INTERNATIONAL STANDARD

ISO/IEC 15444-8

First edition
2007-04-15

Information technology — JPEG 2000 image coding system —

Part 8: Secure JPEG 2000

Technologies de l'information — Système de codage d'image JPEG 2000 —

Partie 8: JPEG 2000 sécurisé

Reference number
ISO/IEC 15444-8:2007(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

CONTENTS

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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

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

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of patents, as indicated in Annex D.

ISO/IEC 15444-8 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information* in collaboration with ITU-T. The identical text is published as ITU-T Rec. T.807.

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

— *Part 1: Core coding system*

— *Part 2: Extensions*

— *Part 3: Motion JPEG 2000*

— *Part 4: Conformance testing*

— *Part 5: Reference software*

— *Part 6: Compound image file format*

— *Part 8: Secure JPEG 2000*

— *Part 9: Interactivity tools, APIs and protocols*

— *Part 10: Extensions for three-dimensional data*

— *Part 11: Wireless*

— *Part 12: ISO base media file format*

— *Part 13: An entry level JPEG 2000 encoder*

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.

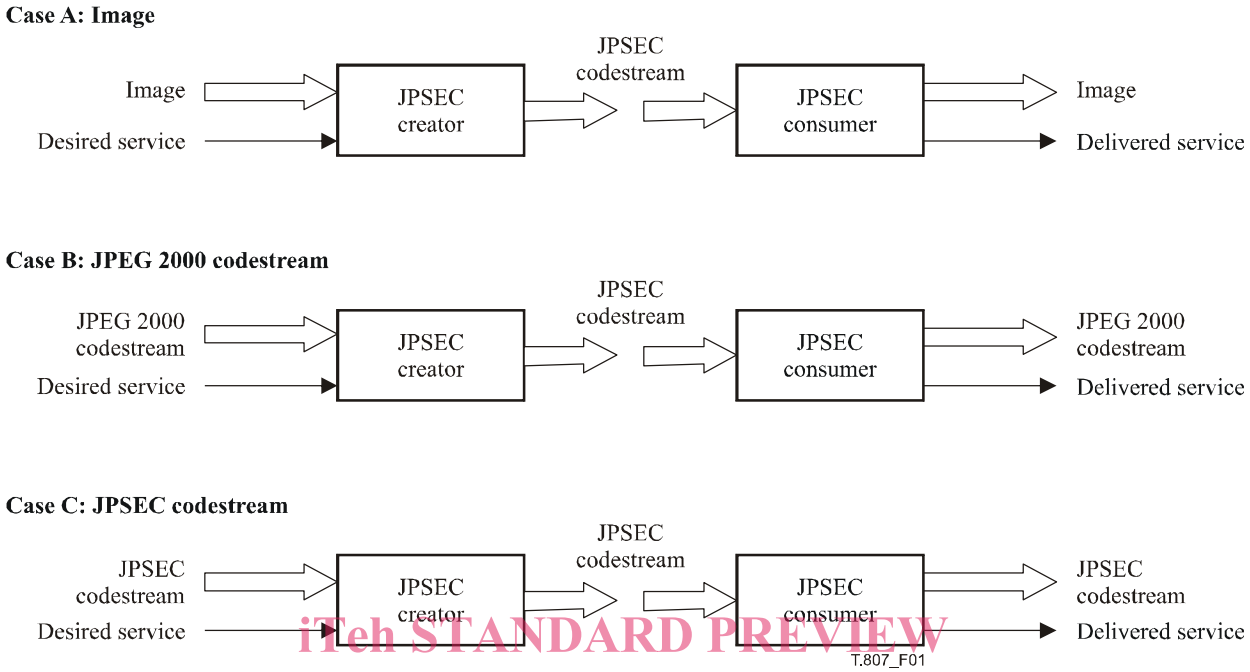

Figure 1 – Overview of the conceptual steps in JPSEC framework

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.

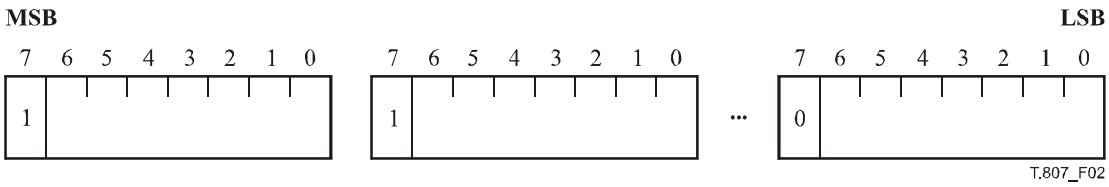

Figure 2 – Byte aligned segment (BAS) structure

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 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