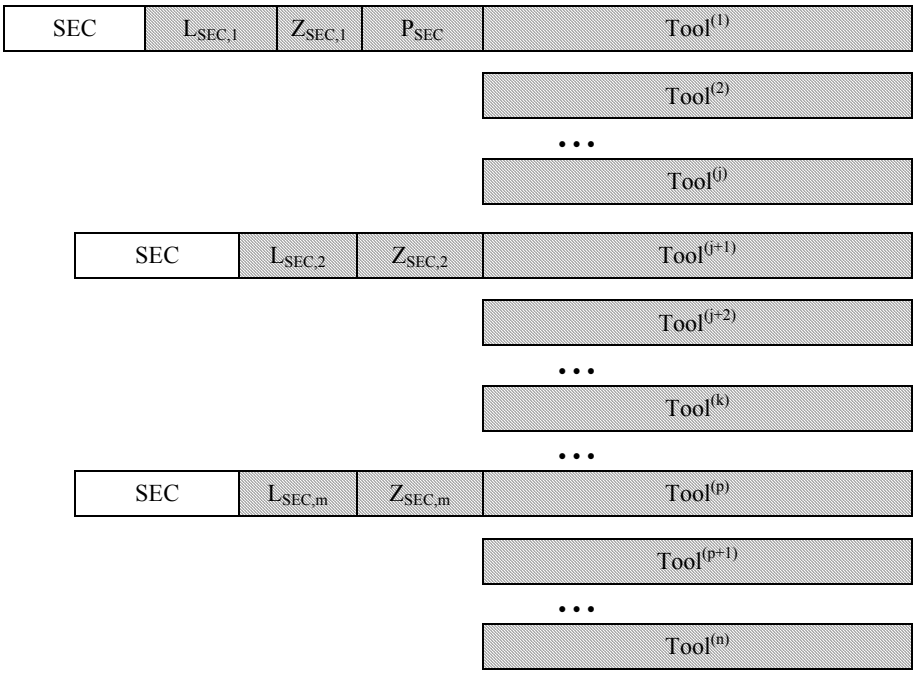


Figure 4 – Main security marker syntax when multiple marker segments are used

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

P_{SEC} is a parameter field that describes security parameters for the entire codestream as opposed to for a particular tool. This is used to indicate events such as JPEG 2000 Part 1 compliance or the use of INSEC markers. The P_{SEC} parameters are shown in Figure 5.

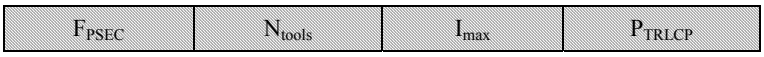


Figure 5 – Codestream security parameters (P_{SEC}) syntax

F_{PSEC}: Flag to indicate if INSEC marker segment is used, if multiple SEC marker segments are used, if the original JPEG 2000 Part 1 codestream data was modified, and if TRLCP tag usage is defined. FBAS structure is used by this field.

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

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

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

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

Parameter	Size (bits)	Values
F_{PSEC}	Variable (FBAS)	See Table 3
N_{tools}	8 + n * 8 (RBAS)	1 ... 2^{7+7*n}
I_{max}	8 + n * 8 (RBAS)	0 ... 2^{7+7*n}
P_{TRLCP}	0, if $F_{TRLCP} = 0$ 32, if $F_{TRLCP} = 1$	See Table 4

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

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

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

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

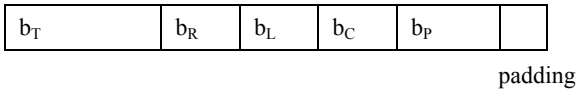


Figure 6 – TRLCP tag descriptor (P_{TRLCP}) syntax

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

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

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

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

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

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

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

The size of each resulting 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 level index, the layer index, the component index, and the precinct index in that order. If extra bits are needed to fill the integer byte size requirement, then the 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 MSBs of the TRLCP tag if they exist.

5.5.2 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 Figures 3, 4 and 7, where N tools are applied. The JPSEC consumer will read the N tools in order of placement in the SEC marker segment shown in Figure 3 or Figure 4, and apply them in that same order to perform the JPSEC consumption of the JPSEC codestream. Note that while the JPSEC consumer applies the JPSEC tools in order 1, 2, ... , N, as read from the SEC marker segment, during creation of the JPSEC codestream these JPSEC tools were applied in the reverse order, i.e., N, N – 1, ...2, 1, as illustrated in Figure 7. Note that the numbering of the tools in the figure was chosen to highlight that the JPSEC consumer applies the JPSEC tools in the reverse order from the JPSEC creator. However, any numbering of the JPSEC tools is acceptable, as long as each JPSEC tool in a JPSEC codestream is given a unique number for identification purposes.

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

In certain applications, it may be desirable for a JPSEC consumer to consume the multiple JPSEC tools in a different manner than described above. For example, the JPSEC consumer may choose to consume the multiple tools in a different order, or to skip certain tools in the consumption. 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.

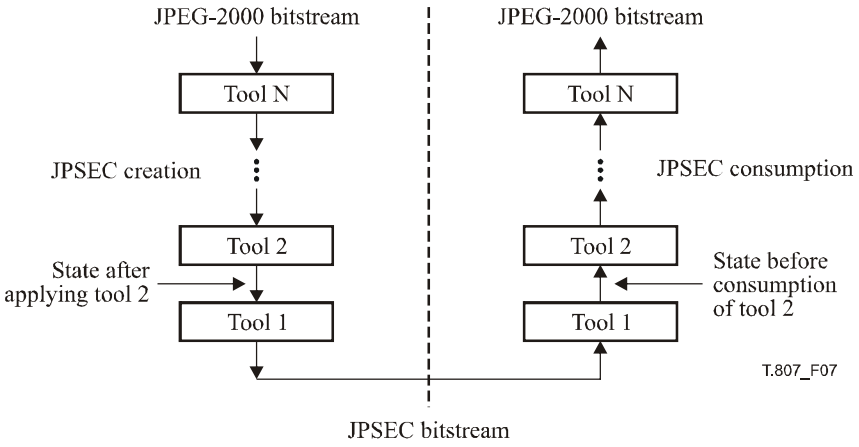


Figure 7 – Use of multiple JPSEC tools

5.6 JPSEC tools

5.6.1 JPSEC tool syntax

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

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

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

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

Figure 8 – JPSEC tool syntax (Tool$^{(i)}$)

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

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

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

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

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

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

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

Table 5 – JPSEC tool parameter values

Parameter	Size (bits)	Values
t	8 + 8 * n (FBAS)	x0xx xxxx $_b$, x1xx xxxx $_b$
i	8 + 8 * n (RBAS)	0 ... (2^{7+7*n} – 2) (2^{7+7*n} – 1), reserved
ID	8, if t = 0 Variable, if t = 1	See Table 6 See Figure 10 and Table 8
L_{ZOI}	16 + 8 * n (RBAS)	0 ... 2^{15+7*n}
ZOI	Variable	See 5.7
L_{PID}	16 + 8 * n (RBAS)	0 ... 2^{15+7*n}
P_{ID}	Variable	Table 7, if t = 0 Managed by JPSEC registration authority, if t = 1

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

The first byte of the tool uses a one-byte FBAS structure whose first BAS bit represents the tool type, t, where 0 specifies a JPSEC normative tool and 1 specifies a JPSEC non-normative tool. This is followed by the instance index, i, which is represented using the RBAS structure. The instance index shall be a unique identifier of the tool within the codestream, and thus shall not be repeated by any other tool in the codestream, even if it is in a different SEC marker segment. The instance index is especially critical (and necessary) when INSEC markers are used, because each INSEC marker segment contains the instance index of the tool to which it applies. It is recommended that the first tool applied at a JPSEC creator has an instance index of 1, and that each additional tool be indexed sequentially as it is applied at the protector.

In addition, each JPSEC tool has an ID number which is 8 bits for JPSEC normative tools and 32 bits for JPSEC non-normative tools. For JPSEC normative tools, the ID number describes which protection method template is used, i.e., it specifies the decryption template, authentication template, or hash template. For JPSEC non-normative tools, the

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

5.6.2 JPSEC normative tool

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

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

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

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

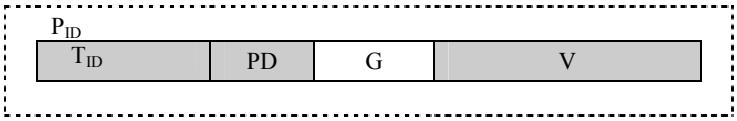


Figure 9 – Parameters (P_{ID}) syntax for JPSEC normative tools (t = 0)

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

PD: Processing domain for JPSEC normative tool.

G: Granularity for JPSEC normative tool.

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

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

Table 7 – JPSEC normative tool parameter values

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

5.6.3 JPSEC non-normative tool

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 5.6.1 and shown in Figure 8, where the tool type t = 1 and the identifier ID_{RA} consists of a name space and an ID number, as defined by Figure 10 and Table 8.

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 $ID_{RA,id}$ identifier shown in Table 9, where the identifiers whose first bit is a 0 are defined by a registration authority, and those whose first bit is a 1 are defined by a particular JPSEC application.

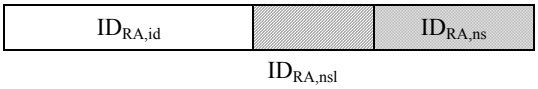


Figure 10 – ID_{RA} syntax

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

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

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

Table 8 – Parameters values in ID_{RA} syntax

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

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

$ID_{RA,id}$	Meaning
0x00 00 00 00 ... 0x7F FF FF FF	JPSEC registration authority tool. Values shall be managed by JPSEC registration authority.
0x80 00 00 00 ... 0xEF FF FF FF	JPSEC user-defined tool. Values can be defined by a particular JPSEC application.
0xF0 00 00 00 ... 0xFF FF FF FF	Reserved for ISO use.

For RA tools, the field $ID_{RA,ns}$ is the name space of the Registration Authority (RA) with which this tool is registered. As each RA has a unique name space, the $ID_{RA,id}$ and $ID_{RA,ns}$ are used together to identify an RA tool. For user-defined tools, the field $ID_{RA,ns}$ is chosen by the developers. In order to limit the risk of ID collisions, it is recommended that the developers seek uniqueness when choosing their name space, for example, by choosing the domain name of their

organization or company. However, note that for user-defined tools, there is no way to guarantee that uniqueness of the name space, so ID collision can occur and should be carefully considered when using user-defined tools.

The P_{ID} field is used to transmit one or more parameters for the JPSEC non-normative tool *i*. The format of the P_{ID} field is not fully given in the scope of JPSEC. If a 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 protection method templates, processing domain, granularity, and value list fields described in 5.8, 5.9, 5.10 and 5.11, respectively.

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

5.7 Zone of Influence (ZOI) syntax

5.7.1 Introduction

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

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

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

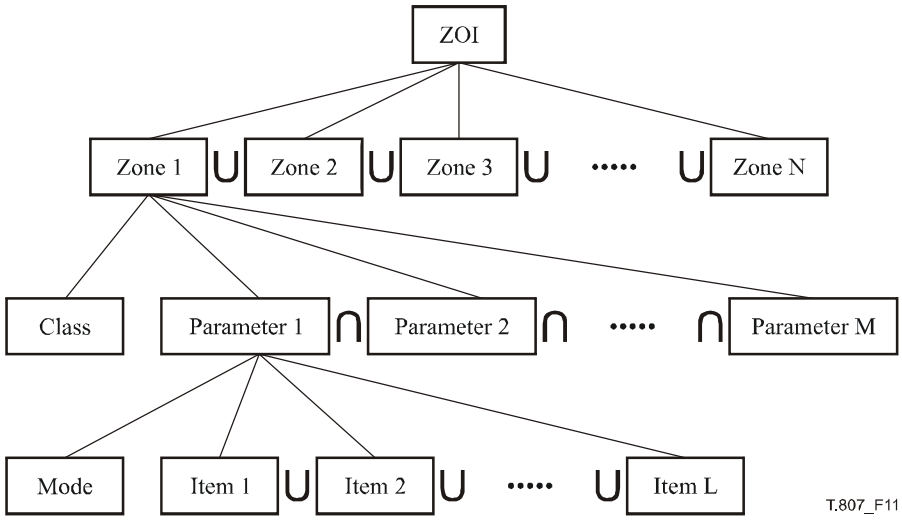


Figure 11 – Zone of Influence conceptual structure

5.7.2 ZOI syntax

Figure 12 shows the ZOI syntax. The ZOI 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.

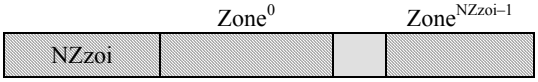


Figure 12 – ZOI syntax

NZzoi: Number of Zones. This field uses the RBAS structure.

Zone^{k}: Zone. Its structure is specified in 5.7.3.

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

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

5.7.3 Zone syntax

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

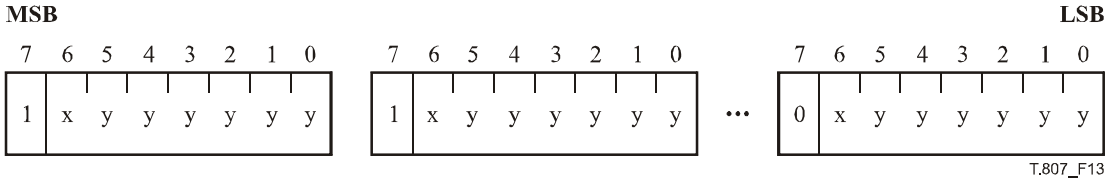


Figure 13 – Zone description class structure (DCzoi)

Figure 14 shows the Zone syntax.

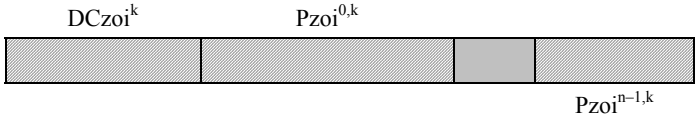


Figure 14 – Zone syntax consists of a description class and one or more parameter sets

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

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

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

Table 11 – Zone parameter values

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

Table 12 – Description class indicator value

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

Table 13 – Image related description class

Bit number	Semantics
1	Image region
2	Tile(s) as defined in 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 (Tile-Resolution-Layer-Component-Precinct) 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	Bit-rate
13	User-defined. The details shall be specified by other means. (e.g., JPSEC ID)
	All other values are reserved

Table 14 – Non-image related description class

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

Packet indices are numbered sequentially within a tile, and therefore may not be unique across tiles. Furthermore, packet indices within a tile may roll over when their maximum value of 65535 is exceeded. For this reason, packet indexing is described in more detail. When the packet indices within a tile do not exceed 65535 packets, then the packet index described in Table 13 is defined by the packet index given by the SOP Nsop parameter as defined in Table A.40

in the JPEG 2000 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 65535 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. Please note that security services which require unique packet indices are vulnerable if the packet index rolls over and repeats.

When the TRLCP tag is used, its format 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 4. This defines the size of TRLCP tags in the ZOI.

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 in the same order. For example, if the zone uses byte ranges and packet ranges, each should have the same number of range items where the first byte range corresponds to the first packet range, and so on.

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

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

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

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

5.7.3.1 Byte range fields

The non-image-related description class allows the ZOI to be described in byte ranges. In general, the 2nd and 3rd elements of Table 14 should be used to represent the byte ranges for most tools such as authentication and encryption/decryption without padding. However, some protection methods, such as encryption/decryption with padding, change the length of the data. When this occurs, it is necessary to specify both the padded byte range and the unpadded or original byte range. In this case, the padded byte range is specified by the 2nd or 3rd element of Table 14 according to the needs of the protection tool. (Note that these two elements cannot be used together.) In addition, the unpadded byte range is specified by the 4th element of Table 14. The unpadded byte range should be specified with the same description mode as the padded byte range and have same number of items. These items should correspond to each other in a one-to-one manner in the same order.

5.7.3.2 Distortion field and relative importance field

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

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

5.7.3.2.1 One-byte distortion field

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

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

where m has a value in the range $0 \le m \le 15$ and exp has a value in the range $0 \le exp \le 15$. 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 $\frac{1}{2} \times (1/2^4) = 1/32$ or about 3%. With 4 bits for the exponent and using base-16 the dynamic range is from 0 to max, where max is given by m = 15 and exp = 15 which corresponds to a distortion of $15 \times 16^{15} = 1.7 \times 10^{19}$.

exp	M

Figure 15 – Distortion field syntax

exp: Exponent of distortion field value (base 16)

m: Matissa of distortion field value

Table 15 – Distortion field parameter values

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

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

5.7.3.2.2 Two-byte distortion field

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

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

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

ε	μ

Figure 16 – Distortion field syntax

ε: Exponent of two-byte distortion field value.

μ: Matissa of two-byte distortion field value.

Table 16 – Distortion field parameter values

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

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

- convert V to a binary number ($12.25_{10} = 1100.01_2$);
- normalize the number; this means there should be a 1 digit to the left of the binary point and multiplication by the appropriate power of two to represent the original value. The normalized form of 1100.01 is 1.10001×2^3;
- the exponent is the power of 2, presented in excess notation. The exponent bias is 15; hence for this example the exponent is represented as 18_{10} (10010_2);
- the mantissa represents the significant bits, *except for the bit to the left of the binary point*, which is always one and therefore does not need to be stored; zeros are possibly appended so as to obtain 11 bits. For this example, the mantissa is 10001000000.

5.7.3.2.3 Relative importance field

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

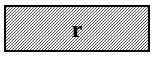


Figure 17 – Relative importance field syntax

r: Relative importance value

Table 17 – Relative importance field parameter values

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

5.7.3.2.4 Additional comments on distortion field and relative importance field

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

Headers can be specified using the distortion or relative importance fields. The loss of various types of data, such as the main and tile-part headers or the SEC header, prevent the decoding of the related image data. The JPSEC creator 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 for the one-byte fields is a byte of all ones (0xFF). Note that this value is the highest possible distortion value for both the one-byte total squared error distortion field and for the one-byte relative importance field. The highest distortion value for the two-byte distortion field is the two bytes of all ones (0xFFFF). The highest importance for the relative importance field of length n-bytes is an n-byte value of all ones.

5.7.3.2.5 Joint use of distortion field and relative importance field

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

5.7.3.3 Bit-rate field

The Bit-rate field is used to specify the protected zone in wavelet coefficient domain. It identifies the most significant bit-planes whose compressed bit-rate is specified by this field. The MSBs are selected using the rate-distortion optimization process specified in Part 1. For example, if the Bit-rate value is 2.5, the protected zone includes the MSBs of all wavelet coefficients whose compressed bit-rate is 2.5 bit per pixel. The syntax of Bit-rate field is shown in Figure 18 and Table 18. The specified bit-rate is given by:

$$R = I_R + F_R/16$$

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

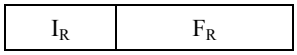


Figure 18 – Bit-rate field syntax

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

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

Table 18 – Bit-rate field parameter values

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

5.7.4 Relationship between multiple parameters

5.7.4.1 Global

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

The non-image related description class 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 5.7.6) where this range contains N elements and when another field f2 is specified by a list of N items. In this case, the field f1, which contains only 1 item (the range) is interpreted as a list of N items. These N items specified by the range in f1 shall correspond one-to-one with the N items listed in f2. Therefore, a range of items can be associated to either a single item or to multiple items (one for each item in the range).

5.7.4.2 Examples

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

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

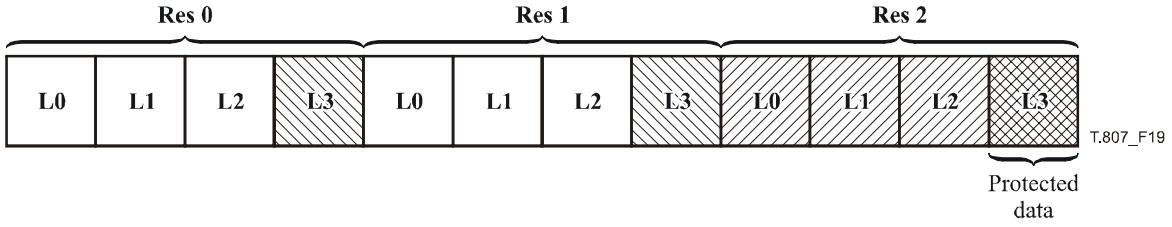


Figure 19 – ZOI example using image related descriptions

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

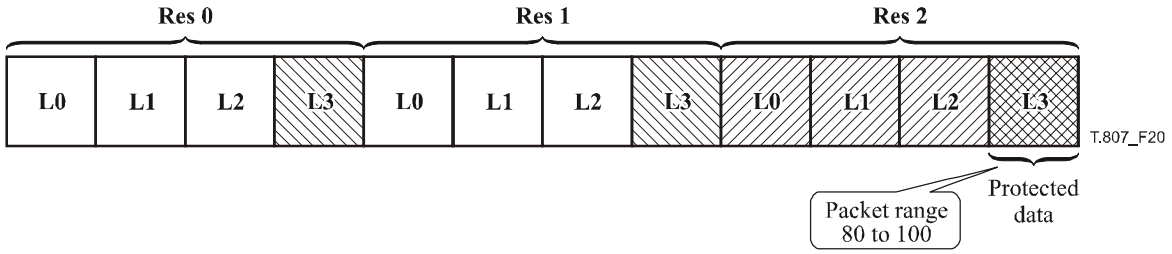


Figure 20 – ZOI example using image related and non-image related descriptions

In the third example, the zone description has two image related descriptions (which are resolution 2 and layer 3) and two non-image related descriptions (which are packet range 80-100 and byte range 856-1250). Once again, the influenced data is the intersection of resolution 2 and layer 3, and the influenced data is contained in packets ranging from 80 to 100. Furthermore these packets and influenced area are located in the byte range 856-1250.

