U/OO/234067-17

Inspection and Sanitization Guidance for JPEG 2000

Version 1.4.4

26 November 2012

National Security Agency
9800 Savage Rd, Suite 6721
Ft. George G. Meade. MD 20755

Authored/Released by:
Unified Cross Domain Capabilities Office
cds_tech@nsa.gov

DOCUMENT REVISION HISTORY

Date	Version	Description
11/26/2012	1.4.4	final
12/13/2017	1.4.4	Updated Contact information, IAC Logo, Cited Trademarks and Copyrights, Expanded Acronyms, and added Legal Disclaimer

Disclaimer

The information and opinions contained in this document are provided “as is” and without any warranties or guarantees. References herein to any specific commercial product, process, or service by trade name, trademark, manufacturer or otherwise, does not constitute or imply its endorsement, recommendation or favoring by the United States Government and this guidance shall not be used for advertising or product endorsement purposes

EXECUTIVE SUMMARY

This document provides guidelines and specifications for developing file analysis and sanitization software for JPEG 2000 image files.

JPEG 2000 was developed by the Joint Photographic Experts Group (JPEG) Committee, adding many features that were not available in the original JPEG specification. JPEG 2000 not only increases the compression efficiency on images, but also provides scalability in resolution, quality and random access into the compressed image data. As a result of these new features, one of the more popular uses of JPEG 2000 is to represent satellite and telescopic imagery, in which it is essential to retain the quality of the original image. For instance, the National Geospatial-Intelligence Agency relies on the JPEG 2000 format for the transmission of satellite images.

Most risks in the JPEG 2000 format occur in data structures with unbounded lengths. These risks can be mitigated by removing all unknown data structures and validating the lengths of all known data structures. This document will go into greater detail on each element of the JPEG 2000 file format that requires additional mitigation.

TABLE OF CONTENTS

.

LIST OF FIGURES

LIST OF TABLES

1. SCOPE

1.1 Purpose of this Document

This document outlines the findings of potential areas of concern that exist within the JPEG 2000 file format. It also provides inspection and sanitization guidance for JPEG 2000 image files to address data attack, hiding, and disclosure risks. This document does not address any potential security risks of the JPEG 2000 compression/encoding algorithm, but focuses more on the actual format as a container for the compressed/encoded image data.

The intended audience of this document includes system engineers, software developers and testers, and security engineers who work with JPEG 2000 filters.

1.2 Introduction

In addition to being an encoding and decoding algorithm, JPEG 2000 also has a defined format for storing the image data. Its use as an image format is not widespread, but is used heavily for some applications such as satellite imagery and georeferencing, i.e. including metadata that connects the image data to exact locations on Earth with accuracy and precision. JPEG 2000 has significant performance advantages over the JPEG 92 specification including its ability for an image to be efficiently rendered at different resolutions using the same codestream.

The JPEG 2000 format has been adopted as the standard algorithm for image data compression in the National Imagery Transmission Format (NITF), which is the digital imagery standard for the US intelligence community and has been adopted by civilian organizations as a standard for commercial imagery [9]. In addition, to facilitate interoperability among communities of interest, the International Organization Organization for Standards/International Electrotechnical Commission (ISO/IEC) JPEG 2000 Part 1 specification document has been tailored via the North Alantic Treaty Organization (NATO) Secondary Imagery Format (NSIF) Basic Image Interchange Format (BIIF) Profile for JPEG 2000 to define a common set of functionality for digital mapping and imagery [8]. NSIF is the standard for formatting digital imagery files and imagery related products and exchanging them among NATO members.

1.3 Background

JPEG 2000, preceded by the original JPEG standard, was initiated from a submission of a compression algorithm called Compression with Reversible Embedded Wavelets (CREW) to the ISO/IEC Sub Committee 29 (SC29) Working Group 1 (WG1). Since this new algorithm provided such a rich set of features that were an improvement over JPEG, a new standardization effort was started and ultimately resulted in the JPEG 2000 standard. The JPEG 2000 requirements were established in 1997 and included incorporation of the following features into the standard, which include but are not limited to:

- Better low bit-rate performance
- Lossless and lossy compression
- Random codestream access and processing
- Increased robustness to bit-errors

The JPEG 2000 format has a stronger internal organization than the original JPEG format and is much richer. For instance, a JPEG 2000 file may include IPR (intellectual property rights) information and Extensible Markup Language (XML) metadata. It also allows greater control over image display, through the use of tiling and coding styles, compared with the original JPEG format. JPEG 2000 images are “encode once, decode many ways”, which means they can be displayed in different ways by using different subsets of the encoded data.

1.4 Standard Sections and ISG Scope

The JPEG 2000 standard, specified by **ISO 15444**, consists of the following parts:

Table 1-1 Sections of the JPEG 2000 Standard

Part	Description	Addressed in this ISG
Part 1: Core coding system	Defines the format and compression algorithms for JPEG 2000 still image files.	**YES**
Part 2: Extensions	Defines flexibility, features, and metadata extensions to Part 1.	**YES**
Part 3: Motion JPEG 2000	Defines the format for JPEG 2000 motion pictures, including audio capability.	NO
Part 4. Conformance	Provides procedures for testing the conformance of files to Part 1.	NO
Part 5: Reference software	Includes source code packages for JPEG 2000 Part 1 implementations in both Java and C.	NO
Part 6: Compound image file format	Defines the JPM format for document imaging.	NO
Part 7: Abandoned	This part has been abandoned by the JPEG.	NO
Part 8: Secure JPEG 2000 (JPSEC)	Addresses security features of JPEG 2000.	**YES**
Part 9: JPEG Interactive Protocol (JPIP)	Defines the client-server protocol to efficient transmission/retrieval of JPEG 2000 data.	NO
Part 10: JPEG 2000 3D	Addresses the extension to 3D images (working draft). It introduces the use of non-uniform grids.	NO
Part 11: JPEG 2000 Wireless	Addresses the robustness against transmission errors that is necessary for wireless transmission.	NO
Part 12: Base media file format	Provides a base format for multimedia.	NO
Part 13: Entry-level encoder	Describes how to write a JPEG 2000 encoder.	NO
Part 14: JPXML	Provides an XML structural representation and reference for JPEG 2000.	NO

1.5 Document Organization

This document provides a taxonomy, gives a high-level overview of the file format, and focuses in detail on those elements and aspects of the format that pose a potential security risk.

The following table summarizes the organization of this document:

Table 1-2 Document Organization

Section	Description
Section 1: Scope	This section provides the scope, background, organization, and limitations of this document.
Section 2: Taxonomy	This section defines the terms and acronyms that are used in this document.
Section 3: JPEG 2000 Overview	This section describes the JPEG 2000 process and modes of operation.
Section 4. JPEG 2000 Format	This section describes the JPEG 2000 file format.
Section 5: JPEG 2000 Constructs and Metadata	This section describes the aspects of the JPEG 2000 format that pose risks to data transfer.
Section 6: Table of Document Constructs	This section lists the constructs that appear in this document.
Appendix A: Referenced Documents	This appendix lists the references that appear in this document.
Appendix B: Summary Table	This appendix summarizes all the filter guidance found in this document.

1.6 Recommendations

The following subsections summarize the categories of recommendation actions that appear in this document along with the associated options.

1.6.1 Actions

Generally, inspection and sanitization programs can perform one or more actions on a specific file format element: *Validate*, *Remove*, *Replace*, *External Filtering Required*, or *Review*. For each area of concern, a recommendation will be provided for every applicable action type. If an action type is not applicable to a given area of concern, this document will indicate "N/A". Table 1-3 Action Descriptions summarizes the actions.

Table 1-3 Action Descriptions

Recommendation Action	Comments
Validate	Verify the data structure's integrity. Determine the adherence of the data to its ISO/IEC standard when possible.
Replace	Replace the data structure or one or more of its elements with alternate values that alleviate the risk (e.g., replacing a user name with a non-identifying, harmless value or just substituting a common name for all authors). A specific function of this action could be transforming the data.
Remove	Remove the data structure or one or more of its elements and any other affected area. If this action is not viable because it would break the file, then indicate this in the action.
External Filtering Required	Identify the type of data and pass the data block onto an external action for handling that type of data. This data will generally be passed to another filter that is designed for handling that data type (e.g. extracting a jpg file from a MS Word document and passing only the jpg data to a filter that can verify/inspect/sanitize jpg data).
Review	Present the data structure or its constructs to a review by a human. This action assumes that a system administrator will be able to make an acceptance/rejection decision about the data by visually reviewing it. The human review assumes that the reviewer has no knowledge of the internal structure of the data. This case is usually reserved for image files.
Reject	Reject the file. This action may be applied when the data is undecipherable, such as encrypted data.

1.7 Document Limitations

1.7.1 Constructs

This document addresses the format-wide areas of concern as well as those individual fields of specific concern within the JPEG 2000 file format. This document does not address individual fields that display no specific concern.

1.7.2 Covert Channel Analysis

It is nearly theoretically impossible to detect or prevent covert channels during communication. It is impossible to identify all available covert channels in any file format. No tool can possibly analyze every channel, so this document highlights the highest risk areas to reduce or eliminate data spills and malicious content. This document does, however, offer guidance specific to the risk of Steganography.

