Inspection and Sanitization Guidance for Moving Picture Experts Group Standards (MPEG-2 with H.264/Advanced Video Coding (AVC))

Version 1.0

19 April 2016

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
4/19/2016	1.0	Initial Release
12/13/2017	1.0	Updated Contact information, IAC Logo, Cited Trademarks and Copyrights, Expanded Acronyms, and added Legal Disclaimer

DISCLAIMER OF WARRANTIES AND ENDORSEMENT

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

The *Inspection and Sanitization Guidance for Moving Picture Experts Group (MPEG) Standards (MPEG-2 with H.264 Advanced Video Coding (AVC))* provides guidelines and specifications for developing file inspection and sanitization software for MPEG-2 video files, which are formally defined by the International Standards Organization (ISO). MPEG-2 is a standard for the generic coding of moving pictures and associated audio information. It describes a combination of lossy compression methods for storage and transmission of audio and video using available storage media and transmission bandwidth. It includes an analysis of the issues with the H.264 advanced video coding, part 10 of MPEG-4. It also provides for inclusion of metadata such as Key-Length-Value, which can be obtained from unmanned aerial vehicle platforms capturing motion imagery. The MPEG-2 standard contains detail down to the bit level, fields that include metadata, conditional information, and variable length content require inspection to ensure data is not hidden or unintentionally disclosed. Given the typically large amount of data contained in MPEG-2 files, inspection and sanitization are critical to ensure that all content within the files can be displayed to end users and that files have no malicious content.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. SCOPE

1.1 Purpose

This document provides guidance for the development of a inspection and sanitization filters for Moving Picture Experts Group (MPEG)-2 files. It analyzes the various elements contained within MPEG-2 files, describes how these elements may contain hidden sensitive data or attempts to exploit a system, and then discusses the risks posed by data hiding, data attack, and data disclosure. The document provides recommendations and mitigations to ensure that MPEG-2 files are safe and conform to the specifications.

The intended audience of this document includes system engineers, designers, software developers, and testers who work on file inspection and sanitization applications that involve processing MPEG-2 files.

1.2 Introduction

File types that act as containers and store a variety of different data introduce a significant amount of risk. Hidden data may be present where information can be stored within the file format and never appear to the end user. Complex file types can often lead to a vulnerable application; this requires the inspection and possible sanitization of these files for correctness.

MPEG-2 files contain audio, video, user data (also called metadata), and other binary data. In the unmanned aerial vehicle (UAV) community, metadata is often embedded in surveillance content to capture scenario parameters such as platform speed and heading. The video content may be compressed by H.264/AVC. Systems that create MPEG-2 files typically input audio, video, and metadata content and multiplex them into a single file. The MPEG-2 file can then be shared or transmitted to other users who can extract the content needed for display.

1.3 Background

The International Standards Organization (ISO) and the International Electrotechnical Commission (IEC) formed the MPEG to set standards for audio and video compression and transmission. The first standard, MPEG-1, was approved in 1993 and was designed to allow encoding of moving pictures and sound into the bitrate of a compact disc for use as low-quality video on DVD video [1]. MPEC-1 includes the popular MPEG-1 Audio Layer III audio compression format.

The MPEG-2 standard, approved in 1995, was designed for video, audio, and transport of broadcast-quality television [2]. The standard was considerably broader in scope and

of wider appeal, supporting interlacing and high definition. MPEG-2 became the compression scheme for over-the-air digital television, digital satellite television (TV) services, and digital cable TV signals. Subsequent standards include MPEG-4 for coding of audio-visual objects [3], MPEG-7 for multimedia description interface [4] and MPEG-21 for multimedia framework [5]. Except for MPEG-4 part 10, better known as H.264/AVC (advanced video coding), this document does not address these subsequent standards.

Given the widespread use of the MPEG-2 standard, the UAV community has adopted MPEG-2 to share and transport surveillance video and audio as well as metadata associated with vehicle parameters such as heading and speed. The most common method of incorporating metadata is Key-Length-Value (KLV), which is an effective, largely self-documenting format. This ISG is not restricted to just the UAV community, it provides guidance for all users of MPEG-2 and H.264.

1.4 Document Organization

Table 1-1 summarizes the organization of this document.

Table 1-1 Document Organization

Section	Description
Section 1: Scope	This section describes the purpose, introduction, background, organization, actions, and limitations related to this document.
Section 2: Constructs and Taxonomy	This section describes the constructs and taxonomy used throughout this document.
Section 3: Overview	This section describes the structure of MPEG-2.
Section 4: MPEG-2 Constructs	This section describes the MPEG-2 constructs that can present risks and the options for mitigation.
Section 5: Acronyms	This section lists the acronyms in this document.
Section 6: Referenced Documents	This section lists the sources used to prepare this document.
Section 7: Summary of Risks	This section maps each construct to the corresponding specifications and risks.

1.5 Actions

Each construct description lists recommended actions for handling the construct when processing a message. Generally, inspection and sanitization programs perform one or more of the following actions on a construct: *Validate*, *Remove*, *Replace*, *External Filtering Required*, *Review*, or *Reject*.

The recommendation section in the description of each construct lists each applicable action along with an explanation specific to the construct. Not all actions are applicable or appropriate for every context. Implementers are not expected to take all the actions for a given risk; instead, they should determine which action—or perhaps actions—applies best to their context. The implementer must define the criteria used to determine which action is "best" and the specific method used to execute the action.

Recommendations such as *remove* and *replace* may alter the integrity of MPEG-2 files. It is important to address these issues in order to retain functionality.

NOTE

The recommendations in this document are brief explanations rather than a How-To Guide. Readers should refer to the construct description or official documentation for additional details.

Table 1-2 summarizes the recommendation actions:

Table 1-2 Recommendation Actions

Recommendation Action	Comments
Validate	Verify the data structure's integrity, which may include integrity checks on other components in the data structure. (This is almost always a recommended action.)
Replace	Replace the data structure or one or more of its elements with values that alleviate the risk (e.g., replace a username with a non-identifying, harmless value or substitute a common name for all authors).
Remove	Remove the data structure or one or more of its elements and any other affected parts.
External Filtering Required	Note the data type and pass the data to an external action for handling that data type (e.g., extract text and pass it to a dirty word search).
Review	Present the data structure or its constructs to a human to review. (This is almost always recommended if the object being inspected can be revised by a human.)
Reject	Reject the file.

NOTE

No recommendations for logging all actions and found data are included here because all activity logging in an inspection application should occur "at an appropriate level" and presented in a form that a human can analyze further (e.g., the audit information may be stored in any format, but must be parsable and provide enough information to address the issue when presented to a human.)

1.6 Document Limitations

1.6.1 Covert Channel Analysis

It is impossible to identify all available covert channels, whether in a file format or a communication protocol. Because the channels contain free-form text, searching for hidden data becomes increasingly difficult. No tool can possibly analyze every channel, so this document highlights the highest risk areas to reduce or eliminate data spills and malicious content.

To clarify the context, a few examples of data tampering must be reviewed first. Steganography embeds a hidden, imperceptible message within an innocuous image or paragraph of a media or text file. A file filter should use other filters that specialize in steganography to handle embedded content such as text, images, video, and audio.

Watermarking techniques embed a pseudo-random pattern into each video frame so that the pattern is imperceptible to a human viewer. Watermarks imprinted with the dual-tree complex wavelet transform (DT CWT) of Kingsbury are robust to many types of attack, including MPEG-4 H.264/AVC compression itself, rotation and cropping (due to rerecording of a large-screen movie with an HD video camera), frame-dropping, and changes in frame-rate and frame-resolution [24]. Though usually used to prevent film piracy, watermarking methods can also be used to hide data in videos. Watermarking may occur outside or inside the MPEG compression format. Blind (non-cooperative) watermark recovery algorithms have recently become available to detect and remove this kind of hidden data [32].

Image forgery, e.g., video content (graphics) editing, manipulation or retouching, is another kind of data tampering that takes place in a functional layer above the MPEG compression format. A well-known example is the "copy-move" or "grafting" technique to remove people or objects from a scene. This type of forgery is accomplished by painting over subimages with a copy of a patch of background material, which covers people and/or objects while blending in with the scene. Video frames in a database that have been overlaid with computed local feature elements, such as from Scale Invariant Feature Transform (SIFT) for frame-content matching, require further modification of those features to conceal the copy-move operation [25].

While methods to detect copy-move tampering are being developed [26], image manipulation designed to mislead the viewer does not involve the MPEG compression formats. This document exposes methods of data tampering that exploit the MPEG compression format itself, as exemplified by recent research [27–32].

1.6.2 Scope

This document covers the MPEG-2 standard (H.262), including video coding, audio coding, and KLV metadata coding. It primarily covers the transport stream format,

used for streaming data over lossy media. The Program Stream is covered in Section 3 but only at a high level. The document also addresses the widely used H.264 AVC from MPEG-4 part 10, but does not examine next-generation standards such as H.265 high efficiency video coding (HEVC) in detail. This document also does not contain a general discussion of forensic methods used to detect the many forms of video/audio data tampering that may occur outside (before or after) the MPEG codec compresses the data.

2. CONSTRUCTS AND TAXONOMY

2.1 Constructs

This document describes many, but not all, of the constructs used in MPEG-2 files, and therefore should not be viewed as a complete reference. Filter developers should consult the official specifications in addition to this document for the full context. For each construct presented, the descriptions contain the following sections:

- **Overview:** An explanation of the construct with examples.
- **Risks and Recommendations:** An explanation of potential risks posed by the construct with corresponding mitigation strategies.
- **Product:** The specifications in which the construct is found.
- **Location:** A textual description of where to find the construct.

2.2 Taxonomy

Table 2-1 describes the terms that appear in this document:

Table 2-1 Document Taxonomy

Term	Definition
Construct	An object that represents some form of information or data in the hierarchy of an MPEG file
Inspection and Sanitization	Activities for processing files and protocols to prevent inadvertent data leakage, data exfiltration, and malicious data or code transmission
ISG (Inspection and Sanitization Guidance)	A document (such as this) that details a file format or protocol and inspection and sanitization activities for constructs within it
Recommendations	A series of actions for handling a construct when performing inspection and sanitization activities

3. MPEG-2 AND H.264 AVC OVERVIEW

The International Standards Organization (ISO) and the International Electrotechnical Commission (IEC) formed the Moving Picture Experts Group (MPEG) as a working group to set standards for audio and video compression and transmission. The MPEG-2 standard is widely used as the format of digital television signals that are broadcast by terrestrial (over-the-air), cable, and direct broadcast satellite TV systems. It also specifies the format of movies and other programs that are distributed on DVD and similar discs.

The MPEG compression methodology is considered *asymmetric* because the encoder is more complex than the decoder. The working group's approach to standardization is novel because it is not the encoder that is standardized, but rather the way a decoder interprets the bitstream. A decoder that can properly interpret the bitstream is termed *compliant.* The advantage of standardizing the decoder is that it functions regardless of the encoder used.

The MPEG standards specify very few requirements regarding structure and operation of the encoder; implementers can supply encoders using proprietary algorithms. This permits competition between different encoder designs, resulting in improved performance and greater user choice because encoders of different levels of cost and complexity can exist.

The Joint Video Team (JVT) maintains a reference encoder-decoder, which implements the H.264/AVC video standard [13]. This is part 10 of MPEG-4, for AVC [3]. According to the JVT website, H.264/AVC "now accounts for roughly half of all communication network traffic world-wide (and over 80% of Internet video)." Therefore, this document addresses H.264/AVC in as well as the MPEG-2 H.262 standard.

Note that H.264/AVC is currently being superseded by High Efficiency Video Coding (HEVC/H.265) [9-10]. This new standard improves compression efficiency and reconstruction quality by using more flexible and complex encoding methods than H.264 [10-11]. According to the HEVC website [9]:

> HEVC is the current joint video coding standardization project of the ITU-T [(International Telecommunication Union - Telecom)] Video Coding Experts Group (ITU-T Q.6/SG 16) and ISO/IEC Moving Picture Experts Group (ISO/IEC JTC 1/SC 29/WG 11). The Joint Collaborative Team on Video Coding (JCT-VC) has been established [in 2013] to work on this project. The Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V) has been established to work on 3D video coding extensions of HEVC and other video coding standards.

The HEVC/H.265 standard was formally published in 2013, but is beyond the scope of this ISG.

3.1 MPEG-2 Part 1: Systems Layer (H.222.0)

The MPEG-2 Standard is published in several parts, as described in ISO/IEC 13818 [2]. Part 1 specifies the Systems Layer of MPEG-2. It defines a multiplexed structure for combining audio, video, and metadata (also known as user data), and the means of representing the timing information needed to replay synchronized sequences in real time. MPEG-2 introduces the concept of a program. A program is a single source of video and/or audio that a user can view. For example, in a cable television system, a television channel represents a program of content. MPEG-2 also introduces the term Elementary Stream (ES). Each single source of video or audio is considered an ES. A program consists of multiple ESs. The MPEG-2 Part 1 standard introduces a Packetized Elementary Stream (PES): a single ES broken up into packets. Every MPEG-2 file or data stream is a series of packets. Some belong to a particular PES, while other packets are used to define the structure of the data and to list each ES present in the data stream. A PES may contain a video or audio source, or contain metadata that pertains to the video source. The Systems Layer defines two distinct, but related container formats: the Transport Stream (TS) and Program Stream (PS). A TS is one or more packetized ESs of data; more specifically, a PES is defined by a specific numerical value known as a packet identifier, or PID. The TS is designed to carry digital content over lossy media. The TS implements error correction and synchronization to handle its transmission. Some example video formats that use TS are Digital Video Broadcasting (DVB), Advanced Television Systems Committee (ATSC), and Internet Protocol Television (IPTV). The PS is a container format designed for reliable, file-based media, e.g., hard disk drives, optical discs (e.g. DVD and HD DVD), flash memory, etc. Blu-ray™[1] is a unique exception to the use of PS for these types of media; it uses MPEG-2 TS. Blu-ray makers decided it was better to use TS because it can provide multiple streams of data (picture in picture, multiple videos, etc.). Unlike TS, a PS can contain only one program worth of content or video and audio. This is used with older movies and regular DVD-based media. Both DVD-Video and Blu-ray implement a container format that is based on several of the MPEG specifications. There are separate DVD-Video and Blu-ray specifications that support extensions to MPEG.

Modern systems using TS allow for multiple channels of simultaneous content encapsulated within one stream of data. A high level diagram of this concept with two channels of information is shown in Figure 3-1 below.

1 Blu ray™ is a trademark of the Blu-ray Disc Association

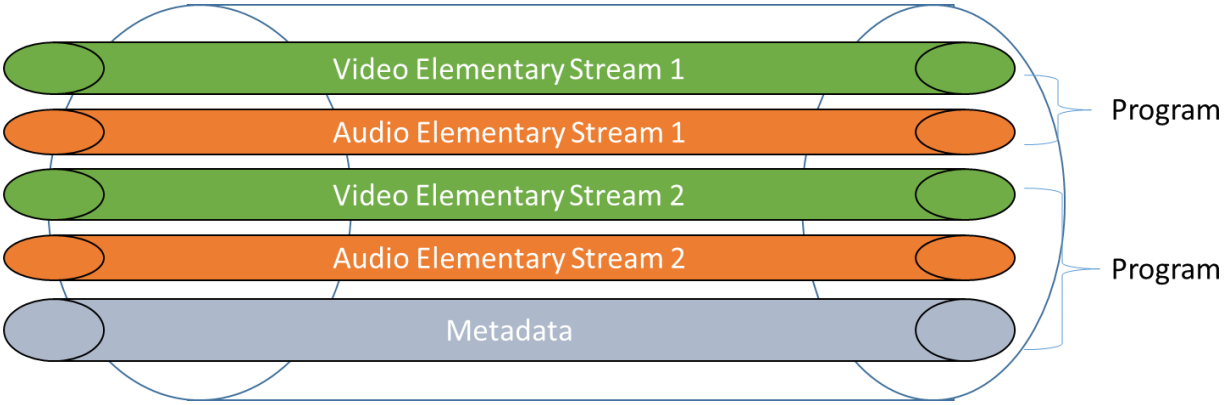


Figure 3-1. High Level Diagram of MPEG-2 TS Systems Layer

When a MPEG-2 TS video file is packetized from each ES (forming several PESs), it resembles the image below in Figure 3-2, with each block being an MPEG-2 TS packet. Packets may not always need to appear in the exact order shown in the image; null packets (an empty packet with no data) may even exist in between packets. Figure 3-2 demonstrates how a sample file would appear on disk. The MPEG-2 TS header packets are a number of packets and will be discussed in more detail later. These packets are used to define each program and each ES in the programs in the data stream.

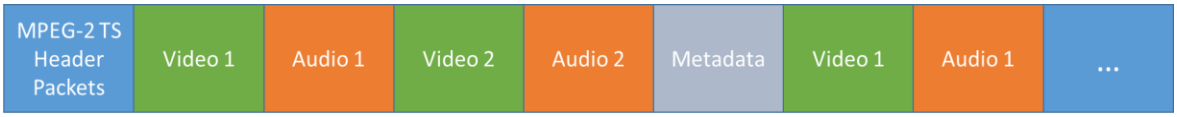


Figure 3-2. MPEG-2 TS Packetization

Figure 3-3 illustrates the steps required in forming both MPEG container formats. The input video, audio, and metadata streams are provided via ESs. After encoding the streams to compress them, the ESs are then packetized into PESs. Each of the PESs consists of a stream of identified packets. The PS or TS results from combining one or more concurrent PESs with a common time base into a single stream. The differences between PS and TS are presented in the next section.

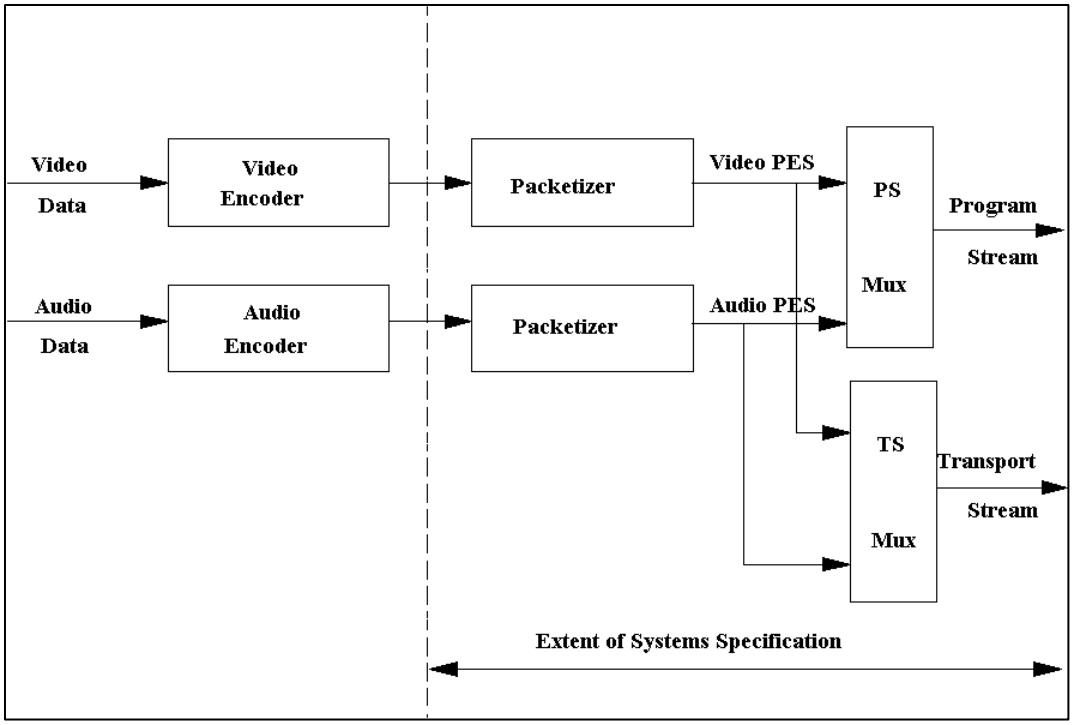


Figure 3-3. MPEG-2 Systems Layer

The TS is designed for use in environments where errors are likely, such as storage or transmission in lossy or noisy media. It combines one or more PESs with one or more independent time bases into a single stream. Multiplexed PESs that share a common time base form a program. Each of the PESs has a packet identifier (PID) that serves as a unique identifier for that stream within the TS.

Video and audio PESs consist of continuous sets of video frames and audio data samples. They are first split into packets to make them suitable for multiplexing. To create a TS, each of these PESs is stored in TS packets, which are 188 bytes long.

3.2 MPEG-2 Transport and Program Stream Formats

The MPEG-2 Systems layer integrates and synchronizes the audio and video elementary streams. A block of an audio or video stream is referred to as an Access Unit (AU). The MPEG-2 systems layer packetizes these AUs to create PESs. Each PES is split across various Transport Stream (TS) packets as shown in Figure 3-4 below. The PES structure is used in both TS and PS formats.

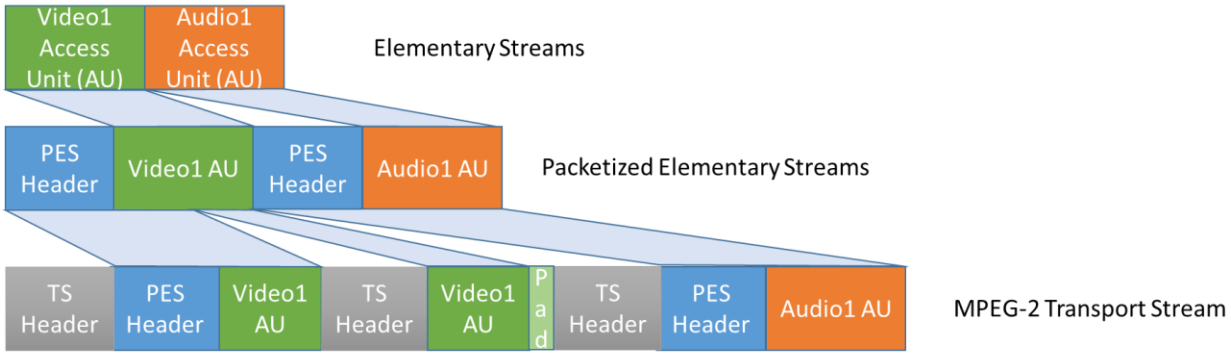


Figure 3-4 MPEG-2 TS Creation [36]

Metadata, such as KLV, is stored in a stream that accompanies the audio and video stream PESs. The stream is given a different PID but is collocated with other streams within MPEG-2 TS. These PESs contain timestamps from a system clock for synchronization. The PESs are multiplexed to form a single output stream for transmission in one of two modes: Program Stream (PS) and Transport Stream (TS). The PS is provided for error-free environments such as storage in DVDs. It is used for multiplexing PESs that share a common time-base, using long variable-length packets. This mode permits multiplexing of streams that do not necessarily share a common time-base. As noted, the TS uses small fixed-length packets (188 bytes) that make them more resilient to packet loss or damage during transmission. The video captured from most UAV platforms utilizes the TS format.