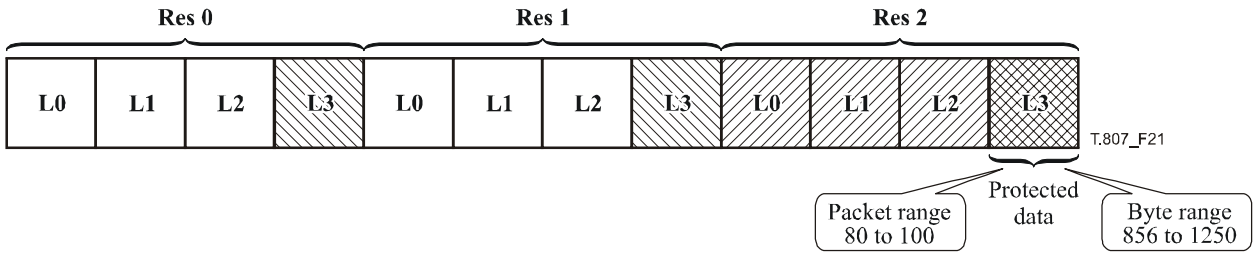


Figure 21 – A second ZOI example using image related and non-image related descriptions

5.7.5 Protecting any data that follows the SEC marker

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

Specifically, the byte range mode of the non-image related description class can be used to specify that a JPSEC tool should be applied to any data following the SEC marker. As described before, the first byte of the SEC header is byte 1 for indexing the byte range. The data that follows the SEC marker and that can be protected includes the SEC segment and most of the main header. Note that all of the JPEG 2000 main header, except for the SIZ marker segment, may 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 consumer to first render the SEC segment data, which can then be used to process the remainder of the codestream.

5.7.6 Zone description parameter syntax (Pzoi)

Figure 22 shows the ZOI description parameter syntax.

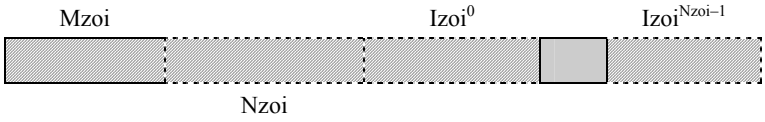


Figure 22 – ZOI description parameter syntax

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

Nzoi: Number of Izoi. This field uses RBAS structure.

Izoi^{i}: Item.

Table 19 – Pzoi^{i} parameter values

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

Table 20 – Mzoi parameter values

FBAS bit number	Values (bits)	Semantics
1	0	The specified zones are influenced by the JPSEC tool
	1	The complement of specified zones are influenced
2	0	Single item is specified
	1	Multiple items are specified
3, 4	00	Rectangle mode. A rectangle region where the first value pair specifies the upper-left corner and the second value pair specifies the lower-right corner such that both corners are inclusive. For each corner, the first value shall be the horizontal position and the second value shall be the vertical position. The indexing shall begin at 0, and shall use the reference grid defined in 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 integer
	01	Izoi^{i} uses 16-bit integer
	10	Izoi^{i} uses 32-bit integer
	11	Izoi^{i} uses 64-bit integer

Table 20 – Mzoi parameter values

FBAS bit number	Values (bits)	Semantics
7, 8	00	$Izoi^i$ is described in one dimension
	10	$Izoi^i$ is described in two dimensions
	01	$Izoi^i$ is described in three dimensions
9	0	Offset with lengths mode is not used
	1	Offset with lengths mode is used: Specifies the initial offset with lengths of contiguous bytes that follows. The existence of this flag shall override the modes specified in bits 3 and 4.
		All other values are reserved

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

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

5.8 Protection method template syntax (T)

5.8.1 General

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

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

Table 21 – Template ID values (ID_T)

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

5.8.2 Decryption template ($T = T_{decry}$, if t = 0 and ID = 1)

The decryption template, T_{decry}, is used to communicate to the decryptor, how to decrypt the received codestream. Figure 23 shows the decryption template syntax. Table 22 shows the sizes and values of the symbols and parameters for the decryption template.

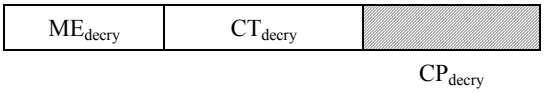


Figure 23 – Decryption template syntax

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. This field uses the FBAS structure.

CT_{decry}: Cipher type identification.

CP_{decry}: Cipher parameter.

Table 22 – Decryption template parameter values

Parameter	Size (bits)	Values
ME_{decry}	8 + 8 * n (FBAS)	Table 23
CT_{decry}	16	Table 24
CP_{decry}	Variable	If $CT_{decry} < 0x6000$, see 5.8.2.1. If $0x6000 \leq CT_{decry} < 0xC000$, see 5.8.2.2. If $CT_{decry} \geq 0xC000$, see 5.8.2.3.

Table 23 – Marker emulation flag values (ME_{decry})

Values	Method type
01xx xxxx	Encrypted data does not contain a false marker emulation
00xx xxxx	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 24 – Cipher identifier values (CT_{decry})

Values	Cipher type
0 ... 0x5FFF	Block cipher (see Table 25)
0x6000 ... 0xBFFF	Stream cipher (see Table 26)
0xC000 ... 0xFFFF	Asymmetric cipher (see Table 27)

Table 25 – Block cipher identifier values (CT_{decry})

Values	Cipher type
0x0000	NULL (no encryption)
0x0001	AES (ISO/IEC 18033-3)
0x0002	TDEA (ISO/IEC 18033-3)
0x0003	MISTY1 (ISO/IEC 18033-3)
0x0004	Camellia (ISO/IEC 18033-3)
0x0005	CAST-128 (ISO/IEC 18033-3)
0x0006	SEED (ISO/IEC 18033-3)
	All other values are reserved for ISO use

Table 26 – Stream cipher identifier values (CT_{decry})

Values	Cipher type
0x6000	SNOW 2 (ISO/IEC 18033-4)
	All other values are reserved for ISO use

Table 27 – Asymmetric cipher identifier values (CT_{decry})

Values	Cipher type
0xC000	RSA-OAEP (ISO/IEC 18033-2)
	All other values are reserved for ISO use

5.8.2.1 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 codestream. Figure 24 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 5.10 and Value list field (V) described in 5.11. Specifically, initialization vectors are only used for modes with ID M_{bc} > 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_{bc}.

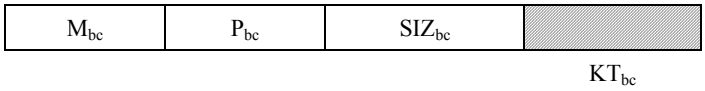


Figure 24 – Block cipher template syntax

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

P_{bc}: Padding mode.

SIZ_{bc}: Size of block in Bytes.

KT_{bc}: Key template (see 5.8.5). It holds information on the keys used by the block cipher.

Table 28 – Block cipher template values

Parameter	Size (bits)	Values
M_{bc}	6	Table 29
P_{bc}	2	Table 30
SIZ_{bc}	8	1 ... 256
KT_{bc}	Variable	See 5.8.5

Table 29 – Block cipher mode values (M_{bc})

Values	Mode type
0	Reserved
0x xxxx	Modes which are used without IV
1x xxxx	Modes which are used with an IV
x0 xxxx	Bits are not padded
x1 xxxx	Bits are padded
0x 0001	ECB (ISO/IEC 10116)
1x 0010	CBC (ISO/IEC 10116)
1x 0011	CFB (ISO/IEC 10116)
1x 0100	OFB (ISO/IEC 10116)
1x 0101	CTR (ISO/IEC 18033-2)
	All other values are reserved for ISO use

NOTE 1 – Careful implementations are required for all modes, because improper implementations may lead to vulnerabilities. Note that even correct implementation of ECB has information leakage when identical blocks appear. Guidelines are contained in ISO/IEC 10116.

NOTE 2 – Values in Table 30 only apply when M_{bc} in Table 29 specifies that Bits are padded. When bits are not padded, P_{bc} shall be set to 00.

Table 30 – Padding mode for block cipher (P_{bc})

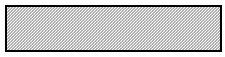
Values	Padding type
00	Ciphertext stealing (RFC 2040)
01	PKCS#7-padding (PKCS#7)
	All other values are reserved for ISO use

NOTE 3 – When using padding, careful system design must be used to avoid potential security vulnerabilities, such as chosen cipher attacks.

5.8.2.2 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 codestream. Figure 25 shows the stream cipher template syntax. Table 31 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 5.10 and Value list field (V) described in 5.11. 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}.



KT_{sc}

Figure 25 – Stream cipher template syntax

KT_{sc}: Key information template (see 5.8.5). It holds information on the keys used by the stream cipher.

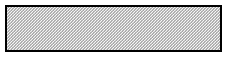
Table 31 – Stream cipher template values

Parameter	Size (bits)	Values
KT_{sc}	Variable	See 5.8.5.

5.8.2.3 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 codestream. Figure 26 shows the asymmetric cipher template syntax. Table 32 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 (N_v) in the Value list field shall be set to 0.



KT_{sy}

Figure 26 – Asymmetric cipher template syntax

KT_{sy}: Key information template (see 5.8.5). It holds information on the keys used by the asymmetric cipher.

Table 32 – Asymmetric cipher template values

Parameter	Size (bits)	Values
KT_{sy}	Variable	See 5.8.5

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

The authentication template, T_{auth}, is used to communicate to the verifier, how to verify the authenticity of received codestream. 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 to as MAC values. The authentication template syntax is shown in Figure 27, and Table 33 shows the sizes and values of the symbols and parameters for the authentication template.

In many security applications, authentication is the most important security service. Even when confidentiality is the targeted security service, it should be augmented by authentication to prevent attacks. In particular, it is recommended to authenticate parts of the SEC marker segment. In addition, the authentication shall be performed on both the authentication template parameters (T_{auth}) and the message to authenticate. Specifically, the zone of influence shall specify that both the content and the authentication template parameters (T_{auth}) are to be authenticated.

Figure 27 – Authentication template syntax

M_{auth}: Authentication method.

P_{auth}: Authentication parameters.

Table 33 – Authentication template parameter values

Parameter	Size (bits)	Values
M_{auth}	8	Table 34
P_{auth}	Variable	If $M_{auth} = 0$, see 5.8.3.1 If $M_{auth} = 1$, see 5.8.3.2 If $M_{auth} = 2$, see 5.8.3.3

Table 34 – Authentication methods (M_{auth})

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

5.8.3.1 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 codestream. Figure 28 shows the hash-based authentication template syntax and Table 35 shows the parameter values.

The MAC values are specified using the tool's granularity field (G) described in 5.10 and Value list field (V) described in 5.11. The size of the MAC value specified in the value list V shall be set to the MAC size defined by SIZ_{HMAC}.

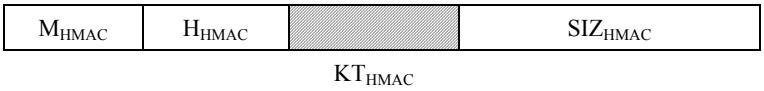


Figure 28 – Hash-based authentication template

M_{HMAC}: Hash-based authentication method identifier.

H_{HMAC}: Hash identifier.

KT_{HMAC}: Key template.

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

Table 35 – Hash-based authentication template parameter values

Parameter	Size (bits)	Values
M_{HMAC}	8	Table 36
H_{HMAC}	8	Table 37
KT_{HMAC}	Variable	See 5.8.5
SIZ_{HMAC}	16	0 ... 65535

Table 36 – Hash-based authentication method identifier (M_{HMAC})

Values	Hash-based authentication method
0	Reserved
1	HMAC (ISO/IEC 9797-2)
	All other values are reserved for ISO use

Table 37 – Hash function identifier (H_{HMAC})

Values	Hash function
0	Reserved
1	SHA-1 (ISO/IEC 10118-3)
2	RIPEMD-128 (ISO/IEC 10118-3)
3	RIPEMD-160 (ISO/IEC 10118-3)
4	MASH-1 (ISO/IEC 10118-4)
5	MASH-2 (ISO/IEC 10118-4)
6	SHA-224 (ISO/IEC 10118-3)
7	SHA-256 (ISO/IEC 10118-3)
8	SHA-384 (ISO/IEC 10118-3)
9	SHA-512 (ISO/IEC 10118-3)
10	WHIRLPOOL (ISO/IEC 10118-3)
	All other values are reserved for ISO use

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

5.8.3.2 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 codestream. Figure 29 is its template and Table 38 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 length and of value 0. The blocksize is the default for the block cipher. Note that if the SIZ_{CMAC} is less than the nominal size of the cipher-based authentication MAC, then it is the truncated version corresponding to the first SIZ_{CMAC} bits of the MAC.

Note that, if the number of data bits is not a multiple of the cipher block size, then the final input block will be a partial block of data, left justified, with zeroes appended to form a full cipher block. Also note that CBC-MAC shall only be applied to data with a fixed and known length.

The MAC values are specified using the tool's granularity field (G) described in 5.10 and value list field (V) described in 5.11. The size of the MAC value specified in the value list V shall be set to the MAC size defined by SIZ_{CMAC}.

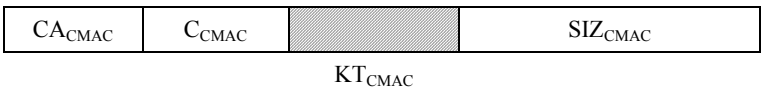


Figure 29 – Cipher-based authentication template syntax

CA_{CMAC}: Cipher-based authentication method.

C_{CMAC}: Block-cipher identifier value.

KT_{CMAC}: Key template.

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

Table 38 – MAC template values

Parameter	Size (bits)	Values
CA_{CMAC}	8	Table 39
C_{CMAC}	8	Table 25
KT_{CMAC}	Variable	See 5.8.5
SIZ_{CMAC}	16	0 ... 65535

Table 39 – Cipher-based authentication method (C_{CMAC})

Values	Method
0	CBC-MAC MAC Algorithm 1 (ISO/IEC 9797-1)
1	CBC-MAC MAC Algorithm 2 (ISO/IEC 9797-1)
2	CBC-MAC MAC Algorithm 3 (ISO/IEC 9797-1)
3	CBC-MAC MAC Algorithm 4 (ISO/IEC 9797-1)
	All other values are reserved for ISO use

5.8.3.3 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 codestream, as well as verifying the identity of the sender for both identity and non-repudiation purposes. Figure 30 defines its template and Table 40 lists the values.

The digital signatures are specified using the tool's granularity field (G) described in 5.10 and value list field (V) described in 5.11. 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 number 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.

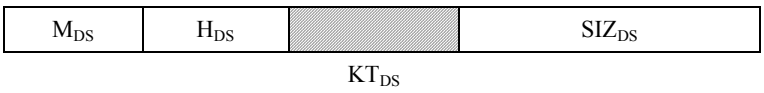


Figure 30 – Digital signature template syntax

M_{DS}: Digital signature method.

H_{DS}: Hash function.

KT_{DS}: Key template (see 5.8.5). 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 40 – Digital signature template values

Parameter	Size (bits)	Values
M_{DS}	8	Table 41
H_{DS}	8	Table 37
KT_{DS}	Variable	See 5.8.5
SIZ_{DS}	16	0 ... 65535

Table 41 – Digital signature methods (M_{DS})

Values	Method
1	RSA (ISO/IEC 14888-2)
2	Rabin (ISO/IEC 14888-2)
3	DSA (ISO/IEC 14888-3)
4	ECDSA (ISO/IEC 14888-3)
	All other values are reserved for ISO use

5.8.4 Hash template ($T = T_{hash}$, if $t = 0$ and $ID = 3$)

The hash template, T_{hash}, is used to communicate the parameters used to compute the hash. Table 42 shows the sizes and values of the symbols and parameters for hash template.

Note that in contrast to the hash-based authentication template discussed in 5.8.3.1 which involves the use of a hash and a secret key, this hash template does not use a key. While this hash template can be used to detect an accidental error or accidental change to the data, 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 5.10 and value list field (V) described in 5.11. The size of the hash value specified in the value list V shall be set to the hash value size defined by SIZ_{hash}.

H_{hash}	SIZ_{hash}

Figure 31 – Hash template syntax

H_{hash}: Hash function identifier.

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

Table 42 – Hash template parameter values

Parameter	Size (bits)	Values
H_{hash}	8	Table 37
SIZ_{hash}	8	0 ... 255

5.8.5 Key information template (KT)

The key information template is used to communicate key information. Figure 32 defines its template and Table 43 lists the values.

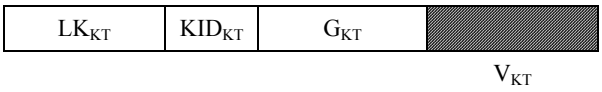


Figure 32 – Key information template syntax

LK_{KT}: Length of key in bits.

KID_{KT}: Key information identifier. It indicates the meaning of the values in the value list V_{KT}. In the decryption template, this value should be set to 2 (URI to retrieve the secret key). In the case of digital signature, the value of this field is free.

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 one or more values using the tool's granularity field (G_{KT}) described in 5.10 and value list field (V_{KT}) described in 5.11. The two fields (G_{KT} and V_{KT}) together determine how the key values in the value list (V_{KT}) are applied to the protected image data, as described in 5.10 and 5.11.

The key information in the value list can take one the forms specified in Table 44. If $KID_{KT} = 1$, then each value is specified with the X.509 certificate template described in 5.8.5.1. If $KID_{KT} = 2$, then each value is specified with a URI for the certificate or secret key.

Table 43 – Key template values

Parameter	Size (bits)	Values
LK_{KT}	16	1 ... 65535
KID_{KT}	8	Table 44
G_{KT}	24	See 5.10
V_{KT}	Variable	See 5.11

Table 44 – Key information identifier values (KID_{KT})

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

5.8.5.1 X.509 certificate template

Figure 33 – 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 45 – X.509 certificate values (KI_{KT} if KID_{KT} = 2)

Parameter	Size (bits)	Values
ER_{KT}	8	0 ... 255 (see Table 46)
$LCER_{KT}$	16	1 ... 65535
CER_{KT}	Variable	–

Table 46 – Encoding rule values (ER_{KT})

Values	Encoding rule identifier
0	Reserved
1	DER (RFC 3217)
2	BER (RFC 3394)
	All other values are reserved for ISO use

5.9 Processing domain syntax (PD)

The processing domain syntax is used to indicate on which domain the JPSEC tool is applied. The possible domains include pixel domain, wavelet coefficient domain, quantized wavelet coefficient domain and codestream domain.

Figure 34 – Processing domain syntax

PD: Processing domain. This field uses the FBAS structure.

F_{PD}: Processing domain field to provide further detailed information about the processing domain. This field uses the FBAS structure.

Table 47 – Processing domain parameters

Parameter	Size (bits)	Values
PD	Variable (FBAS)	See Table 48
F_{PD}	Variable (FBAS)	In wavelet coefficient domain and quantized wavelet coefficient domain, see Table 49. In codestream domain, see Table 50.

Table 48 – Processing Domain (PD) parameter values

FBAS bit number	Values	Semantics
1	1	Pixel domain. Protection method is applied on image pixels.
	0	Otherwise
2	1	Wavelet coefficient domain. Protection method is applied on wavelet coefficients.
	0	Otherwise
3	1	Quantized wavelet coefficient domain: Protection method applied on quantized wavelet coefficient
	0	Otherwise
4	1	Codestream domain: Protection method is applied on codestream generated from arithmetic coder.
	0	Otherwise

Note that the field PD shall have one and only bit set to 1, because each JPSEC tool is applicable to one domain only.

