Received: 2 May 2023 | Revised: 18 June 2023 | Accepted: 3 July 2023

DOI: 10.1111/1556-4029.15335

JOURNAL OF FORENSIC SCIENCES AAFS

ORIGINAL PAPER

Digital & Multimedia Sciences

Forensic integrity verification of video recordings based on MTS files

Kyu-Sun Shim MSc[1,2] | Nam In Park PhD[2] | Seong Ho Lim MSc[2] | Jun Seok Byun MSc[2] | Heejo Lee PhD[1]

[1]Department of Computer Science and Engineering, Korea University, Seoul, Korea

[2]Digital Analysis Division, National Forensic Service, Wonju-si, Korea

Correspondence
Heejo Lee, Department of Computer Science and Engineering, Korea University, Seoul, 02841, Korea.
Email: heejo@korea.ac.kr

Funding information
National Forensic Service, Ministry of the Interior and Safety Republic of Korea, Grant/Award Number: NFS2023DTB04

Abstract

Digital video is used in criminal trials as evidence with legal responsibility because video content vividly depicts events occurring at a crime scene. However, using sophisticated video editing software, assailants can easily manipulate visible clues for their own benefit. Therefore, the integrity of digital video files acquired or submitted as evidence must be ensured. Forensic analysis of digital video is key to ensuring the integrity of links with individual cameras. In this study, we analyzed whether it is possible to ensure the integrity of MTS video files. Herein, we propose a method to verify the integrity of MTS files encoded by advanced video coding high definition (AVCHD), which is frequently used for video recording. To verify MTS file integrity, we propose five features. Codec information, picture timing, and camera manufacture/model are modified AVI and MP4-like format video verification features. *Group of pictures* and *Universally Unique Identifier* patterns were specifically developed for MTS streams. We analyzed the features of 44 standard files recorded using all recording options of seven cameras. We checked whether integrity can be validated on *unmanipulated* videos recorded in various environments. In addition, we considered whether *manipulated* MTS files edited in video editing software could be validated. Experimental results show that all *unmanipulated* and *manipulated* MTS files with known recording devices were discriminated only when all five features were checked. These results show that the proposed method verifies the integrity of MTS files, strengthening the validity of MTS file-based evidence in trials.

KEYWORDS
camera characteristics, digital forensics, forensic video analysis, MTS video, video forgery detection, video integrity verification

Highlights

- The proposed method provides forensic analysis to verify the integrity of MTS video files.
- MTS files are analyzed based on features reflecting the characteristics of the recording camera.
- *Unmanipulated* and *manipulated* MTS files can be distinguished using the proposed method.

1 | INTRODUCTION

Digital video is considered an increasingly important medium for exchanging information between users. According to Cisco, the extent of video interaction over the internet is expected to reach 82% of the total internet traffic by 2022 [1]. As the amount of video distribution increases, an environment is being established that enables individuals to easily produce videos. When an incident occurs, the details, proceedings, and actions of the event are often recorded on digital devices, such as a smartphone, camcorder, or surveillance camera, which can be used as a critical evidence for the investigation and forensic analysis required to solve the case.

An integrity guarantee is essential when using digital video footage as critical evidence because as technology improves, such footage is increasingly susceptible to tampering. Thus, the field of video forensics has developed techniques to examine digital records based on the withstanding characteristic records according to the life cycle of the video [2]. As such, video forensics can be broadly classified into two major categories: authenticity and integrity verification [3]. Until recently, integrity and authentication have been used as synonyms in the field of video forensic research [4]. However, these two terms have completely distinct meanings. According to SWGDE Best Practices for Image Content Authentication, "content authentication is used to determine whether the visual content depicted in imagery is a true and accurate representation of subjects and events," whereas "integrity ensures that the information presented is complete and unaltered from the time of acquisition until its final disposition" [5].

Based on several considerations, numerous prior studies have examined methods for manipulation detection to verify the authenticity and integrity of video files. In [6], techniques for detecting video manipulation and forgery were classified into five major categories according to the features and method types, as depicted in Figure 1. In particular, camera/sensor artifacts, coding artifacts, motion features, and object features are more focused on verifying authenticity in terms of video content. In contrast, the components marked with red boxes in Figure 1 constitute multimedia container features that are essential for establishing the integrity of the video file. Thus, even if the video content is not changed or altered, the information recorded inside the file can be modified and evidence can be manipulated by changing the recording device, the point time when the video was recorded, or other features. Therefore, to use a file as evidence, ensuring the integrity of the file itself is necessary. Research pertaining to the integrity of video files is actively underway.

In [3, 6–8], integrity verification was proposed using container structure and tag values, which are container features. In particular, [7] analyzed the video types of AVI, MOV, MP4, and 3GP using 19 digital camera models, 14 mobile phones, and 6 video editing tools. Moreover, [3, 6] proposed an integrity verification method using the container features for the video files with MP4-like formats, and [8] examined the AVI file format using container features. However, files with MTS extensions constitute data in stream-based format, and existing techniques do not employ methods suitable for the integrity verification video files with MTS extension, which is a commonly used video format.

An MTS extension file is a video clip saved in advanced video coding high definition (AVCHD) format, which is a high definition recording format using MPEG-4 AVC (H.264) video codec. Generally, camcorders and digital cameras store videos on flash memory cards, hard disks, and DVDs in this format [9]. Overall, the AVCHD format is supported in the products of manufacturers such as Canon, Sony, and Panasonic. According to the 2021 digital camera market share report, these three camera manufacturers constitute 75.6% of the global market distribution [10].

Thus, integrity verification for video files in AVCHD format is required, and such video can be verified by analyzing the MTS file because the video can be checked only using the MTS file, even if no other components of the AVCHD format are available.

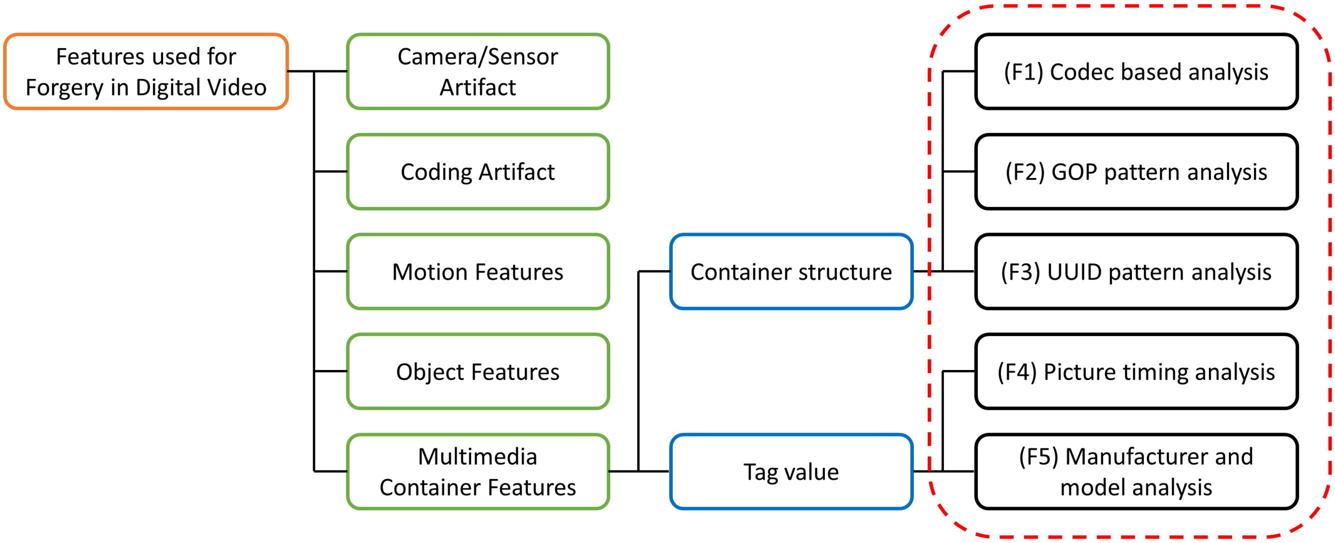

Features used for Forgery in Digital Video
Camera/Sensor Artifact
Coding Artifact
Motion Features
Object Features
Multimedia Container Features
Container structure
Tag value
(F1) Codec based analysis
(F2) GOP pattern analysis
(F3) UUID pattern analysis
(F4) Picture timing analysis
(F5) Manufacturer and model analysis

FIGURE 1 Features used to detect manipulation in digital video [6].