The TS packet consists of a four-byte header followed by 184 bytes shared between the optional and variable-length adaptation field (AF) and the TS packet payload. Packets belonging to the same elementary stream are identified by the packet ID (PID) in the packet header. Figure 3-5 depicts the TS packet header. Note that the shaded box represents the optional AF.

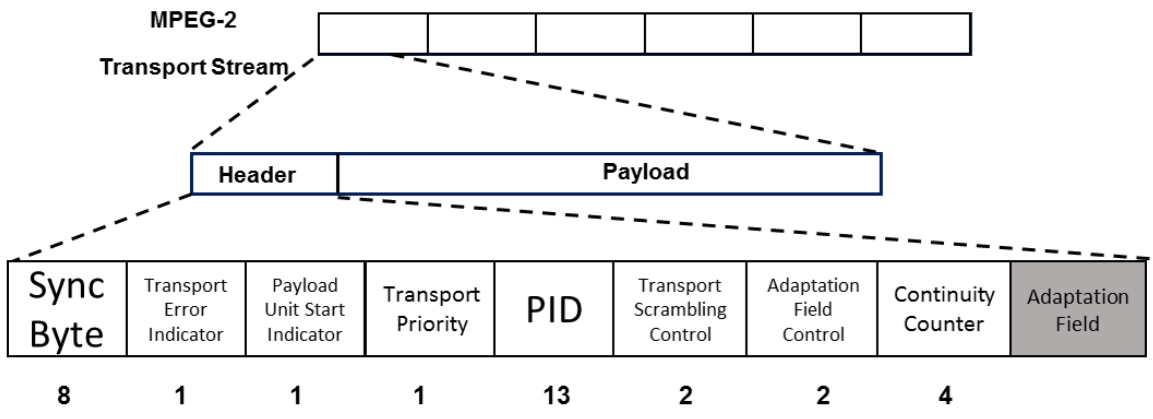


Figure 3-5. MPEG-2 Transport Stream Packet Header Format

The TS packet header contains the following fields:

- The header starts with a *Synchronization Byte (8 bits)*. This has the bit pattern 0x47 (0100 0111).
- A set of three flag bits is used to indicate how the payload should be processed.
    - The first flag indicates the packet has an uncorrectable error.
    - The second flag indicates the start of a payload.
    - The third flag indicates transport priority (1 means higher priority than other packets with the same PID).
- The flags are followed by a 13-bit PID. This is used to uniquely identify the stream to which the packet belongs, as generated by the multiplexer.
- Two Transport Scrambling Control bits may contain the values:
    - 00 – not scrambled
    - 01 – RESERVED for future use
    - 10 – scrambled with even key
    - 11 – scrambled with odd key
- Two AF Control bits may contain the values:
    - 00 – RESERVED for future use
    - 01 – payload only
    - 10 – AF only
    - 11 - AF followed by payload
- A four-bit Continuity Counter is incremented when a payload is present.

The PID allows the receiver to differentiate the stream to which each received packet belongs. Some PID values are predefined and are used to indicate various streams of control information. A packet with an unknown PID, or one with a PID not required by the receiver, is silently discarded. Table 3-1 lists some common PIDs (in hexadecimal).

Table 3-1. Common PID Values

PID (Hex)	Packet Type
0	Program Association Table (PAT)
1	Conditional Access Table
10 or FE	Program Map Table (PMT)
11	MPEG-2 video
14	AC3 audio data
258	MPEG-2 video (standard definition)
259	PCM stereo data
5DE	H.264 video (BBC HDTV)
5E1	AC3 audio (BBC HDTV)
7FF	Filler (no data)

The particular PID value of 0x1FFF is reserved to indicate a null packet. The payload of null packets should not contain meaningful data because the receiver is expected to ignore the contents. A null packet contains 184 bytes of the value 0xFF.

Finally, the optional AF contains additional information that need not be included in every TS packet. The AF is a variable length field with both mandatory and optional sub-fields. Table 3-2 lists the sub-fields that can be implemented in the AF field.

Table 3-2. Adaptation Field Sub-fields

AF Sub-field Name	Length (bits)	Description
Adaptation Field Length	8	Length in bytes of the entire AF (minus this length sub-field byte)
Discontinuity Indicator	1	Set to 1 when there is discontinuity in the packet stream.
Random Access Indicator	1	Set to 1 when the next PES packet to start in the payload contains the first byte of a video sequence header. Used to aid in random access.
Elementary Stream Priority Indicator	1	Set to 1 when the payload has a higher priority than other payloads.
PCR Flag	1	Set to 1 if the PCR field is present.
OPCR Flag	1	Set to 1 if the OPCR field is present.
Splicing point flag	1	Set to 1 if the Splice countdown field is present.
Transport private data flag	1	Set to 1 if private data is present.
Adaptation Field Extension Flag	1	Set to 1 if an extension is present.
PCR (OPTIONAL)	48 (33+6+9)	Program Clock Reference.
OPCR (OPTIONAL)	48 (33+6+9)	Original Program clock reference.
Splice countdown (OPTIONAL)	8	When positive, the number of packets of the same PID until a splicing point is encountered. When negative, this means that the slicing point was in a previous packet.
Stuffing bytes	variable	

One of the most important sub-fields in the AF is the program clock reference (PCR). The PCR is an optional 48-bit field composed of a 9-bit segment incremented at 27 MHz as well as a 33-bit segment incremented at 90 KHz. 6-bits are used for padding. The AF

uses the PCR, along with a voltage controlled oscillator, as a time reference for synchronization of the encoder and decoder clock.

The TS payload may contain structural information referred to as Program Specific Information (PSI). These data structures are found in the TS packet payload following a PES header. PSI includes the Program Association Table (PAT) and the Program Map Table (PMT). At least one PAT structure is present within the TS and is defined with the reserved Packet ID (PID) of zero as shown in Table 3-1. The PAT must exist before any multimedia content is defined and may be sent periodically throughout the stream. The PAT will inform the video player of how many programs (video/audio sources) are present in the stream and will identify the PID that is used for each PMT. Each program in MPEG-2 implements a PMT. The PMT goes into more detail and lists the PID values for each of the video, audio, or metadata ESs included in the MPEG-2 stream. If a program contains 1 video and 2 audio streams, then the PID of those 3 streams is present in the PMT. There are other PSI structures such as the Conditional Access Table (CAT) and the Network Information Table (NIT) that are not addressed in this paper as they are not required per ISO/IEC-13181-1. Both the PAT and PMT are required and are discussed in more detail in Section 4.

MPEG-2 PS is different from MPEG-2 TS because it only implements one channel or one program of multimedia content. MPEG-2 PS was used in older DVDs because it provides a single program of content (the movie). Television systems or satellite TV utilize MPEG-2 TS because it provides multiple channels or programs of content. Both MPEG-2 TS and MPEG-2 PS use a common PES structure that is split across various packets. MPEG-2 PS packets can be significantly larger than the 188 byte TS packet. The MPEG-2 PS structure is divided into multiple packs, which begins with a Pack header and is followed by variable length blocks of multiple PES packets. Figure 3-6 illustrates

how a MPEG-2 PS is divided from multiple packs down into individual PES packets; the same PES packets that are present in MPEG-2 TS.

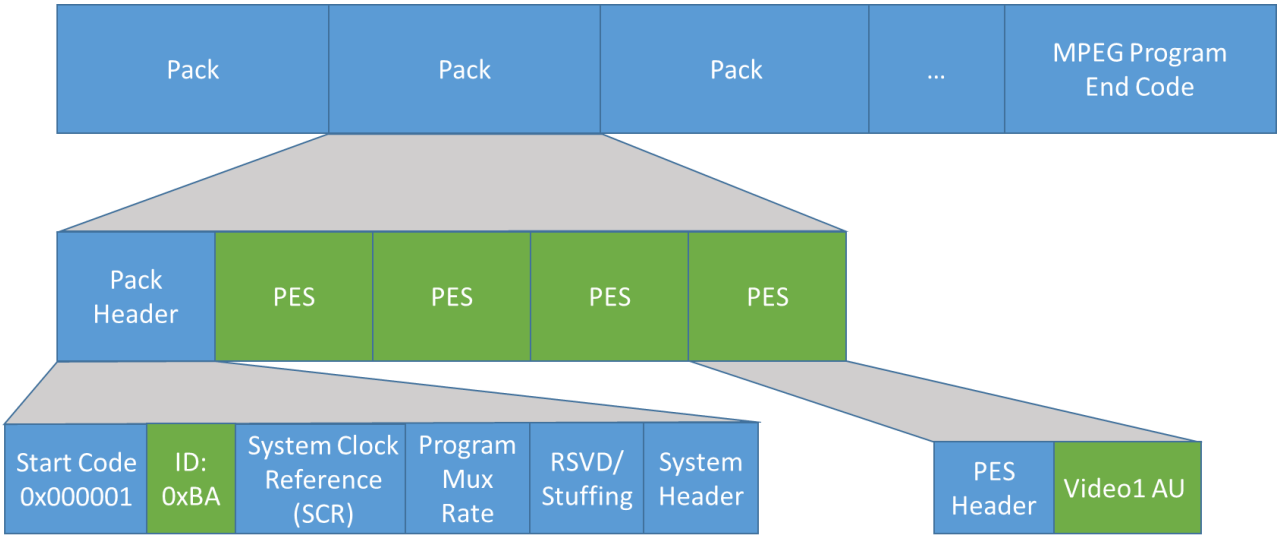


Figure 3-6. MPEG-2 PS Structure

The start of a MPEG-2 PS pack can be identified by the value 0x000001BA. The start code is 0x000001 and the Pack identifier is equal to 0xBA. The header introduces a System Clock Reference (SCR) field which is similar to the PCR field defined in MPEG-2 TS. The Pack header also provides a System header which is identified by the value 0x000001BB. This header provides a listing of the Stream IDs that are present in the PS file and is analogous to the PAT/PMT information that is present in MPEG-2 TS. Following the System Header is a grouping of PES packets similar to MPEG-2 TS. MPEG-2 PS is not covered in greater detail in this document. A filter designed to handle MPEG-2 TS could be extended to handle MPEG-2 PS by covering the structures shown in Figure 3-6.

3.3 MPEG-2 Part 2: Video (H.262)

Part 2 of the MPEG-2 standard defines the video portion. MPEG-2 video resembles MPEG-1, but also supports interlaced video, the format used by analog broadcast TV systems. MPEG-2 Video is formally known as ISO/IEC 13818-2 and as ITU-T Rec. H.262 [6]. It is often referred to as simply H.262 to indicate the video coding methodology.

The MPEG-2 Part 2 specification indicates that raw video frames are compressed into three different types of frames. The first is called an intra-coded frame (I-frame) which is a single frame or full image. This is often referred to as an anchor frame. The second type of frame is called a predictive-coded frame (P-frame) and defines the changes in the image data from the previous frame. Finally, the third type of frame is called a bidirectionally-predictive-coded frame (B-frame). A B-frame is used to compress the

video stream even further by specifying differences between the preceding image frame and the following image frame. Unlike P-frames and B-frames, I-frames do not depend on data in the preceding or the following frames. An I-frame will reset the contents of the frame to the picture defined in the frame. In between each full frame, only the changes or deltas are transmitted to save space.

MPEG-2 provides compression by dividing each frame into a number of macroblocks. H.262 implements three different types of macroblock formats: 4:2:0, 4:2:2, and 4:4:4. A 4:2:0 macroblock consists of 4 Luma values (Y), 1 Chrominance Blue (Cb), and 1 Chrominance Red (Cr) value. A 4:2:2 consists of 4 Y blocks, 2 Cb blocks, and 2 Cr blocks. A 4:4:4 consists of 4 Y, 4 Cb, and 4 Cr blocks. When the frame is reconstructed on the receiving end, the decoder searches for the best match macroblock. Encoded within each macroblock is a motion vector offset which denotes how much the macroblock should move. If there is no motion in the video, the offset will be zero. During periods of movement or change, the motion vector offset will instruct the decoder where to move the macroblock. A P-frame will store only the changes and only the macroblocks in the image that have changed from the previous full frame.

The MPEG-2 standard defines a Group of Pictures (GOP) as a series of frames and their arrangement. This is often a series of frames starting from an I-frame and continuing until the next I-frame (including all P and B-frames), although it is not strictly defined in the standard. The GOP implements a header found within the PES data and its own structure defined later in Section 4.

3.4 MPEG-2 Part 3: Audio

The MPEG-2 Audio section, defined in Part 3 (ISO/IEC 13818-3) of the standard, enhances MPEG-1's audio by allowing coding of audio programs with more than two channels, up to 5.1 multichannel. The MPEG-2 Audio Standard also extends the coding of the MPEG-1 Audio Standard to half sampling rates (16 kHz, 22.05 kHz and 24 kHz) for improved quality with bitrates at or below 64 Kbits/s per channel.

3.5 MPEG-4 Part 10: Advanced Video Coding (H.264)

H.264 is a block-oriented motion-compensation-based codec standard developed by the ITU-T Video Coding Experts Group together with the ISO/ IEC joint working group, i.e., the MPEG. H.264 AVC is part 10 of MPEG-4. (Its predecessor, MPEG4-Visual, is part 2 of MPEG-4). It serves as an industry standard for video compression and is currently one of the formats most commonly used to record, compress, and distribute high-definition video. H.264 is one of the codec standards for Blu-ray discs and is

widely used by streaming internet sources, web software such as the Adobe Flash®[2] Player, and HDTV broadcasts over terrestrial, cable, and satellite network links. It is the preferred video encoding standard for the U.S. Government's Motion Imagery Standards Board (MISB).

H.264 AVC offers better performance than the MPEG-2 H.262 video codec, yielding similar quality (about 2 to 1) in reduced bandwidth situations. This improvement comes with increased complexity in the encoder and decoder, which affects overall cost, but the rapid adoption of H.264 in the commercial world is making this less of an issue. Most modern graphics processors have native support for H.264 decoding.

As illustrated in Figure 3-7, an H.264 video encoder carries out prediction, transform, and encoding processes to produce a compressed H.264 bit stream. An H.264 video decoder carries out the complementary processes of decoding, inverse transform, and reconstruction to produce decoded video sequence.

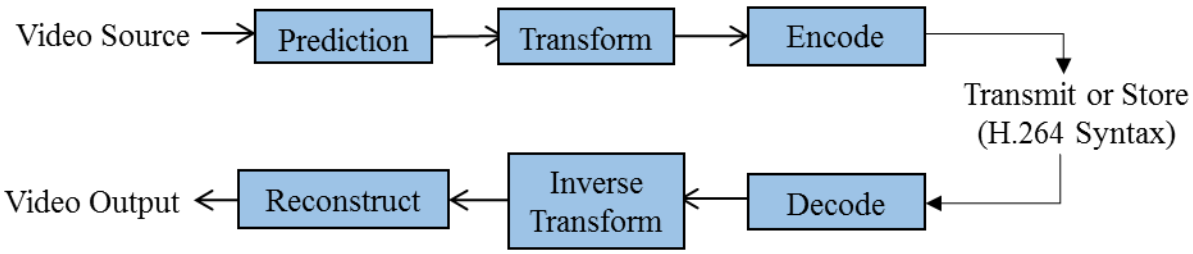


Figure 3-7. H.264 Video Encoding and Decoding

The encoder processes a frame of video in units of a macroblock (16x16 displayed pixels). It forms a prediction of the macroblock based on previously coded data, either from the current frame (intra-prediction) or from other frames that have already been coded and transmitted (inter-prediction). The encoder subtracts the prediction from the current macroblock to form a residual frame.

The prediction methods supported by H.264/AVC are more flexible than those in previous standards, enabling accurate predictions and efficient video compression. Intra-prediction uses 16x16 and 4x4 block sizes to predict the macroblock from surrounding, previously coded pixels within the same frame. Inter-prediction uses a range of block sizes (16x16, 16x8, 8x16, 8x8, 8x4, 4x8, and 4x4) to predict pixels in the current frame from similar regions in previously coded frames. This method of partitioning blocks into motion-compensated sub-blocks of varying size is known as *tree structured motion compensation*.

2 Adobe Flash® is a trademark of Adobe Systems, Inc.

H.264 AVC transforms each block of residual chroma/luma intensity samples using a 4x4 or 8x8 integer transform, a well-defined, efficient approximation of the Discrete Cosine Transform (DCT). After quantization, the DCT-transformed block is often sparsely populated, containing many 0 values. This property of the DCT allows compression of blocks of data. The transform outputs a set of coefficients, each of which is a weighting value for a standard basis pattern. When combined, the weighted basis patterns re-create the block of residual samples.

The output of the transform is a block of transform coefficients, which is then quantized, i.e., each coefficient is divided by an integer value. Quantization reduces the precision of the transform coefficients according to a quantization parameter (QP). Typically, the result is a block in which most or all of the coefficients are zero, with a few non-zero coefficients. When QP is set to a high value, more coefficients are set to zero, resulting in high compression at the expense of poor decoded image quality. When QP is set to a low value, more non-zero coefficients remain after quantization, resulting in better decoded image quality but lower compression.

As previously mentioned, an ES is comprised of a number of Access Units (AU). An AU is defined as the coded data for a picture or block of sound and any stuffing (null values) that follows it. [38] Figure 3-4 illustrated how MPEG-2 can include AUs to construct an ES, which can then form a PES. An AU in H.264 can be further divided into a lower level called the Network Abstraction Layer (NAL). Each block of data in the NAL is called a NAL unit. A series of combined NAL units is a NAL unit stream. The H.264/AVC standard introduces the NAL as a mechanism to provide "network friendliness" to customize the use of the video coding layer (VCL). The NAL permits mapping H.264/AVC data to higher layer protocols and content such as RTP/IP, H.32X protocols, and file formats such as MP4.

A NAL unit can be considered a block of data with a known number of bytes (similar to a packet). Each NAL unit contains a header byte defining the type of data that follows in the unit. The rest of the packet is considered a generic payload that varies in structure based upon the header byte value. NAL units can be used in both packet-oriented (IP/RTP) and bitstream-oriented (MPEG-2) transport systems. All of this information is embedded within the data portion of the PES.

Each NAL unit contains a portion of the embedded video, divided into a structure called a slice. H.264 introduced the concept of a slice; a frame or picture is made up of one or more slices. A slice is defined as an integer number of consecutive macroblocks (often 16x16 pixel blocks). Therefore, a frame is made up of a number of slices which consist of a number of macroblocks. As mentioned earlier, macroblocks are the basic processing unit in H.264 and are the color values associated with the pixels represented in the video stream. The structure of each macroblock in H.264 is shown below in Figure 3-8. The data portion of each macroblock provides the color values Luma (Y), Chrominance Blue (Cb), and Chrominance Red (Cr) for each pixel in the macroblock. If the macroblock is an inter macroblock (P or B macroblock), then the reference

information is provided as well as a motion vector to instruct the decoder how to move the macroblock to the correct location to reflect the movement in the video.

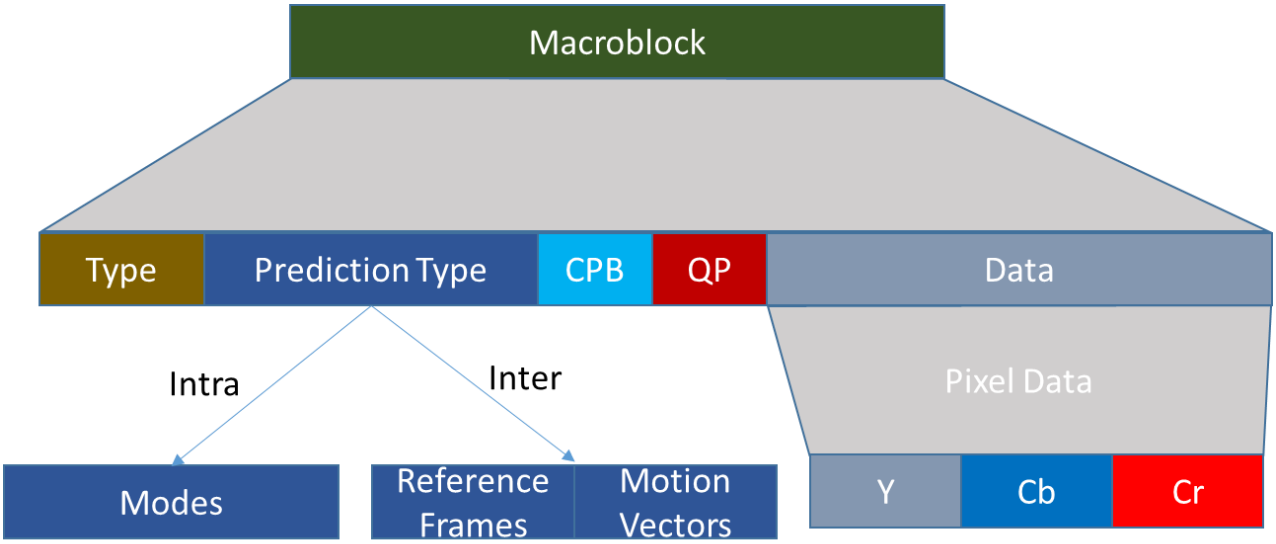


Figure 3-8. H.264 Macroblock Structure[39]

H.264 contains I-, P-, and B-slices that are found within the I-, P-, and B-frames or pictures that were discussed in earlier sections. Since H.264 offers greater compression than H.262, the term I-, P-, and B-slice are often used in place of frames since this compression algorithm operates at a more granular level than the frame level. Since slices are composed of macroblocks, there are also I, P, and B macroblocks. An I-slice contains only I macroblocks. While a P-slice may contain both I and P macroblocks. A B-slice may contain I, P, and/or B macroblocks.