2. GLOSSARY OF TERMS

Term	Definition
Bit Plane	A two dimensional array of bits with all the bits of the same magnitude in all coefficients or samples. Applicable to a component, tile-component, code-block, region-of-interest.
Box	Data structure defined by a length, unique id, and data.
Channel	A logical component of the image.
Codestream	Compressed image data with the signaling info needed to decode.
Coding Passes	The arithmetic coding process to produce a codestream consists of the following 3 passes: • *Significance Propagation Pass*: Coding pass on a single bit plane where the bit is coded if its location is not significant, but at least one of its eight-connected neighbors is significant • *Magnitude Refinement Pass*: Coding pass where all bits from locations that became significant in a previous bit plane are coded • *Clean-up Pass*: Coding pass where any remainder bits not coded in the previous passes are coded
Coefficient	Resulting value of a transformation
Component	A two-dimensional array of samples
Conforming reader	An image viewer/editor that adheres to JPEG 2000
Data attack risk	The ability of a file type to support delivery and execution of malicious content
Data disclosure risk	The ability of a file type to support the unauthorized release of information.
Data hiding risk	The ability of a file type to support the intentional or unintentional hiding of information.
DC level shift	A single value by which all wavelet samples in the image are shifted; the default shift value toggles between signed and unsigned values.
DC level shift, variable	A DC level shift in which the shift value can be different from the default. The DC level value can be varied using the DCO marker segment. Note: DCO is a marker segment defined in ISO/IEC 15444 Part 2.
Filter, wavelet	A mathematical function applied to a set of pixels that transforms it into a set of wavelet samples
ICC	International Color Consortium
ICC profile	An data format for describing the color scheme of an image

ISO/IEC	International Organization for Standardization / International Electrotechnical Commission
ISO/IEC 15444	The family of JPEG 2000 format specification standard documents
JPEG	Joint Photographic Experts Group
LSB	Least Significant Bit(s): the lowest-value bit(s) in a collection of bits; for instance, the rightmost bit in a standard byte is the LSB
Marker	A two-byte value from a predefined enumeration that gives context to some data in the codestream
Marker segment	A codestream metadata structure with a marker and a length field
MSB	Most Significant Bit(s): the highest-value bit(s) in a collection of bits; for instance, the leftmost bit in a standard byte is the MSB
Multiple component transformation	An operation that converts raw component (color bands) data to a different mathematical representation of the components, commonly to reduce the redundancy between components; transformed components may also be subsampled in the process
Packet	A part of the bit stream comprising a packet header and the compressed image data from one layer of one precinct of one resolution level of one tile-component.
Pass, low	A filter that creates a set of wavelet samples that represent the lower frequencies of an image
Pass, high	A filter that creates a set of wavelet samples that represent the higher frequencies of an image; data resulting from the low pass and high pass can be re-combined to reproduce the original image
Precinct	A one rectangular region of a transformed tile-component, within each resolution level, used for limiting the size of packets.
Quantization	A method of reducing the precision of the individual coefficients to reduce the number of bits used to entropy-code them.
Sample	One element in the component.
Subband	A set of wavelet samples obtained from the original image by applying a wavelet filter in the horizontal direction and a wavelet filter in the vertical direction
Subband, LL	The subband obtained from low-pass wavelet filtering in both the horizontal and vertical directions
Tile	A spatial subset of the image data
Wavelet	A linear transform/subband decomposition

Wavelet sample	In the context of JPEG 2000, the representation of a single output value of the wavelet transformation; the number of wavelet samples after transformation is identical to the number of component pixels prior to transformation.
Wavelet transformation	The process through which a bitmap image is iteratively filtered and separated into a series of discrete samples corresponding to different frequency bands

3. JPEG 2000 OVERVIEW

This chapter describes the format of a JPEG 2000 image as specified by ISO 15444. The type of JPEG 2000 formatted data may be specified by or referred to via its file extension (Table 3-1). The formats/extensions covered by throughout the remainder of the ISG are J2C/JPC, JP2, and JPF/JPX.

Table 3-1 JPEG 2000 File Extensions

Extension	Description	Covered in this ISG
J2C/JPC	Contains **only codestream** data. Specified in ISO 15444 Part 1. This format has limited metadata but is capable of representing a still image.	**YES**
JP2	Signifies a **still image file**. Specified in ISO 15444 Part 1. Represents a single image, possibly with some additional metadata for specialized applications.	**YES**
JPF/JPX	Signifies an extension of the JP2 format containing **additional metadata**. Specified in ISO 15444 Part 2. Expands on XML capabilities provided in Part 1 by providing schemas for specific metadata structures. Provides new boxes and marker segments, as well as extensions to some boxes and segments provided in Part 1.	**YES**
MJ2/MJP2	Signifies a JPEG 2000 **motion image file**. Specified in ISO 15444 Parts 3 and 12. Adds motion picture and audio capabilities to the format.	NO
JPM	Signifies an extension of the JP2 format for **multi-page documents**. Specified in ISO 15444 Part 6. Defines additional boxes to specify page information.	NO

3.1 Introduction to JPEG 2000

JPEG 2000 refers to both an encoding/decoding process and also the container for encapsulating JPEG 2000 compressed image data and associated metadata. The security recommendations in this document focus on JPEG 2000 as a container, and not the JPEG 2000 coding process.

JPEG 2000 formatted files have a structure that enables large amounts of data to be potentially hidden from the user of an image viewing application. Data segments that have a fixed or maximum length according to the format specification may be extended simply by changing the value of the length field. This modification may also create a buffer overflow vulnerability because the image reader may allocate a buffer size based on the fixed/maximum length indicated by the ISO/IEC specification, however, the buffer size may be too small if the data length value is increased. In addition, JPEG 2000 allows for fields to be defined and used by a specific vendor's image reader. This vendor uniqueness makes it difficult to generically validate the contents of the fields.

3.1.1 Coding Process

The JPEG 2000 coding process consists of both an encoding and decoding process. Encoding takes as input the source image data and through a series of procedures generates compressed image data. Decoding takes as input compressed image data and through a series of procedures generates the original, reconstructed image data. Figure 3-1 depicts a high level view of the encoding process.

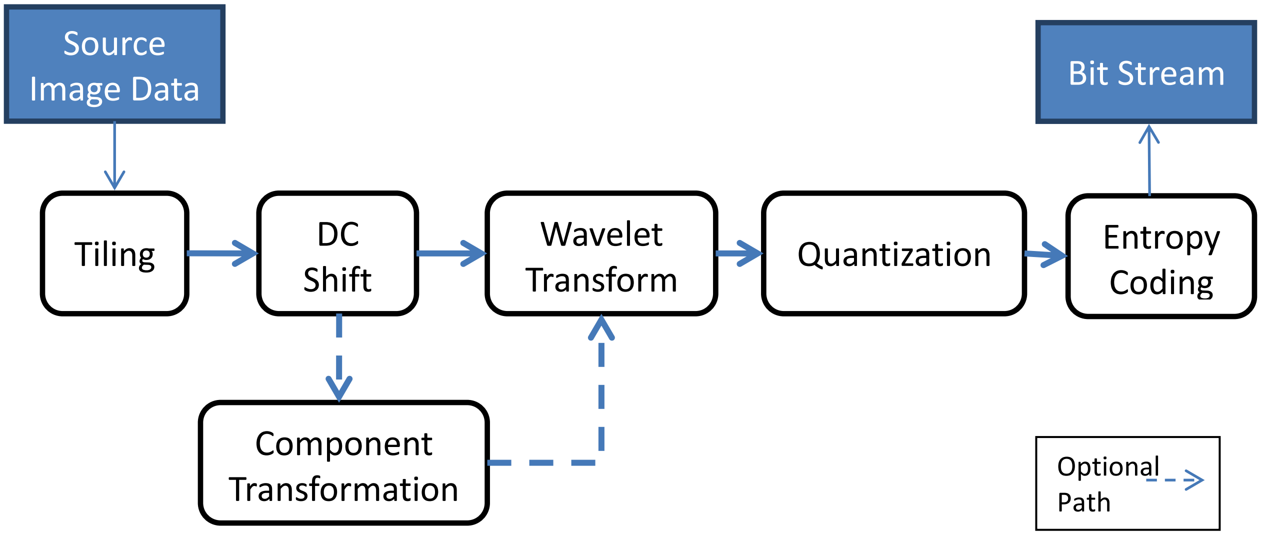

Figure 3-1 JPEG 2000 Encoding Process

An image component is divided into portions (i.e. tiles) through the **Tiling** process. Tiles can be extracted or decoded independently of each other, thus not forcing the image reader to store the entire image into memory at a time. Samples of each tile are then **DC Shifted**, which has the default effect of toggling between sign and unsigned values, and then each tile optionally undergoes **Component Transformation** which modifies the image color space. Each tile is then decomposed into different decomposition levels using a **Wavelet Transformation**. The decomposition levels are comprised of sub-bands of coefficients that describe frequency characteristics of portions of the tile. These coefficients are then **Quantized**, when lossy compression is performed, which reduces the precision of each coefficient. The tile sub-bands are further divided into code-blocks, which are arrays of coefficients. **Entropy coding** (i.e., encoding) occurs on these code blocks via three coding passes (significance propagation, magnitude refinement, and clean-up [See Glossary of Terms]) to produce the resulting bitstream of compressed image data (i.e., JPEG 2000 codestream).

3.1.2 Compression Modes

Image compression is the process of reducing the number of bits used to represent the image, typically done by eliminating redundant information in the image data in order to efficiently store, transmit, and distribute data. Unlike other image coding schemes, JPEG 2000 is capable of performing both lossy and lossless compression on the image data.

A lossy compression scheme discards certain information with the intent of making the resulting image file smaller while still retaining an acceptable approximation of the original image. In order to achieve lossy compression in JPEG 2000 at least one of the coding passes or encoding steps (See Figure 3-1) must be irreversible. One such encoding step is the quantization process, which reduces a continuous or near-continuous signal to a discrete signal, i.e. setting a finite number of possible values for any given signal element. In addition, one of the two component transformation functions is irreversible, so when it is applied to the image data, the result will always be lossy.

Lossless compression does not employ quantization or any form of data removal. It is a reversible function, i.e. it allows the original image to be reconstructed perfectly.

JPEG 2000 employs lossless compression by choosing reversible coding passes and steps throughout the encoding and decoding processes.

3.1.3 Data Format

The JPEG 2000 data formats examined in this document are the JP2, JPC, and JPX formats. JP2 formatted data contains additional metadata associated with the image where the metadata is defined in ISO 15444 Part 1. This metadata is either required for displaying the image or is used by the intended application for viewing the image. JPX formatted data refers to image files that contain ISO 15444 Part 2 extensions. JPC data, the image bitstream aka codestream, is one of the components of both JP2 and JPX formats; however, JPC data can also exist independently outside of the JP2 and JPX formats.

Both JP2 and JPX formatted data are composed of a series of data structures called boxes. Each box has a length field, a type identifier field, and content. The standard length field is 4 bytes long; however, it can be extended with an optional length field that is 8 bytes long. Each box contains some content fields. Some of the content fields have variable length according to the format specification.

J2C formatted data (codestream-only) does not contain boxes, but consists of only markers and marker segments. Markers are special two-byte codes that provide the context and indicate the structure of the data. This document refers to a marker and all the data associated with it collectively as a marker segment.

Many boxes or marker segments have either a definite or a maximum length provided by the JPEG 2000 standard. Some conforming image readers ignore extraneous data beyond that length since it is not required for processing. In addition, there may be extraneous data even within the specified length that is not required for processing. When reading the compressed image data, bytes from the variable length stream are pulled into the arithmetic decoder only as needed. Once the entire image is decoded, any leftover bytes in the codestream are ignored. This aspect of the format specification poses both data hiding and data attack risks.

4. JPEG 2000 DATA FORMAT

The JPEG 2000 JP2 data format (or JPX) is composed of structures called boxes. A box will either contain data or other boxes. Each box contains associated metadata to be used by the image reader for displaying or describing the image. Figure 4-1 depicts a high-level diagram of the JP2 box layout. As seen in the figure, the mandatory boxes are the Signature, Filetype, Header, and Codestream boxes. The optional boxes (Intellectual Property Rights (IPR), XML, Universally Unique Indentifer (UUID), UUID info) are able to appear either before or after the Codestream box.