This study proposes a mechanism to verify the integrity of MTS files using the components of the file container structure and tag values, marked in red box in Figure 1. MTS files store video data in the form of a container based on a communication protocol for audio, video, and data transmission called a MPEG transport stream. Unlike AVI and MP4-like video files, the container structure format is composed of a media stream format. In addition, when an MTS file is edited with features organized in the form of streams, the features that make up the file may change, which can be useful for analysis. Accordingly, we verified the integrity of the MTS file by comparing the order and homogeneity of the media stream based on the consistency of the metadata.

The following analyses are proposed to verify the integrity of MTS video files in this study: We first propose container structure-based analysis on MTS media streams using *codec-based analysis*, *group of pictures (GOP) pattern analysis*, and *Universally Unique Identifier (UUID) pattern analysis*, which are inspired by existing AVI and MP4-like format file analysis techniques.

- In feature 1 (F1), codec-based analysis, we analyze the equivalence of the *sequence parameter set (SPS)* and *picture parameter set (PPS)*, which are in the *network abstraction layer (NAL)* units of H.264, to determine if they match information provided by the recording device.
- In feature 2 (F2), GOP pattern analysis, integrity is verified by analyzing the composition of the encoding of the *group of pictures (GOP)* in the MTS file and comparing whether it is the same as the original provided by the recording devices.
- Feature 3 (F3), UUID pattern analysis, verifies integrity by considering that an MTS file based on a stream structure has *supplemental enhancement information (SEI)* with a *Universally Unique Identifier (UUID)* that is not present in AVI and MP4-like video files. Moreover, the UUID should be written repeatedly in a certain sequence. This pattern should change if the file is manipulated, and the integrity can be verified by considering these factors.

We also use tag value analysis, which is the extraction and comparative analysis of metadata such as *time information* and *camera manufacturer/model*, which are included in the user data along with the UUID.

- Feature 4 (F4), picture timing analysis, is written along with the video frame at regular intervals so that the time of recording can be checked. Moreover, because they are written continuously, the integrity can be verified according to whether the *continuity of the picture* timing is maintained.
- Feature 5 (F5), manufacturer and model analysis, can verify integrity by comparing the manufacturer and model written in the user data of the UUID with those of the actual recording camera.

The present study was conducted to determine whether the integrity of an MTS video file can be guaranteed using the five multimedia container features indicated in Figure 1. Note that features 1 and 2, which relate to the SPS/PPS and GOP, vary according to the hardware encoders provided by the camera, whereas the SEI containing features 3, 4, and 5 is distinguished from the metadata depending on the recording setting. Thus, the three container structure factors and two tag values are valuable features for establishing video integrity.

We conducted experiments with Canon, Sony, and Panasonic cameras that can store video in MTS format. Throughout these experiments, we analyzed the aforementioned features to verify file integrity and compared original and *manipulated* MTS files to validate these features. First, we compiled 44 original MTS files by recording video for all options available on four Sony, one Canon, and two Panasonic models.

In addition, to check the accuracy of the *unmanipulated* MTS verification, we recorded various scenes using the 44 options for the above models five times. To check whether the *manipulated* MTS can be detected, we independently inserted, trimmed, cut, and edited original MTS files using editing software to create a total of 220 test videos. In the experiments, we were able to verify the integrity of the videos in all cases with the method proposed in this study when all five proposed features were checked. By contrast, when we tested only some features, we could not verify all videos. We also suggest an integrity verification procedure when MTS files are submitted as evidence. Therefore, we demonstrated that the integrity of the MTS files can be verified by the proposed model. However, the proposed method only allows comprehensive feature verification given a known recording device or large database of standard MTS files. Accordingly, these measures can improve the effectiveness of the proposed method.

The major contributions of this study can be summarized as follows.

- We recognize features for integrity verification of an MTS file: we suggested and validated five features that can verify the integrity of an MTS file, and we determined whether the MTS file was manipulated based on our analysis.
- We verify and distinguish whether an MTS file has been manipulated: Experimental results verified the accuracy of the proposed strategy to distinguish between *unmanipulated* and *manipulated* MTS files, and an examination procedure was proposed by synthesizing all the results.

The remainder of this study is organized as follows. Section 2 presents the relevant literature addressed in this study. Section 3 describes how the features proposed in this study that characterize MTS files were analyzed. Section 4 validates the features based on experiments, and the corresponding results are discussed. Finally, Section 5 summarizes the conclusions of this study.

2 | RELATED WORK

2.1 | Video forensics techniques

Several previous studies have been conducted to detect tampering, source identification, integrity, and authentication in photos, videos, and audio [3, 6–8, 11].

In particular, numerous methods have been proposed based on the analysis of acquired images, and the same methods have been applied to the frames extracted from videos. Nonetheless, a specialized forensic analysis method for video evidence is required because of the complexities of video data. Earlier, video forensic analysis was performed by analyzing the inconsistencies five of content based on the features of the existing audio–video signal. In [12], the authors detected double encoding or manipulation based on the prediction residual, and in [13, 14], macroblock-type analysis was used.

Ding traced the video frame rate up-conversion to analyze whether video evidence was deliberately processed [15]. Furthermore, [16] proposed a more efficient detection scheme by determining region duplication in a video using an effective algorithm developed on exponential-Fourier moments. The methods proposed in prior research mainly focused on authenticity verification based on signals, which differs from the integrity verification proposed in this paper.

In [6], various types and features applied to video manipulation methods in prior studies were classified into five categories: sensor artifacts, coding artifacts, motion features, object features, and other components. As depicted in Figure 1, a feature that determines integrity can be classified into one of six features, including multimedia container features.

Previous studies also focused on integrity verification by specifying mobile devices [17]. However, the limitation of these studies is that they only considered images and videos generated using components from certain manufacturers, and did not consider similar models from different brands.

2.2 | Video forensics techniques

Source identification is based on metadata, image features, the matrix defects of conformity factor analysis, color interpolation, sensor imperfections, and wavelet transforms. The authors of [18] proposed a method to elaborately compare the results of source identification by segmenting these five features into certain groups.

A technique was proposed to verify the integrity of AVI format video generated by video event data recorders, after which a structural analysis was performed on 296 original videos, and the videos were edited using five types of video editing software in [8]. Consequently, the edited videos exhibited significant variations in structure and metadata values compared to the original. Each editing program contains a specific structure to detect manipulation in the video.

A previous study analyzed the structure of AVI and MP4 format videos recorded using 19 digital camera models, 14 mobile phone models, and six editing programs in [7]. Analyzing the original videos revealed that the structure of each container type was not strictly defined according to the standard specifications. A significant difference was observed in the structure of the videos generated on each device. In addition, after the AVI video was edited using an editing program, the internal information, including the metadata values (which is essential for determining the original video), was deleted or modified.

The authors of [3] implemented an unsupervised method to verify the video integrity based on the variations between the original and edited videos. They further developed a technique to analyze the containers and identify the video acquisition sources. To this end, they used the MP4Parser library to obtain an extensible markup language (XML) file for subsequent analysis. The experimental results were highly significant, demonstrating that the solution used fewer computational resources than alternative solutions.

Conventional methods exist that analyze the internal structure of a multimedia container, but they focus on MP4-like file formats, such as MP4, MOV and 3GP, and AVI formats. To the best of our knowledge, no prior study has focused on verifying the integrity of MTS files, despite the fact that 75.8% of products in the global camera market support MTS files saved in AVCHD format. In this study, we developed a suitable integrity verification method based on the media structures of MTS files.

3 | PROPOSED METHOD

3.1 | Overview

In this study, we propose a method for verifying the integrity of MTS files stored in AVCHD format, as illustrated in Figure 2. Initially, the proposed method analyzes three aspects of the MTS file: *codec-based analysis*, *media stream analysis*, and *GOP analysis*. We then compare the analytical results with the features of a standard unmodified file, validate data consistency, and finally verify file integrity.

First, in the codec-based analysis, the SPS and PPS of H.264 video codec are extracted and compared with those of a standard file that is ensured to be an original from the device. The second step involves extracting and analyzing the UUID pattern with the time information, and the camera manufacturer/model information was caused by the MTS characteristics in the SEI information extracted from the media stream. Among them, the UUID pattern and time information validate the data consistency. In addition, the manufacturer/model name recorded in the SEI can be extracted and compared with the standard file provided by the camera device. Ultimately, we extracted the GOP to verify a consistent pattern in comparison to the GOP in the standard file.