H.264 introduces the SP (switching P) and SI (switching I) slices in the Extended Profile for the video codec. These were developed to improve resiliency to errors in the transmission, allowing switching between different bitstreams, and allowing decoders to begin decoding in the middle of a video stream (i.e., random access). An SP-slice, or sometimes referred to as an SP-frame, is used to decode the same slice with respect to two different slices. This means that two different SP-slices should be equal to one another when they switch between two different referential I- or P-slices. If slices are lost in the process, a decoder may recover and switch between the different SP-slices to recover the lost information. A decoder may also begin decoding video in the middle of a video stream if it is able to capture SP-slices instead of an I-slices. SP-slices can be sent at regular intervals by the video encoder. An SP-slice will consist of P and/or I macroblocks. An SI-slice is defined by the standard as "a slice that is coded using intra prediction only and using quantization of the prediction samples. An SI slice can be coded such that its decoded sample can be constructed identically to an SP slice." The slices are called switching slices because they allow a video decoder to switch between

similar coded sequences. These sequences could be the same video encoded at different bitrates. This allows the decoder to switch between them without waiting for an I-frame to begin decoding on the receiving end.

One important slice is the Instantaneous Decoder Refresh (IDR) picture, shown in Figure 3-9. The IDR is an I-frame that indicates that no slice that follows the IDR may reference any information before the IDR. Not all I-frames are IDRs as future slices may still reference information from previous slices. When an IDR is received, a new video sequence begins. Figure 3-8 illustrates an example NAL stream, which is composed of several NAL units, some are VCL NAL units which are slices of video. Each NAL Unit begins with the Start Code Prefix value of 0x000001. When these are combined they form the PES of the video stream.

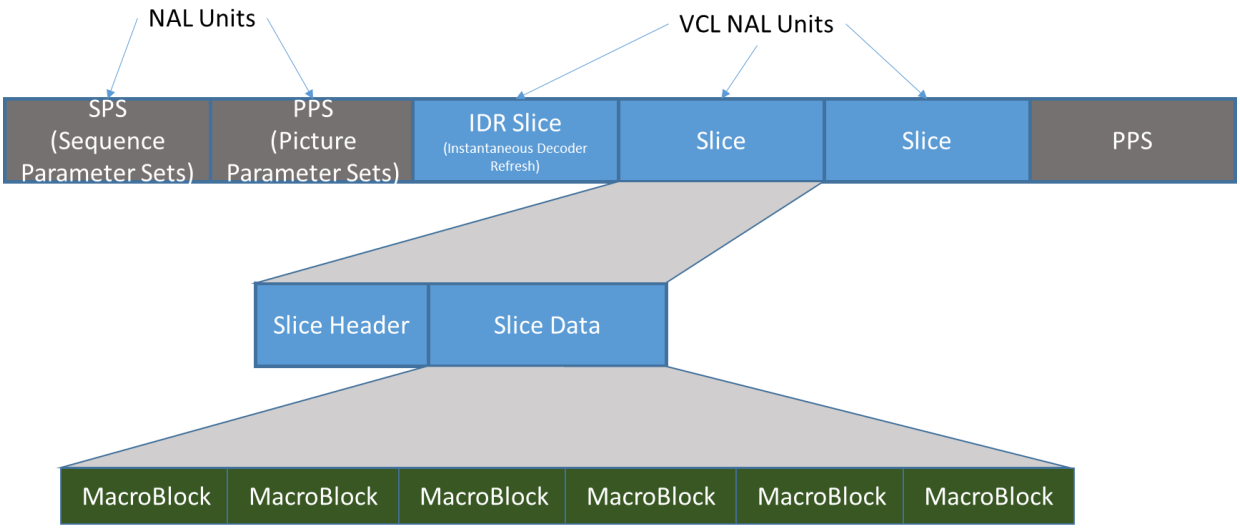


Figure 3-9. Slicing in H.264

3.6 Public-domain H.264 Codecs and Video Analysis Tools

H.264/AVC encoding and decoding (codecs) software in `C` are available to download online [13-15]. The JM (Joint Model) is a reference codec available from the JVT [13]. A real-time codec of high efficiency and quality is `x264` [14]. These codecs also serve as H.264 bitstream analyzers. They can be used to evaluate compressed data rate, video quality, and codec performance by collecting H.264 stream statistics. Commercially available graphical bitstream analyzers can organize and present the high volume of performance statistics to the user [12, Ch.9].

Bitstream analyzers do not directly search for and detect hidden data or data tampering because they focus on compression efficiency and image quality. However, they do verify that an H.264 video complies with the H.264/AVC standard. Non-compliance with the standard may indicate possible video tampering. In addition, the comprehensive video analysis that the analyzers perform may contain relevant video

information and indications of unusual statistics, such as bitrate or peak signal-to-noise ratio (PSNR) to aid the forensic data analyst.

3.7 Motion Imagery Standards Board[3]

Department of Defense (DoD) Directive 5105.60 established the Motion Imagery Standards Board (MISB) "to formulate, review, and recommend standards for motion imagery, associated metadata, audio, and other related systems" for the DoD, Intelligence Community (IC), and National System for Geospatial Intelligence (NSG).

The MISB is the due-process standards body that produces the Motion Imagery Standards Profile (MISP). This standards profile directly expresses the MISB mission and serves as the master baseline standards document developed and managed by the MISB. The MISB also participates in the North Atlantic Treaty Organization (NATO) Standardization Agreement (STANAG) process aimed toward interoperability of coalition forces. This includes working with domestic and international standards bodies to monitor, advocate, and represent DoD/IC/NSG interests for motion imagery, metadata, audio, and related systems to support global interoperability. *MISB Standard 0807*, an important product of the MISB, defines metadata keys that supplement the MPEG data streams for DoD purposes, as explained in the next section.

3.8 Key-Length-Value (KLV) Metadata

The KLV data encoding standard is often used to embed metadata such as platform heading and speed in MPEG-2 video. Items are encoded into KLV triplets, where key identifies the data type, length specifies the data's length, and value is the data itself. KLV metadata is inserted into MPEG-2 TS as its own ES. The PMT for each program will list the PID of each KLV stream embedded within the video stream. A KLV metadata ES is encapsulated in a PES just as any other video or audio ES. Figure 3-10 below illustrates how KLV is encapsulated into a TS. There are two types of KLV: synchronous and asynchronous [20]. Synchronous KLV implements a Presentation Time Stamp (PTS) included in the PES header shown in Figure 3-10. The PTS synchronizes the appearance of the KLV metadata with the other video and audio streams of a particular program. Asynchronous KLV does not implement a PTS field, therefore, it resembles the image in Figure 3-10 minus the PTS field in the PES header. The relationship between asynchronous KLV and the other video and audio streams is based on proximity to the other packets and not based on a time value.

3 Additional information on the MISB can be found at http://www.gwg.nga.mil/misb/index.html

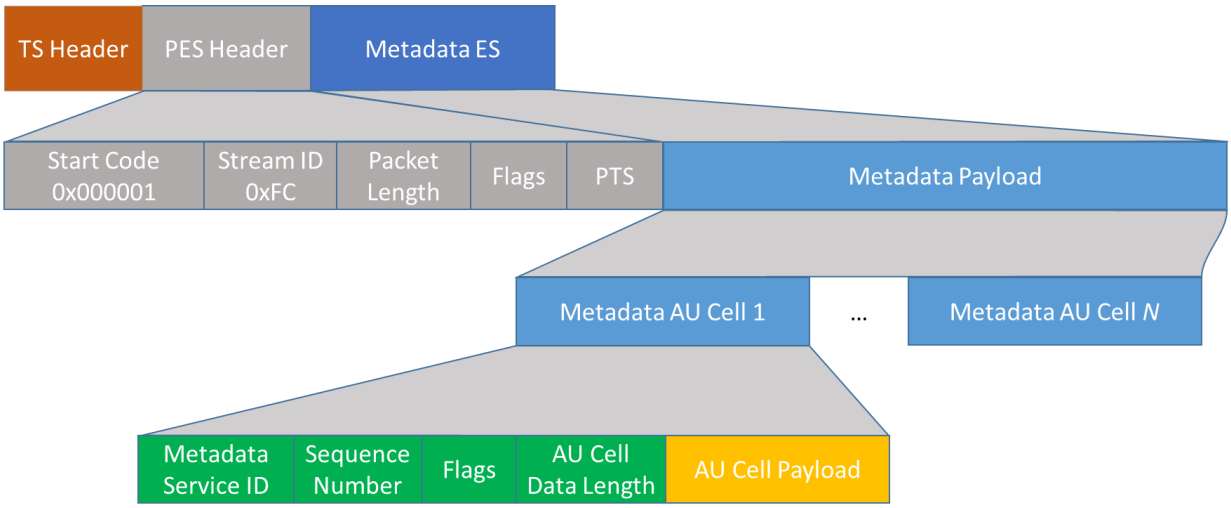


Figure 3-10 Synchronous KLV in MPEG-2 TS [20]

The Society of Motion Pictures and Television Engineers (SMPTE)[4] 336M-2007 (Data Encoding Protocol Using Key-Length-Value) defines the KLV format. The MISB has adopted KLV because it is a largely self-documenting format and promotes robust interoperability among diverse systems.

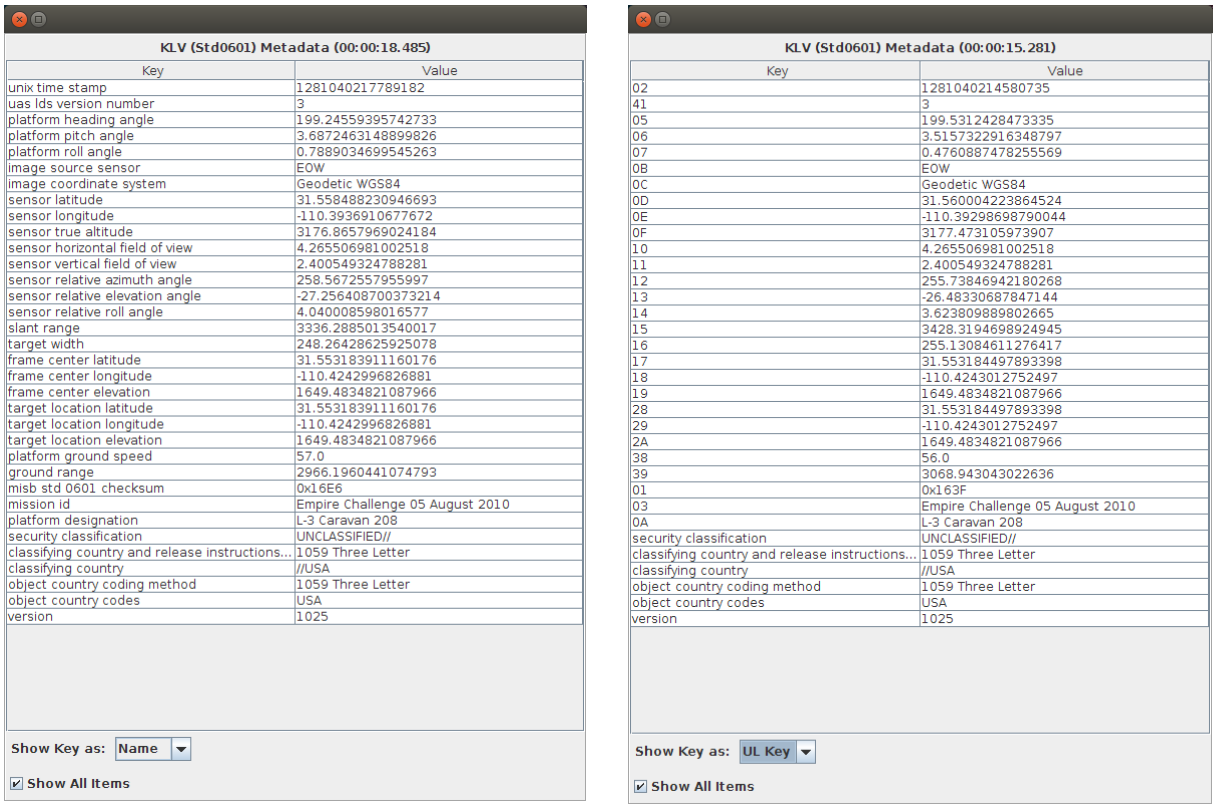
The SMPTE produces and maintains a KLV metadata dictionary (*SMPTE RP 210*). Various organizations may buy part of the KLV domain name-space to maintain private metadata dictionaries. The MISB was the first organization to take advantage of this offer.

Many metadata keys used by the MISB are registered in *SMPTE RP 210*; however, several issues arose. The first concerned the duration: it can take several months for the SMPTE to approve a new KLV metadata key. The second issue was element definition; SMPTE does not tightly define its metadata elements. *MISB Standard 0807* [15] is the metadata dictionary for elements in the DoD private domain space. Unlike SMPTE, the MISB can assign keys quickly when necessary, and can define their meaning and their usage with the necessary precision. The MISB Standard 0807 allows the definition of classified keys.

MISB Standard 0807 has precedence over SMPTE RP 210. All KLV keys are 16 bytes long. All SMPTE keys (including the DoD private keys in *Standard 0807*) begin with the 4-byte sequence 0x06 0x0E 0x2B 0x34 (in hexadecimal). Keys from MISB Standard 0807 have the ninth byte set to 0x0E and the tenth byte set to 0x01, 0x02, or 0x03. A MISB key takes the form 0x06 0x0E 0x2B 0x34 xx xx xx xx 0x0E [0x01, 0x02, or 0x03] xx xx xx xx xx xx.

4Additional information on the SMPTE can be found at https://www.smpte.org

A sample KLV metadata display from an MPEG video is shown in Figure 3-11. The string representation of the KLV keys are shown in the left image, while the key values are shown in the image on the right.



KLV (Std0601) Metadata (00:00:18.485)

Key	Value
unix time stamp	1281040217789182
uas lds version number	3
platform heading angle	199.24559395742733
platform pitch angle	3.6872463148899826
platform roll angle	0.7889034699545263
image source sensor	EOW
image coordinate system	Geodetic WGS84
sensor latitude	31.558488230946693
sensor longitude	-110.3936910677672
sensor true altitude	3176.8657969024184
sensor horizontal field of view	4.265506981002518
sensor vertical field of view	2.400549324788281
sensor relative azimuth angle	258.5672557955997
sensor relative elevation angle	-27.256408700373214
sensor relative roll angle	4.040008598016577
slant range	3336.2885013540017
target width	248.26428625925078
frame center latitude	31.553183911160176
frame center longitude	-110.4242996826881
frame center elevation	1649.4834821087966
target location latitude	31.553183911160176
target location longitude	-110.4242996826881
target location elevation	1649.4834821087966
platform ground speed	57.0
ground range	2966.1960441074793
misb std 0601 checksum	0x16E6
mission id	Empire Challenge 05 August 2010
platform designation	L-3 Caravan 208
security classification	UNCLASSIFIED//
classifying country and release instructions...	1059 Three Letter
classifying country	//USA
object country coding method	1059 Three Letter
object country codes	USA
version	1025

Show Key as: Name

Show All Items

KLV (Std0601) Metadata (00:00:15.281)

Key	Value
02	1281040214580735
41	3
05	199.5312428473335
06	3.5157322916348797
07	0.4760887478255569
0B	EOW
0C	Geodetic WGS84
0D	31.560004223864524
0E	-110.39298698790044
0F	3177.473105973907
10	4.265506981002518
11	2.400549324788281
12	255.73846942180268
13	-26.48330687847144
14	3.623809889802665
15	3428.3194698924945
16	255.13084611276417
17	31.553184497893398
18	-110.4243012752497
19	1649.4834821087966
28	31.553184497893398
29	-110.4243012752497
2A	1649.4834821087966
38	56.0
39	3068.943043022636
01	0x163F
03	Empire Challenge 05 August 2010
0A	L-3 Caravan 208
security classification	UNCLASSIFIED//
classifying country and release instructions...	1059 Three Letter
classifying country	//USA
object country coding method	1059 Three Letter
object country codes	USA
version	1025

Show Key as: UL Key

Show All Items

Figure 3-11. KLV Metadata Example

Extracting the KLV from the TS file requires custom software or a hexadecimal editor to display its contents. Figure 3-12 shows a portion of the KLV metadata from the same TS file shown in Figure 3-11. The red circle indicates the start of the MPEG-2 TS packet header, the synchronization byte (0x47). With the synchronization byte marked, it is clear that the KLV metadata spans several MPEG-2 TS packets. The green circles represents the Society of Motion Picture and Television Engineers (SMTPE) key, note the 4-byte beginning value 0x060E2B34 as defined in Standard 0807. The length field follows the 16-byte key, which is 0x81B3. This uses Basic Encoding Rules (BER) encoding, which means that the length is not 0x81B3 bytes. Since the high bit is set in the first byte (0x80), the lower 7 bits of the first byte (0x01) denote the length (in bytes) of field that follows. This value means that there is one byte for the length field, so the length of the value that follows is 0xB3 bytes (179 bytes). The purple circle is one of the first values seen in the list in Figure 3-12. The value portion of this KLV block is broken into smaller Tag-Length-Value (TLV) data blocks; however, this is not always the case,

it will depend on the specific KLV metadata key in use. In this example, the first tag is 0x02, which is the Unix®[5] timestamp as seen in Figure 3-12. The length is 0x08 bytes and the value is 0x00048D13271DC4C0. This value is equal to 1,281,013,307,000,000 (Thu, 05 Aug 2010 13:01:47 GMT) in base 10. This value is close to the timestamp seen in Figure 3-12. The blue circle represents one of the string values ("EOW") visible in the metadata. By looking at Figure 3-12, the string "EOW" is listed under key 0x0B (Image source sensor). The blue circle below encompasses the value 0x0B 0x03 0x45 0x4F 0x57. The tag type is equal to 0x0B; its length is equal to 0x03 (bytes); and the value is the ASCII string "EOW". Other types and some string values can be seen in Figure 3-12 throughout a number of packets.

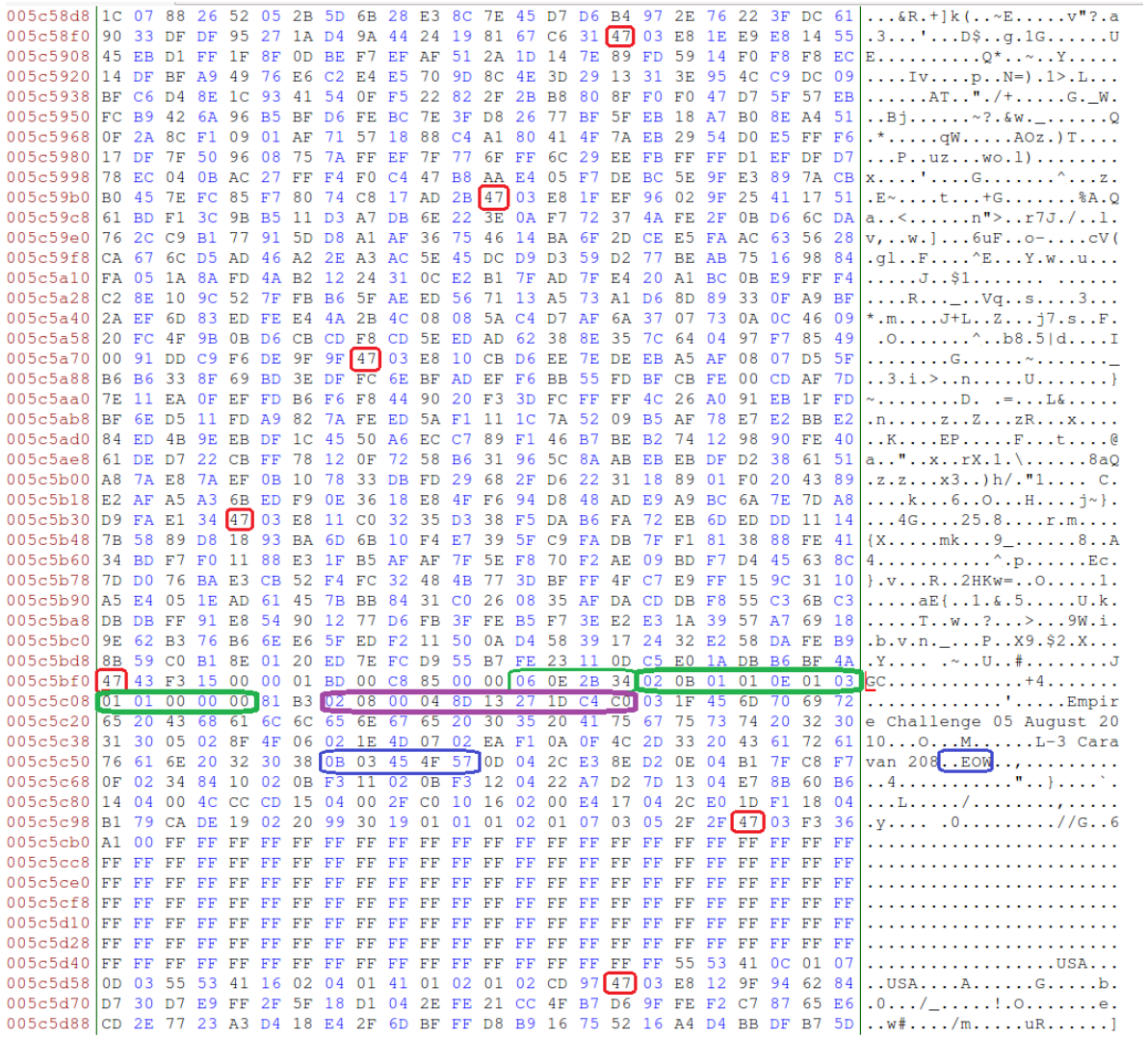


Figure 3-12. Hexadecimal View of KLV Metadata

5 UNIX® is a trademark of The Open Group, Inc.

3.9 MISB Standards for MPEG-2 Inspection and Sanitization

The MISB publishes the Motion Imagery Standards Profile (MISP [34]), Standards (ST), Recommended Practices (RP), Engineering Guidelines (EG) and Technical Reference Material (TRM). The Board recommends that implementers of motion imagery systems adhere to all ST's and RP's that the MISB publishes, but acknowledges that special circumstances and needs may prevent this, and allows for these cases.

Where the MISP term Standard (ST) is used, the MISP item mandates binding technical implementation policy, and as such, should be identified in Government procurement actions as a mandatory compliance item in order for vendor offerings to be accepted by the Government.

A Recommended Practice (RP) should be considered technical implementation policy. They may be identified in Government procurement actions as a mandatory compliance item in order for vendor offerings to be accepted by the Government.