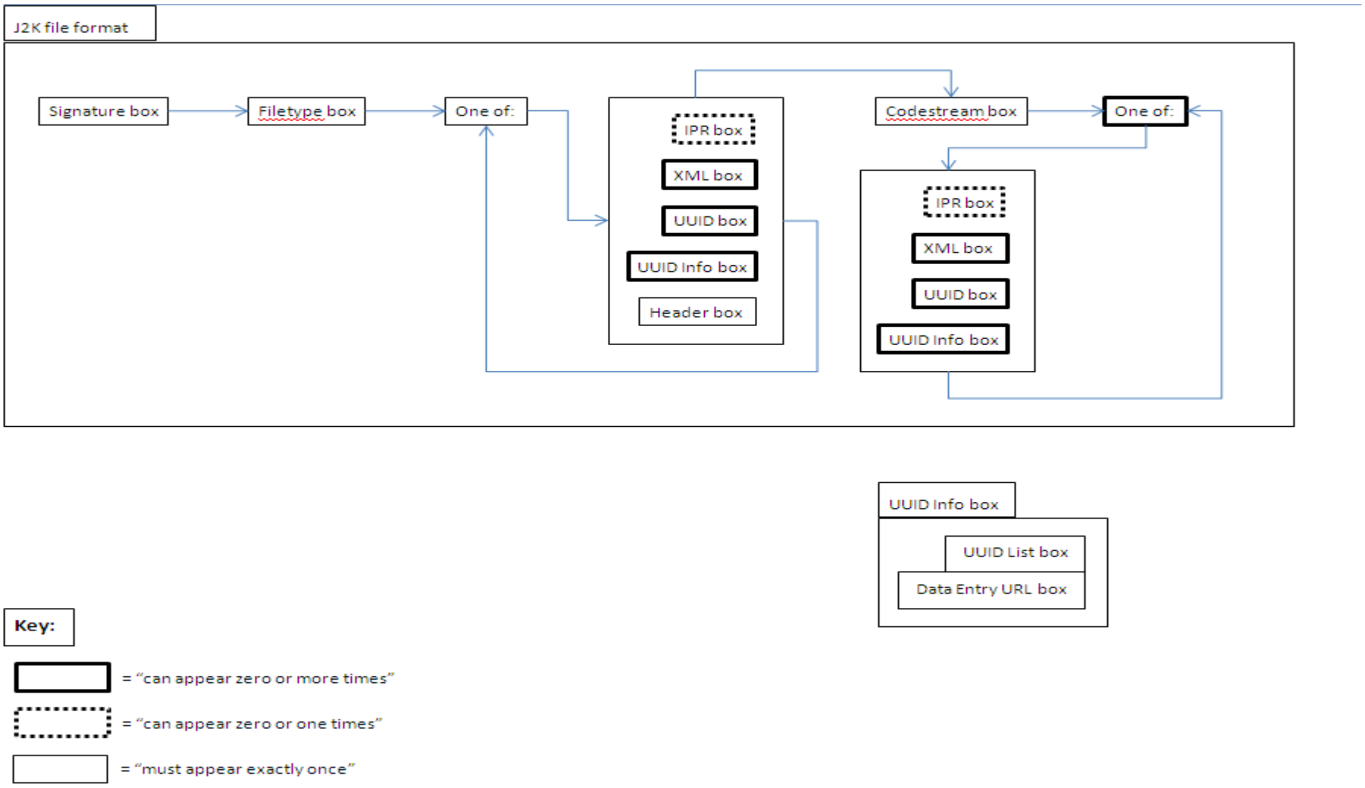

Figure 4-1 Top-level Box Layout of JP2 Formatted-Data

4.1 Signature box

The *Signature* box contains codes for checking common transmission errors and is typically used for the first integrity check of a JPEG 2000 file. It is important to note that the *Signature* box is not a magic number; the values of the error-checking codes will be different if the file has suffered from transmission errors, and in such cases the file is likely to be corrupted beyond repair.

This box must appear first in the file and only once per file. Its identifier field has the value "jP ". If its length field, identifier field, and error-checking codes are not all correct to the specification, then the file is not a valid JPEG 2000 file.

4.2 File Type box

The *File Type* box specifies the version of the ISO 15444-1 Standard that the contents adhere to and may also indicate a list of readers that the file is compatible with. The box contains a **brand** refers to the exact specification to which the file adheres; for instance, JPEG 2000 still images without extensions are identified by the brand "JP2", which adheres to ISO 15444 Part 1. The box may also contain a **minor version** value indicating the Standard's minor version. In addition, the box may include a **compatibility list** that includes a list of one or more standards and or image profiles that the file conforms to.

This box must appear immediately following the *Signature* box. Its identifier field has the value "ftyp".

4.3 Header box

The *Header* box contains image metadata related to the number of components, colourspace, and resolution. This box may contain the following sub-boxes: image header, bits per component, colour specification, palette, component mapping, channel definition, resolution. Figure 4-2 depicts a high-level diagram of the Header box layout.

This box must appear before the *Codestream* box (Section 4.4). Its identifier field has the value "JP2h". It may contain seven specified other types of boxes. Two of these box types (image header, colour specification) are required and the other five box types are optional.

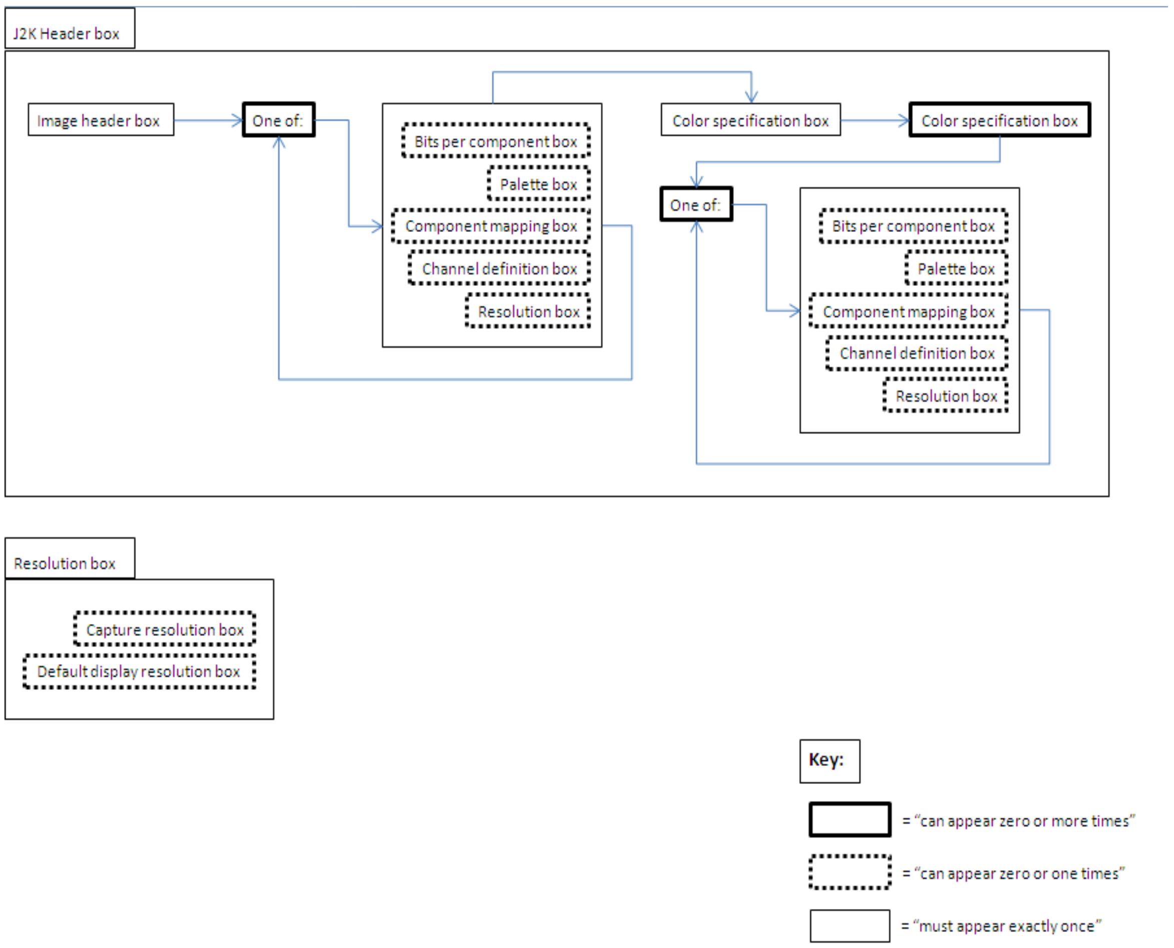

Figure 4-2 Header Box Layout

4.3.1 Image Header box

The *Image Header* box contains 7 fields (Table 4-1) which provide generic information about the image, such as its height and width. It also provides the number of image components and the bit depth of each component. A component is an array of samples of the image. Typically an image will consist of multiple components, for example a separate component may be used to represent red, green, and blue color values. Much of the information in the *Image Header* box is redundant with information that is contained within the codestream. For example, the number of components field in the *Image Header* box should be equal to the the Csiz field value in the codestream SIZ marker.

Field Name	Size (in bits)

Height	32
Width	32
Number of Components	16
Bits per Component	8
Compression Type	8
Colourspace Unknown	8
Intellectual Property	8

Table 4-1 Image Header Fields

This sub-box is contained within the *Header* box and is required as the first sub-box in the *Header* box. Its identifier field has the value “ihdr“.

4.3.2 Color Specification box

The *Color Specification* box specifies the colorspace of the decompressed image by defining the method (with associated metadata) an application should use to interpret the color space of the image data. This method is applied to decompressed and reverse-transformed data, i.e. the bitmap that results from the algorithm applied by conforming readers of JPEG 2000 files. There are two defined methods that can be applied to the image: Enumerated Colorspace or a Restricted International Color Consortium (ICC) profile. There is also flexibility to indicate another profile-defined method that should be applied to the image. If this is indicated, then the entire *Color Specification* box should be ignored.

JPEG 2000 files may contain multiple *Color Specification* boxes, but conforming readers only apply the first one to the image data. Non-conforming readers may use the additional boxes for some other purpose.

This sub-box is contained within the *Header* box and is required. Its identifier field has the value “colr“.