3.2 | Codec based analysis

MTS video files in AVCHD format are compressed and recorded in the H.264 (MPEG-4 AVC/H.264) codec developed by ITU-T and ISO/IEC, and standardized as ISO/IEC 14496–10 [19]. If the video is

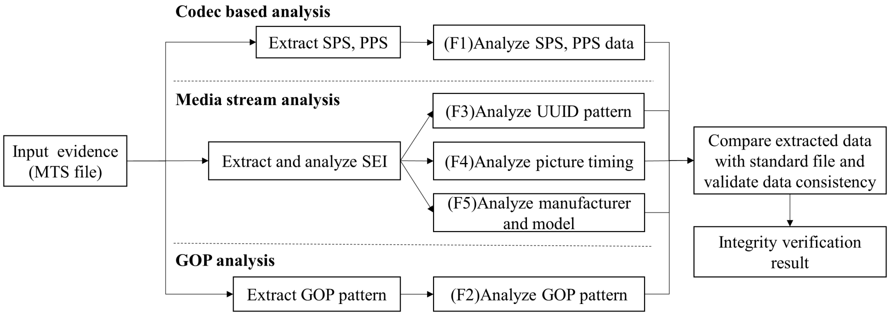
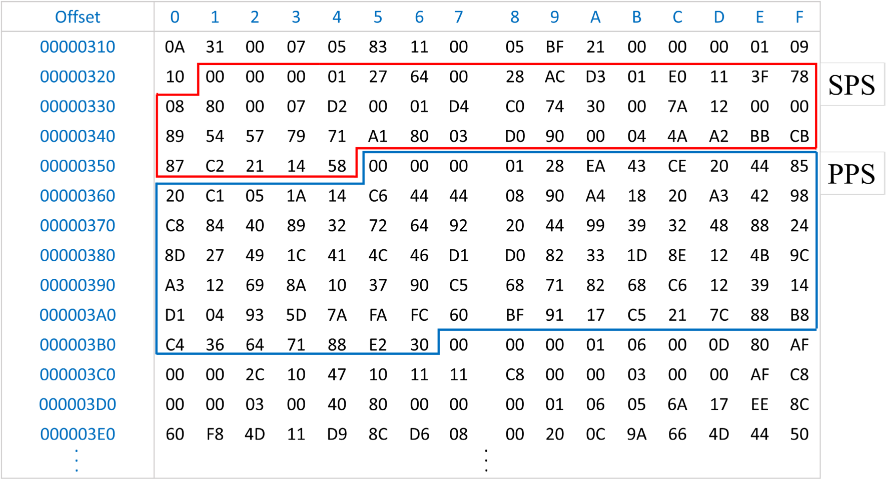

Codec based analysis
Extract SPS, PPS
(F1)Analyze SPS, PPS data
Media stream analysis
(F3)Analyze UUID pattern
Input evidence (MTS file)
Extract and analyze SEI
(F4)Analyze picture timing
(F5)Analyze manufacturer and model
Compare extracted data with standard file and validate data consistency
Integrity verification result
GOP analysis
Extract GOP pattern
(F2)Analyze GOP pattern

FIGURE 2 Overview of proposed MTS integrity verification method.



```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000310   0A 31 00 07 05 83 11 00  05 BF 21 00 00 00 01 09
00000320   10 00 00 00 01 27 64 00  28 AC D3 01 E0 11 3F 78   SPS
00000330   08 80 00 07 D2 00 01 D4  C0 74 30 00 7A 12 00 00
00000340   89 54 57 79 71 A1 80 03  D0 90 00 04 4A A2 BB CB
00000350   87 C2 21 14 58 00 00 00  01 28 EA 43 CE 20 44 85   PPS
00000360   20 C1 05 1A 14 C6 44 44  08 90 A4 18 20 A3 42 98
00000370   C8 84 40 89 32 72 64 92  20 44 99 39 32 48 88 24
00000380   8D 27 49 1C 41 4C 46 D1  D0 82 33 1D 8E 12 4B 9C
00000390   A3 12 69 8A 10 37 90 C5  68 71 82 68 C6 12 39 14
000003A0   D1 04 93 5D 7A FA FC 60  BF 91 17 C5 21 7C 88 B8
000003B0   C4 36 64 71 88 E2 30 00  00 00 01 06 00 0D 80 AF
000003C0   00 00 2C 10 47 10 11 11  C8 00 00 03 00 00 AF C8
000003D0   00 00 03 00 40 80 00 00  00 01 06 05 6A 17 EE 8C
000003E0   60 F8 4D 11 D9 8C D6 08  00 20 0C 9A 66 4D 44 50
   ⋮                            ⋮
```

FIGURE 3 Extraction example of codec information (sequence parameter set and picture parameter set) in an MTS file.

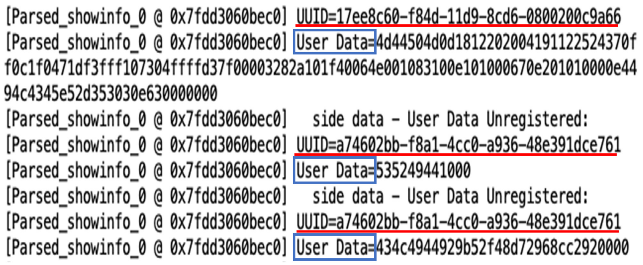
encoded in H.264, this information is stored in the NAL unit, which includes the SPS and PPS recording information required for decoding, such as the video profile/level, resolution, bit depth, and entropy coding mode, whereas the slices record the compressed frame data.

As depicted in Figure 3, the SPS and PPS stored in the MTS file can be reviewed. The red and blue boxes represent decoding header information, which consists of SPS and PPS. The single-byte code following the start code (0x00000001) is 0x27 for SPS and 0x28 for PPS. In the H.264 standard, the last five bits in the first byte of a NAL unit following the start code may be seven to denote SPS, or eight to denote PPS. SPS and PPS constitute information used to decode the video. When the video is manipulated, it is subject to a re-encoding process wherein the SPS and PPS are susceptible to changes caused by editing software, making them necessary elements in verifying video integrity.

After extracting the SPS and PPS, which can be regarded as a component of the video characteristics from the MTS file, they were compared with the characteristics of the standard video file directly obtained from the source camera. Thereafter, the encoding of a unique SPS and PPS pair can be determined according to the recording option provided by the source camera. Although it is not a unique feature, it is a meaningful feature that can reduce the number of cameras that must be sourced.

3.3 | Media stream analysis

The MTS file contains the SEI in the media stream, and SEI messages can contain various types of data indicating the timing of the video pictures or describing the various properties of the coded video or approaches that can be employed for utilization or enhancement. In addition, SEI messages can contain arbitrary user-defined data. The SEI of the MTS file contains UUID and user data, which is meta-information [20]. UUID is a standard protocol for generating unique identities on a network. These identities are regularly stored in stream-based MTS files. In this study, the UUID and user data were extracted using ffmpeg command ("ffmpeg -i filename -vf showinfo -f null – 2>&1"), as depicted in Figure 4.



```
[Parsed_showinfo_0 @ 0x7fdd3060bec0] UUID=17ee8c60-f84d-11d9-8cd6-0800200c9a66
[Parsed_showinfo_0 @ 0x7fdd3060bec0] User Data=4d44504d0d1812202004191122524370f
f0c1f0471df3fff107304ffffd37f00003282a101f40064e001083100e101000670e201010000e44
94c4345e52d353030e630000000
[Parsed_showinfo_0 @ 0x7fdd3060bec0]   side data - User Data Unregistered:
[Parsed_showinfo_0 @ 0x7fdd3060bec0] UUID=a74602bb-f8a1-4cc0-a936-48e391dce761
[Parsed_showinfo_0 @ 0x7fdd3060bec0] User Data=535249441000
[Parsed_showinfo_0 @ 0x7fdd3060bec0]   side data - User Data Unregistered:
[Parsed_showinfo_0 @ 0x7fdd3060bec0] UUID=a74602bb-f8a1-4cc0-a936-48e391dce761
[Parsed_showinfo_0 @ 0x7fdd3060bec0] User Data=434c4944929b52f48d72968cc2920000
```

FIGURE 4 Example of Universally Unique Identifier patterns and recorded user data (meta-information) in an supplemental enhancement information.

For example, in Canon XA20, UUIDs *17ee8c60-f84d-11d9-8cd6-0800200c9a66* and *c0000000-a746-02bb-f8a1-4cc0a93648e3* are regularly recorded. All Sony cameras contain UUIDs *17ee8c60-f84d-11d9-8cd6-0800200c9a66* and *a74602bb-f8a1-4cc0-a93648e391dce761*. Moreover, we determined the presence of another UUID, *dba1adef-b20c-40b4-8c85-8c0b46d5241e*, for all recording options of the FDR-AX700 and certain options of the ILCE-7M2 and HDR-PJ760. The Panasonic models contained only UUID *17ee8c60-f84d-11d9-8cd6-0800200c9a66*.