Documents originally published as Engineering Guidelines (EG) have either been promoted to a ST or RP based on meeting the appropriate criteria listed above. The EG will not be used in future publications. Some legacy documents continue to carry this designation.

Technical Reference Material (TRM) is an informative/educational document that does not contain requirements. A TRM may result from a study or provide additional background to practices promoted by other guidance.

The MISB indicates that, to be considered compliant, any new motion imagery system must:

- Be digital and progressive-scan format.
- Comply with MPEG-2, H.264/AVC, or JPEG 2000 compression standards.
- Comply with the MPEG-2 Transport Stream standard.
- Not allow visually destructive metadata.
- Comply with the minimum metadata set per MISB ST 0902.1 [17].

This ISG is not intended to validate compliance with all MISB STs and RPs, but rather to utilize these standards and practices as a basis for examining MPEG-based files for data hiding, data attack, and data disclosure issues.

3.10 Color in MPEG-2

The representation of color is important in any video or imagery system. Data can often hide within the low order bits of color planes. This section briefly describes how MPEG2 processes color spaces. Color video begins and ends with the capture and

display of superposed red, green, and blue (RGB) images. A video camera typically captures a sequence of frames in tricolor. Each picture element (pixel) in the camera's imaging plane receives its own RGB brightness values. At the consumer end, the video is displayed in RGB colors.

However, the human visual system (HVS) is more sensitive to changes in total brightness or luminance than to changes in color. To reduce the amount of data that is sent or stored, without losing perceptual quality, the raw image frame is immediately converted from RGB color space to YCrCb (or YUV) color space. This conversion is performed by fairly simple algebra:

$$Y = k_r R + k_g G + k_b B$$

$$Cr = \tfrac{1}{2}(R - Y)/(1-k_r)$$

$$Cb = \tfrac{1}{2}(B - Y)/(1-k_b)$$

$$Cg = \tfrac{1}{2}(G - Y)/(1-k_g)$$

for weighting factors k_r, k_g, k_b that sum to 1. Y is called the luminance, or *luma* for short. Cr and Cb are the scaled red and blue color differences with respect to the luminance, called chrominance, or *chroma* for short. Only the red and blue chroma are needed, since for each pixel, the green chroma value can be recovered from the other two chroma, for each pixel. If R,G,B are 8-bit unsigned integers, with range 0–255, then the weighted average Y has the same range, and Cr and Cb are 8-bit signed integers with half the absolute range. These formulas can be solved in order to retrieve the R,G, and B values for display.

Because the color values matter less to the human eye, the two chroma components can be sampled within each frame *less often* than the luma. For example, in the 4:2:0 format, the chroma are sampled only once in every 2x2 sub-array of luma pixel samples. This results in fewer data samples, but preserves the color image quality.

The color space conversion and chroma down-sampling described represent preliminary steps to MPEG compression that forms part of video encoding on the left of Figure 3-3. The MPEG-2 and MPEG-4 standards support various frame sizes and luma/chroma sampling formats for video input and output [12].

4. MPEG-2 CONSTRUCTS

4.1 MPEG-2 Transport Stream Constructs

4.1.1 Transport Stream Programs

A MPEG-2 TS may include a single program or multimedia elementary stream (e.g., a video ES, audio ES, etc.). This type of TS is normally called a Single Program Transport Stream (SPTS). A program is a single stream of video (and possibly audio) that pertains to a particular source. A single program might consist of one video ES, one audio ES, and associated KLV metadata ES. A SPTS contains all the information required to reproduce the program or multimedia elementary stream.

A MPEG-2 TS may also include two or more programs that are combined to form a Multiple Program Transport Stream (MPTS). This larger aggregate also contains all the control information (Program Specific Information (PSI)) required to reproduce each of the programs.

4.1.1.1 Program Association Table

OVERVIEW

MPEG-2 TS permits multiplexing of multiple programs each of which in turn can consist of multiple ESs. The program association table (PAT) lists all programs contained within the TS. The typical TS produced by a UAV platform contains a single program consisting of one video ES and one or two KLV metadata ES. However, a MPEG-2 TS may contain multiple programs, as well as a single program consisting of multiple audio ES's along with a video ES.

Example: A filter may inspect the contents of a TS produced by a UAV platform for validity, normal range-of-values, and consistency of related values. The following example shows a TS that contains a single program consisting of a video ES and a KLV metadata ES:

```
Input #0, mpegts, from 'example.ts':

 Duration: 00:01:10.07, start: 16240.289722, bitrate: 3008 kb/s

 Program 1

 Stream #0.0[0x80]: Video: mpeg2video (Main), yuv420p, 720x480 [PAR 8:9 DAR
4:3], 2870 kb/s, 29.97 fps, 29.97 tbr, 90k tbn, 59.94 tbc

 Stream #0.1[0x90]: Data: KLVA / 0x41564C4B
```

RISKS AND RECOMMENDATIONS

Data Hiding: A MPEG-2 TS consisting of multiple programs could contain hidden data in false programs (programs that are not officially defined in the PAT). The data could then be extracted using a custom application by accessing data in programs that are not formally defined by the transport stream.

1. **Validate:** Identify the number of programs by locating the PMT for each program listed in the PAT.
2. **Remove:** Remove all programs or PIDs in the MPEG-2 TS that are not listed in the Program Association Table (PAT) (and PMT) of the MPEG-2 TS.
3. **Review:** Present the list of programs and associated elementary streams to a human for review.

PRODUCT

MPEG-2

LOCATION

MPEG-2 TS packet.

4.1.1.2 Program Map Table

OVERVIEW

Each MPEG-2 program is described by a Program Map Table (PMT) that includes a list of the Program ID's (PIDs) associated with that program (see Table 3-1). For instance, a single program might consist of one video ES, one audio ES, and associated KLV metadata ES. Each ES, once formated into packets as a Packetized Elementary Stream (PES), is designated in the TS uniquely by its Program ID (PID). Media players decode the payloads of each PID associated with the program, while discarding the contents of all other PIDs.

RISKS AND RECOMMENDATIONS

Data Hiding: A MPEG-2 TS that consists of a single program could contain elementary streams not included in the PMT and not revealed upon playback with media players.

1. **Validate:** Identify the list of PIDs in the PMT.
2. **Validate:** Verify that no additional PIDs not in the PMT are present in the TS.
3. **Remove:** Remove all elementary streams from a Program that are not listed in the Program Map Table (PMT) for that program.
4. **Reject:** Reject the TS if it contains elementary streams with PIDs not listed in the PMT.

PRODUCT

MPEG-2

LOCATION

TS packet.

4.1.1.3 Multiple Audio Streams

OVERVIEW

A MPEG-2 TS program can include a single video ES with multiple associated audio ES's (e.g., multiple languages). The user typically selects an audio ES for playback with media players.

RISKS AND RECOMMENDATIONS

Data Hiding: An audio ES associated with a single video stream that is not selected for playback can contain hidden data.

1. **Validate**: Identify the audio ES's associated with the video ES.
2. **Remove:** Remove all audio ES's other than the audio ES that have been defined a priori.
3. **Review:** Present the list of audio ES's to a human for review.
4. **External Filtering Required:** Extract the audio ES from the TS and present the content to an external filter. If modified, this will require rebuilding and repacketizing the TS with a new audio stream.

PRODUCT

MPEG-2

LOCATION

MPEG-2 TS packet.

4.1.1.4 User Data - Closed Captioning

OVERVIEW

Closed captioning text is not carried in a separate elementary stream with a user-specified PID, but is embedded in the MPEG-2 video stream as "picture user data." This is located by identifying the User Data Header by locating the Start Code Prefix (0x000001) and the value 0xB2. The data following the value 0xB2 is called the private_user_data area or picture user data. The user data is inserted according to ISO/IEC 13818-2 Section 6.2 [2], in the `extension_and_userdata(2)` structure,

which follows the `picture_header()` and `picture_coding_extension()` structures in the video ES packet. Within this structure, the closed captioning data essence is `cc_data()`, as defined in the closed captioning standard Consumer Electronics Association (CEA)-708 Table 2 [8]. (Besides closed captioning data, Bar data and active format description data can also be inserted here in the video stream packets as picture user data.) For more detailed syntax of this picture user data, refer to [7].

Closed Captioning is defined in a separate standard (CEA-708) and there are numerous tables of information that provide information to the receiver on how to properly render the text. CEA-708 was created for use in digital television. It can support captions in any language and supports UTF-32 captions. The standard also controls the appearance of captions. There are 8 different fonts available in CEA-708. One of the fonts is listed as undefined, which might create issues when rendering the text but some systems may use it as a default font. There is also support for changing the text color, background color, font size, text styling, and opacity levels.

RISKS AND RECOMMENDATIONS

Data Hiding: Closed caption text included in the MPEG-2 video packets' picture user data can contain hidden data.

1. **Remove:** If the contents of the User Data area is not closed captioning or data specified by a MISB standard (refer to construct 4.1.4.1) remove, the User Data and rebuild the MPEG stream.

2. **Replace:** Replace any closed captioning text or other hidden data extracted from `cc_data` (expected to be in CEA-708 format) with an empty string.

3. **Replace:** Replace any undefined closed captioning font with a defined font value.

4. **External Filtering Required:** Extract and pass any text found within this section to an external filter.

PRODUCT

MPEG-2 OR H.264/AVC

LOCATION

MPEG-2 TS video packets.

4.1.2 Transport Stream Packet Headers

To create an MPEG-2 TS, one or more ESs of data sources such as coded video or audio data are packetized and multiplexed to form a single output TS for transmission. The TS packet consists of a four-byte header followed by 184 bytes shared between the

variable-length Adaptation Field (AF) and the TS packet payload. Several of the fields in the TS packet may be inspected for valid content as well as potential sanitization.

The MPEG-2 systems layer packetizes all ESs, including audio, video, user data, and control streams, to form PESs. A PES header must always follow the TS header and possible AF. The TS payload may consist of the PES packets or PSI. The PSI provides control and management information used to associate particular ES's with distinct programs.

4.1.2.1 Synchronization Byte

OVERVIEW

All TS decoders search for the beginning of a packet by identifying the Sync Byte (0x47 hexadecimal) in the packet header. Each packet is tested for 0x47 as soon as it arrives. If a packet arrives with an incorrect sync byte, the decoder has lost sync.

Example: Figure 4-1 shows a typical TS hex dump, indicating the synchronization bytes.

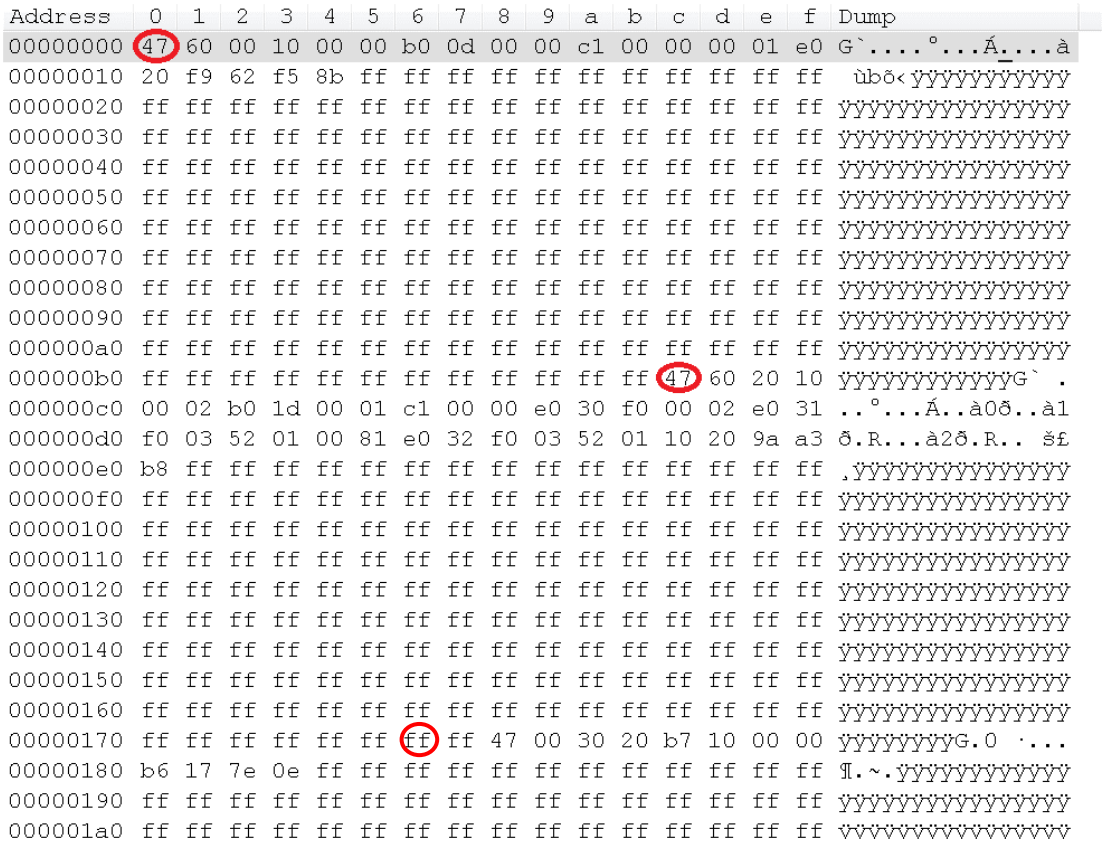


Figure 4-1 Transport Stream Packet Synchronization Byte

RISKS AND RECOMMENDATIONS

Data Hiding and Data Attack: If the decoder loses sync, the packet may contain hidden data or malicious code.

1. **Validate:** Check that the sync byte (0x47 hexadecimal) is present in the first byte of each TS packet, and that it appears every 188 bytes in the TS file.

2. **Reject:** Reject the file if the sync byte is not detected in any TS packet.

PRODUCT

MPEG-2

LOCATION

The first byte of each TS packet header.

4.1.2.2 Null Packets

OVERVIEW

Some transmission schemes impose strict constant bitrate requirements on the TS. To ensure that the stream maintains a constant bitrate, the multiplexer may need to insert some additional packets. The TS PID 0x1FFF hexadecimal is reserved for this purpose. The payload of null packets may not contain any data at all, and the receiver is expected to ignore its contents.

Example: Figure 4-2 shows a typical null packet hex dump, indicating stuffing bytes in the payload.

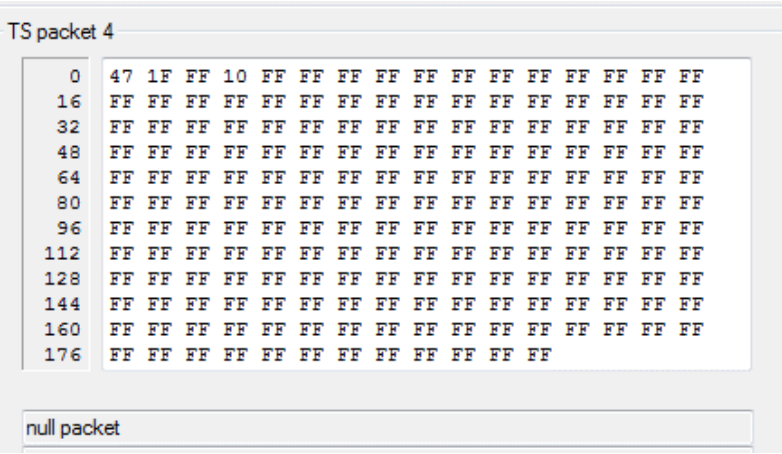


Figure 4-2 Transport Stream Null Packet

RISKS AND RECOMMENDATIONS

Data Hiding: Stuffing bytes in null packets can contain hidden data.

1. **Validate:** For all null packets, check that the contents of the stuffing bytes are 184 bytes of 0xFF in hexadecimal.

2. **Replace**: Replace all stuffing bytes with 0xFF.

3. **Remove:** Remove stuffing packets from TS file; however, this may cause problems for decoders that require a constant bit rate.

PRODUCT

MPEG-2

LOCATION

TS packets with a program ID = 01xFF.

4.1.2.3 Video Compression Format

OVERVIEW

The motion imagery should be compressed using one of the MISB-allowed compression types for video, which include H.262 and H.264. This prevents data hiding by substituting other compression formats.

RISKS AND RECOMMENDATIONS

Data Hiding: Video codecs other than H.262 and H.264 entail associated data hiding risks that can be prevented.

1. **Validate:** Check that the compression format is either H.262 or H.264.

2. **Reject:** Reject the file if the compression format is neither H.262 nor H.264.

PRODUCT

MPEG-2

LOCATION

The MPEG-2 TS packets consist of a four-byte header followed by 184 bytes of payload. The packet header includes a 13-bit PID indicating the video codec type.

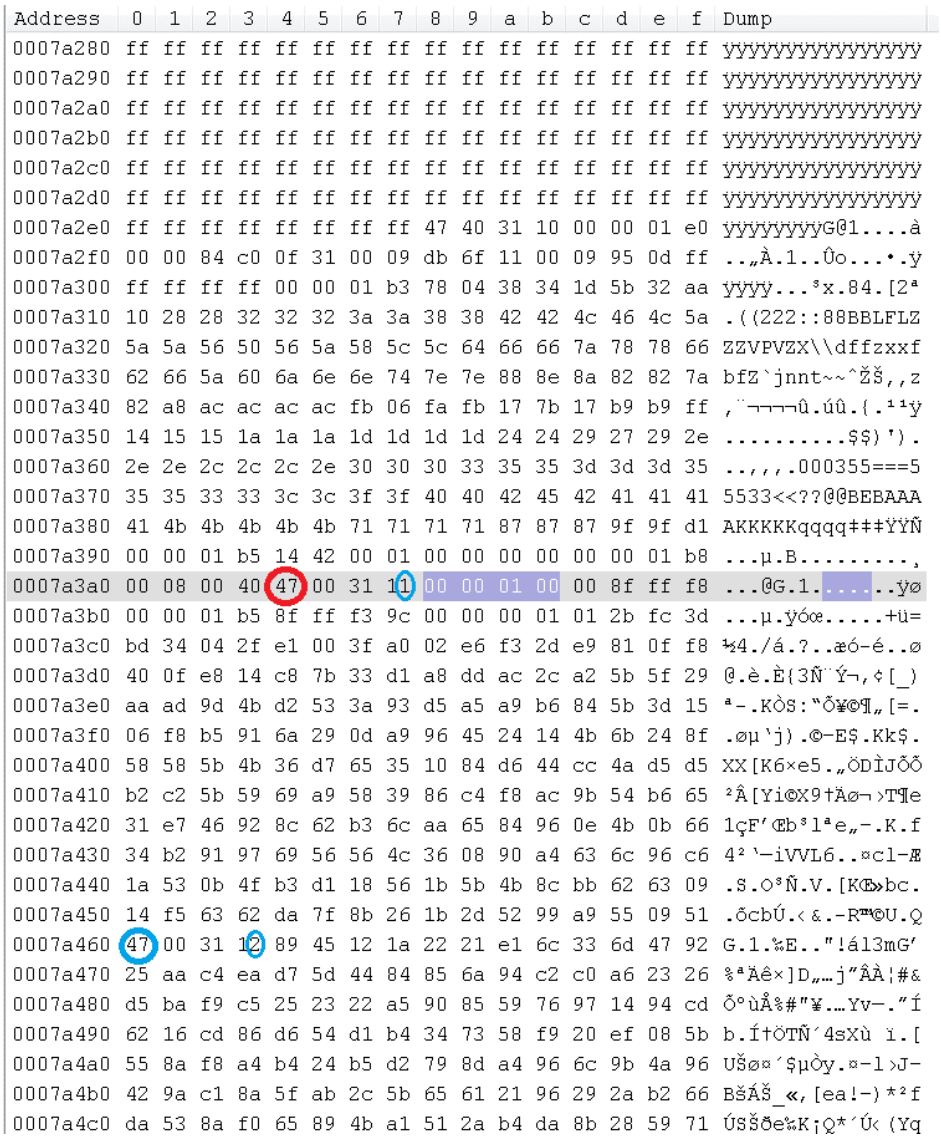
4.1.2.4 Continuity Counter

OVERVIEW

The continuity counter is a 4-bit field in the header that is incremented by 1 each time a packet is encountered with a specific PID. When a PID 'skips' one value of the continuity counter, a 'Continuity Error' has occurred. This indicates that one or more packets were lost, i.e., it identifies when packets are lost, but not how many. Although a

few continuity errors per second are typical for transmission over wireless networks, a rapid increment of the continuity counter indicates suspicious content or a defective file.

Example: Figure 4-3 shows a typical TS hex dump is shown below indicating the continuity counter following the synchronization byte.



Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	Dump
0007a280	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0007a290	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0007a2a0	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0007a2b0	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0007a2c0	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0007a2d0	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0007a2e0	ff	ff	ff	ff	ff	ff	ff	ff	47	40	31	10	00	00	01	e0	ÿÿÿÿÿÿÿÿG@1....à
0007a2f0	00	00	84	c0	0f	31	00	09	db	6f	11	00	09	95	0d	ff	..„À.1..Ûo...•.ÿ
0007a300	ff	ff	ff	ff	00	00	01	b3	78	04	38	34	1d	5b	32	aa	ÿÿÿÿ...³x.84.[2ª
0007a310	10	28	28	32	32	32	3a	3a	38	38	42	42	4c	46	4c	5a	.((222::88BBLFLZ
0007a320	5a	5a	56	50	56	5a	58	5c	5c	64	66	66	7a	78	78	66	ZZVPVZX\\dffzxxf
0007a330	62	66	5a	60	6a	6e	6e	74	7e	7e	88	8e	8a	82	82	7a	bfZ`jnnt~~^ŽŠ,,z
0007a340	82	a8	ac	ac	ac	ac	fb	06	fa	fb	17	7b	17	b9	b9	ff	,¨¬¬¬¬û.úû.{.¹¹ÿ
0007a350	14	15	15	1a	1a	1a	1d	1d	1d	1d	24	24	29	27	29	2e	..........$$)').
0007a360	2e	2e	2c	2c	2c	2e	30	30	30	33	35	35	3d	3d	3d	35	..,,,.000355===5
0007a370	35	35	33	33	3c	3c	3f	3f	40	40	42	45	42	41	41	41	5533<<??@@BEBAAA
0007a380	41	4b	4b	4b	4b	4b	71	71	71	71	87	87	87	9f	9f	d1	AKKKKKqqqq‡‡‡ŸŸÑ
0007a390	00	00	01	b5	14	42	00	01	00	00	00	00	00	00	01	b8	...µ.B.........¸
0007a3a0	00	08	00	40	47	00	31	11	00	00	01	00	00	8f	ff	f8	...@G.1.....ÿø
0007a3b0	00	00	01	b5	8f	ff	f3	9c	00	00	00	01	01	2b	fc	3d	...µ.ÿó.....+ü=
0007a3c0	bd	34	04	2f	e1	00	3f	a0	02	e6	f3	2d	e9	81	0f	f8	½4./á.?..æó-é..ø
0007a3d0	40	0f	e8	14	c8	7b	33	d1	a8	dd	ac	2c	a2	5b	5f	29	@.è.È{3Ñ¨Ý¬,¢[_)
0007a3e0	aa	ad	9d	4b	d2	53	3a	93	d5	a5	a9	b6	84	5b	3d	15	ª-.KÒS:"Õ¥©¶„[=.
0007a3f0	06	f8	b5	91	6a	29	0d	a9	96	45	24	14	4b	6b	24	8f	.øµ'j).©–E$.Kk$.
0007a400	58	58	5b	4b	36	d7	65	35	10	84	d6	44	cc	4a	d5	d5	XX[K6×e5.„ÖDÌJÕÕ
0007a410	b2	c2	5b	59	69	a9	58	39	86	c4	f8	ac	9b	54	b6	65	²Â[Yi©X9†Äø¬›T¶e
0007a420	31	e7	46	92	8c	62	b3	6c	aa	65	84	96	0e	4b	0b	66	1çF'Œb³lªe„–.K.f
0007a430	34	b2	91	97	69	56	56	4c	36	08	90	a4	63	6c	96	c6	4²'—iVVL6..¤cl–Æ
0007a440	1a	53	0b	4f	b3	d1	18	56	1b	5b	4b	8c	bb	62	63	09	.S.O³Ñ.V.[KŒ»bc.
0007a450	14	f5	63	62	da	7f	8b	26	1b	2d	52	99	a9	55	09	51	.õcbÚ.‹&.-R™©U.Q
0007a460	47	00	31	12	89	45	12	1a	22	21	e1	6c	33	6d	47	92	G.1.‰E.."!á13mG'
0007a470	25	aa	c4	ea	d7	5d	44	84	85	6a	94	c2	c0	a6	23	26	%ªÄê×]D„…j"ÂÀ¦#&
0007a480	d5	ba	f9	c5	25	23	22	a5	90	85	59	76	97	14	94	cd	Õºùŵ#"¥.…Yv–.”Í
0007a490	62	16	cd	86	d6	54	d1	b4	34	73	58	f9	20	ef	08	5b	b.Í†ÖTÑ´4sXù ï.[
0007a4a0	55	8a	f8	a4	b4	24	b5	d2	79	8d	a4	96	6c	9b	4a	96	UŠø¤´$µÒy.¤–l›J–
0007a4b0	42	9a	c1	8a	5f	ab	2c	5b	65	61	21	96	29	2a	b2	66	BšÁŠ_«,[ea!–)*²f
0007a4c0	da	53	8a	f0	65	89	4b	a1	51	2a	b4	da	8b	28	59	71	ÚSŠðe‰K¡Q*´Ú‹(Yq

Figure 4-3 Transport Stream Packet Continuity Counter

RISKS AND RECOMMENDATIONS

Data Attack: TS packet loss can result in unpredictable decoder results that cause execution of malicious code.

1. **Validate:** Use the continuity counter to verify that no packet loss has occurred.

2. **Reject:** Reject the file if the continuity counter indicates packet loss above a specified rate.

PRODUCT

MPEG-2

LOCATION

TS packet header.

4.1.2.5 Optional Adaptation Field

OVERVIEW

A MPEG-2 TS packet includes the optional AF if the AF Control bit is set in the packet header. This field can contain important information for the decoder such as the Program Clock Reference (PCR) field. However, the AF also contains a field for Private Data as well as a field for stuffing bytes whose contents should be 0xFF (all ones).

RISKS AND RECOMMENDATIONS

Data Hiding: The optional adaptation field can optionally provide extra data, including stuffing bytes, which may contain hidden data.

1. **Validate:** If the optional AF is present, identify the length of the AF from the first byte.

2. **Validate:** If the AF includes stuffing bytes, check that all values are 0xFF.

3. **Replace:** Replace stuffing bytes with 0xFF.

4. **Reject:** Reject the file if the transport_private_data_flag is set.

PRODUCT

MPEG-2

LOCATION

The end of the TS packet header.

4.1.3 Packetized Elementary Stream Headers

4.1.3.1 PES Start Code

OVERVIEW

As shown in Figures 4-4, 4-5, and 4-6, every PES begins with a 32-bit start code consisting of a start code prefix and a stream ID. Codes 00 through B8 indicate the video stream start codes, while codes B9 through FF indicate stream IDs [20].

byte 0	byte 1	byte 2	byte 3
0000 0000 0000 0000 0000 0001 Start code prefix			Stream ID

Figure 4-4 PES Start Code Prefix and Stream ID [37]

Start code Prefix	used for
0x00	Picture
0x01 - 0xAF	Slice
0xB0	Reserved
0xB1	Reserved
0xB2	User data
0xB3	Sequence header
0xB4	Sequence error
0xB5	Extension
0xB6	Reserved
0xB7	Sequence end
0xB8	Group of Pictures

Figure 4-5 PES Start Codes [37]

Table 4-1. PES Stream IDs [37]

Stream ID	used for
0xB9	Program end (terminates a program stream)
0xBA	Pack header
0xBB	System header
0xBC	Program stream map
0xBD	Private stream 1

0xBE	Padding stream
0xBF	Private stream 2
0xC0 - 0xDF	MPEG-1 or MPEG-2 audio stream
0xE0 - 0xEF	MPEG-1 or MPEG-2 video stream
0xF0	ECM stream
0xF1	EMM stream
0xF2	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Annex A
0xF3	ISO/IEC_13522_stream
0xF4	ITU-T Rec. H.222.1 type A
0xF5	ITU-T Rec. H.222.1 type B
0xF6	ITU-T Rec. H.222.1 type C
0xF7	ITU-T Rec. H.222.1 type D
0xF8	ITU-T Rec. H.222.1 type E
0xF9	Ancillary_stream
0xFA - 0xFE	Reserved
0xFF	Program stream directory

Figure 4-6 shows an example of the PES start code.

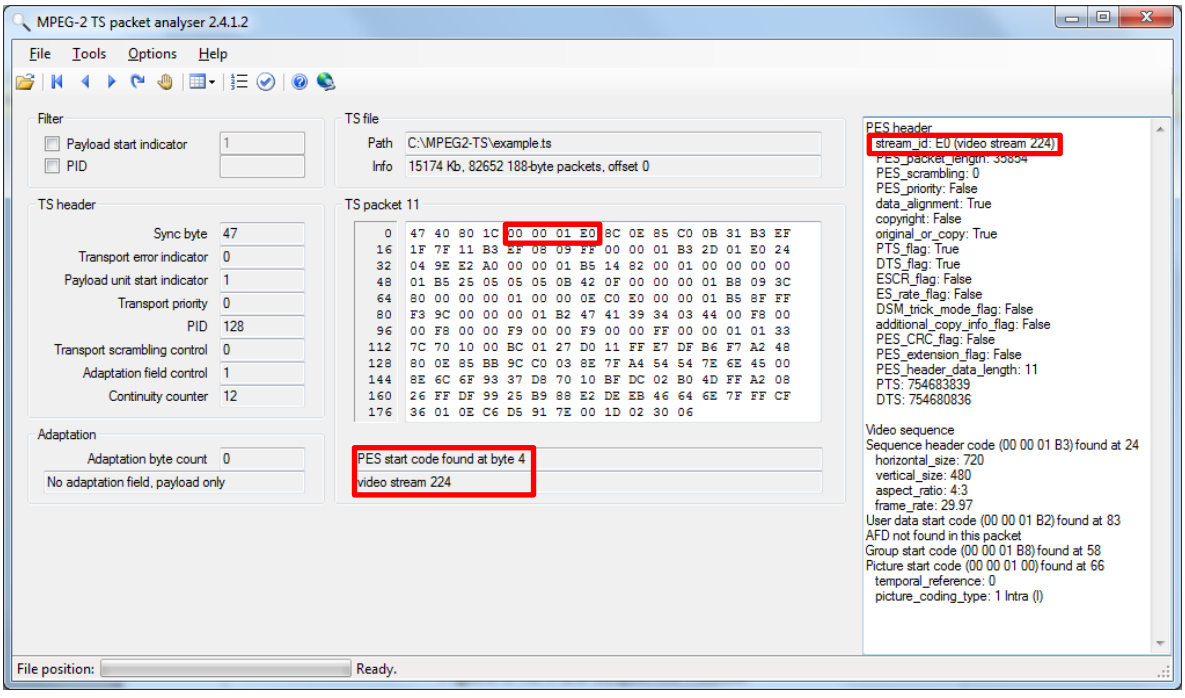


Figure 4-6 Example of PES Start Code

RISKS AND RECOMMENDATIONS

Data Hiding: A packetized elementary stream with a stream ID not corresponding to MPEG-2 video could contain hidden data not revealed upon playback with media players.

1. **Validate:** Check that the stream ID corresponds to an MPEG-2 video (0xE0 - 0xEF).

2. **Reject:** Reject the file if the stream ID is not MPEG-2 video.

PRODUCT

MPEG-2

LOCATION

TS packet.

4.1.3.2 Picture Header

OVERVIEW

The variable-length picture header, as shown in Figure 4-7, is identified by PES packet header start code 0x00 and contains a code to identify the type of each video frame. These include intra-predicted frames (I), predicted frames (P), and bi-directional predicted frames (B).

byte 4								byte 5								byte 6								byte 7							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
temporal sequence number								frame type 1=I, 2=P 3=B, 4=D								VBV delay								---							

Figure 4-7 PES Picture Header [37]

Additional fields may be appended beginning at byte 7, bit 2. If the frame type = 2 (P) or 3 (B) the four bits shown in Figure 4-8 are appended to the header. This field is used by MPEG-1 only; for MPEG-2 it should be set to 0 1 1 1.

3	2	1	0
full_pel_forward_vector	forward_f_code		

Figure 4-8. Additional Fields for PES Header [37]

If the frame type = 3 (B) the additional 4 bits shown in Figure 4-9 are appended to the header.

3	2	1	0
full_pel_backward_vector	backward_f_code		

Figure 4-9. Additional Fields for B-Frames for PES Header [37]

Finally, if the next bit is "1" (extra_bit_picture) it is followed by 8 bits of "extra" data (discarded by decoders). This continues until a "0" bit is encountered.

Figure 4-10 shows an example of the PES picture header.

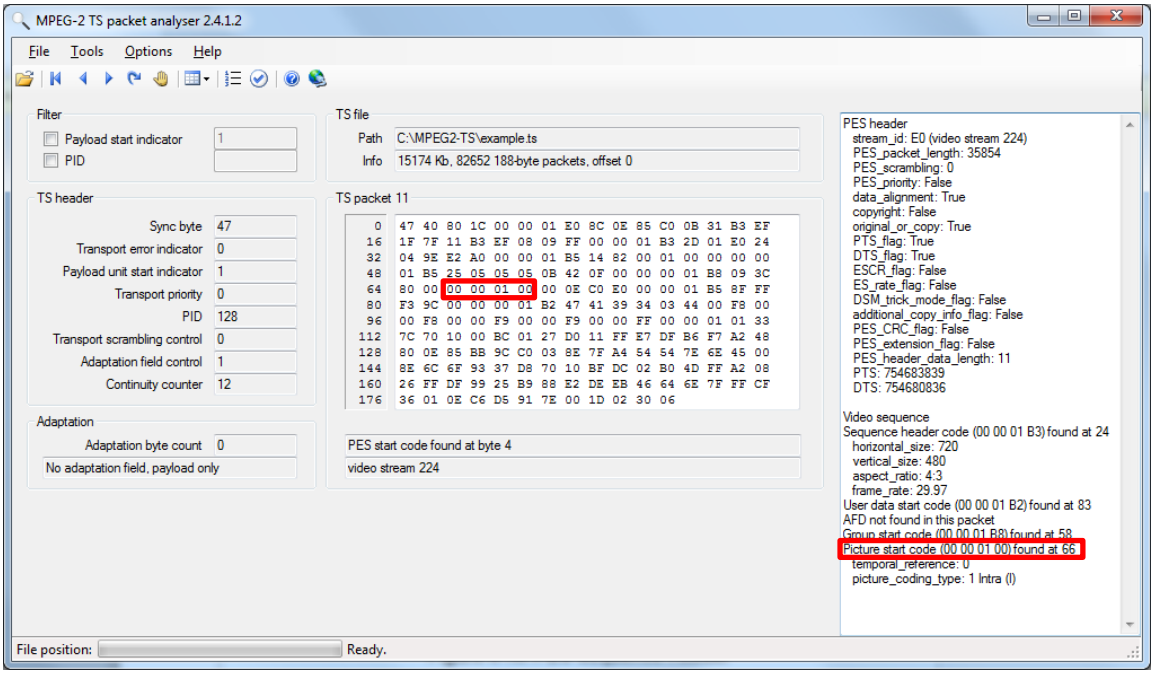


Figure 4-10 Example of PES Picture Header

RISKS AND RECOMMENDATIONS

Data Hiding: Frame types other than I, P, or B may contain hidden data and may be ignored by the receiver.

1. **Validate**: Check that the frame type is intra-coded (I), predicted (P), or bi-directional predicted (B).

2. **Reject**: Reject the file if any frame type is not I, P, or B.

PRODUCT

MPEG-2

LOCATION

Payload of TS packet.

4.1.3.3 Sequence Header

OVERVIEW

The variable length sequence header is identified by PES packet header start code 0x000001B3 (as bytes 0-3) and contains codes to identify the parameters of each video frame as shown in Figure 4-11.

byte 4								byte 5								byte 6								byte 7							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
horizontal size												vertical size												aspect ratio				frame rate			

byte 8								byte 9								byte 10								byte 11							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
bit rate																1	VBV buffer size												constrained parameters flag	load intra quantizer matrix	(load non-intra quantizer matrix)

Figure 4-11 Sequence Header [37]

Table 4-2 shows permissible values for the aspect ratio and frame rate.

Table 4-2 Sequence Header Aspect Ratio and Frame Rate Values [37]

Code	Aspect Ratio	Frame Rate
0	forbidden	forbidden
1	1:1	24000/1001 (23.976)
2	4:3	24
3	16:9	25
4	2.21:1(not used in DVD)	30000/1001 (29.97)

5	reserved	30
6	reserved	50
7	reserved	60000/1001 (59.94)
8	reserved	60
9	reserved	reserved
	:	
15	reserved	reserved

Figure 4-12 shows an example of the sequence header.

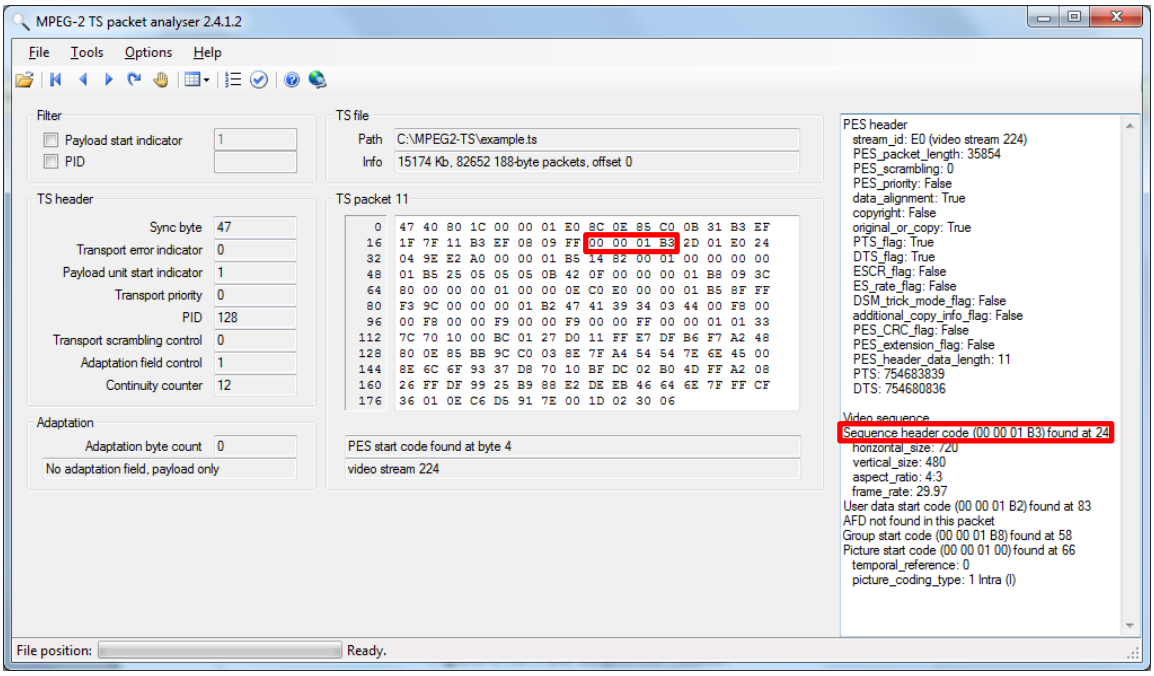


Figure 4-12 Example of Sequence Header

RISKS AND RECOMMENDATIONS

Data Hiding: Frame sizes of unequal dimension or an impermissible aspect ratio or frame rate can be exploited to hide data.

1. **Validate**: Check that all frames have the same horizontal and vertical size.

2. **Validate**: Check that the aspect ratio and frame rate are valid values from Table 4-1.

3. **Validate**: Check that the aspect ratio and frame rate are constant through the TS in each sequence header.

4. **Reject**: Reject the file if not all frames have the same dimensions through the TS, or if the aspect ratio and frame rate are not permitted.

PRODUCT

MPEG-2

LOCATION

Payload of TS packet.

4.1.3.4 Extension Headers

OVERVIEW

PES packet header start code 0xB5 identifies a variety of extension headers, which are denoted by the first four bits as shown in Figures 4-13, 4-14, and 4-15.

byte 4				byte 5				byte 6				byte 7			
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0001				profile and level				progressive_sequence				chroma_format				horizontal size extension				vertical size extension				bit rate extension							

byte 8								byte 9							
5	4	3	2	1	0	7	6	5	4	3	2	1	0		
vbv buffer size extension								low delay		frame rate extension n				frame rate extension d	

Figure 4-13 Sequence Extension Header [37]

byte 4								byte 5								byte 6								byte 7							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0010				video format				color description flag **=0**								(not present)																	
								=1								color primaries								transfer characteristics							
																								matrix coefficients							

byte 5 byte 8								byte 6 byte 9								byte 7 byte 10								byte 8 byte 11							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
display horizontal size									1	display vertical size														0	0	0					

Figure 4-14 Sequence Display Extension Header **[37]**

byte 4								byte 5								byte 6							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
1000				f_code[0][0] (forward horizontal)				f_code[0][1] (forward vertical)				f_code[1][0] (backward horizontal)				f_code[1][1] (backward vertical)				intra_DC_precision		picture_structure	

byte 7							
7	6	5	4	3	2	1	0
Top_Field_First	frame_pred_frame_dct	concealment_motion_vectors	q_scale_type	intra_vlc_format	alternate_scan	Repeat_First_Field	chroma_420_type

byte 8								byte 9								byte 10							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
progressive_frame	composite_display =0	0 0 0 0 0 0						(not present)															
	=1	v_axis	field_sequence		sub_carrier	burst_amplitude		sub_carrier_phase								0	0						

Figure 4-15 Picture Coding Extension Header [37]

RISKS AND RECOMMENDATIONS

Data Hiding: Decoders may ignore an extension header other than the sequence extension, sequence display extension, or picture extension headers. This may permit data hiding.

1. **Validate**: Check that the extension header code is 0001, 0010, or 1000.

2. **Reject**: Reject the file if the extension header code is not 0001, 0010, or 1000.

PRODUCT

MPEG-2

LOCATION

Payload of TS packet.

4.1.3.5 Group of Pictures (GOP) Header

OVERVIEW

A Group of Pictures (GOP) contains a group of I, B, and P frames. A GOP starts with with an I frame. Each GOP starts with a header which can be found in the MPEG-2 payload with the value 0x000001B8. This construct covers the fields that follow the GOP start code. As shown in Figure 4-16, the fixed-length GOP header contains three flags and the time stamp for the first frame. When the transmission buffer is full because of insufficient bandwidth, the streaming scheduler uses the drop frame flag to reduce the transmitted bit rate by dropping video frames. Frame data will follow and may span across multiple packets.

byte 4								byte 5								byte 6								byte 7											
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**		
drop frame flag	hour (0-23)							minute (0-59)								**1**	second (0-59)							frame (0-59)								closed GOP	broken GOP	**0 0 0 0 0**	

Figure 4-16 Group of Pictures (GOP) Header [37]

Figure 4-17 shows an example of the GOP header.