4.3.3 Bits per Component box

The *Bits per Component* box indicates the bit depth of each component. If the bit depth for each component is the same, then this box should not be present. For example of how bit depth is calculated, if an image contained three 8-bit components for red, green, and blue along with a 16-bit alpha channel, the JPEG 2000 file describing that image would have a *Bits per Component* box containing three bytes having the value of seven and one byte having the value of fifteen.

This box is contained within the *Header* box and is optional. Its identifier field has the value "bpcc". Note: The bits per component information can also be specified in the Bits per Component field of the *Image Header* box.

4.3.4 Palette box

The *Palette* box is used as a data transforming function that operates on the codestream. It defines the set of available colors that can be used, in conjunction with the Component Mapping and Channel Definition boxes, for creating multiple channels from a single component. A channel is defined as a logical component of the image; it may be equivalent to a color component such as red, green, or blue, or it may be the result of applying a color palette to a non-color component. Applying a palette to a component results in a wider colourspace used to define the image.

This box is contained within the *Header* box and is optional. Its identifier field has the value "pclr".

4.3.5 Component Mapping box

The *Component Mapping* box defines the method by which image channels are identified from codestream components. The method can either be **direct use** in which the channel is created directly from an actual component in the codestream or **palette mapping** where this box operates in tandem with the *Palette* box to create the means by which the image is interpreted. By separating this function from the *Channel Definition* box (described below), it allows that box to support images with and without palettes.

This box is contained within the *Header* box and is optional. Its identifier field has the value "cmap".

4.3.6 Channel Definition box

The *Channel Definition* box specifies the meaning of the samples in each channel in the image. This box contains an array of channel descriptions. Each description contains the following information: the channel index to associate the definition with, the channel type (level of opacity or transparency), and the channel association with a particular color.

This box is contained within the *Header* box and is optional. It may appear only one time in the Header, with or without the *Palette* and *Component Mapping* boxes. Its identifier field has the value “cdef“.

4.3.7 Resolution box

The *Resolution* box identifies the image capture and default display grid resolutions via its sub-boxes: Capture Resolution and Default Display Resolution.

This box is contained within the *Header* box and is optional. Its identifier field has the value “res “. It may contain two types of boxes: Capture Resolution and Default Display Resolution. Both are optional, however, at least one must appear.

4.3.7.1 Capture Resolution box

The *Capture Resolution* box indicates the resolution at which the image was created. Its identifier field has the value “resc“.

4.3.7.2 Default Display Resolution box

The *Default Display Resolution* box indicates the default resolution at which the image will be displayed. Its identifier field has the value “resd“.

4.4 Codestream box

The *Codestream* box contains the image bitstream. Codestreams are constructed so that an image viewer can display the image without the need for any additional JP2 box data. When JPEG 2000 data only contains the codestream, the data is JPC formatted. The box id field is “jp2c“. Note: When JPC-formatted, the id field is not present.

Codestream boxes may appear one or more times in a JPEG 2000 file. However, a conforming reader should ignore all codestreams after the first one found in the file.

A codestream is composed of markers and marker segments. Both are used for delimiting and error resilience purposes; however a marker segment can also be used to convey additional information about the image.

Marker and marker segments are either found in the codestream main header and/or in the tile-part headers. (A codestream can be comprised of many tiles.) The main header refers to metadata that applies to the entire image. The tile-part header refers to metadata specific to a single tile within the image. Although there are a total of 21 markers and marker segments (Table A.2 ISO 15444-1), the only ones that must appear in the codestream are SOC, SIZ, COD, QCD, and EOC (Table 4-2).

Name	Symbol	Information
Start of Codestream (SOC)	0xFF4F	Required as the first marker to indicate the start of the codestream
Image and Tile Size (SIZ)	0XFF51	Defines the spatial position of the reference grid, the image area, and the tiles. Defines the number of components and the bit depth of each component
Coding Style Default (COD)	0XFF52	Key properties of the default encoding method that was used to create and order the given wavelet samples
Quantization Default (QCD)	0xFF5C	Describes properties that were used to compress the given wavelet samples. In the case of lossy compression, this segment indicates the kind of data that was thrown away
End of Codestream (EOC)	0xFFD9	Required as the last marker to indicate the end of the codestream.

Table 4-2 Required Marker / Marker Segments

ISO 15444 Part 2 allows for the inclusion of additional marker segments, which are described in 4.7.1.

4.5 Intellectual Property Rights (IPR) box

The *Intellectual Property Rights (IPR)* box may appear only once per file. Its identifier field has the value "JP2i". It contains XML that describes intellectual property rights information. The format and types of information are specified by the schema in ISO 15444 Part 2 Annex N.4 and they include the names of right holders and the types of rights that are associated with the image.

4.6 Boxes containing Vendor-Specific Information

The following optional boxes may be used by vendors to add application-specific information to JPEG 2000 JP2-formatted data. The boxes should not contain any information necessary to decode or visually impact the image. The data contained within these boxes can only be validated by the intended reader application.

Box Name	Identifier	Intended Use
XML	"xml " Note: There is a space after the "l".	Contains additional metadata. ISO 15444 Part 2 Annex N defines XML schemas for additional metadata formats.
UUID	"uuid"	Contains vendor-specific information. The UUID format is not defined by ISO 15444.
UUID Info	"uinf"	Provides more info about the UUIDs.
UUID List Required sub-box in UUID Info box	"ulst"	Indicates all the UUIDs associated with the information provided through the URL box.
Data Entry URL Required sub-box in UUID Info box	"url " Note: There is a space after the "l".	Contains a Universal Resource Locator (URL) where more information is located. The URL format is not defined by ISO 15444.

Table 4-2. Boxes containing Vendor-Specific Information

4.7 JPEG 2000 Part 2 Extensions

ISO 15444 Part 2 defines the following extensions to the specification:

- A new file format (JPX) that enables support of JPEG 2000 Part 2 extensions.
- 30 additional box types that may be specified in the image.
- 12 additional codestream marker segments that may be applied to the encoding and/or decoding processes.
- XML schemas for representing image metadata.
- Two transformations that increase compression efficiency.

This section will only discuss the first 4 bullets.

4.7.1 JPX Format

JPEG 2000 files that contain any of the ISO 15444 Part 2 extensions are referred to as JPX-formatted. JPX conforming readers are those that are able to interpret all the Part 2 box and marker segment extensions, while a JP2 conforming reader must only be able to interpret ISO 15444 Part 1 boxes and marker segments.

JPX files should be saved with the .jpf file extension. However, if a JPX file is compatible with JP2 reader requirements, then the file may also be saved with the .jp2 file extension. The File Type box contains the compatibility listing for JP2 and JPX files, meaning that if a JPX file is compatible with the JP2 requirements, then the compatibility listing will contain the value "jp2\040" in the compatibility listing. JPEG 2000 readers can use this listing to determine if it is capable of reading the file.

4.7.2 Boxes

ISO 15444 Part 2 defines an additional 30 box types that may be included in the JPX format. Most of the boxes provide additional information to indicate image properties, such as color information and associated file metadata. Some of the boxes are used for interpreting how to render the image properly, such as the Opacity and Color Group boxes. Other boxes do not have any impact on image rendering, such as the Intellectual Property Rights and Digital Signature boxes.

The listing of JPX-specific boxes can be found in Table 5-2 and 5-3, if specific guidance is provided for a box then that will be identified in the tables as well. Section 5.4 addresses the Part 2 boxes that pose potential data hiding, attack, and/or disclosure risks.

4.7.3 Codestream

4.7.3.1 Codestream Fragmentation

In JP2 formatted data, a single file must contain the entire, contiguous codestream. However, this is not a requirement in JPX formatted files. JPX allows for the codestream to be fragmented within a single file or across multiple files. This feature enables more flexible application use such as faster access to images, more efficient image editing, and limiting access to higher quality or high resolution images. The JPX boxes that enable codestream fragmentation are the Fragment Table, Fragment List, and Media boxes, which are discussed more in Section 5.4.

4.7.3.2 Marker Segments

There are 12 additional markers/marker segments that may be included in the JPX format. The listing of JPX-specific marker segments can be found in Table 5-4, if specific guidance is provided for a marker segment then that will be identified in the tables as well. Section 5.4 addresses the Part 2 marker segments that pose potential data hiding, attack, and/or disclosure risks.

4.7.3.3 Multiple Codestreams

In JPX, it is possible for a single file to contain multiple JP2 images (codestreams). These multiple JP2 images, referred to as composite layers, are combined to form the final rendered result. The JPX format includes information for specifying how these layers should be combined by reader applications to produce the rendered image. This feature also enables the support of animation using JPEG 2000. The JPX boxes that enable this feature are the Codestream Header, Codestream Registration, Component Mapping, Composite Layer Header, and Composition Options boxes.

4.7.4 XML Metadata Extensions

ISO-15444 Part 2 also provides XML schemas that may be used for providing additional metadata about the image. This metadata, however, has no implications on the actual pixel data or how the pixel data should be processed. XML schemas are provided for the following types of metadata:

- Image Creation: Describes information such as the camera and lens info.
- Content Description: Includes descriptions that can be used for categorizing the image for possible searching purposes.
- History: Describes the processing steps and image creation events.
- Intellectual Property Rights (IPR): Includes possible copyright information.
- Image Identifier: Uniquely identifies image.

All the metadata, except the IPR, is provided via the XML box-type. The IPR metadata is contained within the IPR box-type. A list of possible elements in each schema is provided in Appendix C.

Conforming JPX readers should understand and recognize data described using these schemas. However, the specification states "The metadata shall be either correctly interpreted or ignored by a JPX reader."

4.8 Secure JPEG 2000 (JPSEC)

ISO/IEC 15444 Part 8 (JPSEC) defines the framework that is used to secure the JPEG 2000 codestream. When JPSEC features are used, the codestream is then referred to as the JPSEC codestream. The framework, which is extensible, focuses on the following security features.

- Confidentiality via encryption
- Integrity verification via digital signatures, check sums, Message Authentication Codes (MAC) and/or watermarking
- Source authentication via digital signatures or MAC
- Conditional access, which is used to grant or restrict access to image data or certain views of the image (i.e. higher resolution of the image)
- Registered content identification, which allows registration of images and associated metadata
- Secure scalable streaming and secure transcoding which facilitates separation of encryption-type functionality (securing) from decryption-type functionality (unsecuring)

5. JPEG 2000 CONSTRUCTS AND METADATA

This section addresses the aspects of the JPEG 2000 data format that pose data hiding, data disclosure, and/or data attack risks.

5.1 Boxes and Marker Segments

Every data structure in the JPEG 2000 JP2 and JPX formats is contained within a box. Data structures contained within a codestream box (or J2C-formatted) are further decomposed into marker segments. This subsection addresses general issues within boxes or marker segments.