Thus, the UUIDs were regularly recorded in the MTS file, and the user data were stored together in the UUIDs. After extracting the UUIDs, we analyzed the consistency of the UUID pattern, the continuity of the picture timing (i.e., the timestamp of the frame), and the correspondence between the manufacturer/model included in the user data, which were used as features to verify the integrity of the video file.

3.3.1 | UUID pattern analysis

We found that the UUIDs followed certain rules to record the file, depending on the camera manufacturer/model and recording options. For instance, in the case of the Sony FDR-AX700, a total of 29 UUIDs were recorded in a single cycle, *17ee8c60f84d-11d9-8cd6-0800200c9a66* occurred 26 consecutive times, and subsequently, *a74602bb-f8a1-4cc0-a936-48e391dce761* was repeated twice, and *dba1adef-b20c-40b4-8c85-8c0b46d5241e* was sequentially repeated only once. As another example, in the Sony HDR-PJ760, a total of two UUIDs occurred in single cycle, and *17ee8c60-f84d-11d9-8cd6-0800200c9a66* and *a74602bb-f8a1-4cc0-a936-48e391dce761* appeared repeatedly.

Thus, the UUID pattern in a single MTS file bears the characteristic of consistency. In the case of any inconsistency, an integrity issue is highly possible with the video file, thereby requiring further review.

Algorithm 1 describes the method that analyses the UUID pattern in the evidence MTS file, where V verifies whether consistency has been maintained. After extracting the UUID from the MTS file, we created a set of tuples <u_i,c_i>, which is a pair consisting of the consecutive UUID and its count. If the UUID is the same as that in the created pair set, <u_i,c_i> should remain the same. However, if it is not the same, the consistency of the UUID pattern is not validated. If it is the same, the UUID pattern set S({<u_i,c_i>}) is the result of excluding duplicates from the pairs consisting of the UUID and the number of consecutive UUID occurrences analyzed in the MTS file.

For example, in case of Sony FDR-AX700 as illustrated above, the UUID tuples would be *<17ee8c60-f84d-11d9-8cd60800200c9a66,26>, <a74602bb-f8a1-4cc0-a936-48e391dce761,2>, <dba1adef-b20c-40b4-8c85-8c0b46d5241e,1>, <17ee8c60f84d-11d9-8cd6-0800200c9a66,26>, <a74602bb-f8a1-4cc0-a936-48e391dce761,2>, <dba1adef-b20c-40b4-8c85-8c0b46d5241e,1>*. After removing the duplicates, the final UUID pattern set is *<17ee8c60-f84d-11d9-8cd6-0800200c9a66,26>, <a74602bb-f8a14cc0-a936-48e391dce761,2>, <dba1adef-b20c-40b4-8c85-8c0b46d5241e,1 >*.

If the MTS file has not been edited, the final UUID pattern set can validate the file's integrity, as there is only one pair for a particular UUID. If the file has been edited, there may be multiple pairs for the same UUID, or UUIDs that were not present in the original file, depending on the change in UUID.

Algorithm 1 Analysis of UUID pattern

```
Input: $V \leftarrow \mathit{MTS\ video\ file}$
Output: Unique UUID Pattern Set $S = \{<u_i,c_i>\}$, where $u_i$ is the UUID value found in the video file and $c_i$ is the number of times that $u_i$ is repeated consecutively
$U \leftarrow \mathit{ExtractUUID}(V)$ // Extract the UUID from the $V$, storing the result in the set $U$
$\{<u_i,c_i>\} \leftarrow \mathit{countConsecutiveUUID}(U)$ // Count consecutive UUID occurrences in $U$, store as tuple $<u_i,c_i>$
for all $u_i \leftarrow 1$ to $i$ in $\{<u_i,c_i>\}$ do
    for all $u_j \leftarrow 1$ to $j$ in $\{<u_i,c_i>\}$ do
        if $u_i$ equals $u_j$ then
            if $c_i$ not equals $c_j$ then
                return $\mathit{not\ valid}$
            end if
        end if
    end for
end for
$S \leftarrow \mathit{RemoveDuplicatedTuples}(\{<u_i,c_i>\})$
```

3.3.2 | Picture timing analysis

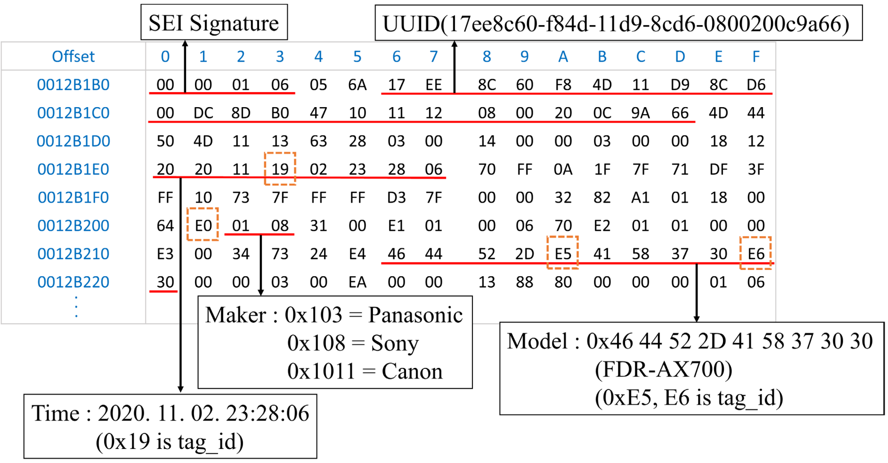
In the MTS file, if the UUID is 17ee8c60-f84d-11d9-8cd6-0800200c9a66, the meta-information-modified digital video pack metadata (MDPM) is recorded as an unregistered user data. As depicted in Figure 5, the picture timing information is displayed as the hexadecimal number 0x2020111902232806 based on tag_id values 0x18 and 0x19, which is expressed as 2020.11.02. 11 23:28:06. The picture timing is recorded at least once per second.

As the frames in the video are continuously stored, the picture time stored in a single MTS file must be continuously recorded. Notably, the integrity of the MTS file cannot be ensured when recording is interrupted. Algorithm 2 describes a procedure to validate the continuity of the picture time recorded in the MTS file. First, the picture time is extracted from an evidence-worthy MTS file. If the difference between picture time and the preceding one is greater than 1 s, the integrity cannot be guaranteed because the continuity of time has not been maintained.

3.3.3 | Manufacturer and model analysis

In the MTS file, the camera manufacturer and model are recorded along with the time information, starting with MDPM (*0x4D44504D*) in the UUID (*0x17ee8c60-f84d-11d9-8cd6-0800200c9a66*), as indicated in Figure 5. For instance, if the camera manufacturer is Sony and the model is FDR-AX700, it is written in hexadecimal as *0x108*

FIGURE 5 Example of the meta-information analysis of the Sony FDR-AX700.



SEI Signature

UUID(17ee8c60-f84d-11d9-8cd6-0800200c9a66)

```
Offset    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
0012B1B0  00 00 01 06 05 6A 17 EE 8C 60 F8 4D 11 D9 8C D6
0012B1C0  00 DC 8D B0 47 10 11 12 08 00 20 0C 9A 66 4D 44
0012B1D0  50 4D 11 13 63 28 03 00 14 00 00 03 00 00 18 12
0012B1E0  20 20 11 19 02 23 28 06 70 FF 0A 1F 7F 71 DF 3F
0012B1F0  FF 10 73 7F FF FF D3 7F 00 00 32 82 A1 01 18 00
0012B200  64 E0 01 08 31 00 E1 01 00 06 70 E2 01 01 00 00
0012B210  E3 00 34 73 24 E4 46 44 52 2D E5 41 58 37 30 E6
0012B220  30 00 00 03 00 EA 00 00 13 88 80 00 00 00 01 06
   ⋮
```

Maker : 0x103 = Panasonic
0x108 = Sony
0x1011 = Canon

Model : 0x46 44 52 2D 41 58 37 30 30
(FDR-AX700)
(0xE5, E6 is tag_id)

Time : 2020. 11. 02. 23:28:06
(0x19 is tag_id)

Algorithm 2 Analysis of picture timing

```
Input: $V \leftarrow MTS\ video\ file$
Output: Result of validation, $vaild$ or $not\ valid$
$T \leftarrow ExtractTimestamp(V)$ // Extract the timestamps from the $V$, storing result in the set $T$
for all $t$ in $T$ do
    $d \leftarrow diff(t[i-1], t[i])$
    if $d > 1\ sec$ then
        return $not\ valid$
    end if
end for
return $valid$
```

and *0x4644522DE541583730E630*, where *0xE0*, *0xE4*, *0xE5*, and *0xE6* denote the tag_id.