In image pixel domain, wavelet coefficient domain and quantized wavelet coefficient domain, the two-dimensional data has to be transformed to one-dimensional in order to apply the security tools. This transformation shall be done by scanning the two-dimensional image data in the raster-scan order.

Table 49 – Processing domain field (F_{PD}) parameter values in wavelet coefficient domain and quantized wavelet coefficient domain

FBAS bit number	Value	Semantics
1	0	Protection method is applied on sign bit
	1	Protection method is applied on most significant bit

Table 50 – Processing domain field (F_{PD}) parameter values in codestream domain

FBAS bit number	Value	Semantics
1	0	Protection method is applied on both packet header and packet body
	1	Protection method is applied on packet body only

The field (F_{PD}) is used to provide further information on processing domain. With different value of PD, this field (F_{PD}) has different semantics. For instance, in wavelet coefficient domain and quantized wavelet coefficient domain, the first bit of F_{PD} is used to indicate whether the JPSEC tool is applied on the most significant bit. In codestream domain, the first bit of F_{PD} is used to indicate whether the JPSEC tool is applied on packet body only or both packet header and body; in the pixel domain, this field (F_{PD}) is reserved.

5.10 Granularity syntax (G)

Granularity is used to indicate the unit of protection for each protection method. Table 53 defines possible granularities. Figure 35 shows the granularity syntax.

PO	GL

Figure 35 – Granularity syntax

PO: Processing order.

GL: Granularity level.

Table 51 – Granularity parameter values (G)

Parameter	Size (bits)	Values
PO	16	See Table 52
GL	8	See Table 53

Table 52 – Processing order values (PO)

Values MSB LSB	Processing order
0 000 000 000 000 000	Order specified by Zone of Influence image-related parameters
1 000 000 000 000 000	Order specified by Zone of Influence non-image-related bitstream parameters
1 000 000 000 000 001	Order specified by Zone of Influence non-image-related packet parameters
0 000 001 010 011 100	Tile-resolution-layer-component-precinct
0 000 011 100 001 010	Tile-component-precinct-resolution-layer
0 000 010 001 011 100	Tile-layer-resolution-component-precinct
0 000 100 011 001 010	Tile-precinct-component-resolution-layer
0 000 001 100 011 100	Tile-resolution-precinct-component-layer
	All other values are reserved

Table 53 – Granularity level values (GL)

Values MSB LSB	Granularity
0000 0000	Tile
0000 0001	Tile-part
0000 0010	Component
0000 0011	Resolution level
0000 0100	Layer
0000 0101	Precinct
0000 0110	Packet
0000 0111	Sub-band
0000 1000	Code-block
0000 1001	Total area identified in ZOI
1000 0000	Item identified in non-image-related ZOI
1000 0001	Zone identified in non-image-related ZOI
	All other values are reserved

In order to process the entire zone specified by ZOI, the granularity level should be "zone identified in ZOI".

5.11 Value list syntax (V)

The Value list field is used to specify values that change 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 5.6.2, 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_{bc} or IV_{sc} depending on whether a block cipher or stream cipher is 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 hash template, it represents the hash value HV_{hash}. 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 36 – Value list field syntax

N_V: Number of values in the value list VL, If $N_V = 0$, then the field terminates. This field uses RBAS structure.

S_V: Size of each value in the value list VL in bytes. This field uses RBAS structure.

VL: List of values.

Table 54 – Value list field (V) parameter values

Parameter	Size (bits)	Values
N_V	16 + 8 * n (RBAS)	0 ... $(2^{15+7*n} - 1)$
S_V	8 + 8 * n (RBAS)	0 ... $(2^{7+7*n} - 1)$
VL	0, if NV = 0 N_V * S_V, otherwise	N/A Determined by template

5.12 Relationships among ZOI, Granularity (G) and Value List (VL)

The ZOI, PO and GL are used together to ensure the unique behaviour of the applied JPSEC tool(s), regardless of the progress order of the JPEG 2000 codestream. In other words, the resulted signature, MAC values and encrypted codestream are independent of the progressive order of the JPEG 2000 codestream. The Zone of Influence (ZOI) specifies, in its entirety, the part of the JPEG 2000 codestream to be protected by the JPSEC tool; The Processing Order (PO), on the other hand, specifies the order in which the JPSEC tool processes the codestream; the Granularity Level (GL) specifies the protection units containing contiguous byte sequence in the re-ordered codestream. Finally, each protection unit corresponds to a value in the Value List (VL), in the order they appear in the re-ordered codestream. The relationship can be illustrated by one example, where the JPEG 2000 codestream has 1 tile, 3 resolution levels and 3 layers, and the number of components and precincts are not important. The progressive order is RLCP in original JPEG 2000 codestream, the Zone of Influence is resolution 0 and 1, and the Processing Order (PO) is TRLCP. Figures 37 and 38 illustrate the re-ordering of the codestream and the mapping from each protection unit to the Value List (VL), when the Granularity Level (GL) is resolution and layer, respectively.

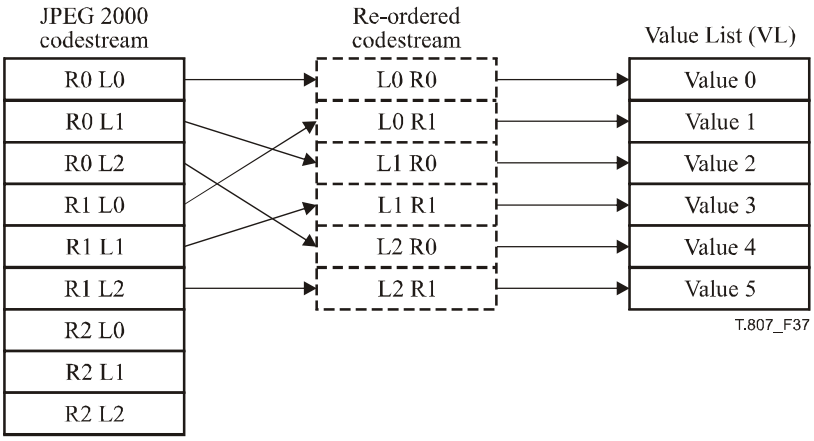


Figure 37 – Granularity Level (GL) is resolution

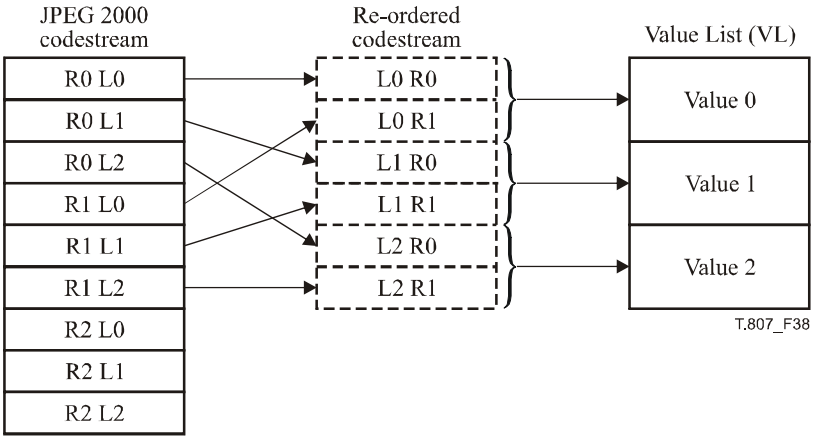


Figure 38 – Granularity Level (GL) is layer

NOTE – The re-ordered codestream is only used to generate the values in Value List (VL). The final JPSEC codestream will have the same progressive order as the original JPEG 2000 codestream.

5.13 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 in 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.

The syntax of the INSEC marker is defined in Figure 39.

Figure 39 – In-codestream security marker syntax

INSEC: Marker code. Table 55 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. This field uses the RBAS structure.

R: Relevance zone for the INSEC information. This field uses the FBAS structure.

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

Table 55 – In-codestream security parameter values (INSEC)

Parameter	Size (bits)	Values
INSEC	16	0xFF94
L_{INSEC}	16	2 ... $(2^{16} – 1)$
i	8 + 8 * n (RBAS)	0 ... $(2^{7+7*n} – 1)$
R	Variable (FBAS)	See Table 56
AP	Variable	Defined by registration authority or application.

Table 56 – Relevance zone values (R)

FBAS bit number	Values	Relevance zone
0	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 subclause contains examples that show 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 subclause 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 57 – ZOI in example 1

Parameter			Size (bits)	Value (in order)	Derived meaning
NZzoi			8 (RBAS)	1	Number of Zones is one
$Zone^0$	DCzoi		1	0_b	The byte aligned segment does not follow
			1	0_b	Image related description class
			6	101000_b	Image regions and resolution levels are specified in order
	$Pzoi^1$	$Mzoi^1$	1	0_b	The byte aligned segment does not follow
			1	0_b	The specified zones are influenced by the JPSEC tool
			1	0_b	Single item is specified
			2	00_b	Rectangle mode
			2	00_b	Izoi uses 8-bit integer
			1	1_b	Izoi is described in two dimensions
		$Izoi^1$	8	$0110\ 0100_b$	Xul is 100
			8	$0111\ 1000_b$	Yul is 120
			8	$1011\ 0100_b$	Xlr is 180
			8	$1101\ 0010_b$	Ylr is 210
	$Pzoi^3$	$Mzoi^3$	1	0_b	The byte aligned segment does not follow
			1	1_b	The complement of the specified zones is influenced by the JPSEC tool
			1	0_b	Single item is specified
			2	11_b	Max mode
			2	00_b	Izoi uses 8-bit integer
			1	0_b	Izoi is described in one dimension
		$Izoi^3$	8	$0000\ 0010_b$	Resolution levels ≤ 2 are specified. (i.e., Resolution levels > 3 are specified with Max mode and complement switch.)

6.1.2 Example 2

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

Table 58 – ZOI in example 2

Parameter			Size (bits)	Value (in order)	Derived meaning
NZzoi			8 (RBAS)	1	Number of Zones is one
$Zone^0$	$DCzoi^1$		1	1_b	The byte aligned segment follows
			1	0_b	Image related description class
			6	001000_b	Resolution levels are specified
	$DCzoi^2$		1	0_b	The byte aligned segment does not follow
			1	0_b	Image related description class
			6	001100_b	Sub-bands and code-blocks are specified
	$Pzoi^3$	$Mzoi^3$	1	0_b	The byte aligned segment does not follow
			1	0_b	The specified zones are influenced by the JPSEC tool
			1	0_b	Single item is specified
			2	10_b	Index mode
			2	00_b	Izoi uses 8-bit integer
			1	0_b	Izoi is described in one dimension
		$Izoi^3$	8	$0000\ 0000_b$	Resolution level index is 0
	$Pzoi^9$	$Mzoi^8$	1	0_b	The byte aligned segment does not follow
			1	0_b	The specified zones are influenced by the JPSEC tool
			1	0_b	Single item is specified
			2	10_b	Index mode
			2	00_b	Izoi uses 8-bit integer
			1	0_b	Izoi is described in one dimension
		$Izoi^8$	8	$0000\ 0001_b$	Sub-band 1 is specified
	$Pzoi^{10}$	$Mzoi^9$	1	0_b	The byte aligned segment does not follow
			1	0_b	The specified zones are influenced by the JPSEC tool
			1	0_b	Single item is specified
			2	00_b	Rectangle mode
			2	00_b	Izoi uses 8-bit integer
			1	0_b	Izoi is described in one dimension
		$Izoi^9$	8	$0000\ 0101_b$	Code-block index for the upper-left corner is 5
			8	$0000\ 1010_b$	Code-block index for the lower-right corner is 10

6.1.3 Example 3

This subclause 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 59 – ZOI in example 3

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

6.1.4 Example 4

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

Table 60 – ZOI in example 4

Parameter			Size (bits)	Value (in order)	Derived meaning
NZzoi			8 (RBAS)	1	Number of Zones is one
$Zone^0$	$DCzoi^1$		1	1_b	The byte aligned segment follows
			1	0_b	Image related description class
			6	001000_b	Resolution levels are specified in order
	$DCzoi^2$		1	0_b	The byte aligned segment does not follow
			1	1_b	Non-image related description class
			6	010000_b	Byte ranges are specified
	$Pzoi^1$	$Mzoi^1$	1	0_b	The byte aligned segment does not follow
			1	0_b	The specified zones are influenced by the JPSEC tool
			1	0_b	Single item is specified
			2	10_b	Index mode
			2	00_b	Izoi uses 8-bit integer
			1	0_b	Izoi is described in one dimension
		$Izoi^1$	8	$0000\ 0000_b$	Resolution level is 0

Table 60 – ZOI in example 4

Parameter			Size (bits)	Value (in order)	Derived meaning
$Zone^0$	$Pzoi^2$	$Mzoi^2$	1	0_b	The byte aligned segment does not follow
			1	0_b	The specified zones are influenced by the JPSEC tool
			1	0_b	Single items specified
			2	01_b	Range mode
			2	01_b	Izoi uses 16-bit integer
			1	0_b	Izoi is described in one dimension
		$Izoi^1$	16	0000 0000 0000 1010_b	Starting byte location is 10th (bytes)
			16	0000 0000 0110 0100_b	Ending byte location is 100th (bytes)

6.1.5 Example 5

This subclause 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 61 – ZOI in example 5

Parameter			Size (bits)	Value (in order)	Derived meaning
NZzoi			8 (RBAS)	2	Number of Zones is two
$Zone^0$	DCzoi		1	0_b	The byte aligned segment does not follow
			1	0_b	Image related description class
			6	01 1000_b	Tiles and resolution levels are specified in order
	$Pzoi^2$	$Mzoi^2$	1	0_b	The byte aligned segment does not follow
			1	0_b	The specified zones are influenced by the JPSEC tool
			1	0_b	Single item is specified
			2	00_b	Rectangle mode
			2	00_b	Izoi uses 8-bit integer
			1	0_b	Izoi is described in one dimension
		$Izoi^2$	8	0000 0000_b	Upper-left tile index is 0
			8	0000 0101_b	Lower-right tile index is 5
	$Pzoi^3$	$Mzoi^3$	1	0_b	The byte aligned segment does not follow
			1	1_b	The complement of the specified zones is influenced by the JPSEC tool
			1	0_b	Single item is specified
			2	11_b	Max mode
			2	00_b	Izoi uses 8-bit integer
			1	0_b	Izoi is described in one dimension
		$Izoi^3$	8	0000 0010_b	Resolution levels ≤ 2 are specified. (i.e., Resolution levels > 3 are specified with Max mode and complement switch.)

Table 61 – ZOI in example 5

Parameter			Size (bits)	Value (in order)	Derived meaning
Zone[1]	DCzoi		1	0_b	The byte aligned segment does not follow
			1	0_b	Image related description class
			6	010100_b	Tiles and layers are specified in order
	Pzoi[2]	Mzoi[2]	1	0_b	The byte aligned segment does not follow
			1	0_b	The specified zones are influenced by the JPSEC tool
			1	0_b	Single item is specified
			2	00_b	Rectangle mode
			2	00_b	Izoi uses 8-bit integer
			1	0_b	Izoi is described in one dimension
		Izoi[2]	8	$0000\ 1010_b$	Upper-left tile index is 10
			8	$0000\ 1111_b$	Lower-right tile index is 15
	Pzoi[4]	Mzoi[4]	1	0_b	The byte aligned segment does not follow
			1	0_b	The specified zones are influenced by the JPSEC tool
			1	0_b	Single item is specified
			2	11_b	Max mode
			2	00_b	Izoi uses 8-bit integer
			1	0_b	Izoi is described in one dimension
		Izoi[4]	8	$0000\ 0101_b$	layers ≤ 5 are specified with Max mode

6.1.6 Example 6

This subclause shows the example that the header segment from bytes 10 to 100 is influenced. In this example, 8 bytes are necessary.

Table 62 – ZOI in example 6

Parameter			Size (bits)	Value (in order)	Derived meaning
NZzoi			8 (RBAS)	1	Number of Zones is one
Zone[0]	DCzoi		1	0_b	The byte aligned segment does not follow
			1	1_b	Non-image related description class
			6	001000_b	Byte ranges after the SEC marker are specified
	Pzoi[3]	Mzoi[3]	1	0_b	The byte aligned segment does not follow
			1	0_b	The specified zones are influenced by the JPSEC tool
			1	0_b	Single item is specified
			2	01_b	Range mode
			2	01_b	Izoi uses 16-bit integer
			1	0_b	Izoi is described in one dimension
		Izoi[3]	16	$0000\ 0000$ $0000\ 1010_b$	Starting byte location is 10th (bytes)
			16	$0000\ 0000$ $0110\ 0100_b$	Ending byte location is 100th (bytes)

6.2 Key information template examples

6.2.1 Example 1

Table 63 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 URI in the decryption stage.

Table 63 – Key information in example 1

Parameter		Size (bits)	Value	Derived meaning
LK_{KT}		16	128	Length of key is 128 bits
KID_{KT}		8	2	URI for secret key is identified
G_{KT}	PO	16	000 001 010 011 100 0_b	Processing order is tile-resolution-layer-component-precinct
	GL	8	0000 1001_b	Unit of protection is the total area identified in ZOI
V_{KT}	N_V	16 (RBAS)	1	Number of values in the value list V is 1
	S_V	8 (RBAS)	19	Length of key information is 19 bytes
	V1	152	https://server/file	Secret key can be retrieved from https://server/file

6.2.2 Example 2

Table 64 shows 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 64 – 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
G_{KT}	PO		16	000 001 010 011 100 0_b	Processing order is tile-resolution-layer-component-precinct
	GL		8	0000 1001_b	Unit of protection is total area identified in ZOI
V_{KT}	N_V		16 (RBAS)	1	Number of values in the value list V is 1
	S_V		8 (RBAS)	Variable	Length of X.509 certificate
	V1	ER_{KT}	8	1	X.509 Certificate is encoded with encoding method DER
		$LCER_{KT}$	16	Variable	Length of CER_{KT}
		CER_{KT}	Variable	Certificate value	Certificate with 1024-bit public key is embedded

6.2.3 Example 3

Table 65 shows that a single public key is used to authenticate a codestream, where the public key is embedded into KI_{KT}.

Table 65 – 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
G_{KT}	PO	16	000 001 010 011 100 0_b	Processing order is tile-resolution-layer-component-precinct
	GL	8	0000 1001_b	Unit of protection is total area identified in ZOI
V_{KT}	N_V	16 (RBAS)	1	Number of values in the value list V is 1
	S_V	8 (RBAS)	256	Length of public key is 256 bytes
	V1	2048	Public key value	Public key is embedded

6.2.4 Example 4

Table 66 shows that multiple secret keys are used to decrypt a codestream, where different secret keys are used for different layers.