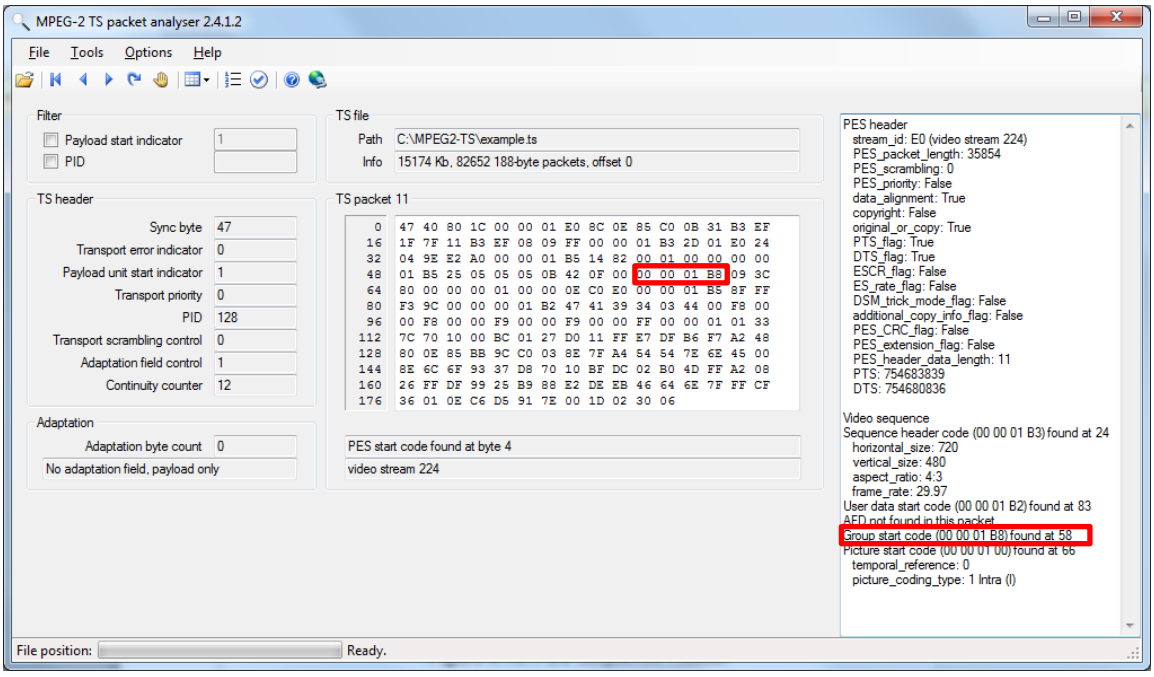


Figure 4-17 Example of PES Group of Pictures (GOP) Header

RISKS AND RECOMMENDATIONS

Data Hiding: Dropped frames can contain hidden data.

1. **Validate**: Check if the drop frame flag is not set.

2. **Reject:** Reject the file if the drop frame flag is set too often, e.g., 10 frames/sec.

Data Attack: Changing the timestamp or frame number to a value that exceeds what a program expects may introduce a data attack risk.

3. **Validate**: Check that the hour, minute, second, and frame fields are within the range shown in Figure 4-16.

PRODUCT

MPEG-2

LOCATION

Payload of TS packet.

4.1.4 TimeStamp Information

4.1.4.1 User Data Header

OVERVIEW

The User Data Header is identified by the value 0x000001B2 in a MPEG-2 compressed elementary stream. This is the start code prefix followed by the user data start code (0xB2). This is the same location used for closed captioning discussed in construct 4.1.1.4. Since this is a user data header, MISB uses this field to provide time stamp information [35]. This field is defined by MISB ST 0604.3 for Precision Time Stamp Information and is represented by 28 bytes. The first 16 bytes represent a Time Stamp Identifier. The Time Stamp Identifier is followed a by a 1 byte Time Stamp Status field. The header concludes with an 11 byte start-code emulation modified Precision Time Stamp. The timestamp will apply for a particular frame of motion imagery so it may be present before a block of image data is defined.

RISKS AND RECOMMENDATIONS

Data Hiding: The user data header can include any information and may be ignored if the application is not designed to handle this header.

1. **Validate:** Check that the 16-byte Time Stamp Identifier is valid and follows the User Data header start code.

2. **Validate:** Check the Precision Time Stamp to ensure that it is a correct time stamp.

3. **Validate:** Check that the field is terminated after 28 bytes as defined by MISB.

4. **Remove**: Remove the User Data header and rebuild the MPEG2-TS stream.

PRODUCT

MPEG-2

LOCATION

TS packet. Located between a picture header and the picture data.

4.1.4.2 Supplemental Enhancement Information

The ITU-T Rec. H.264 implements a Supplement Enhancement Information (SEI) message portion of the H.264 elementary stream data field [35]. The user data unregistered SEI Message field (user_data_unregistered) provides user-defined data and a time code in H.264. The field is defined by a UUID, a 16-byte field, labeled

uuid_iso_iec_11578, which is then followed by a variable length data field. A requirement of MISB ST 0604.3 states that "a Precision Time Stamp consisting of Time Stamp Identifier, a Time Stamp Status and a start-code emulation-modified Precision Time Stamp shall be inserted into the H.264 elementary stream user data unregistered SEI Message Field *uuid_iso_iec_11578*, so that it relates to a specific frame" [35]. This information is similar to the timestamp information in the MPEG-2 user data header.

RISKS AND RECOMMENDATIONS

Data Hiding: The SEI Field is designated as a user data unregistered SEI message which can provide any information, possibly hidden data if the decoder is not capable of parsing this custom data.

1. **Validate:** Check that the 16-byte Time Stamp Identifier is valid and follows the uuid_iso_iec_11578 UUID.

2. **Validate:** Check the Precision Time Stamp to ensure that it is a correct time stamp.

3. **Validate:** Check that the field terminates after the start-code emulation-modified Precision Time Stamp.

4. **Remove**: Remove the SEI from the H.264 stream and rebuild.

PRODUCT

H.264

LOCATION

H.264 elementary stream data field – *uuid_iso_iec_11578*

4.1.5 H.264/AVC

MPEG video presents a rich medium for data hiding because it permits data embedding rates much higher than other digital media. Without changing the perceptual quality of the target video, many video data hiding schemes either exploit motion compensation used in the video codec (interframe data hiding), or insert data into individual video frames (intraframe data hiding). Because it is impractical to detect every possible data hiding scheme applied to MPEG video files, the data hiding risks enumerated in this document are not exhaustive.

Section 4.1.5.1 covers the Network Abstraction Layer (NAL) defined by H.264/AVC. Sections 4.1.5.2 and 4.1.5.3 present two block-oriented interframe and intraframe data hiding schemes. These schemes override the optimum sequence of block types normally selected for suitability to the video frame contents during MPEG encoding and substitute an arbitrary sequence of block types that represent the bits of a hidden

message. This can be done without great change to the original picture contents inside these blocks. These two re-blocking schemes can be defeated by decoding and re-encoding the video frames, which removes the hidden data.

Double-encoding detection is a general forensic tool that indicates data tampering of some form has occurred. Section 4.1.5.4 discusses how to detect double MPEG encoding using DCT coefficient histograms.

Digital watermarking is often employed to embed secondary data in digital video for a variety of applications, including ownership protection, authentication, access control, and annotation. The increased use of watermarking has been matched by a corresponding increase in data hiding applied to video.

Robust watermarking is currently being developed as a global effort. Audiovisual watermarks are designed to be visually and audibly imperceptible; they can withstand sophisticated attacks that attempt to remove the watermarks. The watermarks are indelible in order to prevent commercial video piracy. For example, pirated copies of a video with an identifying watermark can be traced to the owner of the original video.

In principle, watermarking methods can be adapted for a different, non-commercial purpose: to create covert channels for hiding data. In this case, the watermark key or pattern is kept covert. However, so-called blind watermark recovery algorithms are being developed that do not require a key or previous knowledge of the watermark pattern. (They may require knowing the position or format of the watermark.) Section 4.1.5.5 gives an example of such a blind or keyless watermark recovery algorithm that can be used to detect a broad class of covert watermarks.

The last example of covert data attack is motion-vector tampering. This can be universally detected and defeated by the recent add-or-subtract-one (AoSO) algorithm, introduced in Section 4.1.5.6.

4.1.5.1 Network Abstraction Layer (NAL)

OVERVIEW

The NAL is defined by the H.264/AVC Video Coding Standard. The NAL is a packet-based structure that carries video and is used by decoders to interpret the bytestream. The NAL provides the ability to "map H.264/AVC Video Coding Layer (VCL) data to a transport layer such as MPEG-2" [11]. The NAL is designed to be simple and introduce low overhead as it only implements a four byte header: a 3-byte Start Code Prefix (0x000001) plus a single byte NAL unit type value. The NAL header defines the type of data contained within the NAL packet that follows. The data that follows is referred to as the Raw Byte Sequence Payload (RBSP). Table 7-1 of the H.264 standard officially lists the NAL unit types codes [3]. Values of 0, 24-31 are defined as "Unspecified" and do not appear to be used by the standard. A NAL unit type equal to 17, 18, 22, or 23 is defined as "Reserved" by the standard. For reference to a previous construct, a NAL

unit type of 6 implements the SEI structure which was covered previously in construct 4.1.4.2.

RISKS AND RECOMMENDATIONS

Data Hiding: Unspecified or reserved NAL types may introduce hidden data as the decoder may not process this data.

1. **Validate:** Check that the NAL unit type is a valid type from Table 7-1 of the H.264 standard. This includes values 1-16 and 19-21.

2. **Reject:** Reject files that contain unspecified NAL unit types.

3. **Reject:** Reject files that contain reserved NAL unit types.

PRODUCT

H.264 AVC NAL

LOCATION

The first four bytes of the NAL unit in the payload of a TS packet.

4.1.5.2 Interframe Data Hiding by Macroblock Override

OVERVIEW

Interframe data hiding often exploits the video codec by manipulating the frame-to-frame motion vectors to embed hidden data [21]. The video encoder defines video frame pixel macroblocks to capture motion vectors based on the smallest interframe variation.

An adversary may hide data through the H.264 video codec macroblock design by selecting motion compensation macroblocks based not on best match, but on the hidden message to be embedded. For example, an attacker could assign a binary code to the macroblock type as in the sequence depicted in Figure 4-18.

Macroblock Type	Binary Code
16 x 16	00
16 x 8	01
8 x 16	10
8 x 8	11

The hidden message is then converted to binary:

… 0011011001001000 …

Bits are then separated into pairs:

00	11	01	10	01	00	10	00

Pairs are then mapped into macroblock type:

16 x 16	8 x 8	16 x 8	8 x 16	16 x 8	16 x 16	8 x 16	16 x 16

Figure 4-18 Example of H.264 Interframe Data Hiding

The hidden message can then be recovered from the encoded stream by identifying which macroblock was used for interframe prediction.

RISKS AND RECOMMENDATIONS

Data Hiding: The attacker can assign a binary code to certain blocks so that they form a hidden message when combined. This creates a covert channel by spreading hidden data over the selection bits.

1. **Replace**: Decode the video frames and re-encode them with either MPEG-2 or H.264 video encoding.

PRODUCT

MPEG-2 and MPEG-4 part 10 (H.264 AVC)

LOCATION

Across numerous TS packets.

4.1.5.3 Intraframe Data Hiding by I4 Mode Override

OVERVIEW

Intraframe data hiding often exploits the intraframe 4x4 prediction modes (I4 modes) of the video encoder by selecting the I4 mode based not on least distortion, but on the hidden message to be embedded [22].

An attacker can exploit the H.264 video codec intraframe I4 prediction modes by dividing the nine I4 modes into two groups to form the mapping rule between these modes and the bits the attacker wants to conceal. For example, the intraframe prediction blocks are computed based on the I4 modes as shown in Figure 4-19.

0: Vertical	1: Horizontal

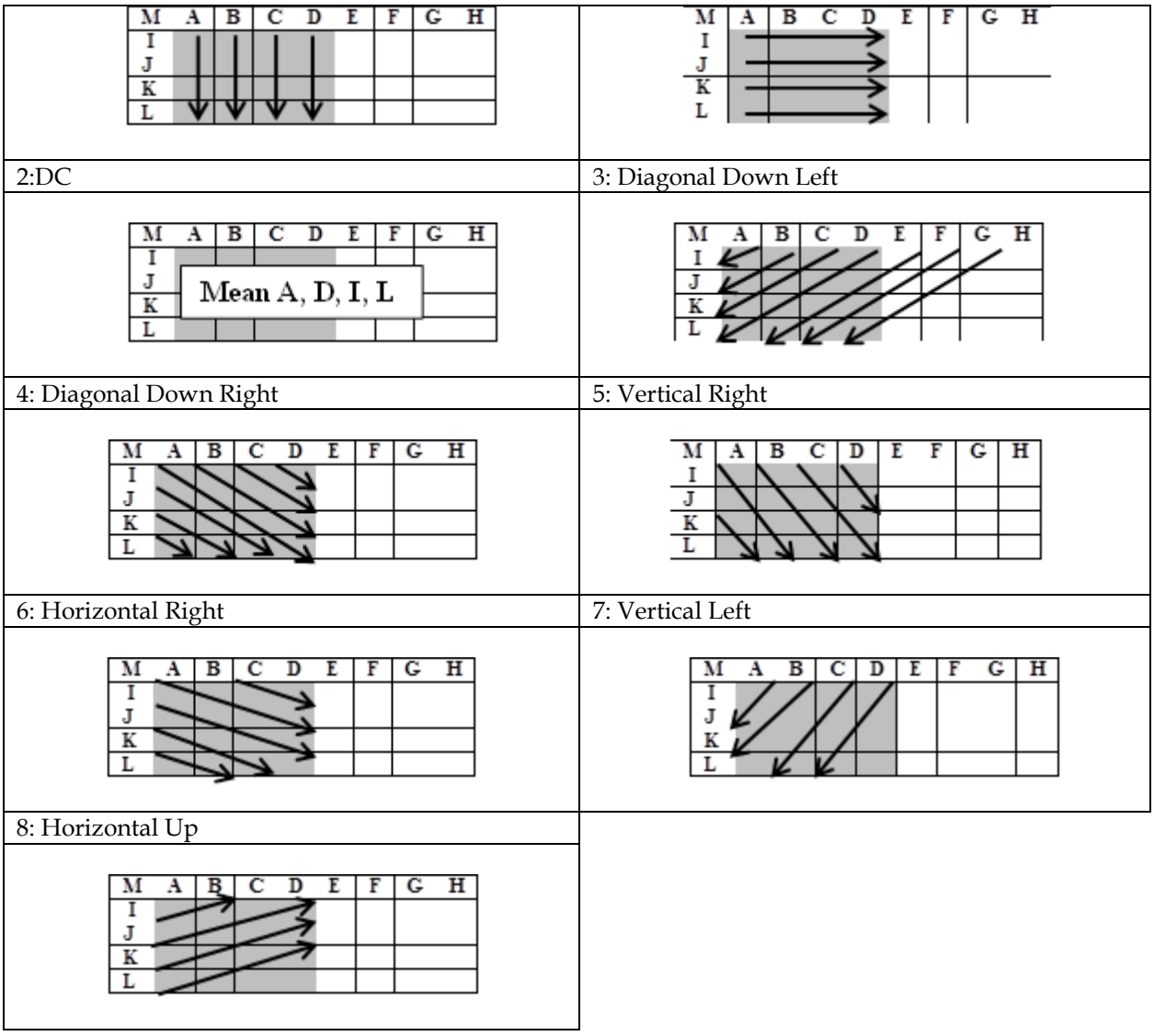


Figure 4-19 H.264 Intraframe Prediction

Data can be hidden by dividing the nine I4 modes into two sets, e.g., A = {0, 1, 3, 8} and B = {2, 4, 5, 6, 7}. The hidden message can then be recovered from the encoded stream by identifying which I4 mode was used for intraframe prediction.

RISKS AND RECOMMENDATIONS

Data Hiding: The attacker selects the 4x4 pixel blocks used by the H.264 video codec for intraframe prediction based not on smallest difference, but rather on hidden data bits. This creates a covert channel by altering the video codec intraframe prediction with imperceptible video quality reduction.

1. **Replace**: Decode the video frames and re-encode them with either H.262 or H.264 video encoding.

PRODUCT

MPEG-2 and MPEG-4 part 10 (H.264 AVC)

LOCATION

Across numerous TS packets.

4.1.5.4 Double MPEG Encoding as Evidence of Data Tampering

OVERVIEW

Most methods for tampering with an MPEG-compressed video require decoding the video first and then tampering with and re-encoding the video. Thus, double encoding of a video, if detected, serves as a general warning or indication of probable data tampering, ranging from data hiding to image forgery attacks. To identify the type of tampering that occurred requires further steganalysis. But using evidence of double encoding to detect that a video has been compromised offers a valuable first line of defense, since tampering attacks can be carried out in many different ways and for many purposes. Without this general defense, a filter would have to rely on a complete set of specialized steganalysis techniques, able to directly detect (and counteract) every specific scheme of video attack. As the authors of [27] explain:

> Every time a video is tampered [with], it should be firstly decoded to frame sequence, and then re-encoded to compressed format after forgery (e.g. frame deletion, frame insertion, and local manipulation). Hence double compression detection can be viewed as the first step of video forensic research.

Double encoding creates video artifacts, such as frame blockiness (discontinuities between block edges) and periodic features in the MPEG discrete cosine transform (DCT) coefficient distribution. These artifacts can be exploited to detect double MPEG encoding of a video.

Software can detect double-encoding of the MPEG video after data hiding or other tampering by applying techniques based on the DCT histograms. To understand how to find double-encoding or double-compression in the DCT histogram, consider how MPEG compresses a video. MPEG uses the first frame in a GOP and subdivides the luma (Y) or chroma (Cr, Cb) component of the frame into 8 x 8 or 16 x 16 blocks for intra-frame (I) prediction, leaving residual data blocks. The approximate DCT is taken of the residual blocks. Then the DCT transform coefficients c_{ij} are quantized:

$$c^q_{ij} = \text{round}(c_{ij} / Q)$$

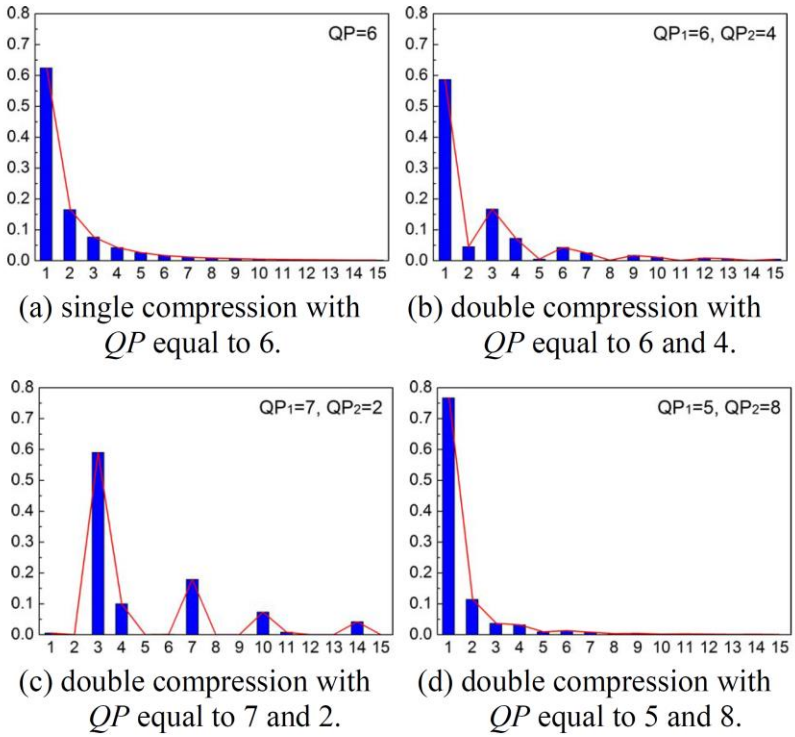
The initial dc (direct current) coefficient of the block is handled separately. The quantization step Q comes from a table, and is basically a linear function of the logarithm of the quantization parameter QP selected to compress the data. The

quantized coefficients c^q_{ij}, ready for entropy encoding, represent the result of a single MPEG compression of one data block of the video frame in a GOP. (The DCT transform of P and B frames is similar, except that motion vectors are also used.)

Now consider an attacker who decodes and encodes the MPEG video a second time. To decode, the attacker multiplies the quantized coefficients c^q_{ij} times Q, and computes the inverse DCT. After re-encoding, even if the picture blocks stay aligned and the coefficients are not tampered with, the attacker requantizes the video using a new step size Q' to obtain a new value,

$$c^{qq'}_{ij} = \text{round}(\text{round}(c_{ij}/Q)\cdot Q/Q')),$$

Thus, after re-encoding, the re-quantized values of the DCT coefficients have changed [28],[29]. By collecting the histograms of these values for all blocks of all I-frames of the video, a filter can observe artifacts such as gaps and periodic spikes in the histogram, as shown in Figure 4-20.



(a) single compression with *QP* equal to 6.

(b) double compression with *QP* equal to 6 and 4.

(c) double compression with *QP* equal to 7 and 2.

(d) double compression with *QP* equal to 5 and 8.

Figure 4-20 DCT Histograms of Single- and Double-encoded Videos [28].

In Figure 4-20, each histogram is averaged over 30 videos, using a Gnu/Xvid MPEG-4 codec. The authors of [28] also collect the histograms of first and second differences of neighboring DCT coefficients. Using the histograms as features, they train a classifier that selects the quantization parameter QP_1 used in the original MPEG compression. If double encoding occurred, this value will probably differ from the known, second

quantization parameter QP_2, which is available from the received MPEG data stream. In [29], DCT block histograms are used to detect and localize double encoding in pictures. The DCT histograms also have other, similar applications, for example, to detect contrast–enhancement in highly compressed JPEG pictures [30].

RISKS AND RECOMMENDATIONS

Data Hiding: In general, data tampering entails double-MPEG encoding to inject hidden data.

1. **Validate:** Implement and run an algorithm, such as the DCT histogram classifier in [27] or methods in [28] and [29], to test for DCT artifacts left by double encoding.

2. **External Filtering Required:** If double-encoding is detected, this indicates probable compromise of the MPEG video. Perform further steganalysis to identify the type of data tampering and to remove all hidden data.

3. **Reject:** Reject the video if analysis reveals double compression.

PRODUCT

MPEG-2 and MPEG-4 part 10 (H.264 AVC)

LOCATION

Video frame DCT blocks

4.1.5.5 MPEG Watermarking for Data Hiding

OVERVIEW

Attackers can also adapt video watermarking techniques, developed to protect against piracy of commercial videos, to embed hidden data [32]. For example, multiple watermarks may be superimposed upon the blocks of DCT coefficients of a frame of an MPEG-2 or H.264/AVC video. This begins by reshaping the 2D-DCT of block m as a vector $\mathbf{x}(m)$. The linear model for adding K watermarks to each block's DCT vector $\mathbf{x}(m)$ (the host vector for the watermarks) for M DCT blocks, is given in [32] as:

$$\mathbf{y}(m) = \sum_{k=1}^{K} A_k b_k(m)\mathbf{s}_k + \mathbf{x}(m) + \mathbf{n}(m), m = 1, 2, \ldots, M,$$

where the watermark k has amplitude A_k, message or data bit $b_k(m)$ carried in block m, and spreading vector $\mathbf{s}_k$ having unit norm. The last term $\mathbf{n}(m)$ is additive noise.