Although the camera manufacturer and model are repeatedly recorded, the model can be modified, which is a vulnerability in the current scope of the investigation. However, considering other analysis features, verifying the untampered recording of the manufacturer/model information compared with the standard file is a crucial factor.

The manufacturer code appears after tag_id *0xE0*, where *0x103* denotes Panasonic, *0x108* refers to Sony, and *0x1011* indicates Canon. The model name is recorded after tag_id *0xE4*, but it is recorded in the MTS file only for Sony devices. In Canon and Panasonic devices, the model name is not recorded.

3.4 | GOP analysis

The GOP refers to a set of video frames composed of the intra-coded frame (I-frame), predictive coded frame (P-frame), and bidirectional coded frame (B-frame) [21]. An I-frame is referred to as a keyframe, which indicates a frame that is encoded independently of all other frames. A P-frame is a frame with motion-compensated difference information related to the previous frame, and the video is decoded with reference to the previous frame. A B-frame is a two-way reference frame with motion-compensated difference information, which is decoded as video by referring the previous and subsequent frames using interpolation.

An I-frame after several P- and B-frames indicates a new GOP. The GOP in the MTS file can be extracted using ffprobe command ("ffprobe -show_frames filename | grep pict_type"), and the GOP structure remains constant in a single MTS file. In addition, the GOPs extracted from MTS files recorded on the same camera under given recording options are identical to each other. The structure of the GOP is represented by two numbers M and N, where M denotes the distance between two anchor frames (I or P) and N indicates the distance between two full images (I-frames).

For instance, the GOP of an MTS file captured with the 1080-50i FX option of the Sony FDR-AX700 is *IBPBPBPBPBPBBIBPBPBPBPBPBB*. In this case, the GOP can be represented as $M=2$ and $N=13$. As another example, when recorded using the 24p FX 24 M option of the Sony ILCE-7 M2, the GOP is *IPPPPPPPPPPPIPPPPPPPPPPP*, which indicates $M=1$ and $N=12$.

The frame distance *N* depends on the frame rate because the frame rate represents the frequency of the I-frame. Thus, the GOP in the MTS file must be consistent and represent a specific value according to the manufacturer and frame rate. At this point, we analyzed and used it as a feature to verify the integrity of the video.

Algorithm 3 describes the procedure for extracting and validating the GOP patterns, and it returns the analysis result in terms of M and N. In particular, it compiles a list from the I-frame to the succeeding I-frame that extracted the GOP list G from the MTS file. Thereafter, G' is extracted by removing the duplicates. If more than one unique GOP exists, the integrity is not verified because the consistency of the GOP pattern has not been maintained. In the case of a unique GOP, (M,N) is the result of the GOP pattern analyzed in the MTS file.

Algorithm 3 Analysis of GOP patterns

```
Input: $V \leftarrow MTS\ video\ file$
Output: GOP Pattern Set $S = (M, N)$, where $M$ is the distance between two anchor frames, $N$ is the distance between full images. And the pattern is from I-frame to the next I-frame, and M and N are calculated.
$G \leftarrow ExtractGOP(V)$ // $G$ is the GOP list by extracting from $V$
$G' \leftarrow RemoveDuplicate(G)$
if count $(G') > 1$ then
    return $not\ valid$
end if
$(M, N) \leftarrow CalDistanceGOP(G')$ // Calculating distance between frames $(M, N)$ from $G'$
```

3.5 | MTS file integrity verification procedure

Based on the above observations, the proposed forensic examination procedure to authenticate the integrity of AVCHD format (MTS extension) files is presented in Figure 6. First, we review the existence of a UUID that stores metadata information in an MTS file. As the UUID exists in the original video but is not related to video playback, it may be deleted when manipulated. Hence, even given one UUID, failure to maintain consistency in the UUID pattern indicates that the file has been edited.

Subsequently, the picture timing and camera manufacturer/model recorded in the SEI, a component of UUID data, are analyzed. Because all images (frames) stored in a video are saved consecutively, discontinuities in picture timing indicate tampering. Furthermore, the analyzed camera manufacturer/model is examined to determine the standard file(s). The SPS and PPS, which constitute the codec information, are then compared with the case wherein the evidence and standard file(s) are in mutual correspondence. For cameras manufactured by Sony, files can be compared one-to-one with the standard file(s) because the camera manufacturer/model details are recorded in the SEI. For cameras manufactured by Canon and Panasonic, files compared with all standard files from the same manufacturer, as the model is not recorded in the SEI. At this point, if no standard file(s) correspond after comparing the codec information, editing can be suspected.

Finally, the GOP pattern is analyzed. If consistency within the GOP pattern is not maintained, editing is suspected, and its correspondence is compared with the GOP pattern of the standard file(s) determined previously. Upon comparing the UUID, picture timing, and codec information with the standard file and GOP pattern, file integrity is verified if all aforementioned criteria are satisfied.

4 | EXPERIMENTAL EVALUATION

We conducted experiments to evaluate the performance of the proposed method in verifying the integrity of MTS video files. Experiments were conducted with seven commercially available Canon, Sony, and Panasonic cameras that save videos in MTS format. In addition, *manipulated* videos were generated via editing software and analyzed for detection according to the proposed method.

4.1 | Experimental setting

We conducted an experiment to verify the integrity of MTS videos using the features described herein. To analyze and verify the features, three types of MTS files were configured: *original*, *unmanipulated*, and *manipulated*. *Original* videos were recorded with cameras from Canon, Sony, and Panasonic using all possible recording configurations, resulting in a total of 44 videos, as listed in Table 1. All videos used the H.264/High codec. However, the resolution and frame rate, which can affect verification features, were different depending on the recording option.

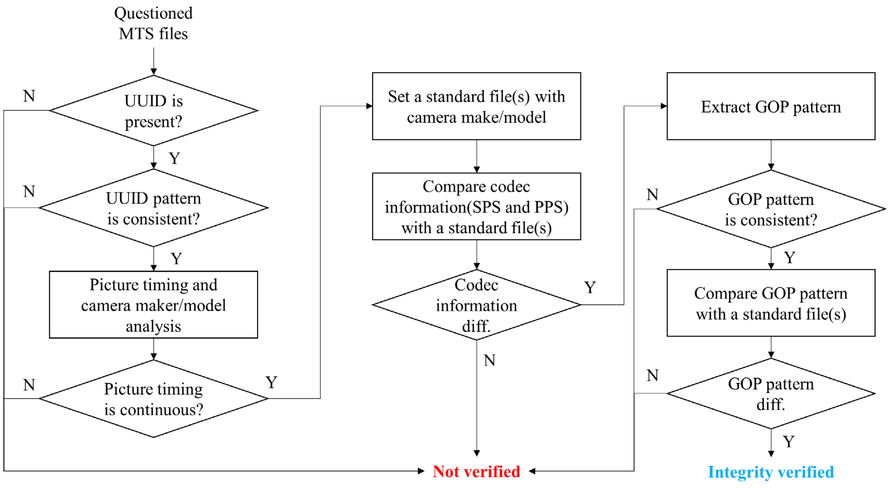

Questioned MTS files
UUID is present?
N
Y
UUID pattern is consistent?
N
Y
Picture timing and camera maker/model analysis
Picture timing is continuous?
N
Y
Set a standard file(s) with camera make/model
Compare codec information(SPS and PPS) with a standard file(s)
Codec information diff.
Y
N
Not verified
Extract GOP pattern
GOP pattern is consistent?
N
Y
Compare GOP pattern with a standard file(s)
GOP pattern diff.
N
Y
Integrity verified

FIGURE 6 Proposal of forensic examination procedure for integrity authentication of advanced video coding high definition format (MTS) files.

TABLE 1 Analysis of the recording option properties of the compared manufacturers and models.