Table 66 – 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
G_{KT}	PO	16	000 001 010 011 1000_b	Processing order is tile-resolution-layer-component-precinct
	GL	8	$0000\ 0100_b$	Unit of protection is layer
V_{KT}	N_V	16 (RBAS)	3	Number of values in the value list V is 3
	S_V	8 (RBAS)	16	Length of each Vn is 16 bytes
	V1	128	https://server/1	Secret key for the 1st layer can be retrieved from https://server/1
	V2	128	https://server/2	Secret key for the 2nd layer can be retrieved from https://server/2
	V3	128	https://server/3	Secret key for the 3rd layer can be retrieved from https://server/3
	V4	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 capability, 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 precincts 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 signalled. Two ZOIs 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 signalled 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 signalled 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 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 67 – SEC marker segment for example 1

Parameter			Size (bits)	Values	Meaning
SEC			16	0xFF65	SEC marker
L_{SEC}			16 (RBAS)	0x82	Length of SEC marker segment is 130 bytes
Z_{SEC}			8 (RBAS)	0	Index of this SEC marker segment
P_{SEC}	F_{PSEC}		1	0_b	The FBAS structure does not follow
		F_{INSEC}	1	0_b	INSEC is not used
		$F_{multiSEC}$	1	0_b	One SEC marker segment is used
		F_{mod}	2	00_b	Original JPEG 2000 data was modified
		F_{TRLCP}	1	0_b	TRLCP tag usage is not defined in P_{SEC}
		F_{TRLCP}	3	000_b	
	N_{tools}		8 (RBAS)	0000001_b	Number of security tool is one
	I_{max}		8 (RBAS)	0000000_b	Maximum tool instance index is zero
t			8 (FBAS)	0	JPSEC normative tool
i			8 (RBAS)	0	Tool instance index
ID_T			8	1	Decryption template
L_{zoi}			16 (RBAS)	0x17	Length of ZOI is 23 bytes
ZOI			184	See Table 68	Zone of Influence for this tool
L_{PID}			16 (RBAS)	0x5e	Length of P_{ID} is 94 bytes
P_{ID}			752	See Table 69	Parameters for this technology

Table 68 – ZOI example

Parameter			Size (bits)	Value (in order)	Derived meaning
NZzoi			8 (RBAS)	2	Number of Zones is one
$Zone^0$	DC_{ZOI}^1		1	1_b	The byte aligned segment follows
			1	0_b	Image related description class
			6	001000_b	Resolution is specified
	DCzoi[2]		1	0_b	The byte aligned segment does not follow
			1	1_b	Non-image related description class
			6	010000_b	Byte ranges after the SOD marker are specified
	$Pzoi^{0,1}$	$Mzoi^1$	1	0_b	The byte aligned segment does not follow
			1	0_b	The specified zones are influenced by the JPSEC tool
			1	0_b	Single item is specified
			2	10_b	Index mode
			2	00_b	Izoi uses 8 bit integer
			1	0_b	Izoi is described in one dimension
		I_{ZOI}	8	$0000\ 0001_b$	Resolution 1 is specified
	$Pzoi^{0,2}$	$Mzoi^2$	1	0_b	The byte aligned segment does not follow
			1	0_b	The specified zones are influenced by the JPSEC tool
			1	0_b	Single item is specified
			2	01_b	Range mode
			2	01_b	Izoi uses 16-bit integer
			1	0_b	Izoi is described in one dimension
		$Izoi^{21}$	16	0x31CC	Starting byte location is 12748 (bytes). (B0)
			16	0xA3E8	Ending byte location is 41960 (bytes). (B1)

Table 68 – ZOI example

Parameter			Size (bits)	Value (in order)	Derived meaning
Zone[1]	DC_{ZOI}[1]		1	1_b	The byte aligned segment follows
			1	0_b	Image related description class
			6	001000_b	Resolution is specified
	DCzoi[2]		1	0_b	The byte aligned segment does not follow
			1	1_b	Non-image related description class
			6	010000_b	Byte ranges after the SOD marker are specified
	Pzoi[0,1]	Mzoi[1]	1	0_b	The byte aligned segment does not follow
			1	0_b	The specified zones are influenced by the JPSEC tool
			1	0_b	Single item is specified
			2	10_b	Index mode
			2	00_b	Izoi uses 8-bit integer
			1	0_b	Izoi is described in one dimension
		I_{ZOI}[1]	8	$0000\ 0010_b$	Resolution 2 is specified
	Pzoi[0,2]	Mzoi[2]	1	0_b	The byte aligned segment does not follow
			1	0_b	The specified zones are influenced by the JPSEC tool
			1	0_{b2}	Single item is specified
			2	01_b	Range mode
			2	10_b	Izoi uses 32-bit integer
			1	0_b	Izoi is described in one dimension
		Izoi[2]	32	0xA3EE	Starting byte location is 41966 (bytes). (B2)
			32	0x21101	Ending byte location is 135425 (bytes). (B3)

Table 69 – P_{ID} example

Parameter		Size (bits)	Values	Meaning
T_{ID}		432	See Table 70	Decryption templates
PD		8 (FBAS)	0_b	The byte aligned segment does not follow
			0_b	Pixel domain is not used
			0_b	Wavelet coefficient domain is not used
			0_b	Quantized wavelet coefficient domain is not used
			1_b	Codestream domain is used
			000_b	Reserved for ISO use
F_{PD}		8 (FBAS)	0_b	The FBAS byte does not follow
			1_b	Only packet body is encrypted
			000000_b	Reserved for ISO use
G	PO	16	$000\ 001\ 010\ 011\ 100\ 0_b$	Processing order is tile-resolution-layer-component-precinct
	GL	8	$0000\ 0011_b$	Unit of protection is resolution level
V	N_V	16 (RBAS)	2	Number of values in the value list V is 2
	S_V	8 (RBAS)	16	Length of each Vn is 16 bytes
	V1	128	IV0	Initialization vector value for R1
	V2	128	IV1	Initialization vector value for R2

Table 70 – Decryption template example

Parameter		Size	Value (in order)	Derived meaning
ME_{decry}		8	0	Marker emulation has occurred
CT_{decry}		16	0001_b	Block cipher (AES)
CP_{decry}	M_{bc}	6	100000_b	CBC mode. Bits are not padded
	P_{bc}	2	00_b	Ciphertext stealing
	SIZ_{bc}	8	16	Block size (16 bytes, 128 bits)
	KT_{bc}	392	See Table 71	Key template

Table 71 – Key template example

Parameter		Size (bits)	Value	Derived meaning
LK_{KT}		16	128	Length of key is 128 bits
KID_{KT}		8	2	URI for secret key
G_{KT}	PO	16	$0\ 000\ 001\ 010\ 011\ 100_b$	Processing order is tile-resolution-layer-component-precinct
	GL	8	$0000\ 0011_b$	Unit of protection is resolution level
V_{KT}	N_V	32 (RBAS)	2	Number of values in the value list V is 2
	S_V	8 (RBAS)	19	Length of each Vn is 19 bytes
	V1	152	https://server/key1	Secret key for resolution level 1 can be retrieved from https://server/key1
	V2	152	https://server/key2	Secret key for resolution level 2 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 signalled 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 precincts 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 signalled. 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 supplying 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 identified 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 72 – The SEC marker segment

Parameter			Size (bits)	Value (in order)	Derived meaning
SEC			16	0xFF65	SEC Marker
L_{SEC}			16	0x0099	Length of SEC marker segment
Z_{SEC}			8 (RBAS)	0	Index of this SEC marker segment
P_{SEC}	F_{PSEC}		1	0_b	The FBAS structure does not follow
		F_{INSEC}	1	0_b	INSEC marker segment is not used
		$F_{multiSEC}$	1	0_b	Only one SEC marker segment in this codestream
		F_{mod}	1	0_b	Original JPEG 2000 data was not modified
		F_{TRLCP}	1	0_b	The TRLCP tag is not used
		Padding	3	000_b	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
$Tool^0$	t		8 (FBAS)	0	JPSEC Normative tool
	i		8 (RBAS)	0	Tool instance index
	ID_T		8	2	This normative tool uses an authentication template
	L_{ZOI}		16 (RBAS)	0x20	Length of ZOI is 32 bytes
	ZOI		256	Table 73	The covered zone of the image
	L_{PID}		16 (RBAS)	0x6c	Length of P_{ID} is 108 bytes
	P_{ID}		928	Table 74	Parameters for JPSEC tool

Table 73 – ZOI signalling

Parameter			Size (bits)	Value (in order)	Derived meaning
NZzoi			8 (RBAS)	1	Number of Zones is 1
$Zone^0$	DC_{zoi}^1		1	1_b	The byte aligned segment follows
			1	0_b	Image related description class
			6	001000_b	Resolution levels are specified in order
	DC_{zoi}^2		1	0_b	The byte aligned segment does not follow
			1	1_b	Non-image related description class
			6	010000_b	Byte ranges are specified
	$Pzoi^{0,1}$	$Mzoi^1$	1	0_b	The byte aligned segment does not follow
			1	0_b	The specified zones are influenced by the JPSEC tool
			1	0_b	Single item is specified
			2	01_b	Range mode
			2	00_b	Izoi uses 8-bit integer
			1	0_b	Izoi is described in one dimension
		$Izoi^1$	8	0	The beginning of the range is 0
			8	2	The end of the range is 2

Table 73 – ZOI signalling

Parameter			Size (bits)	Value (in order)	Derived meaning
$Zone^0$	$Pzoi^{0,2}$	$Mzoi^2$	1	0_b	The byte aligned segment does not follow
			1	0_b	The specified zones are influenced by the JPSEC tool
			1	1_b	Multiple items specified
			2	01_b	Range mode
			2	10_b	Izoi uses 32-bit integer
			1	0_b	Izoi is described in one dimension
		N_{ZOI}	8 (RBAS)	3	Number of I_{ZOI} is 3
		$Izoi^1$	32	104	Starting byte location is 104 (bytes)
			32	12762	Ending byte location is 12762 (bytes)
		I_{ZOI}^2	32	12768	Starting byte location is 12768 (bytes)
			32	41980	Ending byte location is 41980 (bytes)
		I_{ZOI}^3	32	41986	Starting byte location is 41986 (bytes)
			32	135445	Ending byte location is 135445 (bytes)

Table 74 – 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_{HMAC}			8	1	HMAC is used for authentication
		H_{HMAC}			8	1	SHA-1 is used for hashing
		KT_{HMAC}	LK_{KT}		16	128	Length of the key in bits
			KID_{KT}		8	3	KI_{KT} contains the URI for the private key
			G_{KT}	PO	16	0 000 001 010 011 100_b	The order is tile-resolution-layer-component-precinct
				GL	8	00000011_b	Key granularity is resolution
			V_{KT}	N_V	16 (RBAS)	3	There are 3 keys in the list
				S_V	8 (RBAS)	8	Size of each key is 8 bytes
				VL	64	*Key*0	The first key is *key*0, for resolution 0
					64	*Key*1	The second key is *key*1, for resolution 1
					64	*Key*2	The third key is *key*2, for resolution 2
		SIZ_{HMAC}			16	20	Size of MAC is 20
PD					8 (FBAS)	0_b	The byte aligned segment does not follow
						0_b	Pixel domain is not used
						0_b	Wavelet coefficient domain is not used
						0_b	Quantized wavelet coefficient domain is not used
						1_b	Codestream domain is used
						000_b	Reserved for ISO use
F_{PD}					8 (FBAS)	0_b	The FBAS byte does not follow
						0_b	Both packet header and body are encrypted
						000000_b	Reserved for ISO use

Table 74 – P_{ID} signalling parameters

Parameter		Size (bits)	Value (in order)	Derived meaning
G	PO	16	00000101001 11000_b	The order is tile-resolution-layer-component-precinct
	GL	8	00000100_b	Tool granularity is layer
V	N_V	32 (RBAS)	9	There are 9 MACs (3 MACs per resolution)
	S_V	8 (RBAS)	20	Size of each MAC is 20 bytes
	VL	160	*M*0	The first MAC is *M*0
		160	*M*1	The second MAC is *M*1
		160	*M*2	The third MAC is *M*2
		160	*M*3	The fourth MAC is *M*3
		160	*M*4	The fifth MAC is *M*4
		160	*M*5	The sixth MAC is *M*5
		160	*M*6	The seventh MAC is *M*6
		160	*M*7	The eighth MAC is *M*7
		160	*M*8	The ninth MAC is *M*8

6.4 Distortion field examples

This subclause 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 6.1.3 to show how distortion values can be associated with the two data segments signalled by the ZOI in that example. As discussed earlier, example 3 in 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 $Pzoi^2$. Therefore the only changes to the ZOI example 3 in 6.1.3 are to set the distortion field bit in DCzoi, and to add $Pzoi^2$ (the final 9 lines in Table 75).

Table 75 – Associating distortion field to two data segments (extension of ZOI example 3 subclause 6.1.3)

Parameter			Size (bits)	Value (in order)	Derived meaning
NZzoi			8 (RBAS)	1	Number of Zones is one
$Zone^0$	DCzoi		1	0_b	The byte aligned segment does not follow
			1	1_b	Non-image related description class
			6	010001_b	Byte ranges after the SOD marker are specified and associated distortion fields are specified
	$Pzoi^2$	$Mzoi^2$	1	0_b	The byte aligned segment does not follow
			1	0_b	The specified zones are influenced by the JPSEC tool
			1	1_b	Multiple items are specified
			2	01_b	Range mode
			2	01_b	Izoi uses 16-bit integer
			1	0_b	Izoi is described in one dimensions
		$Nzoi^2$	8 (RBAS)	2	Number of data segments is 2
		$Izoi^{2,1}$	16	0000 0000 $0000\ 1010_b$	Starting byte location is 10th (bytes)
			16	0000 0000 $0110\ 0100_b$	Ending byte location is 100th (bytes)

Table 75 – Associating distortion field to two data segments (extension of ZOI example 3 subclause 6.1.3)

Parameter			Size (bits)	Value (in order)	Derived meaning
$Zone^0$	$Pzoi^2$	$Izoi^{2,2}$	16	0010 0111 0001 0000_b	Starting byte location is 10000th (bytes)
			16	0010 1110 1110 0000_b	Ending byte location is 12000th (bytes)
	$Pzoi^6$	$Mzoi^6$	1	0_b	The byte aligned segment does not follow
			1	0_b	The specified zones are influenced by the JPSEC tool
			1	1_b	Multiple items are specified
			2	10_b	Index mode
			2	00_b	Izoi uses 8 bits to represent each distortion value
			1	0_b	Izoi is described in one dimensions
		$Nzoi^6$	8 (RBAS)	2	Number of data segments is 2
		$Izoi^{6,1}$	8	D1 value	Distortion value for the first segment
		$Izoi^{6,2}$	8	D2 value	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. $Pzoi^1$ gives the range of packets and $Pzoi^2$ describes the distortion associated with each of these packets. Notice that since $Pzoi^1$ specifies a range of length 4, and $Pzoi^2$ specifies 4 values, each item in the range is associated with one value, e.g., each packet is associated with one distortion.

Table 76 – Signalling a range of packets and associating distortions for each packet

Parameter			Size (bits)	Value (in order)	Derived meaning
NZzoi			8 (RBAS)	1	Number of Zones is one
$Zone^0$	DCzoi		1	0_b	The byte aligned segment does not follow
			1	1_b	Non-image related description class
			6	100001_b	Packets are specified and associated distortion fields are specified
	$Pzoi^1$	$Mzoi^1$	1	0_b	The byte aligned segment does not follow
			1	0_b	The specified zones are influenced by the JPSEC tool
			1	0_b	Single item is specified
			2	01_b	Range mode
			2	00_b	Izoi uses 8-bit integer
			1	0_b	Izoi is described in one dimension
		$Nzoi^1$	8 (RBAS)	1	Number of data segments is 1
		$Izoi^{11}$	8	0000 0000_b	Starting packet is number 0
			8	0000 0011_b	Ending packet is number 3
	$Pzoi^6$	$Mzoi^6$	1	0_b	The byte aligned segment does not follow
			1	0_b	The specified zones are influenced by the JPSEC tool
			1	1_b	Multiple items are specified
			2	10_b	Index mode
			2	00_b	Izoi uses 8 bits to represent each distortion value
			1	0_b	Izoi is described in one dimensions
		$Nzoi^6$	8 (RBAS)	4	Number of data segments is 4
		$Izoi^{6,1}$	8	D1 value	Distortion value for the first packet
		$Izoi^{6,2}$	8	D2 value	Distortion value for the second packet
		$Izoi^{6,3}$	8	D3 value	Distortion value for the third packet
		$Izoi^{6,4}$	8	D4 value	Distortion value for the fourth packet

7 JPSEC registration authority

7.1 General introduction

The JPSEC registration mechanism provides for the unambiguous identification of non-normative security tools that follow the JPSEC standard and that can be further proposed or developed as non-normative JPSEC tools, adding on to the ones listed in Annex B. This registration is performed by a JPSEC Registration Authority. It shall conform to JTC 1 Directives. The registration of these new JPSEC tools is controlled by the process defined in this subclause.

Applicants may submit technologies that they would like included in the JPSEC reference list. Note that the JPSEC tool use is specified with a JPSEC marker present in the codestream. When an application finds an unknown JPSEC ID, it can hook to a JPSEC RA and get the registered information about the tool.

7.2 Criteria for eligibility of applicants for registration

Eligible applicants shall be organizations acknowledged by their National Bodies.

7.3 Applications for registration

Applications for registering new JPSEC tools shall be published by a JPSEC Registration Authority on a website.

The web site shall contain forms for Registration Application, Request for Update, Notification of Assignment or Update, and Rejection of Application.

All forms shall include:

- name of applicant organization;
- address of applicant organization;
- the name, title, postal/e-mail address, telephone/facsimile number of a contact person within the organization.

Forms for Registration Application and Request for Update shall also include the following entries:

- JPSEC tool name (mandatory).
- Type of JPSEC tool, e.g., digital signature, watermarking, encryption, scrambling, key generation and management, authentication (optional).
- Descriptive technical abstract (mandatory).
- Descriptive overview of the tool (mandatory).
- Operational example use case description (optional).
- Parameters syntax specification, including possible values (optional).
- Guidelines for optimum usage (optional).
- IPR status, e.g., owner, right holder (optional).
- IPR conditions for use (mandatory).
- Restrictions of use, e.g., export conditions (optional).
- Information for downloads of implementations (optional).
- Additional comments, motivation, references, etc. (optional).
- Requirements for confidentiality of selected application entries (optional).
- Requested length of time for tool registration (optional).

The JPSEC Registration Authority shall also provide tutorial material to assist applicants in preparing applications.

7.4 Review and response to applications

This subclause defines the process for the JPSEC Registration Authority to review and respond to applications to ensure fairness.

A technical review committee is set up to review the applications. This committee is composed of ISO/IEC JTC 1/SC 29/WG 1 members and JPSEC Registration Authority members. The review committee examines applications at a WG 1 meeting not later than nine months after the application was submitted.

The review committee accepts or rejects the application, based on the rejection criteria in 7.5.

If accepted, the new JPSEC tool is allocated an identifier (ID) for a specified period of time. The ID syntax shall conform to 5.6.3. The review committee approves the JPSEC tool description information listed in 7.3. The ID shall then be used for signalling in the JPSEC codestream.

Once the application has been reviewed and accepted, the JPSEC RA notifies the applicant of a positive or negative response to the registration request. The response to the applicant shall include a short explanation of the results of the technical review and shall be sent back to applicants no later than nine months after the date of application.

A negative response may be appealed if the registrant believes that there was an error 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 broader WG 1 committee at the next appropriate WG 1 group meeting. He may then be required to provide additional information at the request of the experts, who, under the authority of WG 1, will provide a final, definitive response of acceptance or rejection. In order to have a rejected application reviewed by WG 1, the registrants must re-submit the proposal through their National Body, specifying why the submission requires consideration by WG 1.

7.5 Rejection of applications

The criteria for rejection of an application are the following:

- The applicant is not eligible.
- The proper fees are not paid (when relevant).
- An approved, registered item already exists that contains the identical contents of the submission.
- The information in the application is incomplete or incomprehensible.
- The justification for inclusion in the register is not adequate. The candidate JPSEC tool should demonstrate it provides a useful security service and give examples of use cases when relevant.
- The Authority considers that there is not enough originality in the proposed tool which could easily be implemented with an existing, approved item.
- The submission contains errors or is not compliant with the normative JPSEC specifications or standard;
- The technical description is not sufficient.
- The confidentiality conditions are not appropriate.

7.6 Assignment of identifiers and recording of object definitions

The review process and above syntax ensure that the assigned ID is unique within the register and that the same ID is not assigned to another object.

After the assignment has been made, the ID and associated information shall be included in the register and the JPSEC Registration Authority shall inform the applicant of the assignment within nine months.

The JPSEC tool definition shall be recorded in the register at the time when the ID is assigned.

7.6.1 Reuse of IDs

Identifiers may be reused by a Registration Authority. For example identifiers become available for reuse after their expiration or when they are given up voluntarily or reclaimed.

ID owners may voluntarily give up their ID through a Request for Update.

7.6.2 Reclamation

A JPSEC Registration Authority may reclaim an identifier for technical reasons or for tool misuse. When this occurs, Identifier owners will be notified by a Notification of Update.

7.7 Maintenance

For the purpose of maintenance of the register, a JPSEC Registration Authority shall implement mechanisms for maintaining the integrity of register including adequate backup for retaining records.

An ID owner may update the associated JPSEC tool information through a Request for Update.

A JPSEC Registration Authority shall provide mechanisms for maintaining confidentiality of entries as granted in the application.

7.8 Publication of the register

Generally, the interests of the community of information technology users is best served if the register information is made public. In certain cases, however, there may be a need for confidentiality of some or all of the data pertinent to a particular registration, either permanently or for some portion of the registration process.

A JPSEC Registration Authority shall publish the registry information in a manner that is consistent with the confidentiality requirements of the JPSEC tool.

Where publication is required, electronic and printed paper versions are mandatory. If a JPSEC Registration Authority is to provide publication, it shall keep accurate distribution records pertaining to its publications.

7.9 Register information requirements

The JPSEC Registration Authority shall electronically publish the list of non-normative JPSEC tools in its register, as well as the information associated to them, in a manner that is consistent with the confidentiality requirements of the JPSEC tool.