JPEG2000:1: Box General (Identifier, Length, Order)

DESCRIPTION:

A box is a container-type data structure with a reported size, a type identifier, and type-specific content. Boxes are the central building blocks of JPEG 2000 JP2-formatted files.

CONCERNS:

Since the length of a box is self-reported, some viewer applications parse JPEG 2000 files using the given length information, even if that length is not valid according to the specification. This poses a data hiding risk because data may be appended to any box and the length adjusted accordingly. This also poses a data attack risk because a reader or parser may use a fixed-length buffer to store the contents of a box and one cannot assume that bounds are checked properly in all cases. In addition, some readers may ignore boxes that it cannot identify (while still retaining the data), so this also poses a data hiding risk. ISO-15444 states: “If a conforming reader finds a box that it does not understand, it shall skip and ignore that box.” Also, since most boxes report their own length, it is possible to create a covert channel by reporting a length that exceeds the typical use of the box and using the remainder of the box data segment to filtrate extra data.

Where further guidance has not been provided for the box, the below recommendations can be applied. Boxes that have additional guidance are because they pose additional risks that are more explicitly described in subsequent ISG constructs.

Table 5-1 lists the boxes that have a defined length.

Table 5-1 Boxes With Constant Lengths

Box	Specified Length in Bytes
Signature Box	12
Image Header Box	22
Capture Resolution Box	18
Default Display Resolution Box	18
Composition Options Box	17

Table 5-2 lists the ordering requirements placed on boxes.

Table 5-2 Box Order & Identifier Requirements

Box	**ID**	**Ordering**	**JPX Extension**	**Additional ISG Guidance**
Signature	“jP “ 2 spaces	Must appear first in file	No	**Yes**
Filetype	ftyp	Must appear immediately after Signature	No	**Yes**
Reader Requirements	rreq	Must appear immediately after Filetype box	Yes	**Yes**
JP2 Header	jp2h	Must appear before Codestream box	Yes	No
Label	“lbl “ 1 space	Must not appear outside JP2 Header box, Codestream Header box, Compositing Layer Header box, or Association box	Yes	No
Image Header	ihdr	Must not appear outside JP2 Header box or Codestream Header box	No	**Yes**
Bits Per Component	bpcc	Must not appear outside JP2 Header box or Codestream Header box	Yes	No
Color Specification	colr	Must not appear outside JP2 Header box, Codestream Header box, or Color Group	Yes	**Yes**
Palette	pclr	Must not appear outside JP2 Header box or Codestream Header box	No	No
Component Mapping	cmap	Must not appear outside JP2 Header box or Codestream Header box	No	No
Channel Definition	cdef	Must not appear outside JP2 Header box or Compositing Layer Header box	No	No
Color Group	cgrp	Must not appear outside Compositing Layer Header box	Yes	No
Opacity	opct	Must not appear outside Compositing Layer Header box	Yes	No
Capture Resolution	resc	Must not appear outside Resolution box	No	No
Default Display Resolution	resd	Must not appear outside Resolution box	No	No
Fragment List	flst	Must not appear outside Fragment Table box or Cross-Reference box	Yes	No
Composition Options	copt	Must not appear outside Composition box	Yes	No
Instruction Set	inst	Must not appear outside Composition box	Yes	No
Graphics Technology Standard Output	gtso	Must not appear outside Desired Reproductions box	Yes	No
Number List	nlst	Must not appear outside Association box	Yes	No

UUID List	uuid	Must not appear outside UUID Info box	No	**Yes**
Data Entry URL	“url “ 1	Must not appear outside UUID Info box	No	**Yes**
Association	asoc	Any box may appear inside	Yes	No

Table 5-3 Other Box Identifier Requirements

The following boxes do not have any ordering requirements except that they must appear after the Signature, Filetype, and Reader Requirements boxes.

Box Name	ID	JPX Extension	Additional Guidance
Codestream	jp2c	No	**Yes**
XML	“xml “ 1 space	No	**Yes**
IPR	jp2i	Yes	**Yes**
UUID Info	uinf	No	No
Resolution	“res “ 1 space	No	No
UUID List	ulst	No	No
Data Reference	dtbl	Yes	No
Fragment Table	ftbl	Yes	No
Cross Reference	cref	Yes	No
Codestream Header	jpch	Yes	No
Compositing Layer Header	jplh	Yes	No
Codestream Registration	creg	Yes	No
Media Data	mdat	Yes	**Yes**
Composition	comp	Yes	No
Binary Filter	bfil	Yes	**Yes**
Desired Reproductions	drep	Yes	No
ROI Description	roid	Yes	No
Digital Signature	chck	Yes	**Yes**
MPEG-7 Binary	mp7b	Yes	**Yes**
Free	free	Yes	**Yes**

PRODUCT: ANY JPEG 2000 IMAGE FILE

LOCATION:

J2C and JPX formatted data

RECOMMENDATIONS:

1 Validate: Verify that the box in question adheres to the length bounds indicated by the specification as referenced in Table 5-1.

2 Validate: If no length bound is specified, then verify box length against a predefined maximum length value.

3 Validate: Verify that the order of boxes in the file does not violate the specification as referenced in Table 5-2.

4 Validate: Verify that each box in the file has a valid identifier.

5 Remove: Remove all data beyond the length bound for the given box that is determined from the specification, as referenced in Table 5-1.

6 Remove: Remove all boxes that do not have a valid identifier.

7 Remove: Remove boxes with identifiers that are not on an approved list of box identifiers. Note: This may affect the user's experience with the image and degrade reader capabilities with the image.

8 Remove: Remove boxes whose length exceeds a pre-defined value. Note: This does not apply to the required boxes.

9 Replace: Replace boxes with pre-defined, known good box values.

10 External Filtering Required: N/A

11 Review: N/A

12 Reject: Reject files that have invalid box identifiers, length, and/or ordering.

REFERENCE:
N/A

JPEG2000:2: **Marker Segments**

DESCRIPTION:

A marker segment is a data structure in the codestream beginning with a 2-byte identifier followed by a 2-byte length value and corresponding segment data. Marker segments are the building blocks of JPEG 2000 codestreams.

Table 5-4 Marker and Marker Segments

Marker Segment Name	Symbol	Code	JPX Extension
Start of Codestream	SOC	0xFF4F	No
Start of Tile Part	SOT	0xFF90	No
Start of Data	SOD	0xFF93	No
End of Codestream	EOC	0xFFD9	No
Image and Tile Size	SIZ	0xFF51	No
Coding Style Default	COD	0xFF52	No
Coding Style Component	COC	0xFF53	No
Region of Interest	RGN	0xFF5E	No
Quantization Default	QCD	0xFF5C	No
Quantization Component	QCC	0xFF5D	No
Progression Order Change	POC	0xFF5F	No
Tile Part Lengths	TLM	0xFF55	No
Packet Length, Main	PLM	0xFF57	No
Packet Length, Tile	PLT	0xFF58	No
Packed Packet Headers, Main	PPM	0xFF60	No
Packed Packet Headers, Tile	PPT	0xFF61	No
Start of Packet	SOP	0xFF91	No
End of Packet	EPH	0xFF92	No
Component Registration	CRG	0xFF63	No
Comment	COM	0xFF64	No
Variable DC Offset	DCO	0xFF70	Yes
Visual Masking	VMS	0xFF71	Yes
Downsampling Factor Style	DFS	0xFF72	Yes
Arbitrary Decomposition Style	ADS	0xFF73	Yes
Arbitrary Transformation Kernels	ATK	0xFF79	Yes
Component Bit Depth	CBD	0xFF78	Yes
Multiple Component Transformation Definition	MCT	0xFF74	Yes
Multiple Component Collection	MCC	0xFF75	Yes
Multiple Component Transformation Ordering	MCO	0xFF77	Yes
Non-Linearity Point Transformation	NLT	0xFF76	Yes
Quantization Default, Precinct	QPD	0xFF5A	Yes
Quantization Component, Precinct	QPC	0xFF5B	Yes

CONCERNS:

All the markers in the codestream except for the SOC (start of codestream), SOD (start of data), EOC (end of codestream), and EPH (end of packet header) markers are followed immediately by a 2-byte field specifying the length of the marker segment. This poses a data hiding risk because data can be added after the end of a marker segment provided that its length field is updated to account for it. Additionally, marker segments that use arrays pose a data attack risk when the array size is based on the segment length and the maximum specified array length is small; the length can be increased beyond the expected maximum, which could cause a buffer overflow.

PRODUCT: ANY JPEG 2000 CODESTREAM

LOCATION:

This issue exists in every marker segment in a codestream box.

RECOMMENDATIONS:

1 Validate: Verify that each marker segment has a valid code as according to Table 5-4.

2 Validate: Verify the data in each marker segment.

3 Validate: Verify that marker segment is within the codestream data.

4 Remove: Remove all data beyond the length bound for each marker segment length that is defined by the specification. Modify the length field accordingly. Note: This recommendation cannot be employed for all marker segments.

5 Remove: Remove Comments (COM) marker segment since it is optional.

6 Replace: Replace the entire segment with pre-defined, known good marker segment data that does not conflict with other data in the file. Note: This operation may alter the original, intended display of the image.

7 External Filtering Required: Pass COM marker segment data to filter that is capable of processing text.

8 Review: N/A

REFERENCE:

See ISO 15444 Part 1 and Part 2

5.2 File Header

This subsection addresses issues that exist in specific boxes and/or fields within the file header segments, which contains the Signature, File type, and JP2 Header boxes.

JPEG2000:3: **Signature Box**

DESCRIPTION:
The *Signature* box identifies the data as JPEG 2000 JP2/JPX formatted data.

CONCERNS:
Although not specifically a data hiding, attack, or disclosure risk, if the values in the *Signature* box do not match the specified value of 0x0D0A870A, then the data is not valid JP2-formatted data and should not be processed by image readers.

PRODUCT: ANY JPEG 2000 IMAGE FILE

LOCATION:
The Signature box is the first box in JP2 and JPX formatted data, however it will not appear in J2C formatted data.

RECOMMENDATIONS:

1 Validate: Verify that the value of the error-check field is 0x0D0A870A. Note: This validate should not be used when the data is JPC formatted.

2 Remove: N/A

3 Replace: N/A

4 External Filtering Required: N/A

5 Review: N/A

6 Reject: Reject images that do not match the specified value of 0x0D0A870A.

REFERENCE:
See ISO 15444 Part 1, Annex I.5.1.

JPEG2000:4: **Filetype Box**

DESCRIPTION:

The *Filetype* box identifies the specific JPEG2000 format(s), such as JP2, JPX, etc., and versions that the data adheres to. This box is used by the readers to determine how or if the reader is capable of displaying the data.