The authors of [32] demonstrate that their multicarrier, iterative generalized least squares (M-IGLS) recipe can separate and recover the watermarks' message bits $b_k(m)$

and scaled spreading (carrier) vectors $\mathbf{v}_k = A_k\mathbf{s}_k$. They alternately solve for the message bits (±1's) given the spreading vectors, then solve for the spreading vectors given the bits, etc., until the values converge. They use the block vectors' autocorrelation matrix R_{xx} to optimally weight the least squares solutions.

RISKS AND RECOMMENDATIONS

Data Hiding: An MPEG video may carry covert data in the form of imperceptible watermarks, which are difficult to remove by simple counterattacks such as re-encoding.

1. **Remove:** Remove the watermark carriers/messages from each block by subtracting their estimated values from the block DCT coefficients.

2. **External Filtering Required:** Implement and run a blind recovery algorithm, such as multicarrier, iterative generalized least squares (M-IGLS) [32], using several initializations, to recover multiple watermark carrier vectors and their hidden message bits for every block in each video frame.

PRODUCT

MPEG-2 and MPEG-4 part 10 (H.264 AVC)

LOCATION

Video frame 2D-DCT blocks

4.1.5.6 MPEG Motion Vector Tampering

OVERVIEW

Interframe data hiding often exploits the MPEG video format by manipulating the frame-to-frame motion vector (MV) content in various ways to embed hidden data. For example, an attacker can exploit MPEG-2 or H.264/AVC video by tampering with the motion vectors' horizontal and vertical components V^h and V^v :

$$\begin{cases} SV_{k,l}^h = V_{k,l}^h + \alpha_{k,l}^h \eta_{k,l}^h \\ SV_{k,l}^v = V_{k,l}^v + \alpha_{k,l}^v \eta_{k,l}^v \end{cases}$$

where the stego noise η is -1, 0, or 1, and α is 1 or 0 to include the noise value η or not.

The authors of [33] recognized that:

- The untampered MVs are almost always locally optimum; i.e., the MVs leave the minimum prediction error (PE) the residual block to transform by DCT.
- The PE (DCT block) must also be adjusted to agree with the tampered MV.

They then developed two Add or Subtract One (AoSO) features that provide a universal test for motion vectors modified in various ways [33].

RISKS AND RECOMMENDATIONS

Data Hiding: MPEG video can carry covert data stored by adding steganographic noise to motion vectors.

1. **Remove:** Remove the steganographic noise from the MV for each block and restore the block PE to be near optimal.
2. **External Filtering Required:** Implement and run the AoSO algorithm [33] to test each non-I (P,B) frame for motion vector tampering.

PRODUCT

MPEG-2 and MPEG-4 part 10 (H.264/AVC)

LOCATION

Video P and B frame pixel macroblocks.

4.2 Program Stream and Non-Standard Constructs

This section presents unique PS constructs and non-standard constructs that are not covered by the TS file format. Some of the TS constructs can be generally applied to data in a PS, taking into account the differences between the streams discussed in Section 3. This section focuses on some unique features that are mix between the MPEG PS and DVD-Video/Blu-ray specifications.

4.2.1 Multiple Angles and Additional Video Sources

OVERVIEW

Multiple angles is a feature of several DVD-Video movies but is not covered by the MPEG specification. Multiple angles allows a user to switch the view from the primary video stream. One example of this feature is a DVD video of a concert, where each video stream focuses on a particular member of the band. Some movies provide alternate angles for different scenes to display text in different languages. Otherwise, this is not a widely used feature.

Blu-ray videos support a feature called Bonusview(video and audio), which support a type of picture in picture. This was originally called Blu-Ray Profile 1.1. This is a separate stream of data from the primary video/audio stream. This also requires a special player in order to utilize this feature.

All of these features are not defined formally in any MPEG specifications, but their data formats utilize the video encoding defined in an open specification. Many of these features take advantage of multiple video streams and private data streams, although they are not defined in an open specification. They appear alongside the primary MPEG video and can be switched to a certain times by the video player.

RISKS AND RECOMMENDATIONS

Data Hiding: Any type of alternate angle, or additional video/audio streams can introduce a form a data hiding. Many users may not be aware or additional video and audio streams and it could be used as a covert channel.

1. **Remove:** Remove any additional video/audio streams that are not pertinent for the playback of the primary video/audio.

PRODUCT

MPEG, DVD-Video, and Blu-ray

LOCATION

Additional streams can be present through the file alongside the primary video/audio stream.

4.2.2 Subpictures

OVERVIEW

Subpictures is a feature within DVD-Video, supported by MPEG, that displays an overlaid object on top of the primary video stream. This can be used for user menus, sub-titles, and basic animations. They are encapsulated into a structure or sequence of Sub-Picture Units (SPU). They are located in private stream 1 in an MPEG stream. In some cases, such as a menu button, SPUs are critical for functioning DVD videos as this allows the user to navigate the contents of the video. The SPU implements a data structure shown below in Table 4-3. [41] The SPU contains a Run Length Encoded (RLE) compressed bitmap of pixels as well as a table of commands that are sent to the video player. Examples of commands are SET_COLOR, Start Display, and setting the display area.

Table 4-3. Sub-Picture Data Structure

Field	Description
SPUH	Sub-Picture Unit Header, contains the size of the SPU and the starting address of the Display Control Sequence Table (DCSQT)
PXD	Pixel Data. RLE compressed pixel data.
SP_DCSQT	Sub-Picture Display Control SeQuence Table. This table contains a list of commands that are sent to the decoder. Each entry contains a header defining a delay value (wait before execute command) and an offset value to the next table entry.

RISKS AND RECOMMENDATIONS

Data Hiding: It is possible that subpictures may not appear to the user or the user has disabled them. They are an external data source outside of the primary video stream. They could be used to store hidden information. Conversely, subpictures can be used to obscure information in the original video stream since it is displayed above the primary content.

1. **Validate:** Check that the size of the sub-picture data matches the size of the SPU.
2. **Validate:** Check that any command in the SP_DCSQT that sets the display area is visible.
3. **Remove**: Remove all SPU structures from the MPEG stream. This may impact the usability of certain videos.
4. **External Filtering Required:** Extract the RLE bitmap encoded within the SPU and pass to an external filter.

PRODUCT

MPEG, DVD-Video, and Blu-ray

LOCATION

Subpictures are spread out across private stream 1 in MPEG data.

4.3 MPEG-2 Metadata Constructs

4.3.1 KLV Metadata

This subsection addresses the KLV metadata typically embedded in MPEG-2 TS files containing motion imagery. Metadata is collected by the mission computer or input from other sources (e.g., an operator). All metadata must abide by the MISB standards for syntax and semantics.

All MISB metadata is encoded using KLV. This binary format was chosen because of its efficiency, extendability and decode robustness. The MISB maintains a listing of all metadata keys in a dictionary [15] with corresponding EGs [16] that describe the processes for requesting, assigning, approving, and managing metadata identifiers (KLV keys).

While many KLV elements supported by the MISB describe important mission data, not all are mandated to meet MISB compliance; however, certain mission and security related metadata must be present [17].

4.3.1.1 Metadata Key Present in MISB Standard 0807 Dictionary

OVERVIEW

The MISB maintains a listing of all metadata keys in MISB STD 0807 – KLV Metadata Dictionary [15], with a corresponding document, MISB EG 0607 – MISB Metadata Registry and Processes [16], that describes the fundamentals of the metadata dictionary. A filter should inspect MPEG-2 files to ensure that all the metadata keys are included in the dictionary.

A MISB metadata key might implement a form of nested KLV values, referred to as Tag-Length-Value (TLV). This is a similar form nested within the value of a higher level KLV metadata entry. Keys and Tags should be treated similarly with regards to filtering.

RISKS AND RECOMMENDATIONS

Data Hiding: Unknown metadata keys may introduce a hidden data risk because they could include arbitrary unused data.

1. **Validate**: Check that the metadata key and tags are on defined whitelist.

2. **Validate**: Check that the length is correct and after that number of bytes, the next valid key or tag is present.

3. **Validate**: For each key or tag value present in the stream, validate its data type. If defined as an ASCII string, validate each byte of the value is a valid ASCII character. If date or time, validate it is in the correct format.

4. **Remove**: Remove any metadata key or tag not found in the whitelist.

5. **Remove**: Remove the entire KLV metadata stream if any length field is incorrect. This may require rebuilding the transport stream since it spans several MPEG-2 TS packets.

6. **Replace:** Replace any metadata packet in MPEG-2 TS with a NULL packet.

7. **Reject**: Reject the file if the number of metadata keys not included in the dictionary exceeds a specified threshold.

8. **Reject**: Reject the file if the length field for the KLV is incorrect.

Data Disclosure: Metadata may introduce data that the media player overlooks and may be accidentally included in the file.

9. **Remove:** Remove the entire KLV metadata stream and rebuild the file.

10. **Remove**: Remove any free-text KLV or TLV items.

11. **External Filtering Required**: Pass the value of the KLV or TLV metadata to an external filter.

PRODUCT

MPEG-2

LOCATION

KLV embedded metadata

4.3.1.2 Metadata Checksum

OVERVIEW

KLV metadata can provide a checksum that ensures the metadata has not been altered. An application can verify this by computing the checksum and then comparing it to the checksum value provided. MPEG-2 files should be inspected to ensure that the metadata item computed checksum is identical to the included checksum.

The Unmanned Aircraft System (UAS) Local Set KLV metadata contains a 16-bit checksum. This checksum value covers the entire KLV block starting with the 16-byte

key, BER length, and everything in the value field except for the actual checksum value. The value in the KLV will contain TLV, so the final Type and Length of the checksum TLV are included in the computation of the checksum as shown in **Figure 4-21**.

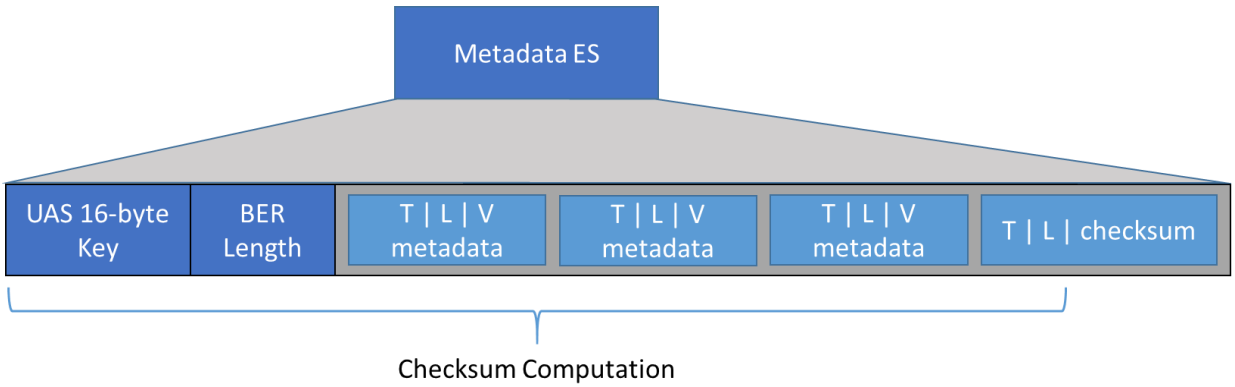


Figure 4-21. KLV Checksum Computation [18]

The checksum is calculated simply by adding every 16-bit chunk in the area from Figure 4-21 to produce a final 16-bit value, the checksum. An example algorithm is given in MISB ST 0601.8 [18].

RISKS AND RECOMMENDATIONS

Data Hiding and Data Attack: An invalid checksum indicates that the data stream was modified at some point, and may include hidden data or malicious content.

1. **Validate**: Check that the metadata item computed checksum is identical to the included checksum.

2. **Remove:** Remove metadata items whose computed checksum differs from the included checksum.

3. **Reject:** Reject the file if the number of metadata items with invalid checksum exceeds a specified threshold.

PRODUCT

MPEG-2

LOCATION

KLV embedded metadata from 0601.8 UAS Datalink Local Set. The checksum is tag #1 within metadata set.

4.3.1.3 Metadata Value

OVERVIEW

MPEG-2 files should be inspected to ensure that all metadata item values are consistent with those in the KLV Metadata Dictionary [15]. This section generically applies to any metadata value found across numerous MISB standards.

RISKS AND RECOMMENDATIONS

Data Hiding: A KLV metadata item value outside the acceptable range indicates that a field many include hidden data. A KLV metadata item value not on an allowable list can be an attempt to hide data.

1. **Validate**: Check that the metadata item value is in the proper numerical range if applicable.

2. **Validate**: Check that the metadata item value is on an allowable list or enumerated type defined by an appropriate MISB standard.

3. **Validate**: Check that the metadata item value is correctly formatted according to its definition in the MISB standard.

4. **Remove**: Remove metadata items with numerical values outside the proper range, if applicable.

5. **Remove**: Remove metadata items with invalid or incorrectly formatted values.

6. **Replace**: Replace invalid or incorrect metadata items with padding or null bytes (0xFF or 0x00). This should be performed cautiously depending on the field type.

7. **Reject**: Reject the file if the number of metadata items with values outside the proper range exceeds a specified threshold, if applicable.

Data Hiding and Data Disclosure: The value field might contain free-text, sensitive data, or data included accidentally.

8. **External Filtering Required:** Pass each free-text value field to an external filter.

PRODUCT

MPEG-2

LOCATION

KLV embedded metadata

4.3.1.4 Security Markings

OVERVIEW

MISB ST 0102.11 – Security Metadata Universal and Local Sets for Digital Motion Imagery specifies the metadata keys that implement security markings [40]. These are a disclosure risk should the data be passed to a destination not allowed to process videos at the security level. There are several metadata fields that describe the video content. The following image in Figure 4-22 is an example MPEG-2 TS video with its KLV data highlighted for this section. The red circles highlight the synchronization byte (0x47), the start of a TS packet. The green circle represents the start of an SMPTE key with its 4 byte identifier 0x060E2B34. The orange highlighted content is the string "UNCLASSIFIED//". This is the value from the 16-byte key 0x060E2B34010101030208020100000000 from MISB ST 0102.11. The next byte is the length of the value, which is equal to 0x0E. This is the correct length of the value field.

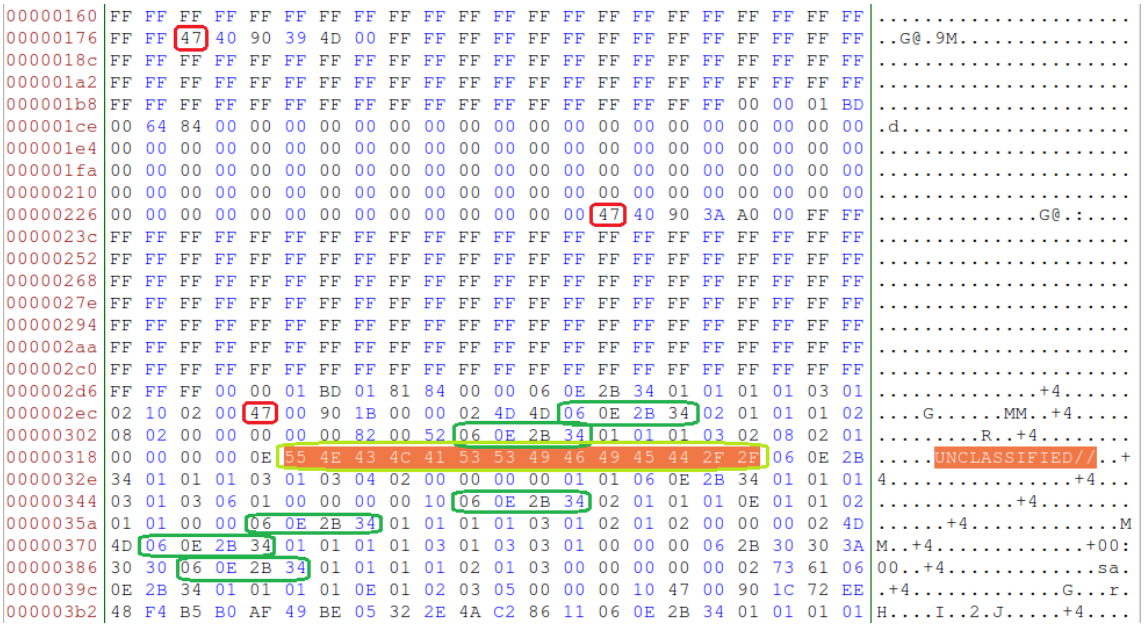


Figure 4-22. KLV Security Metadata

According to MISB ST 0102.11, this key has a list of allowed values. The string "UNCLASSIFIED//" is one of those allowed values so this metadata reflects and Unclassified video stream. Table 1 from MISB ST 0102.11 lists all the 16-byte keys that comprise the Security Metadata Universal Set Elements. Many of these fields have a list of allowed values or reference to other standard documents for a list of allowed abbreviations such as Country Codes. There are several date fields with respect to declassification and some free-text comment fields that allow an operator to enter any text. Several fields are required, some are optional, and some are required based upon context (e.g., if information is classified, a declassification date should then exist).

RISKS AND RECOMMENDATIONS

Data Disclosure – The fields present in this section define the classification and sharing guidelines for this content. If improperly marked, it may be difficult to determine if this video can be shared.

1. **Validate:** If the metadata field contains a list of allowed values in MISB ST 0102.11, check that the value is allowed.

2. **Validate:** Check that all required metadata fields are present.

3. **Validate:** Check for other metadata fields that have a specific structure (e.g., date or time) that it follows the correct structure from Table 1 in MISB ST 0102.11

4. **Reject:** Reject any file with an invalid or improper classification marking.

Data Hiding and Data Disclosure – Free-text information can be a form a hidden data if the viewer does not render the information and may also accidently release information if not checked.

5. **Remove:** Remove any free-text field in this metadata section.

6. **External Filtering Required:** Pass any free-text field to an external filter.

4.3.1.5 All MISB Standard 0902 Minimum Metadata Items Present

OVERVIEW

MISB STD 0902 - Motion Imagery Sensor Minimum Metadata Set [17] specifies the mandated set of KLV elements that characterize many of the dynamic parameters collected during a mission. This set is drawn from a more complete set defined in MISB STD 0601 - UAS Datalink Local Metadata Set [18]. The minimum set enables basic discovery and retrieval functionality in exploitation. Table 4-4 shows the MISB minimum metadata set from Standard 0902.1. The Tag-Length-Value (TLV) encoded data are shown as hexadecimal bytes.

Table 4-4 Minimum Metadata Set

Tag	Name	Value	Interpretation	TLV Hex Bytes
2	UNIX Time Stamp	1,231,798,102,000,000 microseconds	Mon Jan 12 2009 22:08:22 (UTC)	02 08 00 04 60 50 58 4E 01 80
5	Platform Heading Angle	0x71C2	159.9744 Degrees	05 02 71 C2
6	Platform Pitch Angle	0xFD3D	-0.4315251 Degrees	06 02 FD 3D
7	Platform Roll Angle	0x08B8	3.405814 Degrees	07 02 08 B8
13	Sensor Latitude	0x5595B66D	60.17682296 Degrees	0D 04 55 95 B6 6D
14	Sensor Longitude	0x5B5360C4	128.42675904 Degrees	0E 04 5B 53 60 C4
15	Sensor True Altitude	0xC221	14190.72 Meters	0F 02 C2 21
16	Sensor Horizontal FoV	0xCD9C	144.5713 Degrees	10 02 CD 9C
17	Sensor Vertical FoV	0xD917	152.6436 Degrees	11 02 D9 17
18	Sensor Rel. Azimuth Angle	0x724A0A20	160.71921147 Degrees	12 04 72 4A 0A 20
19	Sensor Rel. Elevation Angle	0x87F84B86	-168.79232483 Degrees	13 04 87 F8 4B 86
20	Sensor Rel. Roll Angle	0x00000000	0.0 Degrees	14 04 00 00 00 00
21	Slant Range	0x03830926	68590.98 Meters	15 04 03 83 09 26
22	Target Width	0x1281	722.8199 Meters	16 02 12 81
23	Frame Center Latitude	0xF101A229	-10.54238863 Degrees	17 04 F1 01 A2 29
24	Frame Center Longitude	0x14BC082B	29.15789012 Degrees	18 04 14 BC 08 2B
25	Frame Center Elevation	0x34F3	3216.037 Meters	19 02 34 F3
65	UAS LDS Version	0x02	MISB Standard 0601.2	41 01 02
1	Checksum	0xC84C	0xC84C	01 02 C8 4C

The TLV bytes are appended end-to-end, and together become the value portion of the enclosing KLV packet. Figure 4-23 shows 97 bytes of TLV data, encoded as the length of the KLV packet. The entire set starts with the 16-byte Universal Label (UL) key, followed by the length 0x61, followed by all the TLV bytes in order. In hex, the entire KLV packet is:

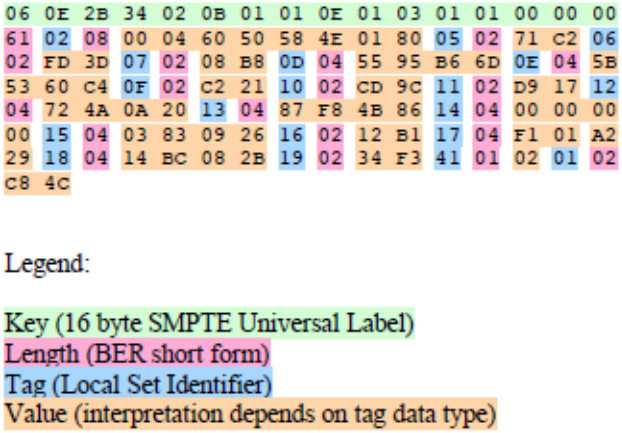


Figure 4-23 Example KLV Packet

RISKS AND RECOMMENDATIONS

Data Hiding and Data Attack: KLV metadata packets that do not include all of the metadata items in the Minimum Metadata Set indicate possible hidden content or embedded malicious content.

1. **Validate**: Validate that the embedded KLV metadata contain all minimum metadata items.

2. **Reject**: Reject the MPEG-2 file if any minimum metadata items are not present.

PRODUCT

MPEG-2

LOCATION

KLV embedded metadata

4.3.1.6 KLV Metadata Geolocation ID and Video Imagery

OVERVIEW

The KLV embedded metadata shown in Table 4-4 identify the time and geolocation of image frames. If a UAV camera captures important events on video , or if a UAV mission was deployed to observe a sensitive geographical region, then the UAV's

MPEG-2 video stream, together with its identifying KLV metadata, may disclose sensitive information.