The following information shall be contained in the register for each JPSEC tool:

- the assigned ID;
- name of initial applicant;
- address of initial applicant;
- date of original assignment;
- date of last transfer of assignment, if allowed (updatable);
- name of current owner (updatable);
- address of current owner (updatable);
- the name, title, postal/e-mail address, telephone/facsimile number of a contact person within the organization (updatable);
- date of last update (updatable).

It shall also contain the information provided by the applicant on its JPSEC tool as specified in 7.3 as well as the approval information.

Annex A

Guidelines and use cases

(This annex forms an integral part of this Recommendation | International Standard)

A.1 A class of JPSEC applications

A.1.1 Introduction

This subclause 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.1.2 Overview of a secure JPEG 2000 image distribution

Figure A.1 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 image source and content.

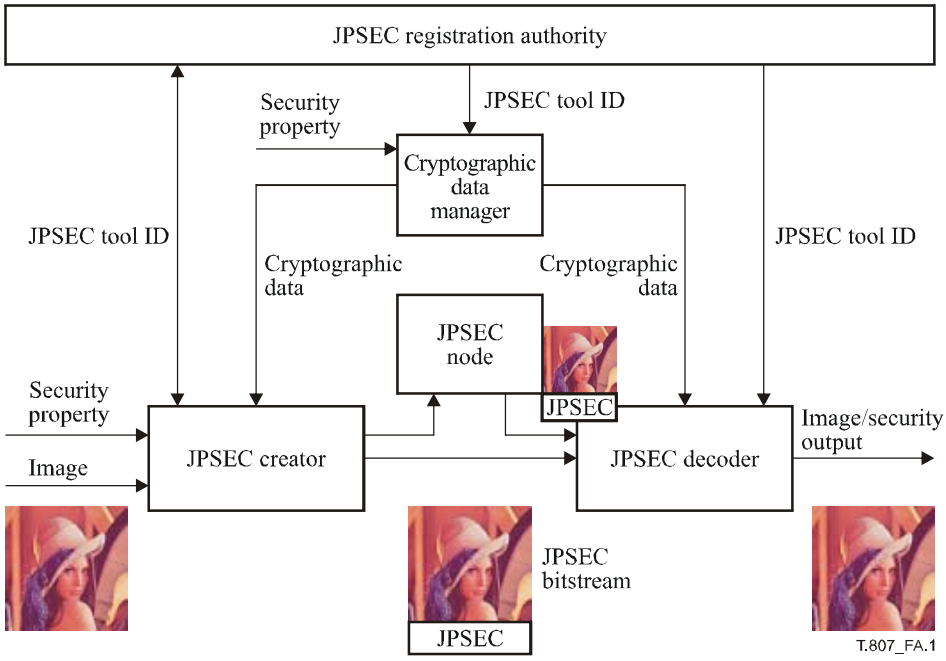


Figure A.1 – Overview of a secure JPEG 2000 image distribution application

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

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

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

Step 3: The JPSEC codestream is received and rendered by a JPSEC consumer.

Step 1: JPSEC codestream creation

The creator is in charge of creating the secure JPEG 2000 codestream. This codestream 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 codestream delivery

A JPSEC codestream 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 codestream.

When required by the JPSEC security tool methods in the Security Property Parameter of the JPSEC codestream (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 codestream consumption rendering

A JPSEC codestream 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.2 shows the legend description to be used.

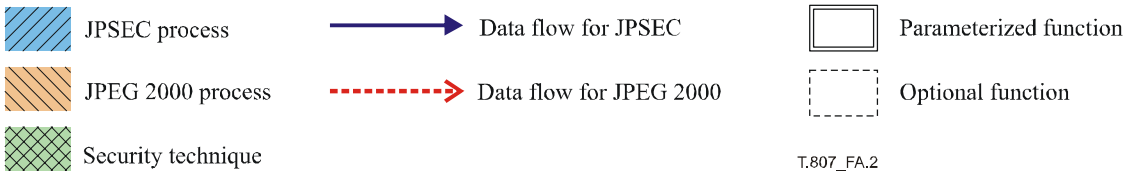


Figure A.2 – Legend description

- **JPSEC process**: A process that uses tools defined in this Recommendation | International Standard.
- **JPEG 2000 process**: A process defined in ITU-T Rec. T.800 | ISO/IEC 15444-1 (JPEG 2000 Part 1).
- **Security technique**: A well-known security technique either defined in this Recommendation | International Standard or in some other standard or document.
- **Data flow for JPSEC**: A data flow that communicates information defined in this Recommendation | International Standard. A dashed line indicates optional.
- **Data flow for JPEG 2000**: A data flow defined in ITU-T Rec. T.800 | ISO/IEC 15444-1 (JPEG 2000 Part 1).
- **Parameterized 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.1.3 Encryption end description procedure

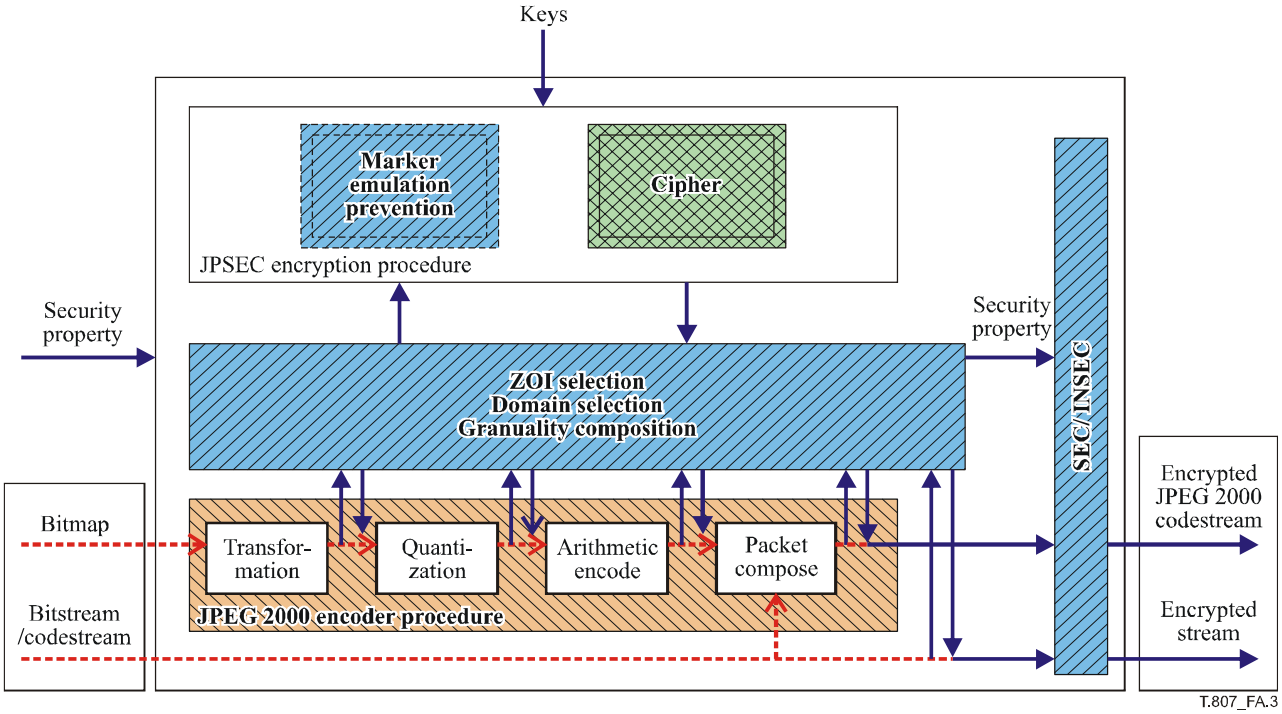


Figure A.3 – Encryption procedure

Figure A.3 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 plaintext data with encrypted data;
- (optionally) apply a marker emulation prevention mechanism;
- compose the Security Parameter Property in the SEC and/or INSEC marker segment.

Note that in general the JPSEC encryption procedure generates JPSEC codestream that is not backward compatible with JPEG 2000 Part 1. The image data is intended to be passed on to a Part 1 compliant decoder after appropriate decryption. It is possible to apply a marker emulation prevention mechanism to avoid marker segment emulation within the encrypted codestream.

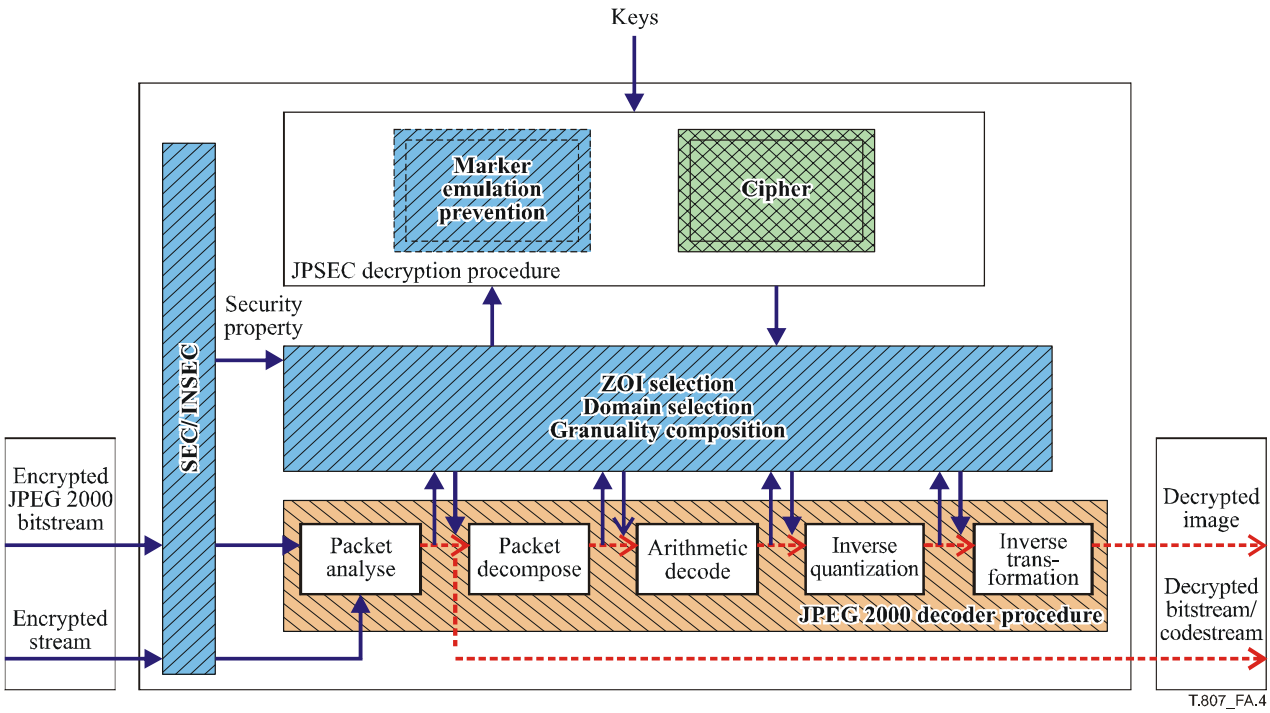


Figure A.4 – Decryption procedure

Figure A.4 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 signalled 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 signalled 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.1.4 Signature generation and authentication procedure

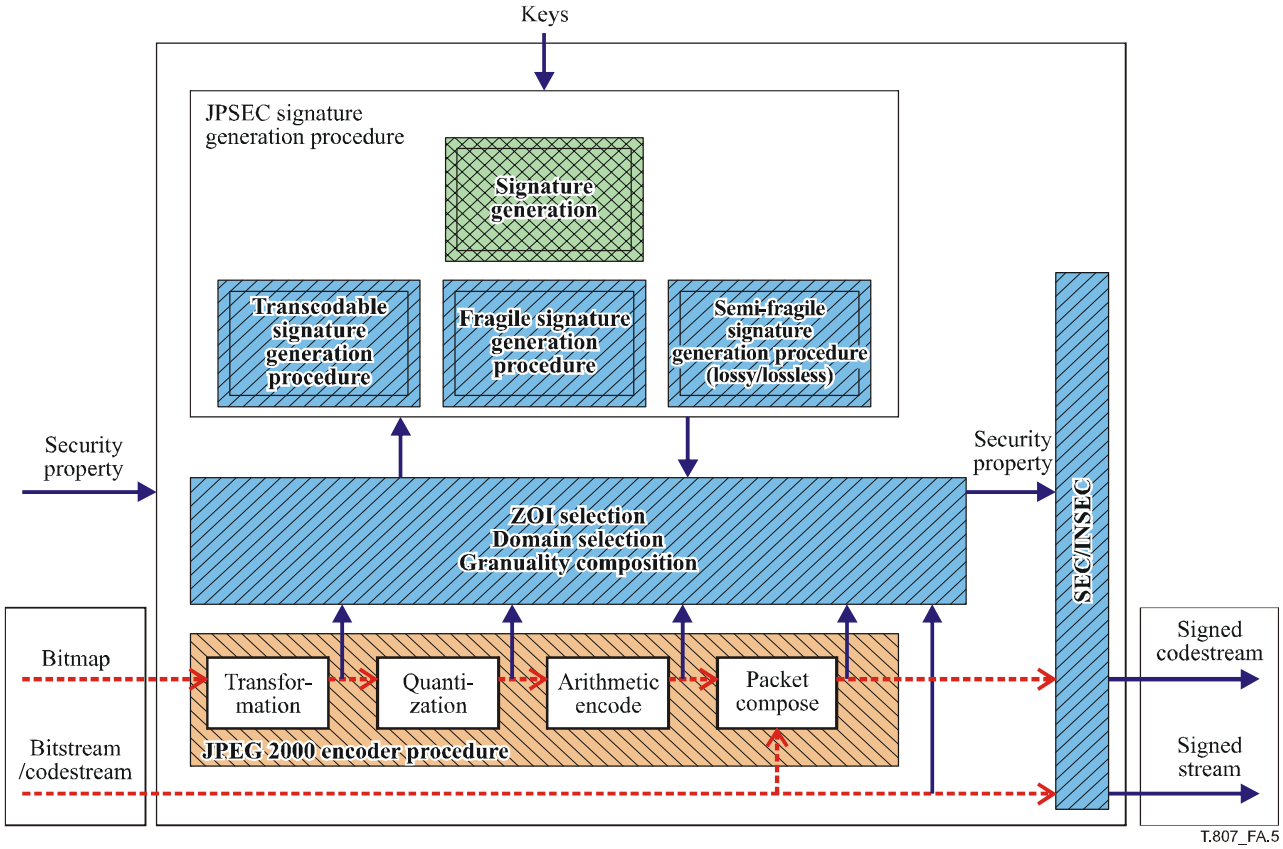


Figure A.5 – Signature generation procedure

Figure A.5 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 predetermined extent. Further, a "transcoding mode" authentication can verify the source part of the codestream.

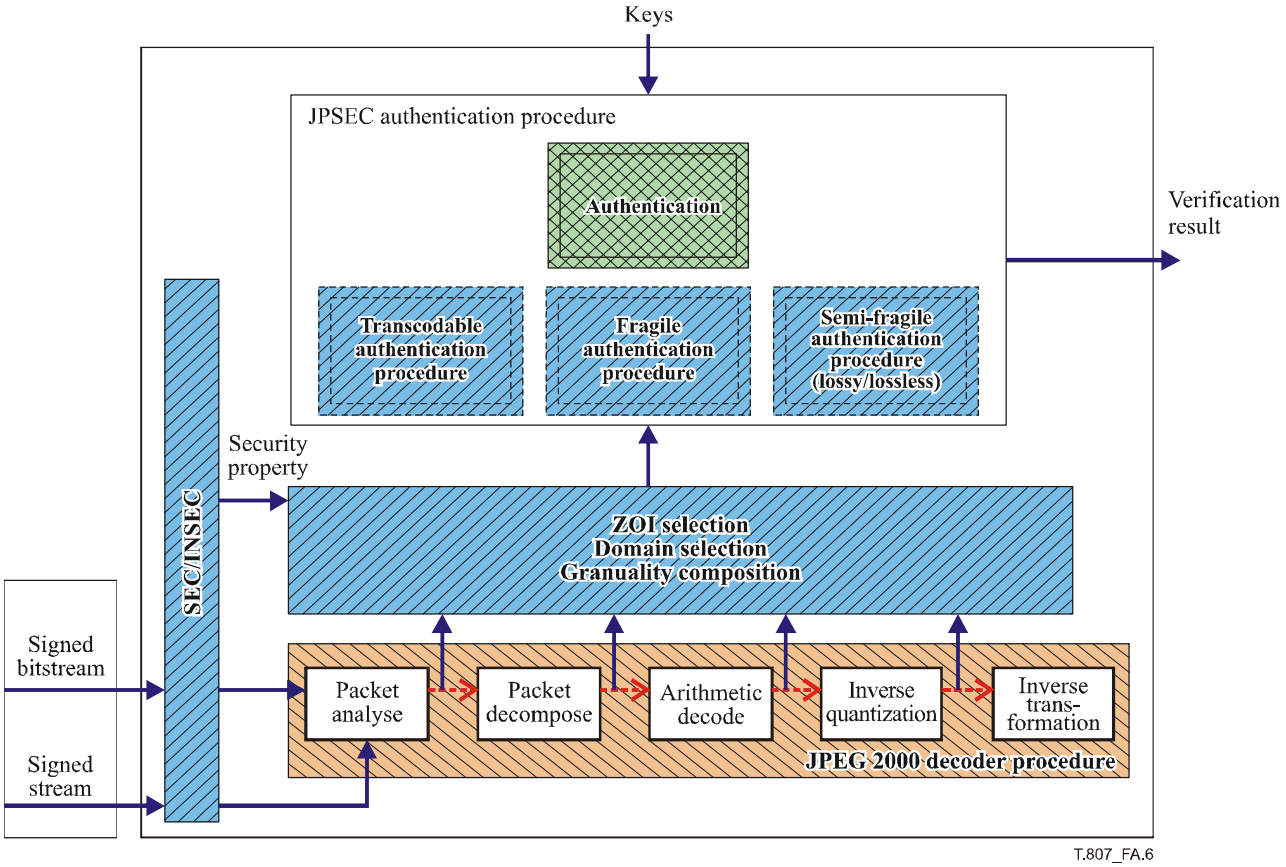


Figure A.6 – Authentication procedure

Figure A.6 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 technique. Further, it is possible to verify the selected data in a unit based on the Granularity.

A.1.5 ICV (Integrity Check Value) generation and integrity check procedure

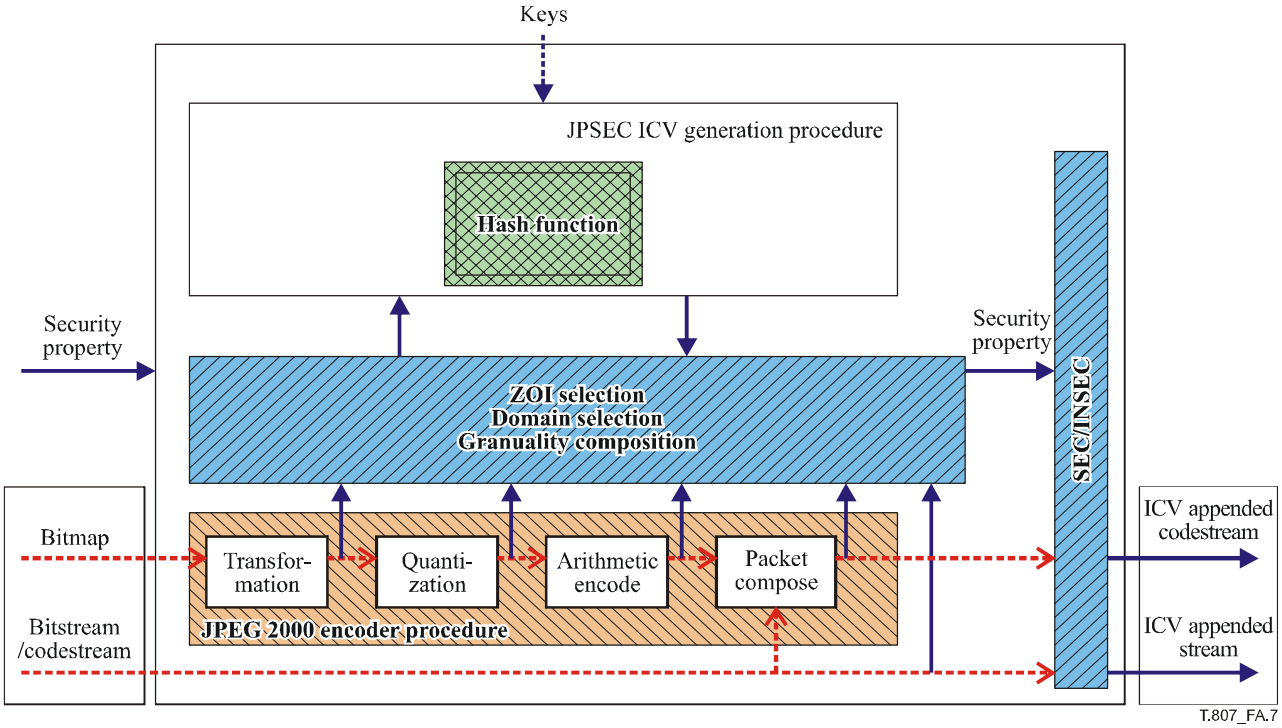


Figure A.7 – ICV (Integrity Check Value) generation procedure

Figure A.7 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.