No	Manufacture	Model	Recording option	Codec	Resolution	Frame rate
1	Sony	FDR-AX700	1080-60i FX	H.264/High	1920×1080	29.970
2			1080-60i FH	H.264/High	1920×1080	29.970
3			1080-60i LP	H.264/High	1440×1080	29.970
4			1080-50i FX	H.264/High	1920×1080	25.000
5			1080-50i FH	H.264/High	1920×1080	25.000
6			1080-50i LP	H.264/High	1440×1080	25.000
7		HDR-PJ760	60i Highest Quality FX	H.264/High	1920×1080	29.970
8			60i Highest Quality FH	H.264/High	1920×1080	29.970
9			60i Standard Quality HD	H.264/High	1440×1080	29.970
10			60i Long Time LP	H.264/High	1440×1080	29.970
11			60p Quality PS	H.264/High	1920×1080	59.940
12			24 M Highest Quality FX	H.264/High	1920×1080	23.976
13			24 M High Quality FH	H.264/High	1920×1080	23.976
14		ILCE-7 M2	60i FX 24 M	H.264/High	1920×1080	29.970
15			60i FH 17 M	H.264/High	1920×1080	29.970
16			60p PS 28 M	H.264/High	1920×1080	59.940
17			24p FX 24 M	H.264/High	1920×1080	23.976
18			24p FH 17 M	H.264/High	1920×1080	23.976
19		ILCE-5000	60i FX 24 M	H.264/High	1920×1080	29.970
20			60i FH 17 M	H.264/High	1920×1080	29.970
21			24p FX 24 M	H.264/High	1920×1080	23.976
22			24p FH 17 M	H.264/High	1920×1080	23.976
23	Canon	XA20	28 Mbps LPCM	H.264/High	1920×1080	59.940
24			28 Mbps 59.94p	H.264/High	1920×1080	59.940
25			24 Mbps LPCM 59.94i	H.264/High	1920×1080	29.970
26			24 Mbps LPCM PF 29.97	H.264/High	1920×1080	29.970
27			24 Mbps LPCM 23.98p	H.264/High	1920×1080	23.976
28			24 Mbps 59.94i	H.264/High	1920×1080	29.970
29			24 Mbps PF 29.97	H.264/High	1920×1080	29.970
30			24 Mbps 23.98p	H.264/High	1920×1080	23.976
31			17 Mbps 59.94i	H.264/High	1920×1080	29.970
32			17 Mbps PF 29.97	H.264/High	1920×1080	29.970
33			17 Mbps 23.98p	H.264/High	1920×1080	23.976
34			5 Mbps 59.94i	H.264/High	1440×1080	29.970
35			5 Mbps PF 29.97	H.264/High	1440×1080	29.970
36			5 Mbps 23.98p	H.264/High	1440×1080	23.976
37	Panasonic	LX100II	FHD 28 M 60p	H.264/High	1920×1080	59.940
38			FHD 17 M 60i	H.264/High	1920×1080	29.970
39			FHD 24 M 60i	H.264/High	1920×1080	29.970
40			FHD 24 M 24p	H.264/High	1920×1080	23.976
41		LX10	FHD 28 M 60p	H.264/High	1920×1080	59.940
42			FHD 17 M 60i	H.264/High	1920×1080	29.970
43			FHD 24 M 60i	H.264/High	1920×1080	29.970
44			FHD 24 M 24p	H.264/High	1920×1080	23.976

In addition, a total of 220 *unmanipulated* MTS files were examined, with five videos for each of the 44 recording options in Table 1. For each option, five types of scenes (indoor, outdoor, grid, flat, and white) were recorded to determine whether the change in scene affects the feature analysis for MTS file integrity verification.

Next, a total of 220 *manipulated* MTS files were generated by applying the main editing functions to the original files with Sony PlayMemories [22] and Vegas [23]. In the case of PlayMemories, only MTS videos saved by Sony cameras can be stably edited, and only the insert and trim functions are provided. Unlike other editing software, PlayMemories is characterized by the fact that the structure of the MTS file before editing is preserved even after editing. For this reason, 44 files were created with two edits (insert and trim) for 22 recording options of the four models manufactured by Sony.

Video editing software normally provides four editing functions—*insert*, *trim*, *cut*, and *edit*—to modify video content. Using Vegas, we created 176 files by independently applying the four editing functions for all 44 recording options. As with any video editing software, once editing is complete, the video is re-encoded, resulting in changes to the file structure. Although videos are typically saved in MP4 format after editing, Vegas allows edited videos to be saved in MTS format, making it appropriate for our study.

4.2 | MTS file verification

The MTS files were tested based on the features described in Section 3, that is, codec information (SPS, PPS), UUID, picture timing, camera manufacturer/model, and GOP structure. We analyzed the MTS files for 44 recording options of seven models.

The MTS file was verified based on Figure 1. First, SPS and PPS were extracted from the MTS files, and the camera manufacturer/model were extracted from the SEI. Subsequently, we searched the corresponding manufacturer/models and recording options from the 44 MTS files (Table 1) stored as standard files. In addition, the integrity of each MTS file was verified based on the consistency of the UUID pattern, the continuity of picture timing, and the consistency of the GOP pattern.

The results of verifying the integrity of 220 *unmanipulated* MTS files using the proposed method are listed in Table 2. Among the file features, if the SPS and PPS were extracted from 220 files, all were detected within the 44 standard files in Table 1. Moreover, the extracted camera manufacturer/model precisely corresponded to the recording device. In addition, we confirmed that the UUID pattern, picture timing, and GOP pattern maintained consistency and continuity throughout the recordings. In particular, the *unmanipulated* MTS files were taken in five scenes, and diversity between the scenes did not affect integrity verification through the features proposed in this study. Thus, the *unmanipulated* MTS files can be deemed to be entirely unmanipulated.

The analysis results of five features for 44 *manipulated* MTS files edited with PlayMemories and 176 *manipulated* MTS files edited with Vegas are presented in Table 3. We observed that the files were more affected when edited with a professional editing program such as Vegas. When edited with PlayMemories, the UUID, camera manufacturer/model, and detailed meta-information were saved corresponding to that in the original.

First, as listed in Table 3, when editing using the insert video function in PlayMemories, the SPS, PPS, camera manufacturer/model, UUID pattern, and GOP remain the same without any alterations. In particular, if the SPS and PPS were compared with the dataset of the standard files, the standard file(s) with the same SPS and PPS were detected. However, when the video was inserted, the continuity of the picture timing was not maintained in the inserted portion.

After trimming the video in PlayMemories, the identity of the camera manufacturer/model, consistency of the UUID, and continuity of the picture timing were unaffected. When a segment of the video was trimmed, the same UUID and picture timing were maintained in the trimmed section. However, the SPS and PPS were all altered to the same value in all the trimmed videos, which were not detected in the standard files. Overall, the GOP pattern varied from that in the standard file.

In all the videos edited with Vegas, the SPS and PPS were determined according to the rendering settings, and all UUID information was deleted. Thus, the UUID pattern, picture timing, and camera manufacturer/model could not be further inspected. Although the GOP pattern remained consistent and the videos edited with Vegas were consistent with each other, they differed from those in the standard file.

Thus, if the five features of *manipulated* MTS files were verified, one or more unverified features were detected in 44 *manipulated*

TABLE 2 Verification results of *unmanipulated* MTS files.

Model (recording options)	SPS, PPS	Camera manufacturer/ model	UUID pattern	Picture timing	GOP pattern	Verification result
Sony FDR-AX700(6)	Existence	Sony/FDR-AX700	True	True	True	**Verify**
Sony HDR-PJ760(7)	Existence	Sony/HDR-PJ760	True	True	True	**Verify**
Sony ILCE-7 M2(5)	Existence	Sony/ILCE-7 M2	True	True	True	**Verify**
Sony ILCE-5000(4)	Existence	Sony/ILCE-5000	True	True	True	**Verify**
Canon XA20(14)	Existence	Canon/none	True	True	True	**Verify**
Panasonic LX100II(4)	Existence	Panasonic/none	True	True	True	**Verify**
Panasonic LX10(4)	Existence	Panasonic/none	True	True	True	**Verify**

TABLE 3 Verification results of the *manipulated* MTS files.

Editing S/W	Function	SPS, PPS	Camera manufacturer/ model	UUID pattern	Picture timing	GOP pattern	Verification result
PlayMemories (44)	Insert	Existence	Matched	True	False	True	**Not verify**
	Trim	**Non-existence**	Matched	True	True	**False**	**Not verify**
Vegas (176)	Insert	**Non-existence**	–	–	–	**False**	**Not verify**
	Trim	**Non-existence**	–	–	–	**False**	**Not verify**
	Cut	**Non-existence**	–	–	–	**False**	**Not verify**
	Edit	**Non-existence**	–	–	–	**False**	**Not verify**

TABLE 4 Clustered results of analyzed GOP structural patterns.

Types		Cluster results
Original	Unique	1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23,24, 27, 30, 33, 36 (in Table 1)
	Duplicated	9–10, 25–26, 28–29, 31–32, 34–35, 37–41, 38–42, 39–43, 40–44 (in Table 1)
Unmanipulated		Matched with original files
Manipulated	Sony PlayMemories (Inserting)	Matched with original files
	Sony PlayMemories (Trimming)	Not matched with original files
	Vegas	Not matched with original files and matched each other manipulated by Vegas

MTS files edited with PlayMemories. In the case of the 176 *manipulated* MTS files edited with Vegas, all features could not be verified, and thus, the integrity of the 220 *manipulated* MTS files could not be verified with 100% accuracy.