Alternatively, the KLV metadata stream and the video and audio streams can be separated and stored in separate files with unrelated names. The stream files can each be disclosed separately, but the association of their filenames is kept as sensitive data. Note that, with some effort, an attacker could still re-associate the timestamps in both files. This can be prevented by removing or shifting the timestamps by several hours in one of the separated stream files (KLV or video/audio) .

RISKS AND RECOMMENDATIONS

Data Disclosure: A filter should screen UAV video/audio streams that cover sensitive events or scenes or mission destinations. Analyzing the time and geolocation features present in the associated KLV embedded metadata can prevent unintentional disclosure of sensitive information.

1. **Validate**: Check that the recorded timestamps and geolocations (latitude/longitude, altitude) in the frame KLV metadata or given indirectly by the sensor and pointing KLV metadata do not lie within sensitive timeframes and regions of interest (ROIs) specified, e.g., by latitude/longitude polygons or by circular error probable (CEP) circles.

2. **Remove**: Remove video/audio frames and associated KLV metadata that cover sensitive events or ROIs whose space and time coordinates are known and should not be disclosed.

3. **Replace:** Separate the sensitive video/audio streams and their associated embedded KLV metadata streams into two distinct MPEG-2 stream files, using dissociated filenames. Only the association of the separated MPEG streams' filenames is then kept as sensitive information.

PRODUCT

MPEG-2

LOCATION

KLV embedded metadata

Video, audio TSs

4.3.1.7 KLV Metadata Geolocation Consistency Checks

OVERVIEW

The KLV embedded metadata shown in Table 4-4 identify the geolocation and orientation of both the UAV sensor and the object(s) viewed by the sensor in the image frames. Figures 4-24 and 4-25 present the basic sensor-object geometry from a local viewpoint and a global viewpoint, respectively. The geometric parameters represented in the KLV packet values are interrelated. In some cases, it is possible to compute one KLV packet value from the others, based on geometric relationships. The original and computed values should agree, assuming that all of the KLV values are consistent with each other and comprise a valid description of the actual UAV sensor-object scenario.

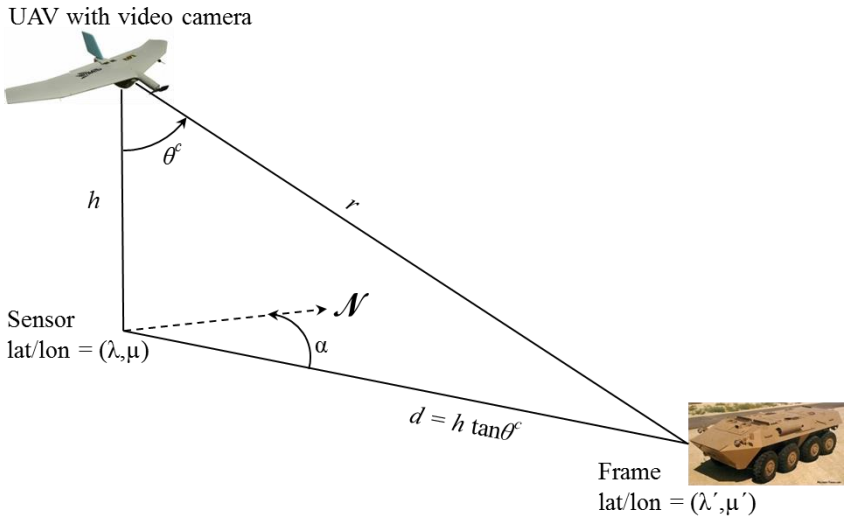


Figure 4-24 UAV Sensor and Frame Geometry for KLV Metadata Checks

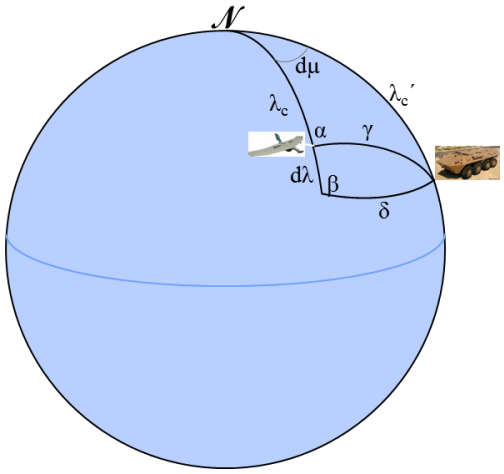


Figure 4-25 Global Geometry of UAV Sensor and Frame Object

However, if the KLV packet values relevant to the sensor-object geometry are overwritten or otherwise altered to convey hidden data, then the geometrical consistency of those values may be destroyed or impaired. This subsection presents two

examples of consistency checks to test for data hiding of this kind. In the first case, a filter should verify the given KLV value for *slant range r* from sensor to object and compare this value against the value computed from the other KLV packet values available. In the second case, a filter should verify the *relative latitude and longitude*. These are defined by the lat/lon pair differences $(\Delta\lambda, \Delta\mu) = (\lambda', \mu') - (\lambda, \mu)$ between the sensor and object positions, given by the respective KLV packet values for lat/lon. In addition, $\Delta\lambda$, $\Delta\mu$ can be recovered using only one latitude and other KLV values from the formula recipe given in Figure 4-26.

$$
\begin{aligned}
\lambda_c &= \tfrac{\pi}{2} - \lambda \\
\lambda'_c &= \tfrac{\pi}{2} - \lambda' \\
\gamma &= d/R_{\oplus} \\
\sin\Delta\mu &= \sin\alpha \sin\gamma / \sin\lambda'_c \\
\cos\delta &= \cos^2\lambda'_c + \sin^2\lambda'_c \cos\Delta\mu \\
\sin\beta &= \sin\alpha \sin\gamma / \sin\delta \\
A &= \sin^2\delta \cos^2\beta - \cos^2\gamma \\
B &= 2\cos\delta \sin\delta \cos\beta \\
C &= \cos^2\delta - \cos^2\gamma \\
At^2 + Bt + C &= 0 \\
t \equiv -\tan\Delta\lambda &= (-B + \mathrm{sign}(AC)\sqrt{B^2 - 4AC})/(2A)
\end{aligned}
$$

$$
\begin{aligned}
\cos\lambda'_c &= \cos\lambda_c \cos\gamma + \sin\lambda_c \sin\gamma \cos\alpha \\
\Delta\lambda &= \lambda_c - \lambda'_c \\
\sin\Delta\mu &= \sin\alpha \sin\gamma / \sin\lambda'_c
\end{aligned}
$$

Figure 4-26 Recipes to Recover Δ lat/lon from Azimuth, Distance Away, and One Latitude: (i) given only the "frame" lat λ´; (ii) given only the "sensor" lat λ

RISKS AND RECOMMENDATIONS

Data Hiding: To prevent data hiding, a filter should screen UAV video/audio streams that cover sensitive events or scenes or mission destinations by analyzing the time and geolocation features present in the associated KLV embedded metadata.

1. **Validate**: Check for slant range consistency. Estimate the slant range from sensor elevation θ and altitude *h:*

$$r = h / \cos \theta^c$$

Check that it is nearly equal to the KLV slant range value.

2. **Validate**: Check for sensor and object latitude/longitude consistency. Compute

$$\Delta \text{ latitude/longitude} = (\Delta\lambda, \Delta\mu) = (\lambda', \mu') - (\lambda, \mu)$$

as a function $f(\lambda', a, d)$ only, using the available KLV values specified in the recipes outlined in Figure 4-24 above. Check that these results are nearly equal to the given KLV lat/lon packet value differences.

3. **Remove**: Remove all KLV metadata packets of the tag types reported and verified to be inconsistent.

PRODUCT

MPEG-2

LOCATION

KLV embedded metadata

4.3.1.8 KLV Metadata Constant Types With Dynamic Values

OVERVIEW

To compress the KLV embedded metadata stream, metadata types that do not require fast updates over time are considered constant, and updated only intermittently to economize on KLV metadata packets. Metadata types that vary rapidly over time and thus require fast updates are considered dynamic. Figure 4-27 shows an example bitstream segment that contains two packet types: (i) complete KLV packets (yellow) containing constant-plus-dynamic tags that need only to be sent once every 10 seconds, and (ii) KLV packets (blue) containing only the dynamic tags (those with values that change rapidly in time), which are sent at maximum rate.

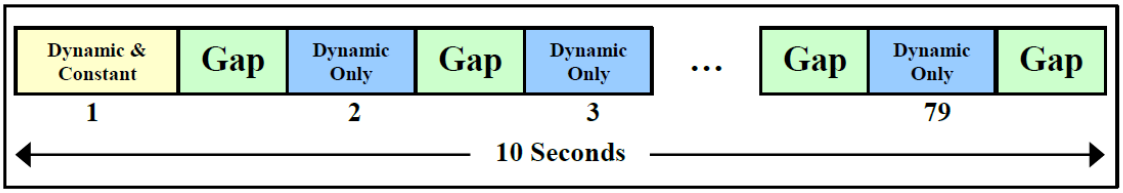


Figure 4-27 Example Bitstream (from [18]) for a 10-second Time Interval, Showing Two Kinds of Embedded KLV Metadata Packets

Table 4-5, taken from [18], recommends KLV minimum metadata set (MMS) tagged types for either slow or fast updates. "Fast" implies the maximum update rate.

Table 4-5 Constant and Dynamic Minimum Metadata Types

Tag #	Tag Name	Max Size (Bytes)	Rec Update Interval
1	Checksum	2	Fast
2	UNIX Time Stamp	8	Fast
3	Mission ID	127	10 s
5	Platform Heading Angle	2	Fast
6 | 90	Platform Pitch Angle (Short) Platform Pitch Angle (Full)	2 4	Fast Fast
7 | 91	Platform Roll Angle (Short) Platform Roll Angle (Full)	2 4	Fast Fast
10	Platform Designation	127	10 s
11	Image Source Sensor	127	10 s
12	Image Coordinate System	127	10 s
13	Sensor Latitude	4	Fast
14	Sensor Longitude	4	Fast

RISKS AND RECOMMENDATIONS

Data Hiding: KLV metadata streams, which transmit their constant tag types only intermittently (i.e., once every 10 seconds) to reduce data-link bandwidth, should be screened for any constant tags that vary over time, especially if they are updated as dynamic tags at the fast rate.

1. **Validate**: Check the embedded KLV metadata stream for any constant tags in Table 4-4 that change in value over time.

2. **Validate**: Check for constant tags that are updated at the same maximum rate as dynamic tags, except when no KLV metadata reduction is needed and every tag (constant or dynamic) is updated at the fast rate. But constants should not normally change.

3. **Validate**: Most constant types in Table 4-4 relate to security classification, and many are large (1024 bytes). Check for inappropriate contents, e.g., absence of expected ASCII text keywords, such as “UNCLASSIFIED” “SECRET,” etc.

4. **Remove**: Remove KLV metadata constant types that change.

5. **External Filtering Required**: Pass KLV metadata text fields to an external filter.

6. **Review**: Review the KLV metadata constant types that change.

PRODUCT

MPEG-2

LOCATION

KLV embedded metadata

5. ACRONYMS

Table 5-1 Acronyms

Acronym	Denotation
2on2	DoD format: MPEG-2 compression on MPEG-2 stream format
AF	Adaptation Field
AoSO	Add-or-subtract-one algorithm
ATM	Asynchronous Transfer Mode
ATSC	Advanced Television Systems Committee, Inc.
AU	Access Unit
AVC	Advanced Video Coding
BER	Basic Encoding Rules
CAT	Conditional Access Table
CEA	Consumer Electronics Association
CEP	Circular Error Probable
CPB	Coded Picture Buffer
DCSQT	Display Control Sequence Table
DCT	Discrete Cosine Transform
DT-CWT	Dual-Tree Complex Wavelet Transform
DVB	Digital Video Broadcasting
EG	Engineering Guidelines
ES	Elementary Stream, a data source, e.g., coded video, audio, or metadata
GOP	Group of Pictures
HEVC	High Efficiency Video Coding
HVS	Human Visual System
IDR	Instantaneous Decoder Refresh
IEC	International Electrotechnical Commission
IPTV	Internet Protocol Television
ISG	Inspection and Sanitization Guidance
ISO	International Standards Organization
ITU-T	International Telecommunications Union – Telecom
JVT	Joint Video Team
KLV	Key-Length-Value format for a metadata elementary stream (ES)
M-IGLS	Multicarrier Iterative Generalized Least Squares

Acronym	Denotation
MISB	Motion Imagery Standard Board (DoD)
MISP	Motion Imagery Standards Profile
MPEG	Moving Picture Experts Group
NAL	Network Abstraction Layer
NATO	National Atlantic Treaty Organization
NIT	Network Information Table
NSG	National System for Geospatial Intelligence
OPCR	Original Program Clock Reference
PAT	Program Association Table
PCR	Program Clock Reference
PES	Packetized Elementary Stream, divided into 188-byte MPEG packets (TS)
PID	Packet Identifier
PMT	Program Map Table
PS	Program Stream, the multiplexed PES's data sources in storage format
PSI	Program Specific Information
PTS	Presentation Time Stamp
QP	Quantization Parameter
RLE	Run Length Encoding
RP	Recommended Practices
SCR	System Clock Reference
SEI	Supplement Enhancement Information
SIFT	Scale Invariant Feature Transform
SMPTE	Society of Motion Picture and Television Engineers
SPTS	Single Program Transport Stream, a complete video/audio TS package
SPU	Sub-Picture Unit
STANAG	Standardization Agreement
TLV	Tag-Length Valve
TRM	Technical Reference Material
TS	Transport Stream, the multiplexed PES's in communication format
UAS	Unmanned Aircraft System
UAV	Unmanned Aerial Vehicle
VCL	Video Coding Layer

6. REFERENCED DOCUMENTS

1. *Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s* (ISO/IEC 11172), 1993.
2. *Generic coding of moving pictures and associated audio information* (ISO/IEC 13818-1,2 4th ed.), 2013.
3. ITU-T Rec. H.264 (04/2013) Advanced Video Coding for Generic Audiovisual Services.
4. *Multimedia content description interface* (ISO/IEC 15938), 2002.
5. *Multimedia framework (MPEG-21)* (ISO/IEC 21000), 2001.
6. *H.262: Information technology - Generic coding of moving pictures and associated audio information: Video,* International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) February 2000.
7. *ATSC Digital Television Standard: Part 4 – MPEG-2 Video System Characteristics,* Document A/53 Part 4:2009, 7 August 2009.
8. *Digital Television (DTV) Closed Captioning,* Doc. CEA-708-D, Consumer Electronics Association, Arlington, VA, August 2008.
9. High-Efficiency Video Coding (HEVC) homepage at:

   https://hevc.hhi.fraunhofer.de/
10. G.J. Sullivan; J.-R. Ohm; W.-J. Han; T. Wiegand, "*Overview of the High Efficiency Video Coding (HEVC) Standard*", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22 , No. 12, December 2012 , pp. 1649-1668.
11. T. Wiegand, G. J. Sullivan, G. Bjøntegaard and A. Luthra, "*Overview of the H.264/AVC Video Coding Standard,*" IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 7, July 2003, pp. 560-576.
12. Iain E. G. Richardson, *The H.264 advanced video compression standard,* Hoboken, NJ : Wiley, 2010.
13. Joint Video Team (JVT), *Joint Model (JM 18.6).* Available online at:

    http://iphome.hhi.de/suehring/tml/
14. `x264`, available online at:

    http://www.videolan.org/developers/x264.html
15. *MISB ST 0807.14 - KLV Metadata Dictionary*, Oct 2014
16. *MISB ST 0607.2 - Metadata Registry and Processes*, Feb 2014
17. *MISB ST 0902.4 - Motion Imagery Sensor Minimum Metadata Set*, Oct 2014

18. *MISB ST 0601.8 - UAS Datalink Local Metadata Set*, Oct 2014

19. *MISP-2015.1 – Motion Imagery Standards Profile, Oct 2014*

20. *MISB ST 1402 - MPEG-2 Transport of Compressed Motion Imagery and Metadata*, Feb 2014. Online reference available at http://www.gwg.nga.mil/misb/docs/standards/ST1402.pdf

21. S. Kapotas and A. Skodras, "*A New Data Hiding Scheme for Scene Change Detection in H.264 Encoded Video Sequences*," IEEE International Conference on Multimedia and Expo, 2008.

22. H. Zhu, R. Wang, D. Zu, X. Zhou, "*Information Hiding Algorithm for H.264 Based on the Prediction Difference in Intra_4x4*," 3rd International Conference on Image and Signal Processing (CISP), 2010.

23. MPEG Headers Quick Reference. Online reference available at http://dvd.sourceforge.net/dvdinfo/mpeghdrs.html

24. M. Asikuzzaman, M.J. Alam, A.J. Lambert, M.R. Pickering, "*Imperceptible and Robust Blind Video Watermarking using Chrominance Embedding: A Set of Approaches in the DT CWT Domain,*" IEEE Trans. Information Forensics and Security, vol.9 no.9,. Sept. 2014.

25. A. Costanzo, I. Amerini, R. Caldelli, and M. Barni, *"Forensic Analysis of SIFT Keypoint Removal and Injection,"* IEEE Trans. Information Forensics and Security, vol.9 no.9, Sept. 2014.

26. G. Chierchia, G. Poggi, C. Sansone, L. Verdoliva, *"A Bayesian-MRF Approach for PRNU-Based Image Forgery Detection,"* IEEE Trans. Information Forensics and Security, vol.9 no.4, April 2014.

27. D. Xu, R. Wang, and Y.Q. Shi, "*Data Hiding in Encrypted H.264/AVC Video Streams by Codeword Substitution,"* IEEE Trans. Information Forensics and Security, vol.9 no.4, April 2014.

28. W. Wang, X, Jiang, S. Wang, T. Sun, *"Estimation of the Primary Quantization Parameter in MPEG Videos,"* Visual Communications and Image Processing (VCIP), 2013, DOI: 10.1109/VCIP.2013.6706413

29. Y. Chen and C. Hsu, "*Detecting Recompression of JPEG Images via Periodicity Analysis of Compression Artifacts for Tampering Detection*," IEEE Trans. Information Forensics and Security, vol.6 no.2, June 2011.

30. T. Bianchi, A. Piva, "*Image Forgery Localization via Block-Grained Analysis of JPEG Artifacts*," IEEE Trans. Information Forensics and Security, vol.7 no.3, April 2012.

31. G. Cao, Y. Zhao, R. Ni, X. Li, "*Contrast Enhancement-Based Forensics in Digital Images,"* IEEE Trans. Information Forensics and Security, vol.9 no.3, March 2014.

32. M. Li, M. K. Kulhandjian, D.A. Pados, S.N. Batalama, M.J. Medley, "*Extracting Spread-Spectrum Hidden Data From Digital Media,*" IEEE Trans. Information Forensics and Security, vol.8 no.7, July 2013.

33. K. Wang, H. Zhao, H. Wang, "*Video Steganalysis Against Motion Vector-Based Steganography by Adding or Subtracting One Motion Vector Value,*" IEEE Trans. Information Forensics and Security, vol.9 no.5, May 2014.

34. Motion Imagery Standards Profile (MISP 2015.1), Oct 2014

35. *MISB ST 0604.3 – Time Stamping Compressed Motion Imagery*, Feb 2014

36. Kim, K. MPEG-2 ES/PES/TS/PSI. Online reference available at: *http://cmm.khu.ac.kr/korean/files/02.mpeg2ts1_es_pes_ps_ts_psi.pdf*

37. MPEG Headers Quick Reference: Online Reference available at: *http://dvd.sourceforge.net/dvdinfo/mpeghdrs.html*

38. Tektronix®[5]. What is an MPEG-2 Access Unit (AU)? Online Reference available at: http://www.tek.com/support/faqs/what-mpeg-2-access-unit-au?ct=FAQ&cs=faq&ci=6366&lc

39. Phor, V. "Video decoding: SDI interface implementation &H.264/AVC bitstreamdecoder hardware architecture design and implementation," online reference available at: http://www.slideshare.net/phorvicheka/kkschoolslidesphorm2camsi20132014

40. *MISB ST 0102.11 – Security Metadata Universal and Local Sets for Digital Motion Imagery*, October 2014.

41. *Sub-Pictures.* Online reference available at http://www.mpucoder.com/DVD/spu.html

5 Tektronix® is a trademark of Tektronix Inc

7. SUMMARY OF RISKS

Table 7-1 summarizes the risks of each construct as well as the combined number of risks for the entire document.

Table 7-1. Summary of Risks for MPEG-2 TS and KLV Metadata

Construct	Data Attack	Data Hiding	Data Disclosure
4.1.1.1. Program Association Table		X	
4.1.1.2. Program Map Table		X	
4.1.1.3. Multiple Audio Streams		X	
4.1.1.4. Closed Captioning		X	
4.1.2.1. Synchronization Byte	X	X	
4.1.2.2. Null Packets		X	
4.1.2.3. Video Compression Format		X	
4.1.2.4. Continuity Counter	X		
4.1.2.5. Optional AF		X	
4.1.3.1. Packetized Elementary Stream (PES) Header Start Code		X	
4.1.3.2. PES Picture Header		X	
4.1.3.3. PES Sequence Header		X	
4.1.3.4. PES Extension Headers		X	
4.1.3.5. Group of Pictures (GOP) Header	X	X	
4.1.4.1. User Data Header		X	
4.1.4.2. Supplemental Enhancement Information		X	
4.1.5.1 Network Abstraction Layer		X	
4.1.5.2. H.264 Interframe Data Hiding		X	
4.1.5.3. H.264 Intraframe Data Hiding		X	
4.1.5.4. Double MPEG Encoding as Evidence of Data Tampering		X	

4.1.5.5. MPEG Watermarking for Data Hiding		X	
4.1.5.6. MPEG Motion Vector Tampering for Data Hiding		X	
4.2.1.1. KLV Metadata Key Present in MISB Standard 0807 Dictionary		X	X
4.2.1.2. Metadata Checksum	X	X	
4.2.1.3. Metadata Value		X	X
4.2.1.4. Security Markings		X	X
4.2.1.5. All MISB Standard 0902 Minimum Metadata Items Present	X	X	
4.2.1.6. KLV Metadata Geolocation ID and Video Imagery			X
4.2.1.7. KLV Metadata Geolocation Consistency		X	
4.2.1.8. KLV Metadata Constant Types with Dynamic Values		X	
Total:	**5**	**28**	**4**
Combined Total:			**37**