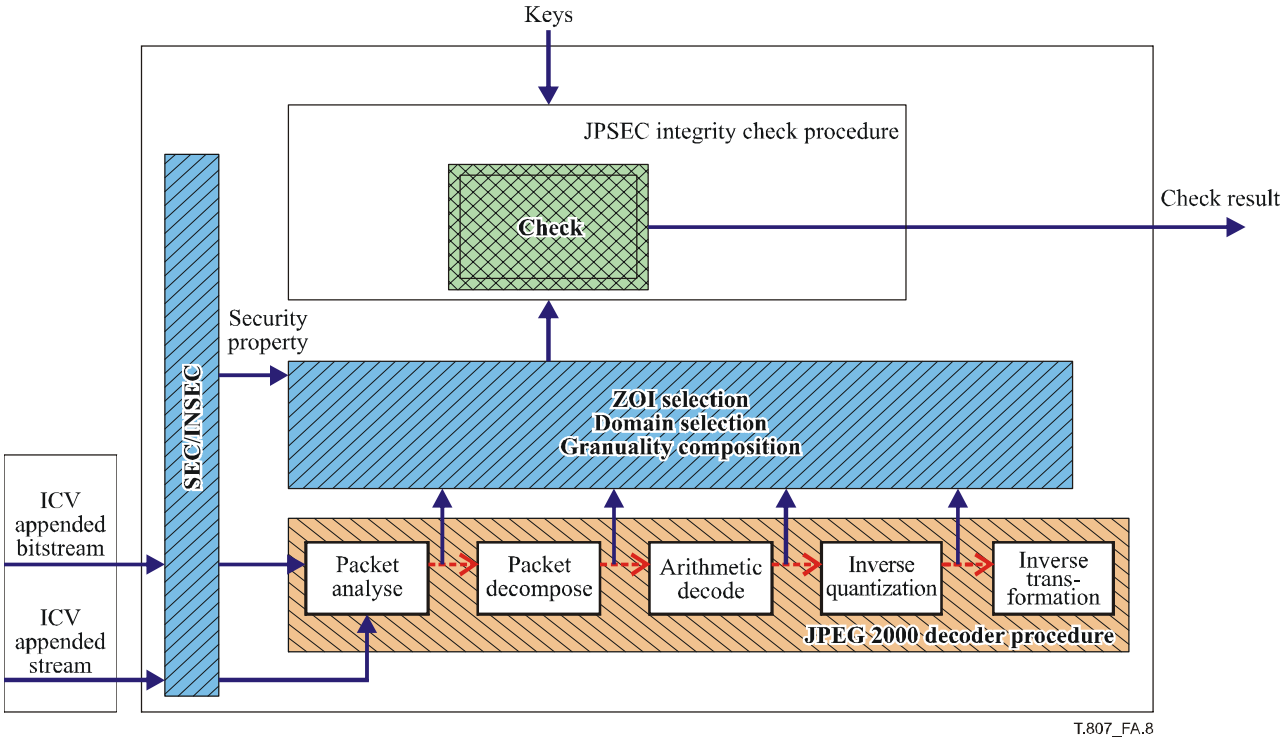


Figure A.8 – Integrity check procedure

Figure A.8 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.

Annex B

Technology examples

(This annex forms an integral part of this Recommendation | International Standard)

B.1 Introduction

The JPSEC syntax allows normative and non-normative security tools to be applied to JPEG 2000 images. This subclause 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;
- A unified authentication framework for JPEG 2000 images;
- A simple packet-based encryption method for JPEG 2000 codestreams;
- 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 codestreams;
- JPEG 2000 data confidentiality and access control system based on data splitting and luring;
- Secure scalable streaming and secure transcoding.

B.2 A flexible access control scheme for JPEG 2000 codestreams

B.2.1 Security service

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

B.2.2 Typical application

It provides protection of content delivery via variable media, e.g., Internet, digital 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.2.3 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 codestream sends his/her request to the key server. The key server, in turn, responds with appropriate decryption keys according to the user's privilege. The user can access the allowed sub-images.

B.2.4 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 $k^t = h(K \,|''T''|t)$, for each tile $t = 0, 1, \ldots, n_T - 1$, 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 $r = n_R - 2, \ldots, 1, 0$, where $k^{n_R - 1} = h(k^t \,|''R'')$ and "*R*" denotes the ASCII code of the letter *R*.

3) Compute key $k^{rl} = h(k^{r(l+1)})$, for each $r = n_R - 1, \ldots, 1, 0$, $l = n_L - 2, \ldots, 1, 0$, where $k^{r(n_L-1)} = h(k^r|"L")$ and where "*L*" denotes the ASCII code of the letter *L*.

4) Calculate key $k^{rlc} = h(k^{rl}|"C"|c)$, for each $r = n_R - 1, \ldots, 1, 0$, $l = n_L - 1, \ldots, 1, 0$, $c = 0, 1, \ldots, n_C - 1$, where "*C*" denotes the ASCII code of the letter *C* and *c* denotes the index of this component.

5) Produce keys $k^{rlcp} = h(k^{rlc}|"P"|p)$, for each $r = n_R - 1, \ldots, 1, 0$, $l = n_L - 1, \ldots, 1, 0$, $c = 0, 1, \ldots, n_C - 1$, $p = 0, 1, \ldots, n_R - 1$, 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 corresponding 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.2.5 Codestream 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 this 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 – Example parameters for this scheme

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

Parameter		Size (bits)	Values	Meaning
t		8 (FBAS)	1	Registration authority protection tool
i		8 (RBAS)	*Instance value*	Tool instance identifier
ID_{RA}	$ID_{RA,id}$	32	*Tool ID value*	Registered ID
	$ID_{RA,nsl}$	8 (RBAS)	21	The length of $ID_{RA,ns}$ in bytes
	$ID_{RA,ns}$	168	*Namespace*	The namespace of the RA with which this tool is registered
L_{ZOI}		16 (RBAS)	$[2 \ldots 2^{16} - 1]$	Length of ZOI
ZOI		Variable	See 5.7	Zone of influence for this scheme
L_{PID}		16 (RBAS)	$[2 \ldots 2^{16} - 1]$	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_{decry}	TP_{ID}

Parameter	Size (bits)	Values	Meaning
$ID_T = 1$	8	Always set to 1	Tag for Decryption template
T_{decry}	Variable	Decryption template values	Decryption template
TP_{ID}	Variable	See Table B.3	Additional information for 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 codestream progression 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} field is presented
AK_{Info}	Variable	See Table B.4	Information on the 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 24	Cipher used to encrypt the access keys
N_{ak}	16	0 ... (2^{16} – 1)	Number of access keys
AK	N_{ak} * K_{bc}	NaN	Access keys

B.2.6 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*".

B.3 A unified authentication framework for JPEG 2000 images

B.3.1 Operational description

This JPSEC tool provides the following JPSEC services: image data/content integrity 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 than 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.3.2 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.

Figure B.1 illustrates how the tool can be used to protect images.

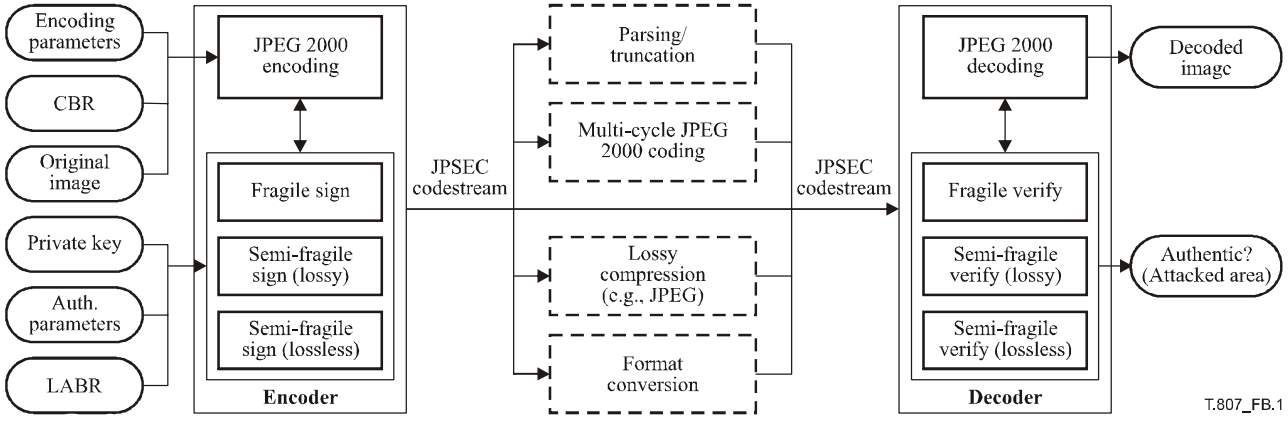


Figure B.1 – Image protection using unified authentication framework for JPEG 2000

This tool can use different signalling syntaxes depending on the chosen authentication method. For fragile authentication, it uses the JPSEC normative tool syntax, as defined in 5.8.3. For semi-fragile authentication, it uses the JPSEC non-normative tool syntax, as illustrated in Table B.5. In addition, the F_{INSEC} should be set to 0 as INSEC marker is not used by this tool, and F_{mod} should be set to 1 because the resulted codestream 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				8 (FBAS)	1	Non-normative tool syntax is used
i				8 (RBAS)	0 ... (2^7 – 1)	Tool instance index
ID_{RA}	$ID_{RA,id}$			32	0 ... (2^{32} – 1)	ID number to be assigned by RA
	$ID_{RA,nsl}$			8 (RBAS)	21	The length of $ID_{RA,ns}$ in bytes
	$ID_{RA,ns}$			168	*namespace*	The namespace of the RA with which this tool is registered
L_{ZOI}				16 (RBAS)	0 ... (2^{16} – 1)	Length of ZOI
ZOI				Variable	*ZOI values*	The covered zone in the image protected by the tool
L_{PID}				16 (RBAS)	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 21
	T_{auth}	M_{auth}		8	2	Digital Signature Method is used, as defined in Table 34
		P_{auth}	M_{DS}	8	See Table 41	Digital signature algorithm used, such as DSA or RSA
			H_{DS}	8	See Table 37	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

Table B.5 – Syntax for semi-fragile authentication

Parameter			Size (bits)	Value	Derived meaning
P_{ID}	PD		1	0_b	The FBAS structure is terminated
			1	0_b	Pixel domain is not used
			1	0_b	Wavelet coefficient domain is not used
			1	1_b	Quantized wavelet coefficients domain is used
			1	0_b	Codestream domain is not used
			3	000_b	Reserved for ISO use
	G	PO	16	*Processing order values*	Processing order
		GL	8	0000 1001	Granularity level: Unit of protection is total area identified in ZOI
	V	N_V	16	1	Number of digital signatures in the list is 1
		S_V	8 (RBAS)	1 ... $(2^8 – 1)$	Size of the digital signature in bytes
		VL	8* 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.3.3 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.

B.4 A simple packet-based encryption method for JPEG 2000 codestreams

B.4.1 Operational description

This subclause 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 confidentiality 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 functionality 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 has 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.4.2 Technical overview

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

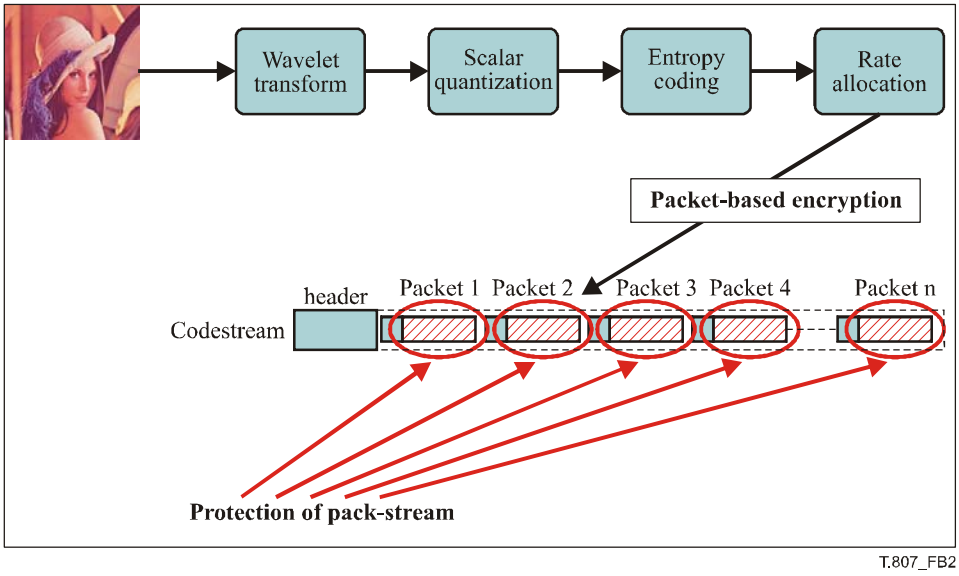


Figure B.2 – Packet-based encryption principle

This JPSEC tool can take several image-related parameters as an input: resolution levels, quality layers, components, precincts or tiles. Only the packet payloads corresponding 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 consumer 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 [22] 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.4.2.1 Signalling example

The technique can be signalled with the template based syntax of the normative clause. Below is an example of signalling for this technique (see Table B.6), which specifies one zone for the ZOI, but of course there could be more, following the same syntax as Zone^0.

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

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

Table B.7 – Decryption template description, in the case of AES-192/CBC

Parameter					Size (bits)	Value	Derived meaning
P_{PM}	ME_{decry}				8	0000 0000	NULL: no marker emulation prevention method
	CT_{decry}				16	0x0003	Cipher identifier: AES (block cipher)
	CP_{decry}	M_{bc}			6	10 0000	Cipher mode: CBC
		P_{bc}			2	01	Padding mode (PKCS#7-padding)
		SIZ_{bs}			8	0001 0000	Size of block: 16 bytes (128 bits)
		KT_{bc}	LK_{KT}		16	0x00C0	Size of key: 192 bits
			KID_{KT}		8	0000 0011	Key information is a URI
			LKI_{KT}		16	0x0021 (=33)	Length of the URI: 33 bytes
			KI_{KT}		264	https://server/ path/secretkey .pem	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	16	0 000 001 010 011 100	Processing order is TRLCP
				GV	8	0000 1001	Granularity of key is total area in ZOI
			V_{KT}	Nv	16	0x0001	Single key value in KI_{KT}; Values not specified in V_{KT}
				Sv	16	0010 0001	Length of the URI: 33 bytes
				VL	264	https://server/ path/secretkey .pem	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.

Table B.8 – Processing domain syntax

Parameter	Size (bits)	Value	Derived meaning
PD	1	0_b	The byte aligned segment does not follow
	1	0_b	Not in pixel domain
	1	0_b	Not in wavelet coefficient domain
	1	0_b	Not in quantized wavelet coefficient domain
	1	1_b	Processed in codestream domain
	3	000_b	Not used

Table B.9 – Granularity and value list syntax

Parameter		Size (bits)	Value	Derived meaning
G	PO	16	0 000 001 010 011 100	Processing order is TRLCP
	GV	8	0000 0110	Unit of protection is packet
V	N_V	16	1	Number of IV values specified
	Sv	8	16	IV size in bytes
	VL	128	*Value*	IV value

B.4.3 Conclusion

The technique presented in this subclause 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 5.8 and supports various levels of complexity.

B.5 Encryption tool for JPEG 2000 access control

B.5.1 Security services addressed

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

B.5.2 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 thumbnail, a low quality image, and a partially scrambled image.

B.5.3 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.5.4 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, then 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 0xFF90 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.

If the length of plaintext is odd, an exception to the processing is necessary; the last byte is not encrypted or padded with one extra byte.

(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.5.5 Signalling method

Table B.10 shows example parameters for 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 – Example parameters for this technology

Parameter		Size (bits)	Value	Meaning
SEC		16	0xFF65	SEC marker
L_{SEC}		16	Variable	Length of SEC marker segment
Z_{SEC}		8	1 (example)	Index of this SEC marker segment
P_{SEC}		1	0	FBAS byte does not follow
	F_{INSEC}	1	1 (example)	INSEC is used
	$F_{multiSEC}$	1	0_b	One SEC marker segment is used
	F_{mod}	1	1_b	Original JPEG 2000 data was modified
	F_{TRLCP}	1	0_b	TRLCP tag usage is not defined
	Padding	3	000_b	Not used
	N_{tools}	8 (RBAS)	1	Number of security tool is one
	I_{max}	8 (RBAS)	0	Maximum tool instance index is zero
t		8 (FBAS)	1	RA protection JPSEC non-normative tool
i		8 (RBAS)	0000000_b	Tool instance index
ID_{RA}	$ID_{RA,id}$	32	0	Registered ID
	$ID_{RA,nsl}$	8 (RBAS)	21	The length of $ID_{RA,ns}$ in bytes
	$ID_{RA,ns}$	168	*namespace*	The name space of the RA with which this tool is registered
L_{zoi}		16	9	Length of ZOI is 9 bytes
ZOI		Variable	See Table B.11 (example)	Zone of Influence for this tool
L_{PID}		16	Variable	Length of L + T + PD + G
P_{ID}		Variable	See Table B.12 (example)	Parameters for this technology

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

Parameter			Size (bits)	Value (in order)	Derived meaning
NDzoi			8	1	Number of Zones is one
$Zone^0$	Dczoi		1	0_b	The byte aligned segment does not follow
			1	0_b	Image related description class
			6	101000_b	Image regions and resolution levels are specified in order
	$Pzoi^1$	$Mzoi^1$	1	0_b	The byte aligned segment does not follow
			1	0_b	The specified zones are influenced by the protection method
			1	0_b	Single item is specified
			2	00_b	Rectangle mode
			2	00_b	Izoi uses 8-bit integer
			1	1_b	Izoi is described in two dimensions
		$Izoi^1$	8	$0110\ 0100_b$	Xul is 100
			8	$0111\ 1000_b$	Yul is 120
			8	$1011\ 0100_b$	Xlr is 180
			8	$1101\ 0010_b$	Ylr is 210
	$Pzoi^3$	$Mzoi^3$	1	0_b	The byte aligned segment does not follow
			1	1_b	The specified zones are not influenced by the protection method
			1	0_b	Single item is specified
			2	11_b	Max mode
			2	00_b	Izoi uses 8-bit integer
			1	0_b	Izoi is described in one dimension
		$Izoi^3$	8	$0000\ 0010_b$	Resolution levels > 3 are specified

Table B.12 – P_{ID} for this technology

Parameter		Size (bits)	Value	Meaning
T		Variable	See Table B.13	Decryption templates
PD		8	$0000\ 1000_b$	FBAS byte does not follow. Processed in codestream domain.
G	PO	16	$0\ 000\ 001\ 010\ 011\ 100\ 0_b$	Processing order is tile-resolution-layer-component-precinct.
	GL	8	$0000\ 0110_b$	Unit of protection is packet
Skip		8	0	*Skip* parameter for this tool

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

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

B.5.6 Conclusion

This subclause described an encryption technology for a JPEG 2000 codestream. The significant advantage of this technology is to prevent marker emulation from occurring in the encrypted codestream.