4.3 | Effectiveness of features

In this study, we identified five features to verify the integrity of MTS files. Each feature was analyzed to determine its effectiveness in integrity verification.

4.3.1 | Effectiveness of codec-based analysis

First, we analyzed the impact of codec-based analysis and found that the codec information—SPS and PPS—vary with respect to camera model and recording option. Specifically, we found that 26 of the 44 recording options offered by the seven models were unique. Of the remaining 18 recording options, 10 (two for HDR-PJ760 and eight for XA20) had matching SPS and PPS values. Options 9–10, 25–26, 28–29, and 34–35 in Table 1 matched according to similar provided options within the same device. For the two Panasonic models, although no matching SPS and PPS values were found within the same model, these values were found to match for the same recording options on different models (options 37–41, 38–42, 39–43, and 40–44 in Table 1).

As shown in Table 4, the *unmanipulated* MTS files were confirmed to match the originals in terms of SPS and PPS. However, for the aforementioned 18 recording options, two duplicate model and recording option matches were found.

Videos edited with Sony PlayMemories can be categorized among two types. In the case of insertion, the SPS and PPS of the original video were maintained, matching the originals taken from the same camera. However, in the case of trimming, no SPS and PPS were found to match the original files for any of the 44 recording options. Furthermore, none of the files matched each other.

The 176 MTS files manipulated in Vegas did not have the same SPS and PPS as the 44 original files in all cases of inserting, trimming, cutting, and editing. However, it was confirmed that the SPS and PPS of the 176 files are not affected by the recording device, and MTS files saved with the same option in Vegas following modification were determined to have the same SPS and PPS.

Thus, the codec information analysis confirms that the *unmanipulated* MTS files had at least one matching SPS and PPS with those of the 44 original files in all cases. In the case of *manipulated* MTS files, tampering could be determined in all cases with the exception of insertions performed via PlayMemories.

4.3.2 | Effectiveness of GOP pattern analysis

When analyzing the impact based on GOP pattern, the pattern was observed to be affected by the frame rate of the model and video. The results of analyzing the GOP structure of the 44 files detailed in Table 1 are listed in Table 5. According to the analyzed results, the GOP is related to the manufacturer and frame rate. Both Canon and

TABLE 5 Clustered results of analyzed GOP structural patterns.

Manufacture	Frame rate	*M*	*N*	No. in Table 1
Sony	23.976	1	12	12, 13, 17, 18, 21, 22
	59.940	1	30	11, 16
	25.000	2	13	4, 5, 6
	29.970	2	15	1, 2, 3, 7, 8, 9, 10, 14, 15, 19, 20
Canon, Panasonic	23.976	3	12	27, 30, 33, 36, 40, 44
	29.970	3	15	25, 26, 28, 29, 31, 32, 34, 35, 38, 39, 42, 43
	59.940	3	30	23, 24, 37, 41

Panasonic exhibited an M of three, and Sony models exhibited an M of either one or two. In addition, *N* was analyzed as 12 for a frame rate of 23,976 fps, 30 for 59,940 fps, 13 for 25,000 fps, and 15 for 29,970 fps.

In the case of *unmanipulated* MTS files, the GOP patterns were observed to be identical irrespective of scene. However, as the GOP patterns were clustered as shown in Table 5, it is not possible to specify the manufacturer and model of a camera solely using the GOP pattern, although it is possible to determine if the video originated from a camera.

In the case of inserting a *manipulated* MTS file into Sony PlayMemories, the GOP pattern was maintained consistently within one file and validated. As shown in Table 5, it was also included in the cluster that the original MTS belongs to. In the trimmed video, various GOP patterns were found in one file, so the GOP pattern was not validated as shown in Table 3.

We found that the GOP patterns of videos manipulated with Vegas are determined by the re-encoding option provided by the editing software, regardless of the manufacturer and model. For example, when the frame rate was re-encoded to 29.970, we found *M* = 2 and *N* = 15 m to be the same as those for Sony. However, when we checked the frame sequence, we found that the original MTS was *IBPBPBPBPBPBPBBIBPBPBPBPBPBPBB*, whereas that of the *manipulated* video was *IBPBPBPBPBPBPBPIBPBPBPBPBPBPBP*.

When analyzing the GOP patterns, it was found that in the case of PlayMemories, wherein edited videos maintain the original structure, the GOP patterns of the merged files were confirmed to match the originals. However, the trimmed files could not be validated because multiple GOP patterns were found. In addition, video files manipulated and re-encoded with Vegas were consistent, and depending on the set frame rate, the GOP patterns are included in the clusters in Table 5, so it was not possible to confirm manipulation using only GOP patterns.

4.3.3 | Effectiveness of media stream analysis

Through the media stream data, we analyzed video files by UUID pattern, picture timing, and manufacturer and model.

In the case of UUID patterns, the consistency of the pattern in which the UUIDs of *<17ee8c60-f84d-11d9-8cd6-0800200c9a66>* and *<c0000000-a746-02bb-f8a1-4cc0a93648e3>* repeatedly appear once in the case of Canon cameras was maintained. For Sony cameras, the UUIDs *<17ee8c60-f84d-11d9-8cd6-0800200c9a66>*, *<a74602bb-f8a1-4cc0-a936-48e391dce761>*, and *<dba1adef-b20c-40b4-8c85-8c0b46d5241e>* were found in a specific pattern that remained consistent. However, the UUID *<dba1adef-b20c-40b4-8c85-8c0b46d5241e>* was only found in all recording options of the FDR-AX700 model, the 60p Quality PS option of the HDR-PJ760 model, and the 60p PS 28 M option of the ILCE-7M2 model. In both models, it was observed only when the frame rate was 59,940. Panasonic cameras were consistent, with *<17ee8c60-f84d-11d9-8cd6-0800200c9a66>* appearing once.

Because consistency of the UUID pattern was maintained in *unmanipulated* MTS files as well as files manipulated via PlayMemories, integrity validation was not validated by the UUID pattern. No UUIDs were found in videos manipulated with Vegas. It was confirmed that the UUID information and all additional information disappeared in the process of re-encoding after editing.

Picture timing can be found in the User Data of the UUID, namely *<17ee8c60-f84d-11d9-8cd6-0800200c9a66>*. By analyzing the time information in the User Data, we found that each frame was recorded more than once per second. The original and *unmanipulated* MTS file exhibited continuity in the timing of the recorded picture within 1 s.

Among the *manipulated* MTS files, the picture timing information of the videos manipulated with PlayMemories was confirmed. However, for the merged MTS files, the continuity of picture timing was not confirmed at the merged point. Because the trimmed MTS file was trimmed from the front and back of the video, the continuity of picture timing was validated as an integrity measure. Videos edited with Vegas could not be validated, as all UUID information is removed.

In regards to manufacturer and model information, both types of information are stored in video files taken by Sony cameras, whereas only manufacturer information is stored in videos taken by Canon and Panasonic cameras. In *unmanipulated* MTS files, this information is stored in an identical format as that in the original files for all three manufacturers.

For files manipulated with PlayMemories, the manufacturer and model information were maintained, validating this feature's integrity. Videos edited with Vegas could not be validated, as all UUID information is removed during the editing process.

In conclusion, it was possible to validate the integrity of MTS files through the analysis of the aforementioned features using media stream information. Although UUID pattern consistency and picture timing consistency cannot be used to identify the manufacturer and model of a camera, they can confirm the presence of manipulation. In addition, videos recorded by Sony cameras can be identified in terms of manufacturer and model, whereas those recorded by Canon and Panasonic cameras can only be identified by manufacturer. However, as shown in Figure 5, features 4 and 5 are recorded in hexadecimal and text format, making them susceptible to editing. Picture timing is a complex feature because temporal information must be edited throughout the video file. In contrast, manufacturer and model information is easy to edit because it is constant throughout the video. For this reason, although media stream analysis cannot be a strong basis for identifying manipulation, it can be used as a supplement.

4.4 | Discussion

To verify the integrity of MTS files encoded in AVCHD, this study analyzed five features: SPS/PPS, GOP pattern, UUID pattern, picture timing, and camera manufacturer/model. To examine the effectiveness of the features, the MTS files for 220 *unmanipulated* MTS files of each of five videos of 44 recording options from seven models were analyzed. Additionally, we experimented whether the *manipulated* MTS files edited with Sony PlayMemories and Vegas can be verified. Consequently, based on the experiment, we were able to observe the following for each type of MTS file.