CONCERNS:

This box, like many of the other JP2 boxes, reports its own length so could be used as a covert channel. The only unrestrained data within the box is a compatibility list, which is used to specify the codes, standards, and/or profiles that the data conforms to. The length of the box determines the number of fields in this list. However, the data within this box is capable of being somewhat restrained because the data tends to fall within a specific enumeration of values.

PRODUCT: ANY JPEG 2000 IMAGE FILE

LOCATION:

The Filetype box is the second box in JP2 formatted data.

RECOMMENDATIONS:

1 Validate: Verify that the data adheres to defined ISO-15444 values. The enumeration of values is (in ASCII):   “jp2\040”, “J2P0”, “J2P1”.

2 Remove: N/A

3 Replace: Replace the compatibility list with pre-defined, known values.

4 External Filtering Required: N/A

5 Review: N/A

REFERENCE:

See ISO 15444 Part 1, Annex I.5.2.

JPEG2000:5: **UUID Box**

DESCRIPTION:

Each *UUID* box contains an ID field containing a 16-byte UUID value (format defined by ISO/IEC 11578) and a variable length data field that can contain any amount of arbitrary data. The data field is intended to store vendor-specific or application-specific information within a JPEG 2000 file. The format of data found in a UUID box is determined externally by the vendor or application developer and is referenced by the UUID which is represented by a 32 hexadecimal string in the form: 8-4-4-4-12 such as `654e8400-e32c-5624-5ca3-876234239231`.

CONCERNS:

The data field in a UUID box has no bounds or format defined in ISO 15444. This poses a data hiding risk because any type and length of data can be inserted in the field. This poses a data attack risk because if the software parsing the JPEG 2000 file employs a fixed-length buffer and does not properly check the bounds, then the data can be crafted in a way that creates a buffer overflow.

PRODUCT: ANY JPEG 2000 IMAGE FILE

LOCATION:

JP2 / JPX format

RECOMMENDATIONS:

1 Validate: N/A (Application-specific data cannot be validated).

2 Remove: Remove the entire UUID box. This may cause unintended effects when the file is loaded in the end user application.

3 Replace: N/A

4 External Filtering Required: N/A

5 Review: N/A

REFERENCE:

See ISO 15444 Part 1, Annex I.7.2. Also, see this ISG, Chapter 4.7.

JPEG2000:6: Data Entry URL Box

DESCRIPTION:

A *Data Entry URL* box provides a URL where data associated with a UUID is stored. The URL field is formatted as a null-terminated string.

CONCERNS:

This poses a data hiding risk because data may be added after the null termination which image readers may ignore. This also poses a data attack risk if readers use a fixed length array to store the string and may not check their array bounds before copying data, which can lead to a buffer overflow exploit. It is also possible that a valid URL could point to sensitive data or that a non-sensitive location referenced by a URL could become sensitive, which is a data disclosure risk.

PRODUCT: ANY JPEG 2000 IMAGE FILE

LOCATION:
JP2 / JPX format

RECOMMENDATIONS:

1 Validate: Verify that the *Data Entry URL* box is contained within a *UUID Info* box.

2 Validate: Verify that the URL string is shorter than a predefined length.

3 Validate: Verify that the string is null-terminated.

4 Remove: Remove data that follows the null termination.

5 Remove: Remove the entire box. This may cause unintended effects when the file is loaded in the end user application.

6 Replace: Replace URL string with a pre-defined value.

7 External Filtering Required: Pass the URL string to a filter that can validate the URL.

8 Review: N/A

REFERENCE:
See ISO 15444 Part 1, Annex I.7.3.2. Also, see this ISG, Chapter 4.8.

JPEG2000:7: **ICC (International Color Consortium) Profile**

DESCRIPTION:

An ICC profile is a data format for describing the color attributes of any device that captures or displays color [7]. Essentially, an image's ICC profile determines its color scheme. ICC profiles are used across a variety of data formats such as Bitmap, JPEG, and PDF. Within the JPEG 2000 JP2 format, the ICC profile is found in the *Color Specification* box. The *Color Specificatio*n box Method field is used to determine the type of ICC profile that should be used by the image reader. If the box Method field is the value 2 then the ICC profile is called "restricted" because it must conform either to the Monochrome Input or to the Three-Component Matrix Input profile class. If the Method field is 3 then any ICC profile is permitted. If the Method field is 4 then the ICC profile is a vendor specific color method.

CONCERNS:

An ICC profile is an embedded object adhering to an independent data format. As such, its data disclosure, hiding, and attack risks must be subjected to the same level of scrutiny as the JPEG 2000 file that contains it.

PRODUCT: ANY JPEG 2000 IMAGE FILE

LOCATION:

JP2 / JPX format. The ICC profile may exist at the end of the *Color Specification* box.

RECOMMENDATIONS:

1 Validate: Verify the ICC profile against the ICC format specification.

2 Remove: Remove the ICC profile. This may result in a color scheme that renders the image unviewable.

3 Replace: Replace the ICC profile with a 4-byte enumerated color space (ECS) field and set the Method value to 1. This may result in a color scheme that renders the image unviewable.

4 External Filtering Required: Pass the contents of the ICC Profile field to a filter capable of handling ICC profiles.

5 Review: N/A

REFERENCE:

See ISO 15444 Part 1, Annex I.5.3.3. Also, see this ISG, Chapter 4.3.2.

5.3 Codestream

The codestream contains the compressed and encoded image data and all the metadata required to decode and display it, which is conveyed via marker segments.

JPEG2000:8: Image Data

DESCRIPTION:

Image data in the codestream can be contained within one or more tiles that can then be decomposed into one or more tile-parts. Each tile-part starts with a tile-part header segment that provides the length of the tile-part as a 4-byte integer. The last tile-part in the codestream may specify a length of 0, indicating it contains all data until the end of codestream (EOC) marker.

CONCERNS:

It is possible to modify the length value in the tile-part header and append arbitrary data, which would pose a data hiding risk, as well as a data attack risk because the arbitrary data itself may be intended as an attack vector against the image reader. Although this is possible, this arbitrary data would either break the decoding process resulting in an image reader error or break the image itself making it unviewable. In addition, data (not contained within a valid box) added past the EOC marker will have no impact on the reconstructed pixel values and thus will be hidden from the viewer.

PRODUCT: ANY JPEG 2000 IMAGE FILE

LOCATION:

Codestream within JP2 and JPX-formatted data

RECOMMENDATIONS:

1 Validate: N/A

2 Remove: Remove data that appears after the EOC marker, unless another valid JP2/JPX box is found.

3 Replace: Re-encode the codestream in a lossy manner.

4 Replace: Re-encode the codestream with different coding values, such as tile height and width or progression order.

5 External Filtering Required: N/A

6 Review: N/A

REFERENCE:

See ISO 15444 Part 1, Annex A.4.2.

JPEG2000:9: **Multiple Codestreams**

DESCRIPTION:

The JPEG 2000 specification allows multiple codestream boxes to be present in one image. This means that one actual file can contain multiple images. Each codestream contains all the necessary metadata to display an image properly, with the caveat that color information may be present in the image header. Conforming JP2 readers are required to ignore all codestreams after the first, so there is no reason to have multiple codestreams in a single JP2 file. JP2 files that conform to ISO 15444-1 have no functionality for multiple codestreams built into the format. However, multiple codestreams may be used in the JPX format.

CONCERNS:

There is no programmatic way to verify which codestream was intended for display. This poses a data hiding risk because a JP2 reader may only display one of the codestreams as the image.

PRODUCT: ANY JPEG 2000 IMAGE FILE

LOCATION:

JP2 and JPX-formatted data

RECOMMENDATIONS:

1 Validate: Verify that the data has only one codestream box. Note: Applicable to only JP2-formatted data.

2 Remove: Remove all codestreams after the first.

3 Replace: N/A

4 External Filtering Required: N/A

5 Review: Present each codestream image for human review. (In the case of JPX-formatted files, this may be the only option since it is legitimate to have multiple codestreams in a single file.)

REFERENCE:

See ISO 15444 Part 1, Annex I.5.4.

JPEG2000:10: **Resolution Levels**

DESCRIPTION:

The JPEG 2000 coding process uses progressive resolution encoding, which allows an image to be displayed at a lower resolution using a subset of data bits. Using more data bits increases the image's resolution. This is possible because the image data is organized by resolution. Each JPEG 2000 image has a base resolution level denoted R_0. All subsequent resolution levels are composed of data that augments R_0 so that the augmented image has increased quality.

CONCERNS:

The progressive resolution capability poses a data disclosure risk because certain levels of resolution may reveal sensitive data. It is also possible that the existence of image data at a given resolution level discloses sensitive information about the capabilities of an image capture device.

PRODUCT: ANY JPEG 2000 IMAGE FILE

LOCATION:

Codestream of any JPEG 2000 file.

RECOMMENDATIONS:

1 Validate: N/A

2 Remove: Remove all data that exists beyond a given resolution level by determining the progression order and removing only those packets that are associated with the image resolution.

3 Replace: Re-encode the codestream in a lossy manner to reduce overall quality of the image, thus degrading the quality of the resolution levels.

4 External Filtering Required: N/A

5 Review: Pass the image data to human review to examine the image at different resolution levels.

REFERENCE:

See ISO 15444 Part 1, Annex B.12.1.

JPEG2000:11: **Steganography**

DESCRIPTION:

Image steganography commonly employs the use of hiding data in the least significant bits (LSBs) in the image data. Additionally, when reading the compressed image data, bytes from the variable length stream are pulled into the arithmetic decoder only as needed. Once the entire image is decoded, any leftover bytes in the codestream are ignored. This aspect of the format specification poses both data hiding and data attack risks.

CONCERNS:

Steganography poses a data hiding risk because it is a covert channel. The hidden data may also introduce malicious content as well.

PRODUCT: ANY JPEG 2000 IMAGE FILE

LOCATION:

This problem exists in the codestream of any JPEG 2000 file.

RECOMMENDATIONS:

1 Validate: N/A

2 Remove: N/A

3 Replace: Perform lossy compression of the image data. For imagery that requires fidelity to a very high resolution, this may not be acceptable.

4 Replace: Invert or randomly set the LSBs of each byte in each image tile. This will alter the image.

5 Replace: Restructure the image in a way that significantly changes the underlying data without affecting the visual display of the image. For example, this can be done by recoding with a different progression order and tile height/width. Use this guidance instead of 3 or 4 when the image quality must remain pristine. This method may not meet operational performance requirements due to the intensive calculations required to apply this type of transformation.