B.6 Key generation tool for JPEG 2000 access control

B.6.1 Security services addressed

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

B.6.2 Typical applications

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

B.6.3 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.6.4 Technical overview

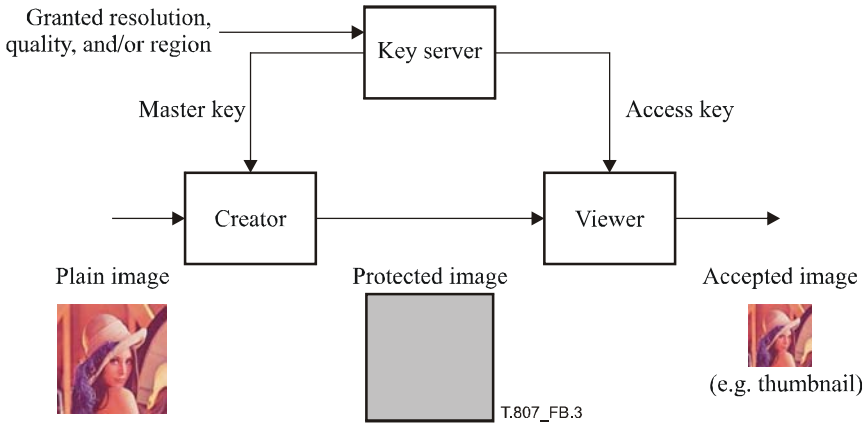


Figure B.3 – Overview of this technology

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 region. 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 cannot 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.6.5 Signalling 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 the 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	0	Index of this SEC marker segment
P_{SEC}		1	0	FBAS byte does not follow
	F_{INSEC}	1	1	INSEC is used
	$F_{multiSEC}$	1	0_b	One SEC marker segment is used
	F_{mod}	1	1_b	Original JPEG 2000 data was modified
	F_{TRLCP}	1	0_b	TRLCP tag usage is not defined
	Padding	3	000_b	Not used
	N_{tools}	8 (RBAS)	1	Number of security tool is one
	I_{max}	8 (RBAS)	0	Maximum tool instance index is zero
t		8 (RBAS)	1	JPSEC non-normative tool
i		8 (RBAS)	0	Instance index for this tool

Table B.14 – Recommended parameter in this technology

Parameter		Size (bits)	Values	Meaning
ID_{RA}	$ID_{RA,id}$	32	5	Registered ID for this tool
	$ID_{RA,nsl}$	8 (RBAS)	21	The length of $ID_{RA,ns}$ in bytes
	$ID_{RA,ns}$	168	*namespace*	The namespace of the RA with which this tool is registered
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			Size (bits)	Value (in order)	Derived meaning
NDzoi			8	1	Number of Zones is one
$Zone^0$	Dczoi		1	0_b	The byte aligned segment does not follow
			1	0_b	Image related description class
			6	101000_b	Image regions and resolution levels are specified in order
	$Pzoi^1$	$Mzoi^1$	1	0_b	The byte aligned segment does not follow
			1	0_b	The specified zones are influenced by the protection method
			1	0_b	Single item is specified
			2	00_b	Rectangle mode
			2	00_b	Izoi uses 8-bit integer
			1	1_b	Izoi is described in two dimensions
		$Izoi^1$	8	$0110\ 0100_b$	Xul is 100
			8	$0111\ 1000_b$	Yul is 120
			8	$1011\ 0100_b$	Xlr is 180
			8	$1101\ 0010_b$	Ylr is 210
	$Pzoi^3$	$Mzoi^3$	1	0_b	The byte aligned segment does not follow
			1	1_b	The specified zones are not influenced by the protection method
			1	0_b	Single item is specified
			2	11_b	Max mode
			2	00_b	Izoi uses 8-bit integer
			1	0_b	Izoi is described in one dimension
		$Izoi^3$	8	$0000\ 0010_b$	Resolution levels > 3 are specified

Table B.16 – P_{ID} for this technology

Parameter		Size (bits)	Values	Meaning
T		Variable	See Table B.17	Decryption templates
PD		8	$0000\ 1000_b$	FBAS byte does not follow. Processed in codestream domain.
G	PO	16	$0\ 000\ 001\ 010\ 011\ 100_b$	Processing order is tile-resolution-layer-component-precinct
	GL	8	$0000\ 0110_b$	Unit of protection is packet
H		16	See Table 37 in 5.8.3.1	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_{bc} in T)

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

Parameter		Size (bits)	Value (in order)	Derived meaning
ME_{decry}		8	1	Marker emulation has not occurred
CT_{decry}		16	3	Block cipher (AES)
CP_{decry}	M_{bc}	6	10 0010	OFB mode is used. (Bits are not padded.)
	SIZ_{bc}	16	128	Block size (128 bits)
	KT_{bc}	Variable	See 5.8.5	Key template
	IVsc	128	*Initial vector value*	Initial vector value

B.6.6 Conclusion

This subclause 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.

B.7 Wavelet and bitstream domain scrambling for conditional access control

B.7.1 Summary

Access control to an image is an important functionality 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 subclause, a technique for conditional access control is presented. The method was initially presented in [23]. Basically, it adds pseudo-random noise to the image. Authorized users know the pseudo-random sequence and thus can remove this noise. On the other hand, unauthorized users only have access to severely 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 codestream. 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.7.2 Technical overview

The system is composed of three main components:

- Scrambling: two approaches are supported. The scrambling is either performed on the quantized wavelet coefficients, or directly on the bits in the codestream. 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 codestream 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 algorithm [24] 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 codestream. In a preferred embodiment of the technique, the RSA algorithm is used for encryption [25]. Other encryption algorithms could be used as well. The length of the key can be selected at the time the image is protected.

Figures B.4 and B.5 correspond to the two cases of wavelet and bitstream domain scrambling.

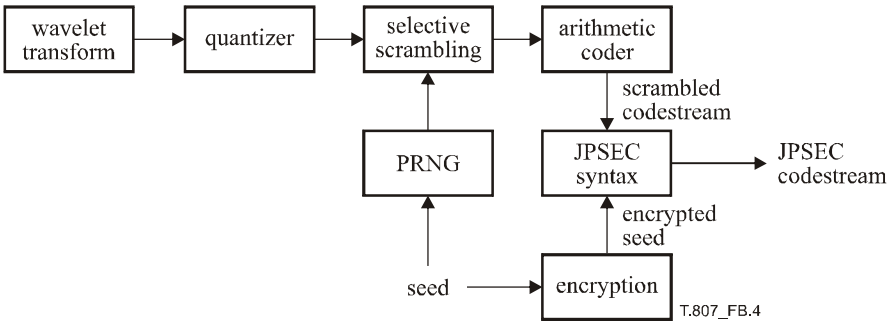


Figure B.4 – Block diagram for wavelet domain scrambling

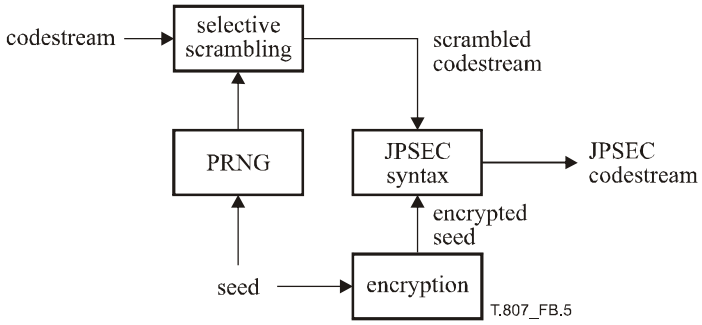


Figure B.5 – Block diagram for bitstream domain scrambling

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. The syntax given below is very flexible and supports the usage of multiple seeds and multiple keys.

B.7.3 Codestream syntax

In this example, both the SEC and INSEC marker segments are used. The codestream syntax is defined below. The SEC marker segment is using the tool syntax for non-normative tools. The INSEC marker segment is used to signal which codeblocks are scrambled and which seeds are used.

B.7.3.1 Syntax for SEC marker segment

The tool syntax for non-normative tools is used. 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 corresponding to a different key identification $KeyID^{(i)}$. This is illustrated below. See Figure B.6.

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 semantic 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.7.3.2 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 added 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:

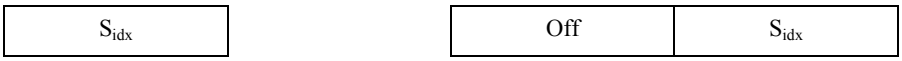


Figure B.7 – Syntax for AP: Wavelet domain scrambling (left), Bitstream domain scrambling (right)

With the following semantic:

Table B.19 – Syntax and semantic for AP

Parameter	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.7.4 Conclusions

In this subclause, 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 codestream. Consequently, 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 pseudo-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.

B.8 Progressive access for JPEG 2000 codestream

B.8.1 Security services addressed

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

B.8.2 Typical applications

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

B.8.3 Potential users, implementation model and motivations

The challenge in the design of the access control scheme is to strike a delicate 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 codestream. Therefore, only users with the right security clearance can decrypt the packets corresponding to the granted image in the codestream.

B.8.4 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.

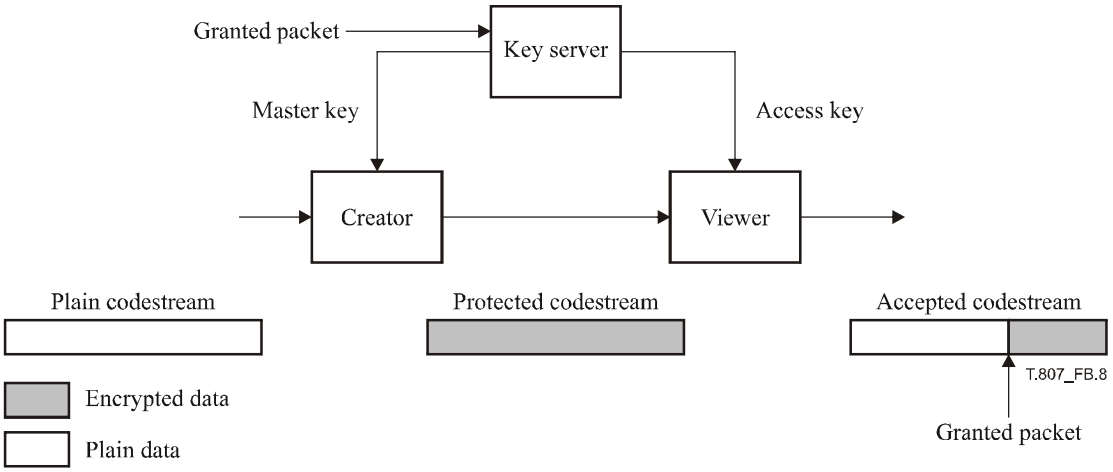


Figure B.8 – Technical overview of this technology

B.8.5 Signalling method

Table B.20 shows recommended parameters in this technology. Any parameters 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.20 – Example parameters for this tool

Parameter		Size (bits)	Values	Meaning
SEC		16	0xFF65	SEC marker
L_{SEC}		16	Variable 0 ... 255	Length of SEC marker segment
Z_{SEC}		8	0	Index of this SEC marker segment
P_{SEC}		1	0	FBAS byte does not follow
	F_{INSEC}	1	1_b	INSEC is used
	$F_{multiSEC}$	1	0_b	One SEC marker segment is used
	F_{mod}	1	1_b	Original JPEG 2000 data was modified
	F_{TRLCP}	1	0_b	TRLCP tag usage is not defined
	Padding	3	000_b	Not used
	N_{tools}	8 (RBAS)	1	Number of security tools is one
	I_{max}	8 (RBAS)	0	Maximum tool instance index is zero

Table B.20 – Example parameters for this tool

Parameter		Size (bits)	Values	Meaning
t		8 (RBAS)	1	RA protection tool
i		8 (RBAS)	0	Instance index
ID_{RA}	$ID_{RA,id}$	32	7	Registered ID
	$ID_{RA,nsl}$	8 (RBAS)	21	The length of $ID_{RA,ns}$ in bytes
	$ID_{RA,ns}$	168	*namespace*	The namespace of the RA with which this tool is registered
L_{zoi}		16 (RBAS)	Variable	Length of ZOI
ZOI		Variable	See Table B.21 (example)	Zone of Influence for this tool
L_{PID}		16 (RBAS)	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			Size (bits)	Value (in order)	Derived meaning
NDzoi			8	1	Number of Zones is one
$Zone^0$	DCzoi		1	0	The byte aligned segment does not follow
			1	1	Non-image related description class
			6	000100	Packets are specified
	$Pzoi^4$	$Mzoi^4$	0	1	The byte aligned segment does not follow
			1	1	The specified zones are not influenced by the protection method
			1	1	Multiple items are specified
			11	2	Max mode
			00	2	Izoi uses 8-bit integer
			00	2	Izoi is described in one dimension
		$Izoi^{11}$	8	0000 1010	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		8	$0000\ 1000_b$	Subsequent BAS byte does not exist. Codestream domain.
G	PO	16	$0\ 000\ 001\ 010\ 011\ 100_b$	Processing order is tile-resolution-layer-component-precinct
	GL	8	$0000\ 0110_b$	Unit of protection is packet
H		16	See Table 37 in 5.8.3.1	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_{bc} in T)

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

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

B.8.6 Conclusion

This subclause 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.

B.9 Scalable authenticity of JPEG 2000 codestreams

B.9.1 Security service

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

B.9.2 Typical application

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

B.9.3 Motivation

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

B.9.4 Technical overview

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

B.9.4.1 Signing module

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

- Reads a JPEG 2000 codestream.
- Constructs 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.
- Signs the *root* value of the hash tree with a private key based on a signature algorithm.
- Creates the SEC parameters. Insert the parameters into the SEC segment so as to produce an authentic codestream.

B.9.4.2 Transcoding module

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

- Reads the discarded packets which are not included in the transcoded codestream.
- Constructs the hash sub-trees with the discarded packets.
- Inserts root values of the sub-trees into SEC segment.

The transcoded codestream includes the updated SEC segment and codestream excluding the discarded packets.

B.9.4.3 Verifying module

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

- Reads the received codestream.
- Constructs the hash tree with the received packets and codestream headers from bottom to top. If some packets are discarded, replaces the sub-tree with the corresponding SIT. Thus the *root′* value is constructed.
- Checks the *root′* value against the signature in the SEC segment based on the specific signature system. If it matches, the codestream is accepted; otherwise, reject the received packets.

B.9.5 Codestream syntax

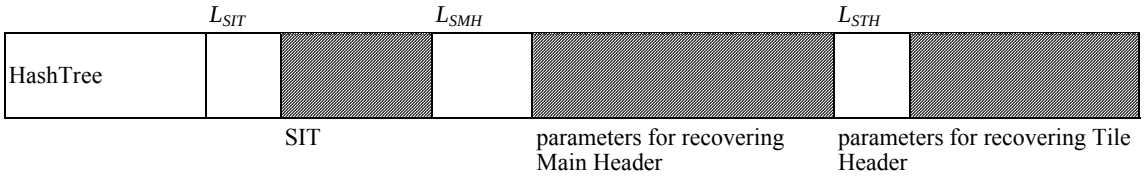
The SEC structure is shown in Table B.24. It includes the SEC marker, tool ID, and ZOI, the authentication template, and the security parameters for verification. The security parameters include data for recovering the codestream headers.

Table B.24 – Non-normative tool syntax

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

Parameter		Size (bits)	Values	Semantic
t		8 (RBAS)	1	Registration authority protection tool
i		8 (RBAS)	*Instance value*	Tool instance identifier
ID_{RA}	$ID_{RA,id}$	32	*ID value*	Registered ID
	$ID_{RA,nsl}$	8 (RBAS)	21	The length of $ID_{RA,ns}$ in bytes
	$ID_{RA,ns}$	168	*namespace*	The namespace of the RA with which this tool is registered
L_{ZOI}		16	$[0 \ldots 2^{16} - 1]$	Length of parameters for ZOI
ZOI_{ID}		Variable	ZOI values	Zone parameters
L_{ID}		16	$[19 \ldots 2^{16} - 1]$	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



Parameter	Size (bits)	Values	Meaning
HashTree	8	0 ... (2^8 – 1)	Hash Tree order. It may be different from the codestream 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
a) For Keyed-MAC authentication, the (verification) key should be delivered separately. b) NaN: Not a Number. c) L_{hash} is the size of hash value, e.g., 160 for SHA-1.			

B.9.6 Conclusion

The technology provides a flexible authentication mechanism for JPEG 2000 codestream. It has the property of "sign once, verify many ways". Concretely, after an original JPEG 2000 codestream is signed once, various codestreams transcoded from the original codestream 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.

B.10 JPEG 2000 data confidentiality and access control system based on data splitting and luring

The described system in this clause is based on the splitting, 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 the 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.10.1 Operational description

B.10.1.1 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 through 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 will not be allowed to 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 prerogative, possibly by combining this system with watermarking or fingerprinting.

B.10.1.2 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 plus 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 management 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.10.1.3 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_jp2file, 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 only after authentication and rights management protocols are passed.

B.10.2 Technical overview

Figure B.9 shows an overview of the system.

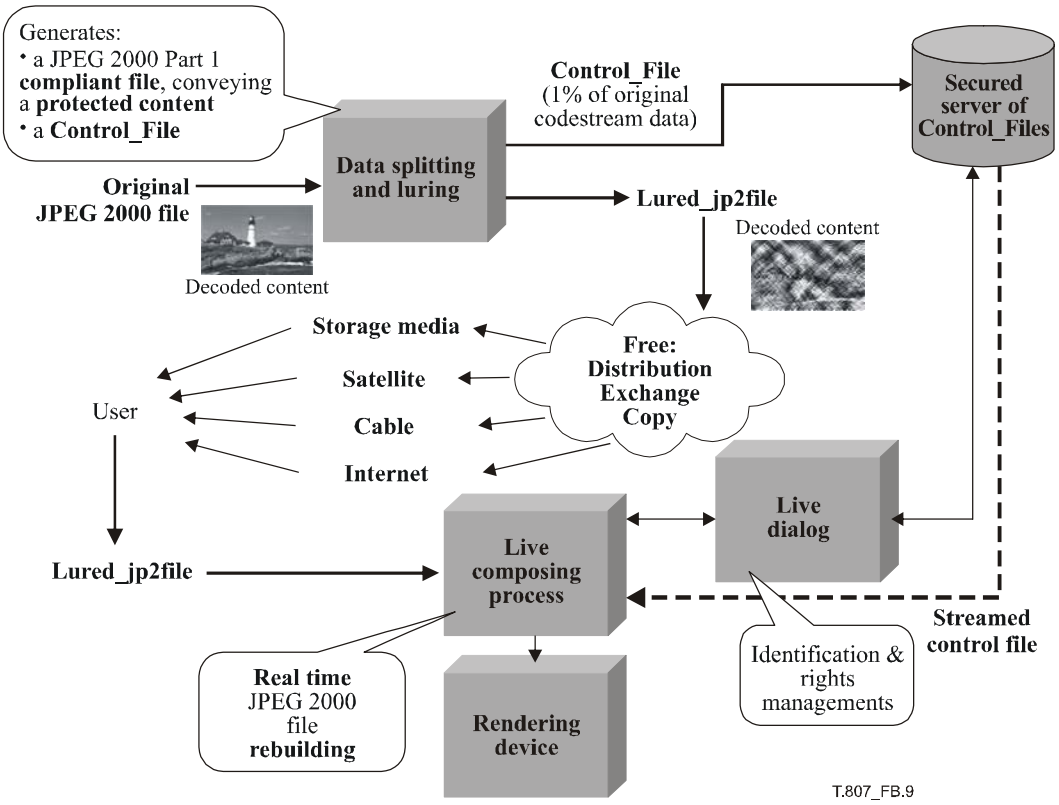


Figure B.9 – System overview

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 colour 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:

- *Splitting of the original* 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 *visual scrambling*;
- JPEG 2000 Part 1 *compliance and file size preserving*;
- *Low bit-rate and low computational* 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:

Table B.26 – Parameter values for this tool

Parameter					Size (bits)	Value (in order)	Derived meaning
SEC					16	0xFF65	SEC marker
L_{SEC}					16	0xXXXX	Length of the SEC marker segment
Z_{SEC}					8	1 ... 255	Index of the marker segment
P_{SEC} (if Z_{SEC} = 1)	F_{INSEC}				1	0	INSEC is not used
	$F_{multiSEC}$				1	0	One SEC marker segment is used
	F_{J2K}				2	1	JPSEC stream compliant with JPEG 2000 Part 1
	F_{TRLCP}				1	0	TRLCP tag usage is not defined in this field
	N_{tools}				7	1	One security tool is used in the codestream
	I_{max}				7	1	Maximum tool instance index value used
	Padding				5	0	Padding
$Tool^{(0)}$	t				8 (RBAS)	1	Non-normative protection tool
	i				8 (RBAS)	0	Tool instance index
	ID_{RA}	$ID_{RA,id}$			32	ID	RA is used to deliver the ID number
		$ID_{RA,nsl}$			8 (RBAS)	21	Length of $ID_{RA,ns}$ is 21 bytes
		$ID_{RA,ns}$			168	*namespace*	The namespace of the RA with which this tool is registered
	L_{ZOI}				16	*Length value*	Length of L_{ZOI} + ZOI
	ZOI	NZ_{ZOI}			8	0...254	Number of Zones
		$Zone^0$	DC_{ZOI}		1	0	The byte aligned segment does not follow
					1	1	Non-image related description class
					6	000010	Packet indexes are specified
			$Pzoi^{0,0}$	Mzoi	1	0	The byte aligned segment does not follow
					1	0	The specified zones are influenced by the protection method
					1	1	Multiple items are specified
					2	10	Index mode
					2	xx	Izoi uses 8- or 16- or 32-bit integer
					1	0	Izoi is described in one dimension
				Nzoi	8	Variable	2 ... 255 (number of packets indexes)
				$lzoi^i$	xxx Nzoi	variable	Packet index
	L_{PID}				16	0 ... $(2^{16} – 1)$	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.