- *Unmanipulated* MTS analysis: The results confirmed that SPS and PPT corresponded with the standard files. The camera manufacturer/model and GOP pattern also corresponded. The UUID pattern was consistent and the picture timing was continuous. Overall, the integrity of the unedited *unmanipulated* MTS file could be verified.
- *Manipulated* MTS analysis: PlayMemories maintained the UUID data prior to editing such that certain features were the same as the standard file. However, all features were not verified owing to variations. Because editing programs such as Vegas perform the re-encoding process, the UUID information is removed and codec information (SPS, PPS) varies; thus, the GOP pattern varied, and all features differed from those of the standard file. Overall, the integrity of all *manipulated* MTS files was not verified.

In this study, we proposed and validated a method as depicted in Figure 6, to verify the integrity of a video file through analysis. The manipulated video was suspected of being edited because its integrity was not verified according to the result of the abovementioned feature analysis. Thus, the method proposed in this paper can accurately validate the integrity of MTS files submitted as evidence by comparison with the original files, assuming that the camera used to record the videos is known. In contrast, as shown in Figure 6, codec information and GOP patterns cannot be used for verification without standard file(s) if the camera model is not accurately known. Therefore, maximizing data representation within standard files is necessary to improve accuracy in all possible situations. If so, the proposed approach is suitable for practical applications, and the results of the current study will be useful in AVCHD format-related forensics.

5 | CONCLUSIONS

The proposed method verified the integrity of AVCHD encoded MTS files. In the field of digital forensics, the most basic approach is to acquire all digital evidence from storage media in a forensic manner. Importantly, a chain of custody (CoC) must be secured so that a file may be used as a legal evidence in trials because videos can reproduce the situation at the scene.

Video files can be modified by insertions, trimming, cutting, and editing using editing tools. Thus, its integrity must be established to strengthen the power of proof as evidence. The proposed method analyses the features that can evince the integrity of unedited MTS files and explain the mechanism of identifying the submitted video file as the original or *manipulated* MTS file based on feature comparisons.

Accordingly, we analyzed 44 AVCHD encoded MTS files captured using cameras manufactured by Sony, Canon, and Panasonic. The analysis results revealed consistencies in UUID pattern, picture timing continuity, camera manufacturer/model analysis, equal codec information (SPS and PPS), and GOP consistency, which are the five major features that can establish the integrity of MTS files.

Consequently, the variation between *unmanipulated* and *manipulated* MTS files could be analyzed using these five features, and we confirmed that the features were valid through comparison with MTS files captured from actual devices. It is particularly important that the results of the technique proposed herein can obtain a chain of custody (CoC) by verifying the integrity of AVCHD encoded video files using MTS file analysis, which has not been proposed before.

In the future, we will extend our methodology to determine whether a transmitted or shared file maintains the form and integrity of the original file, which will require an expanded standard file database. Furthermore, we intend to extend the proposed technique to identify the editing tools used to modify MTS files.

FUNDING INFORMATION

This work was supported by National Forensic Service (NFS2023DTB04), Ministry of the Interior and Safety, Republic of Korea and Institute of Information & Communications Technology Planning & Evaluation (IITP) (No.2022–0-00277, Development of SBOM Technologies for Securing Software Supply Chains) grant funded by the Ministry of Science and ICT, Republic of Korea.

CONFLICT OF INTEREST STATEMENT

The authors have no conflicts of interest to declare.

ETHICS STATEMENTS

All study participants provided informed consent. The authors confirm that the study complies with the guidelines included in the JFS Information for Authors.

ORCID

Kyu-Sun Shim https://orcid.org/0000-0001-9926-7378

REFERENCES

1. Cisco visual networking index: forecast and trends, 2017–2022. San Jose, CA: Cisco; 2017.

2. Piva A. An overview on image forensics. Int Sch Res Notices. 2013;2013:496701. https://doi.org/10.1155/2013/496701
3. Iuliani M, Shullani D, Fontani M, Meucci S, Piva A. A video forensic framework for the unsupervised analysis of MP4-like file container. IEEE Trans Inf Forensics Secur. 2018;14(3):635–45. https://doi.org/10.1109/tifs.2018.2859760
4. Korus P. Digital image integrity–a survey of protection and verification techniques. Digit Signal Process. 2017;71:1–26. https://doi.org/10.1016/j.dsp.2017.08.009
5. SWGDE. Best practices for image authentication. 2018. https://drive.google.com/file/d/1IhMWleu-i-jW4LBSOIQIJnejNOfdCdJs/view?usp=sharing. Accessed 22 Apr 2023.
6. Huamán CQ, Orozco ALS, Villalba LJG. Authentication and integrity of smartphone videos through multimedia container structure analysis. Future Gener Comput Syst. 2020;108:15–33. https://doi.org/10.1016/j.future.2020.02.044
7. Gloe T, Fischer A, Kirchner M. Forensic analysis of video file formats. Digit Invest. 2014;11:S68–S76. https://doi.org/10.1016/j.diin.2014.03.009
8. Song J, Lee K, Lee WY, Lee H. Integrity verification of the ordered data structures in manipulated video content. Digit Investig. 2016;18:1–7. https://doi.org/10.1016/j.diin.2016.06.001
9. AVCHD information. http://www.avchd-info.org/. Accessed 22 Apr 2023.
10. Digital Camera World. Camera market share. https://www.digitalcameraworld.com/news/camera-market-share-canon-owns-48-sony-22-nikon-drops-to-14. Accessed 22 Apr 2023.
11. Park NI, Shim KS, Lee JW, Kim JH, Lim SH, Byun JS, et al. Advanced forensic procedure for the authentication of audio recordings generated by voice memos application of iOS14. J Forensic Sci. 2022;67(4):1534–49. https://doi.org/10.1111/1556-4029.15016
12. Shanableh T. Detection of frame deletion for digital video forensics. Digit Investig. 2013;10(4):350–60. https://doi.org/10.1016/j.diin.2013.10.004
13. Vazquez-Padin D, Fontani M, Bianchi T, Comesaña P, Piva A, Barni M. Detection of video double encoding with GOP size estimation. In: Proceedings of the 2012 IEEE international workshop on information forensics and security (WIFS) IEEE; 2012 Dec 2–5; Costa Adeje, Spain. Piscataway, NJ: IEEE. 2012. p. 151–6. https://doi.org/10.1109/wifs.2012.6412641
14. Gironi A, Fontani M, Bianchi T, Piva A, Barni M. A video forensic technique for detecting frame deletion and insertion. In: Proceedings of the 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP); 2014 May 4–9; Florence, Italy. Piscataway, NJ: IEEE; 2014. p. 6226–30. https://doi.org/10.1109/icassp.2014.6854801
15. Ding X, Yang G, Li R, Zhang L, Li Y, Sun X. Identification of motion-compensated frame rate up-conversion based on residual signals. IEEE Trans Circuits Syst for Video Technol. 2017;28(7):1497–1512. https://doi.org/10.1109/tcsvt.2017.2676162
16. Su L, Li C, Lai Y, Yang J. A fast forgery detection algorithm based on exponential-Fourier moments for video region duplication. IEEE Trans Multimedia. 2017;20(4):825–40. https://doi.org/10.1109/tmm.2017.2760098
17. Thing VL, Ng KY, Chang EC. Live memory forensics of mobile phones. Digit Investig. 2010;7:S74–S82. https://doi.org/10.1016/j.diin.2010.05.010
18. Orozco AS, González DA, Corripio JR, Villalba LG, Hernandez-Castro J. Techniques for source camera identification. In: Proceedings of the 6th international conference on information technology; 2013 May 8–10; Amman, Jordan. Piscataway, NJ: IEEE; 2013. p. 1–9.
19. ISO. ISO/IEC Standard 14496-10:2022. Information technology—Coding of audio-visual objects—Part 10: Advanced video coding. 2022. https://www.iso.org/standard/83529.html. Accessed 3 Jul 2023.
20. Vetro A, Wiegand T, Sullivan GJ. Overview of the stereo and multiview video coding extensions of the H. 264/MPEG-4 AVC standard. Proc IEEE. 2011;99(4):626–42. https://doi.org/10.1109/jproc.2010.2098830
21. Group of pictures. https://en.wikipedia.org/wiki/Group_of_pictures. Accessed 22 Apr 2023.
22. Sony PlayMemories. https://support.d-imaging.sony.co.jp/www/disoft/int/download/playmemories-home/win/en/. Accessed 22 Apr 2023.
23. Vegas. https://www.vegascreativesoftware.com/. Accessed 22 Apr 2023.

How to cite this article: Shim K-S, Park NI, Lim SH, Byun JS, Lee H. Forensic integrity verification of video recordings based on MTS files. J Forensic Sci. 2023;00:1–14. https://doi.org/10.1111/1556-4029.15335