6 External Filtering Required: N/A

7 Review: N/A

REFERENCE:

N/A

5.4 ISO 15444-2 JPX Constructs

This subsection addresses issues with the boxes and marker segments defined in ISO 15444 Part 2. All constructs in this section may occur only in JPX-formatted files.

JPEG2000:12: **Reader Requirements Box**

DESCRIPTION:

The Reader Requirements box indicates the features that are present in the image file, as well as what features must be supported by a reader application so that the file's content can be used fully. This box indicates information or features such as if the data has been signed or compressed, presence of multiple codestreams, if the codestream has been fragmented, any codestream extensions, and colorspace details.

CONCERNS:

All but one of the fields in this box have a constrained length of no more than 2 bytes. The exception is the variable length Vendor Features list that should contain UUIDs. Since the UUID information may not be validated by a reader, this field can pose a potential data hiding risk.

PRODUCT: ANY JPX IMAGE FILE

LOCATION:

JPX format. This is a required box.

RECOMMENDATIONS:

1 Validate: Verify all Vendor Features entries follow the guidance in the UUID box construct.

2 Remove: Remove all Vendor Features information from the file and set the Number of Vendor Features field to a value of zero.

3 Replace: N/A

4 External Filtering Required: N/A

5 Review: N/A

REFERENCE:

See ISO 15444 Part 2, Annex M.11.1.

JPEG2000:13: **Media Data Box**

DESCRIPTION:

Media Data boxes may contain JPEG 2000 codestreams when the codestream has been fragmented. The box may also contain other media data (e.g. MPEG-4). There can be multiple boxes of this type in the file. The Fragment Table box indicates the type of data that is in the Media Data box(es).

CONCERNS:

The data inside this box is defined by ISO 15444 only in the case where the data is JPEG 2000 codestream, thus the risks are the same as all the risks associated with the codestream. When the data is not a part of the codestream, thus assumed to be some other media type, then the risks associated with the data are the same as that data format. Because the data format is independent of the JPEG 2000 format, the risks cannot be determined.

PRODUCT: ANY JPX IMAGE FILE

LOCATION:

JPX format

RECOMMENDATIONS:

1 Validate: Verify the data as codestream when it is specified as such by the Fragment Table box.

2 Remove: Remove the box if the data is not a codestream. This may break the intended viewing of the file.

3 Replace: N/A

4 External Filtering Required: If the data is not a JPEG 2000 codestream and the type can be determined, send it to a filter that can handle that type.

5 Review: N/A

REFERENCE:

See ISO 15444 Part 2, Annex M.11.9.

JPEG2000:14: **Digital Signature Box**

DESCRIPTION:

The Digital Certificate box may contain either a checksum or a digital signature, which can be applied to either the entire file or a specified range of bytes within the file. The box may exist one or more times. The box also specifies the algorithm that was used to generate the checksum/signature using the following values:

0 : checksum using the MD5 algorithm
1 : checksum using the SHA-1 algorithm
2 : digital signature using DSA
3 : digital signature using RSA with MD5
4 : digital signature using RSA with SHA-1
5 : Cryptographic Message Syntax

CONCERNS:

Although a conforming JPX reader should correctly interpret the digital signature box, it may not have any key management support to actually validate the contents of the box, thus it may just ignore the box altogether. This poses a potential data disclosure risk in that the origin and dissemination of the image may not be able to be controlled. In addition, if the box is ignored then there may also be a possible data hiding risk.

PRODUCT: ANY JPEG 2000 IMAGE FILE

LOCATION:

JPX format

RECOMMENDATIONS:

1 Validate: Verify that the checksum(s)/signature(s) is (are) valid and that the payload has not been modified.

2 Validate: When the box indicates that a digitial signature was used (versus a checksum), verify that the digital signature is on an approved list.

3 Remove: Remove the box. This results in loss of authenticity.

4 Replace: N/A

5 External Filtering Required: Send the entire file (or specified range of bytes) to an application that can perform the checksum and digital signature validation.

6 Review: N/A

REFERENCE:

See ISO 15444 Part 2, Annex M.11.17 and Part 8, Chapter 5.8.3.3.

JPEG2000:15: **Binary Filter Box**

DESCRIPTION:

The Binary Filter box is a mechanism used to store either compressed non-image data (thus decreasing the overall size of the file) or encoded data (for security purposes). The box contains a UUID indicating whether GZIP compression or Data Encryption Standard (DES) encryption was used, followed by a binary blob of transformed data. Since the box is optional, the box shall not contain any data needed to render the image data. The box may also appear more than one time in the file. One use for the Binary Filter box is to store large amounts of XML data in a compressed format.

CONCERNS:

This poses a data hiding risk and a data attack risk because the contents of the transformed data cannot be determined without decompressing or decrypting it. Readers that do not process this box may ignore the box altogether, which poses an additional data hiding risk.

PRODUCT: ANY JPX IMAGE FILE

LOCATION:

JPX format

RECOMMENDATIONS:

1 Validate: Verify that the UUID field contains one of the two specified values (GZIP or DES).

2 Validate: Verify if the data can successfully be decompressed or decrypted.

3 Remove: Remove the box.

4 Replace: N/A

5 External Filtering Required: If the data is compressed or if the format of the decrypted data can be determined, then send the data to a filter than can handle its format.

6 Review: N/A

REFERENCE:

See ISO 15444 Part 2, Annex M.11.14.

JPEG2000:16: **MPEG-7 Binary Box**

DESCRIPTION:

This box contains MPEG-7 Binary Moving Picture Expert Group (MPEG) format for XML (BiM). BiM is a binary encoding scheme for XML data. It was originally part of the MPEG-7 standard but it can be used for any XML metadata. The format of this data is specified in ISO 15938 (Multimedia Content Description Interface), which specifies the language and description format for describing multimedia. It specifies the use of encoded XML (aka BiM) to store these descriptions.

The use of this box within JPEG 2000 is to support Motion JPEG 2000 (ISO 15444 Part 3). In order to maintain compatibility with MPEG, Motion JPEG 2000 uses the MPEG-4 data format, which in turn uses MPEG-7 for describing the multimedia. Since Motion JPEG 2000 is not within the scope of this ISG, the BiM format was not further explored.

CONCERNS:

This poses a data hiding and data disclosure because the box may be ignored by readers that do not support Motion JPEG 2000.

PRODUCT: ANY JPX IMAGE FILE

LOCATION:

JPX / MJ2 / MJP2 formats

RECOMMENDATIONS:

1 Validate: N/A

2 Remove: Remove the box.

3 Replace: N/A

4 External Filtering Required: Send the box data to a filter that can handle MPEG-7 data.

5 Review: N/A

REFERENCE:

See ISO 15444 Part 2, Annex M.11.19. Also see ISO 15938.

JPEG2000:17: **Free Box**

DESCRIPTION:

Free boxes specify sections of the file that are not currently used and may be overwritten when editing the file using some application. According to the ISO specification, the contents are undefined and meaningless and shall be ignored by reader applications.

CONCERNS:

These boxes pose a data hiding risk because they are ignored by readers.

PRODUCT: ANY JPX IMAGE FILE

LOCATION:

JPX format

RECOMMENDATIONS:

1 Validate: N/A

2 Remove: Remove all Free boxes.

3 Replace: N/A

4 External Filtering Required: N/A

5 Review: N/A

REFERENCE:

See ISO 15444 Part 2, Annex M.11.20.

JPEG2000:18: **XML Extended Metadata Box**

DESCRIPTION:

The XML box may contain information that can be used for describing extended metadata associated with the image. See Appendix C for the types of metadata that can be included.

CONCERNS:

The actual metadata contained within the XML data may contain sensitive information thus pose a data disclosure risk. Additionally, there are XML schemas provided for each extended metadata type that are used to validate the extended metadata. These schemas are loose in the sense that they support input flexibility, which may pose data hiding and data attack risks as described in [6].

PRODUCT: ANY JPEG 2000 IMAGE FILE

LOCATION:

JPX format

RECOMMENDATIONS:

1 Validate: Verify the data against the corresponding schema as defined in ISO 15444-2 Annex N. (If possible, replace the provided extended metadata schemas with schemas that are more secure prior to validation of the XML instances.)

2 Remove: Remove all XML boxes.

3 Replace: N/A

4 External Filtering Required: Pass the contents of each XML box to a filter that can process XML.

5 Review: N/A

REFERENCE:
ISO 15444 Part 2, Annex M.11.18 and Annex N.

“Security Guidance for the use of XML Schema 1.0/1.1 and RELAX NG”, NSA Cross Domain Products and Technologies Branch [6]

JPEG2000:19: **Multiple JPX Codestreams**

DESCRIPTION:
The JPX format allows for multiple codestreams to be contained in the same file. The codestreams can either be independent of one another or may be a single codestream fragmented into multiple pieces. The Reader Requirements box will indicate if multiple codestreams are present.

CONCERNS:
Each codestream is susceptible to the risks described in Section 5.3. Thus the recommended actions are the same.

PRODUCT: ANY JPX IMAGE FILE

LOCATION:
JPX format

RECOMMENDATIONS:
See recommendation listings in Section 5.3.

REFERENCE:
See ISO 15444 Part 2

5.5 ISO 15444-8 JPSEC Constructs

JPEG2000:20: JPSEC Tools

DESCRIPTION:

The intent of ISO 15444 Part 8 JPEG-200 Security (JPSEC) is to standardize tools and solutions that ensure the security of transaction, protection of contents (IPR), and protection of technologies (IP), and to allow applications to generate, consume, and exchange JPEG 2000 Secured bitstreams. JPSEC tools can either be normative or non-normative. A normative tool is one that uses a predefined tool template for decryption, authentication, or hashing. A non-normative tool is one that is either a user-defined application or specified by the JPSEC registration authority. The JPSEC codestream can be protected via one or more of these tools. In addition, each tool can be specified to only apply to a specific coverage area of the image (i.e., “Zone Of Influence (ZOI)”). Note: Use of ZOI is how resolution levels can be access controlled.

The specification of JPSEC tools is contained within one or more SEC marker segments. The syntax for the SEC marker segment is:

- SEC marker code: 0xFF65
- Length of marker segment (Lsec)
- Index of this marker segment relative to any other SEC marker segments present (Zsec)
- Codestream security parameters (only present in first SEC marker segment) (Psec)
- Tool parameters (Tool)

Each Tool described in the SEC segment has an ID associated with it, which defines the JPSEC protection mechanism that is applied to the file.

CONCERNS:

Use of a confidentiality tool for image encryption will make it impossible to validate the image data unless the decryption password is available, thus the file could be susceptible to data hiding and disclosure risks. Use of an authentication tool could make the file susceptible to data attack in that the tool could specify a URI to use to obtain the necessary certificate or secret key information. In addition, the certificate data itself could be an avenue for data attack, disclosure, and attack risks.