B.11 Secure scalable streaming and secure transcoding

B.11.1 Summary and motivation

This subclause describes a method for providing the protection services of confidentiality and authentication of JPEG 2000 codestreams in a manner that:

1) allows a (potentially untrustworthy) entity to securely transcode or adapt JPSEC protected streams without 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 with 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 transcoding is defined as *transcoding without unprotecting (decrypting) the content*. This is achieved with secure scalable streaming, which combines scalable coding, encryption, and signalling in a manner that allows low-complexity, secure transcoding 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. For example, in Figure B.10 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.

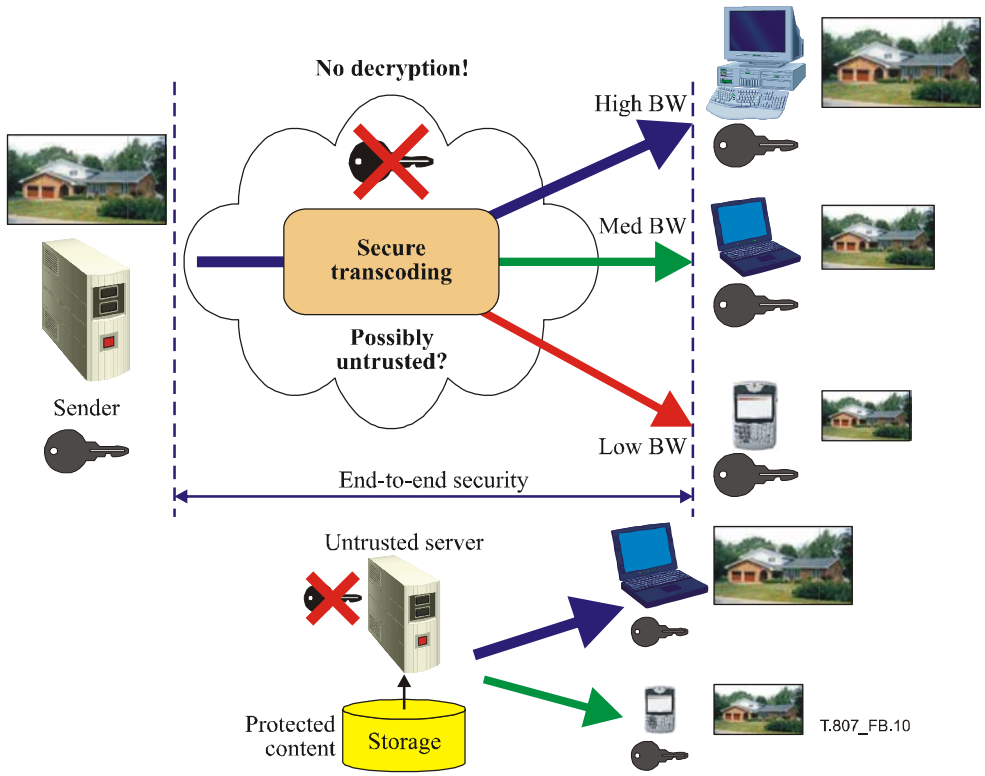


Figure B.10 – JPSEC enables end-to-end security and mid-network secure transcoding

B.11.2 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 resolution 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, codestream segment, and encryption 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 template.

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 values 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 ways. 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 the 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 signalled 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 signalling 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.

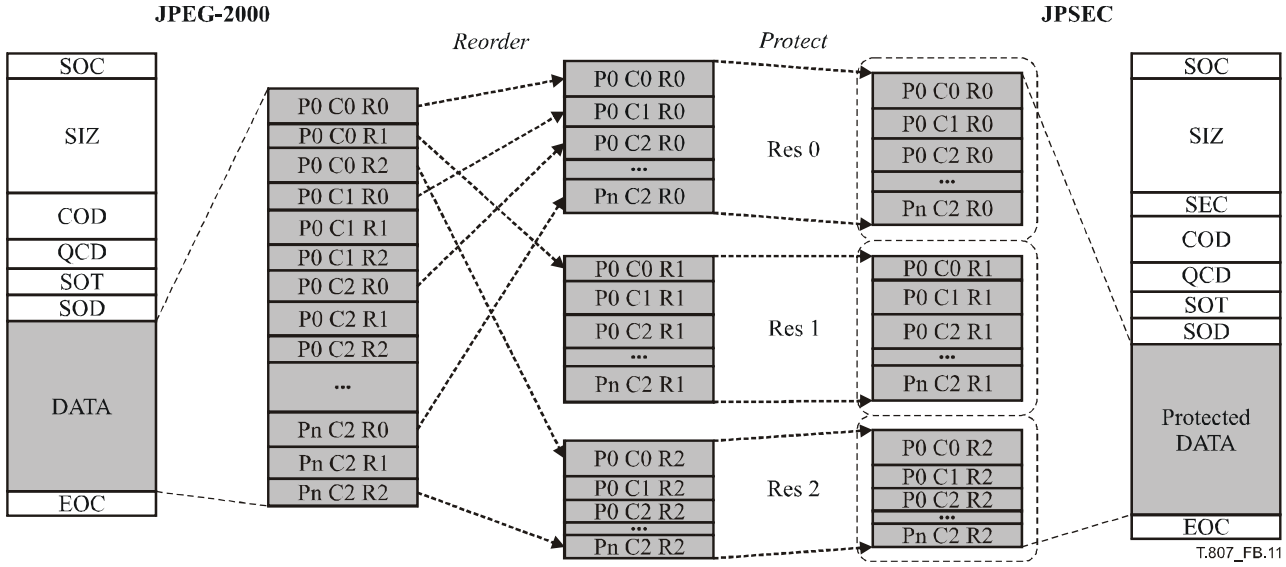


Figure B.11 – An example of forming a JPSEC codestream

B.11.3 Codestream syntax

The JPSEC syntax can be used to create a secure scalable streaming 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 encrypted bits for each zone. The signalling syntax for the decryption protection template, processing domain, and granularity are shown in Table B.27. The decryption method is signalled 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 headers 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 template. The syntax for the authentication template is shown in Table B.28 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 (see 5.7.3.2) to enable rate-distortion (R-D) optimized secure streaming and secure transcoding [26], [27] and [28].

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

Parameter			Size (bits)	Value (in order)	Derived meaning
T_{decry}	ME_{decry}		8	0	Marker emulation flag is NULL
	CT_{decry}		16	1	AES encryption
	CP_{decry}	M_{bc}	6	$10\ 0101_b$	CTR and no padding
		P_{bc}	2	0	Padding is not used for CTR mode
		SIZ_{bc}	8	128	Block size is 128 bits
		KT_{bc}	Variable	*Key template*	Key information template
PD			1	0_b	Byte aligned segment (BAS) does not follow
			1	0_b	Not in pixel domain
			1	0_b	Not in wavelet coefficient domain
			1	0_b	Not in quantized wavelet coefficient domain
			1	1_b	Processed in codestream domain
			3	000_b	Not used
G		PO	16	$0\ 0000\ 0101\ 0011\ 100_b$	Processing order is TRLCP
		GL	8	$0000\ 1001_b$	Granularity is total area identified by the ZOI
V		N_V	16	1	One value is specified
		Sv	8	16	Size is 16 bytes
		VL	128	*Nonce value*	Counter value for CTR mode

Table B.28 – Parameter values for authentication template protection tool

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

B.11.4 Conclusions

This subclause 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 untrustworthy) 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 permissible manner.

Annex C

Interoperability

(This annex forms an integral part of this Recommendation | International Standard)

C.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 normative tool for authentication inserts message authentication code values that are 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 codestream will begin decoding the JPSEC stream, it will then skip the SEC marker segment, and continue to decode the JPSEC stream as if it were a JPEG 2000 part 1 stream. The JPSEC normative tool for authentication 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 cause 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 codestreams 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 to 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 which has the 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.

C.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.

C.3 JPIP

C.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, ITU-T Rec. T.802 | ISO/IEC 15444-3 and ITU-T Rec. T.805 | 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 ITU-T Rec. T.802 | ISO/IEC 15444-3, Motion JPEG 2000, and ITU-T Rec. T.805 | 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 codestream, 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 codestreams, between servers and clients. This subclause considers how JPIP can be used to transport JPSEC content.

C.3.2 Specific issues on interactivity between JPIP and JPSEC

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

In A.3.5 "Main header data-bin" of ITU-T Rec. T.808 | ISO/IEC 15444-9, 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 ITU-T Rec. T.808 | ISO/IEC 15444-9 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 ITU-T Rec. T.808 | ISO/IEC 15444-9, 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 ITU-T Rec. T.808 | ISO/IEC 15444-9, 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 ITU-T Rec. T.808 | ISO/IEC 15444-9 describe the support to the tile-part header and tile-part data. However, A.3.4 and A.3.5 of ITU-T Rec. T.808 | ISO/IEC 15444-9 do not specify if they support 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.

C.3.3 Summary

Generally speaking, JPSEC makes itself suitable to be transported by JPIP. INSEC marker is used in the codestream 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 interactivity between JPIP and JPSEC.

C.4 JPWL

Wireless JPEG 2000 or JPWL (ITU-T Rec. T.810 | ISO/IEC 15444-11) extends the baseline JPEG 2000 specification to achieve the efficient transmission of JPEG 2000 imagery over an error-prone transmission environment. More specifically, JPWL defines a set of tools and methods to protect the codestream against transmission errors. It also defines means to describe the sensitivity of the codestream to transmission errors, and to describe the locations in the codestream of residual transmission errors.

JPWL is notably addressing the protection of the image header, Forward Error Correcting (FEC) codes, Unequal Error Protection (UEP), joint source-channel coding, data partitioning and interleaving, and robust arithmetic coding. JPWL is not linked to a specific network or transport protocol, but provides a generic solution for the robust transmission of JPEG 2000 imagery over error-prone networks.

The main functionalities of JPWL are:

- to protect the codestream against transmission errors;
- to describe the degree of sensitivity of different parts of the codestream to transmission errors;
- to describe the locations of residual errors in the codestream.

JPWL defines four marker segments: Error Protection Capability (EPC), Error Protection Block (EPB), Error Sensitivity Descriptor (ESD) and Residual Error Descriptor (RED).

The EPC marker segment indicates which JPWL normative and informative tools are used in the codestream. More specifically, EPC signals whether the three other normative marker segments defined by JPWL, namely the error sensitivity descriptor (ESD), the residual error descriptor (RED) and the error protection block (EPB) are present in the codestream. Furthermore, EPC signals the use of informative tools which have been previously registered with the JPWL RA. EPC is mandatory in a JPWL codestream.

The primary function of EPB is to protect the Main and Tile-part header. However, it can also be used to protect the remaining of the codestream. The EPB marker segment contains information about the error protection parameters and redundancy data used to protect the codestream against errors.

The ESD marker segment contains information about the sensitivity of codestream to errors. This information can be exploited when applying an Unequal Error Protection (UEP) technique. Straightforwardly, more powerful codes are used to protect the most sensitive portion of the codestream. This information can also be used for selective retransmissions. Finally, the information carried in ESD could also be used for other non-JPWL applications such as efficient rate transcoding or smart prefetching.

The RED marker segment signals the presence of residual errors in the codestream. Indeed, a JPWL decoder may fail to correct all the errors in a codestream. RED allows signalling the location of such residual errors. This information can then be exploited by a JPEG 2000 decoder in order to better cope with errors. For instance, the decoder could request retransmission, conceal the errors or discard the corrupted information.

C.4.1 General relationship between JPWL and JPSEC

The combination of JPWL and JPSEC is required whenever JPEG 2000 images need to be secured and transmitted over an error-prone wireless channel.

At the transmitter side, JPWL error sensitivity is typically generated during JPEG 2000 encoding. JPSEC tools are then applied to the codestream in order to secure it. Finally JPWL encoding tools are used to make the codestream more robust to transmission errors.

At the receiver side, JPWL decoding tools are first applied to correct possible transmission errors. During this step, JPWL may also generate residual errors information. Finally, JPSEC tools are applied in order to fulfil the selected security services.

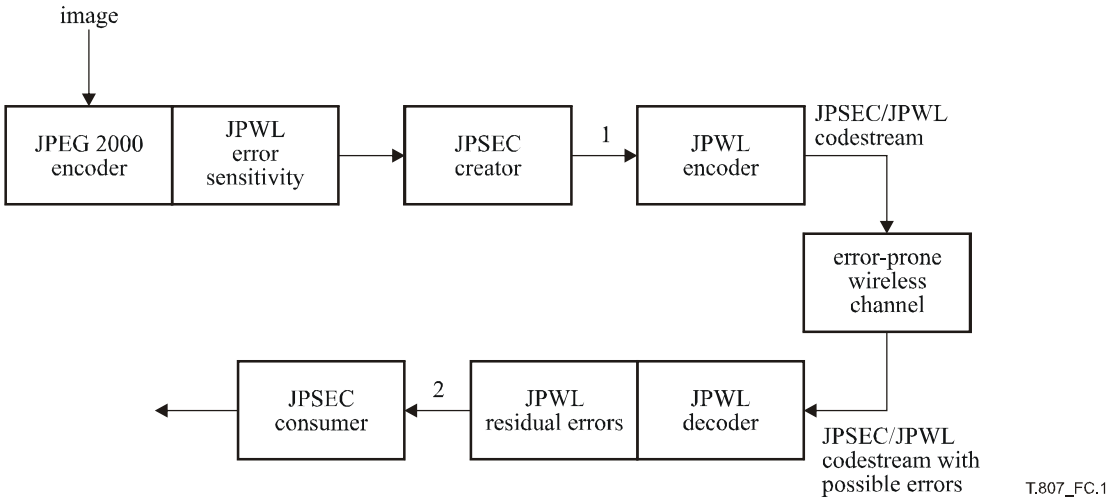


Figure C.1 – Typical JPWL and JPSEC combination

C.4.2 Specific issues on interoperability between JPWL and JPSEC

A number of issues have to be considered for interoperability between JPWL and JPSEC, as detailed hereafter:

1) JPWL Error Protection Capability (EPC): The presence of this marker segment will affect byte ranges. Note that this marker segment is mandatory in a JPWL codestream.
2) JPWL Error Protection Block (EPB): This marker segment is typically added as the last step at the transmitter and removed as the first step at the receiver. In principle, it should not affect JPSEC.
3) JPWL Error Sensitivity Descriptor (ESD): This marker segment is typically added during JPEG 2000 part 1 encoding, in which case it will be transparent to subsequent JPSEC operations. However, JPSEC could adversely affect the use of ESD in JPWL. In particular, JPSEC should not change byte ranges whenever ESD uses byte ranges. In addition, the JPSEC operations should not affect distortion values; otherwise the information carried by ESD becomes irrelevant. In the latter case, the JPSEC creator has the option to remove the ESD marker segment.
4) JPWL Residual Error Descriptor (RED): This marker segment can be inserted after JPWL decoding. It may therefore affect JPSEC byte ranges. It may also impact JPSEC authentication techniques. In case of a corrupted codestream, the RED information can be useful for a JPSEC consumer to appropriately handle it.
5) JPSEC SEC: The presence of this marker segment will affect byte ranges. Note that this marker segment is mandatory in a JPSEC codestream.
6) JPSEC INSEC: The presence of this marker segment will affect byte ranges. Note that this marker segment appears in the codestream data.

In the case when there are no residual errors, the JPWL encoder and decoder should ideally be transparent. In other words, in this case, the streams at points 1 and 2 in the above figure should be strictly identitical.

As a general recommendation, when used in combination with JPWL, it is preferable for JPSEC to use byte ranges beginning after SOD marker in order to minimize problems with byte ranges. In addition, it is preferable to restrict the presence of JPWL marker segments to the Main header and to avoid their presence in the Tile-part headers.

Annex D

Patent statements

(This annex does not form an integral part of this Recommendation | International Standard)

NOTE – Annex D is an ISO/IEC annex only. Companies submitting patent declarations to ITU concerning this text are listed in the IPR database. See: http://itu.int/ITU-T/ipr/.

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 ISO 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 D.1 – List of statements

Number	Submitting entity
1	Canon Inc
2	Columbia University
3	EMITALL Surveillance
4	HP
5	Institute for Infocomm Research
6	MediaLive
7	New Jersey Institute of Technology

BIBLIOGRAPHY

[1] ITU-T Recommendation X.800 (1991), *Security architecture for Open Systems Interconnection for CCITT applications.*

ISO/IEC 7498-2:1989, *Information Processing Systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture*.

[2] ISO/IEC 9796-2:2002, *Information technology – Security techniques – Digital signature schemes giving message recovery – Part 2: Integer factorization based mechanisms*.

[3] ISO/IEC 9797-1:1999, *Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher*.

[4] ISO/IEC 9798-1:1997, *Information technology – Security techniques – Entity authentication – Part 1: General*.

[5] ISO/IEC 10118-1:2000, *Information technology – Security techniques – Hash-functions – Part 1: General*.

[6] ISO/IEC 10118-2:2000, *Information technology – Security techniques – Hash-functions – Part 2: Hash-functions using an n-bit block cipher*.

[7] ISO/IEC 10118-3:2004, *Information technology – Security techniques – Hash-functions – Part 3: Dedicated hash-functions*.

[8] ISO/IEC 10118-4:1998, *Information technology – Security techniques – Hash-functions – Part 4: Hash-functions using modular arithmetic*.

[9] ISO/IEC 11770-1:1996, *Information technology – Security techniques – Key management – Part 1: Framework*.

[10] ISO/IEC 11770-2:1996, *Information technology – Security techniques – Key management – Part 2: Mechanisms using symmetric techniques*.

[11] ISO/IEC 11770-3:1999, *Information technology – Security techniques – Key management – Part 3: Mechanisms using asymmetric techniques*.

[12] ISO/IEC 13335-1:2004, *Information technology – Security techniques – Management of information and communications technology security – Part 1: Concepts and models for information and communications technology security management*.

[13] ISO/IEC TR 13335-4:2000, *Information technology – Guidelines for the management of IT Security – Part 4: Selection of safeguards*.

[14] ISO/IEC 14888-1:1998, *Information technology – Security techniques – Digital signatures with appendix – Part 1: General*.

[15] ISO/IEC 14888-3:1998, *Information technology – Security techniques – Digital signatures with appendix – Part 3: Certificate-based mechanisms*.

[16] ISO/IEC 15946-2:2002, *Information technology – Security techniques – Cryptographic techniques based on elliptic curves: Part 2 – Digital signatures*.

[17] ISO/IEC 15946-3:2002, *Information technology – Security techniques – Cryptographic techniques based on elliptic curves: Part 3 – Key establishment*.

[18] ISO/IEC 15946-4:2004, *Information technology – Security techniques – Cryptographic techniques based on elliptic curves: Part 4 – Digital signatures giving message recovery*.

[19] ISO/IEC 18033-2:2006, *Information technology – Security techniques – Encryption algorithms – Part 2: Asymmetric ciphers*.

[20] ISO/IEC 18033-3:2005, *Information technology – Security techniques – Encryption algorithms – Part 3: Block ciphers*.

[21] ISO/IEC 18033-4:2005, *Information technology – Security techniques – Encryption algorithms – Part 4: Stream ciphers*.

[22] DWORKIN (Morris): Recommendation for Block Cipher Modes of Operation, Methods and Techniques, *NIST Special Publication 800-38A*.

[23] GROSBOIS (R.), GERBELOT (P)., EBRAHIMI (T.): 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 Diego, 29 July-3 August, 2001.

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

[25] RIVEST (R.L.), SHAMIR (A.), ADLEMAN (L.M.): A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM (2) 21*, 1978, Page(s): 120-126.

[26] WEE (S.), APOSTOLOPOULOS (J.): Secure Scalable Video Streaming for Wireless Networks, *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.

[27] WEE (S.), APOSTOLOPOULOS (J.): Secure Scalable Streaming Enabling Transcoding Without Decryption, *IEEE Inter. Conf. on Image Processing (ICIP)*, http://lib.hpl.hp.com/techpubs/2001/HPL_2001_320.html Sept. 2001.

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

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	**Terminals for telematic services**
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2007