PRODUCT: ANY JPSEC CODESTREAM

LOCATION:

SEC segments appear in JPSEC-enabled codestreams.

RECOMMENDATIONS:

1 Validate: If the tool ID is the value “2” (Authentication Template), then validate the data using the appropriate authentication method.

2 Validate: If the tool ID is the value “3” then rehash the specified data and verify that the hashes match.

3 Remove: If the tool ID is the value “1” (Decryption Template) and it is only specified for a specific ZOI, then remove the bytes associated with that ZOI and remove the associated SEC data.

4 Replace: N/A

5 External Filtering Required: If the tool ID is the value “2” and indicates the use of a X.509 certificate, then pass the certificate to a filter capable of handling that data format.

6 External Filtering Required: If the tool ID is the value “2” and indicates the use of a URI, then pass the data to a filter capable of text.

7 Review: N/A

8 Reject: If the tool ID is the value “1” (Decryption Template), then reject the file since the data is encrypted.

REFERENCE:
ISO 15444 Part 8

JPEG2000:21: Byte Aligned Segments (BAS)

DESCRIPTION:

To provide extensible signaling to support different modes of security services, JPSEC uses a variable length data structure referred to as a byte aligned segment (BAS). A BAS is a sequence of one or more BAS bytes. When the most significant bit (MSB) of the BAS is a 1, then another BAS follows. When the MSB is 0 then there is no following BAS. In addition, one type of BAS, the Range BAS (RBAS) is able to extend the number of bits used to represent a value. Several values in JPSEC are specified using BAS or RBAS such as the Zsec and Tool fields of the SEC marker segment.

CONCERNS:

The use of BAS and RBAS pose a data hiding risk because RBAS can hold an arbitrary amount of data. The use of BAS/RBAS may also pose a data attack risk because if a long sequence of BAS are used or a RBAS overly extends the number of bits then the large amount of data could potentially crash the reader application.

PRODUCT: ANY JPSEC CODESTREAM

LOCATION:
BAS segments appear in JPSEC-enabled codestreams.

RECOMMENDATIONS:

1 Validate: Verify that the data in each BAS adheres to the format for that specific field.

2 Remove: Remove BAS after a certain predefined threshold.

3 Replace: N/A

4 External Filtering Required: N/A

5 Review: N/A

REFERENCE:
ISO 15444 Part 8, Chapter 5.4

6. TABLE OF DOCUMENT CONSTRUCTS

This section contains an index of all the constructs that appear in this document.

APPENDIX A. REFERENCED DOCUMENTS

Referenced Documents

The following publications were referenced in this document or used to prepare the document.

1. "Information technology — JPEG 2000 image coding system: Core coding system", ISO/IEC 15444 Part 1

2. "Information technology — JPEG 2000 image coding system: Extensions", ISO/IEC 15444 Part 2

3. "Information technology — JPEG 2000 image coding system — Part 8: Secure JPEG 2000", ISO/IEC 15444 Part 8

4. "Information technology — Multimedia content description interface", ISO/IEC 15938

5. "Image technology color management – Architecture, profile format and data structure", ISO/IEC 15076 Part 1

6. "Security Guidance for the use of XML Schema 1.0/1.1 and RELAX NG", NSA Cross Domain Products and Technologies Branch, 11May 2011, version 1.0.1

7. "Image technology colour management -- Architecture, profile format and data structure -- Part 1", ISO 15076-1:2010

8. "BIIF Profile for JPEG 2000 Version 01.10", ISO/IEC BIFF Profile BPJ2K01.10

9. "JPEG 2000 Profile for the National Digital Newspaper Program", XEROX Global Services (2006)

APPENDIX B.   SUMMARY OF RISKS TABLE

Table B-1 Summary Of Risks

JPEG 2000 Feature	Attack	Hiding	Disclosure
Box General (Identifier, Length, Order)	1	1	0
Marker Segments	1	1	0
Signature Box	0	0	0
UUID Box	1	1	0
Data Entry URL Box	1	1	1
ICC Profile	1	1	1
Image Data	1	1	0
Multiple Codestreams	0	1	0
Resolution Levels	0	1	1
LSB Steganography	0	1	0
Reader Requirements Box	0	1	0
Media Data Box	1	1	1
Digital Signature Box	1	1	0
Binary Filter Box	1	1	0
MPEG-7 Binary Box	0	1	1
Free Box	0	1	0
XML Extended Metadata Box	1	1	1
Multiple JPX Codestreams	1	1	1
JPSEC Tools	1	1	1
Byte Aligned Segments	1	1	0
Total	**13**	**19**	**8**

APPENDIX C. XML METADATA DESCRIPTION

Image creation metadata

The root element, IMAGE_CREATION, has two attributes and a sequence of five elements that are all optional. The first attribute is a timestamp and the second attribute tells the language used.

The first element, GENERAL_CREATION_INFO, specifies generic information about how the image was created. This element is defined as a sequence of six elements that are all optional: the creation time, the image source, the type of scene, who created the image, what organization conducted the image capture, and which person conducted the image capture.

The second element, CAMERA_CAPTURE, specifies the properties of the camera that captured the image. This element is defined as a sequence of six elements that are all optional: the camera info, the capturing software info, the capturing lens info, technical data about the capture device, the capture-time camera settings, and technical data about the accessories used with the camera at capture time.

The third element, SCANNER_CAPTURE, provides information about the scanner used to obtain the image. This element is defined as a sequence of three elements that are all optional: the scanner info, the capturing software info, and the capture-time scanner settings.

The fourth element, SOFTWARE_CREATION, provides information about the software used to create the image. This element is defined as a sequence of one element that is required: data about the software that created the image.

The fifth element CAPTURED_ITEM describes the type of medium used to capture the original image. This element is defined as a sequence of two elements that can appear any number of times in any order: information about a reflection print and information about a film.

Content description metadata

The root element, CONTENT_DESCRIPTION, has two attributes and a sequence of twelve elements. The first four elements may at most one time, but may not be present. The next seven elements each may occur an unbounded number of times but must remain in element order. The last element may occur once or may not be present at all.

The first element, GROUP_CAPTION, describes the subject or purpose of the image. This element is defined as an internationalized string. The second element CAPTION is described the same way; in theory, they are meant to describe different concepts, but this is not clarified in ISO15444 Part 2.

The third element, CAPTURE_TIME, indicates when the image was originally generated. This element is defined as a complex data structure that provides information about the date and time.

The fourth element, LOCATION, describes the physical location where the image was captured. This element is defined as a complex object that describes the location.

The fifth element, PERSON, specifies a person who appears in the image. This element is defined as a sequence of three elements describing the person's position in the image, the location of the image itself, and any key properties of the person.

The sixth element, THING, specifies a tangible object depicted in the image. This element is defined as a sequence of elements describing the name of the object depicted in the image, a comment about the object, its position within the image, the location of the image, properties about the object, and a list of objects contained within the main object, each element of which obeys the same schema as the main object.

The seventh element, ORGANIZATION, specifies an organization with physical property that is displayed in the image. This element is defined as a sequence of three elements describing the organization's position within the image, the location of the image, and properties about the organization.

The eighth element, EVENT, specifies an event captured in the image. This element is defined as a sequence of elements describing the event type, a description of the event, the location of the event, the date and time of the event, the duration of the event, a comment about the event, a list of the participants in the event, a list of relationships with other events, and a list of sub-events defined directly or by reference.

The ninth element, AUDIO, specifies audio streams associated with an image. This element is defined as a sequence of elements describing a reference to an audio stream, the format of the audio stream, the MIME type of the audio stream, a description of the audio stream, and a comment about the audio stream.

The tenth element, PROPERTY, describes a property of the image or of an object within the image. This element is defined as a sequence of elements describing the property name, the property value, a comment about the property, and a list of sub-properties.

The eleventh element, DICTIONARY, specifies a dictionary of terms associated with a property. This element is defined as a sequence of elements describing the dictionary name and a comment about the dictionary.

The twelfth element, COMMENT, specifies miscellaneous information defined by an application or by a user. This element is defined as an internationalized string with a timestamp.

History metadata

The root element, HISTORY, has two attributes and a sequence of three elements that are all optional. The first attribute is a timestamp and the second attribute tells the language used.

The first element, PROCESSING_SUMMARY, specifies the list of operations previously applied to the image during its workflow; for example, it indicates whether an image has been cropped or transformed. This element is defined as a sequence of elements whose existence or absence functions as a flag.

The second element, IMAGE_PROCESSING_HINTS, specifies the list of operations previously applied during editing of the image. It describes the same set of possible operations as the PROCESSING_SUMMARY element, but it also describes which application performed each operation and allows for data specific to that operation to be recorded. For example, it is possible to indicate in this section that an image was cropped by MS Paint and that the bottom 100 rows were deleted. This element is defined as a sequence of tuples, each of which provides the descriptive details of the editing application, the operation that was performed, and any available verbose description of the operation.

The third element, METADATA, specifies a previous version of the metadata that may describe portions of the image that were deleted or cropped. This element is defined as a sequence of five elements, each of which points to one of the five metadata formats, thus completely encompassing a previous version of the extended image metadata.

Intellectual property rights metadata

The root element IPR has two attributes and a sequence of seven elements that are all optional. The first attribute is a timestamp and the second attribute tells the language used.

The first element, IPR_NAMES, specifies the names of people or organizations associated with the image. This element is defined as an unbounded list of name elements that are people, organizations, or references to people or organizations.

The second element, IPR_DESCRIPTION, provides a description of the image content. This element is defined as a sequence of four elements that are all optional: the image title, a description of the image content, a caption for the image, and a copyright notice.

The third element, IPR_DATES, specifies dates associated with the IPR in the image. This element is defined as an unbounded sequence of dates.

The fourth element, IPR_EXPLOITATION, specifies the protections, restrictions, and obligations associated with an image. This element is defined as a sequence of four elements that are all optional: an indication of the type of IPR protection, the usage restrictions, the associated obligations for use, and an indication of what IPR management system is used.

The fifth element, IPR_IDENTIFICATION, specifies a link to additional information. This element is defined as a sequence of two elements that are both optional: a generic IPR identifier and a "license plate" containing a set of registration values that make up a globally unique identifier (GUID).

The sixth element, IPR_CONTACT_POINT, specifies the point of contact for the right holder. This element is defined as a single choice among three contact types: a person, an organization, or a reference to a person or organization.

The seventh element, IPR_HISTORY, contains an unbounded sequence of previous IPR elements.

Image identifier metadata

The root element IMAGE_ID has a sequence of two elements. The first element is a unique identifier with the string type. The second element is a URI used to specify the type to which the unique identifier string conforms.