International Journal of Internet, Broadcasting and Communication Vol.10 No.3 128-135 (2018)
http://dx.doi.org/10.7236/IJIBC.2018.10.3.128

IJIBC 18-3-17

Design and Implementation of Video File Structure Analysis Tool for Detecting Manipulated Video Contents

Yun-Seok Choi

Dept. of Computer Science, Dongduk Women's University, Korea
cooling@dongduk.ac.kr

Abstract

The various video recording device, like car black box and cctv, are used currently and video contents are used as evidence of traffic accidents and scenes of crime. To verify integrity of video content, there are various study on manipulated video content analysis. Among these studies, a study based on analysis of video file structure and its variables needs a tool which can be used to analyze file structure and extract interested attributes. In this paper, we proposed design and implementation of an analyzing tool which visualizes video file structure and its attributes. The proposed tool use a model which reflects commonality of various video container format, so it is available to analyze video structure with regardless of the video file types. And the tool specifies interested file structure properties in XML and therefore we can change target properties easily without modification of the tool.

Keywords: *Video Container Format, File Structure, visualization, XML*

1. Introduction

Video contents are used as part of important evidence about traffic accidents or scenes of crime in current and therefore various video recording devices are used widely[1, 2]. Use of video editing tools has increased as the use of video contents has become common[3]. Thus the possibility of manipulating video contents has also increased. To verify the integrity of video contents, there are various studies on manipulated video content detection. Among the studies, there is an approach to verify the integrity of video content by analyzing the structure of the video file[4, 5]. In the study, the integrity verification for video content is performed by characteristics that video recording tools or editing tools make a unique file structure and properties in the video file. A tool is required that performs identification and visualization for video file structure and extracts interested properties. In this paper, we propose design and implementation of video file structure analysis tool which identifies the video file structure depending on its video container format and extract interested variables of the structure. The proposed tool use a model which reflects commonality of video container formats, so the tool is available to analyze video structure with regardless of the video file

Manuscript Received: July. 24, 2018 / Revised: July. 30, 2018 / Accepted: Aug. 9, 2018
Corresponding Author: cooling@dongduk.ac.kr
Tel: +82-2-940-4683, Fax: +82-2-940-4170
Dept. of Computer Science, Dongduk Women's University, Korea

types. And the tool specifies interested structure properties in XML and therefore we can change target properties easily without modification of the tool. The rest of this paper is organized as follows. Section 2 we review the backgrounds of our work. Design and implementation of the video file structure analysis tool is shown in Section 3. The implementation and discussion about the tool are shown in Section 4. Section 5 concludes this paper.

2. Background

2.1 Video container format

A video container format, such as the AVI or MP4 is a type of multimedia file that defines storing structure of video, audio and metadata. The AVI container format proposed by Microsoft is one of the multimedia file standards using Riff file specification. It can be used to capture, edit, and playback audio or video sequences[7, 8]. Multiple data streams of audio and video with different data type are stored in the form of the AVI container. In the AVI Container format, FOURCCs codes are used to identify stream types, data chunks, index entries and so on. The AVI structure consists of RIFF Header and data which is set of lists and chunks. The RIFF header consist of FOURCCs code 'RIFF', data size and file type. The data size is total size of the data except 'RIFF' and size itself. The file type represents type of video file and has the value FOURCCs code 'AVI '. In the data, a chunk consists of a chunk id of FOURCCs code, size, and data. A List consists of list id, list type, size, and data. The list id has FOURCCs code predefined value 'LIST'. The 'hdrl' list, which defines the format of the stream, and the 'movi' list, which represents stream data, are required lists of an AVI file. Among the various lists, 'hdrl' list contains main AVI header 'avih' and information about stream header and stream format, so it is major concerns of the study of detecting manipulated video contents based on file structure analysis.

The MP4 (MPEG-4 Part 14) is a multimedia container format standard defined as part of ISO / IEC 14496-14: 2003[9]. A multimedia file which uses the MP4 container format has object-oriented based structure and consists of a set of boxes. The box is a basic data unit in MP4 container format and it consists of size, type, and data[9,10]. The size represents the total size of the box including itself. The type specifies the type and format of the data being stored. The type value uses four-character ASCII to represent its type and stores data with big-endian byte ordering. A box can contain other boxes. Boxes that can contain other boxes are 'moov', 'trak, 'mdia', 'minf', 'dinf', 'udta', 'edts', 'stbl' and etc. 'ftyp', 'free', 'mvhd', 'iods', 'tkhd', 'mdhd', and 'hdlr' are the primary target boxes of the study of detecting manipulated video contents based on file structure analysis.

2.2 Analysis of video file structure for detection of manipulated video contents.

Most methods for detection of manipulated video contents are based on the analysis of visual elements, so it is difficult to identify manipulated video contents by editing programs. To solve this problem, [4] proposed a method to verify manipulated video contents based on the fact that each recording device or editing software makes video contents with a unique file structure. They researched that an editing software uses a unique structure of ordered fields in a video file and what editing software is used to edit video contents can be identified by structure comparisons. In [5], video file structures are analyzed with forensic viewpoints. There are diverse structural variations of video contents depending on recording device or editing software because the AVI or MP4 container format did not define the file structure strictly. In case of editing video contents, editing software add a specific information to the structure of video file. When editing video content, some of the structure properties of the original file are replaced by specific information of the editing software. So, if specific information of an editing software exist in a video file structure, it can be

found what kind of editing software is used to edit the video content. To analyze video file structure, it is necessary to use a tool that identify the file structure of video content and extract specific properties. So, in this paper we proposed design and implementation of the tool.

3. Design and implementation of Analysis Tool

A video container format defines unique components that make up the structure of the file. In this chapter, we will research a model for representing components of the container format and propose the architecture of analyzing tool.

3.1 Video structure model

The structure of the video file is determined by what video container format is used. The container format can be divided into a header representing the meta-data of the video file and a body representing the video and audio data. Each part is a set of components defined by the container. In the AVI, a video file is a set of Riff header, lists and chunks. The MP4 file is a set of boxes. Figure 1 shows components of AVI and MP4. The value in parentheses indicates the byte size.

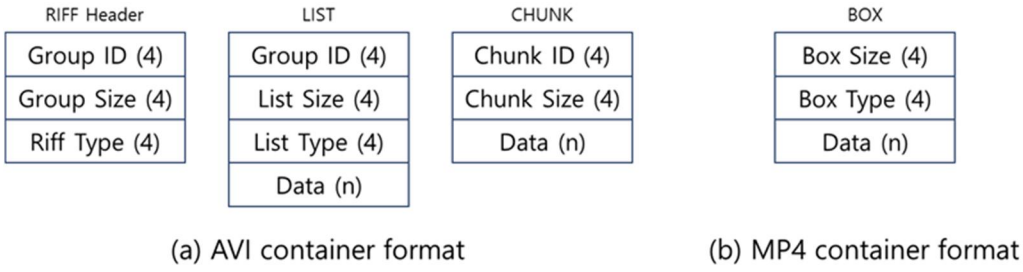


Figure 1. Video container format components

The components of any container format also have size and identifier that identifies the component. The Riff Type, List Type, Chunk ID, and Box Type is an identifier. The Group ID of Riff Header and the Group ID of List store a predetermined value to indicate the type of the component, and the Data field is optional. Therefore, component fields of the AVI and MP4 container format can be expressed as follows.

Table 1. Component fields of video container formats

Fields	Description
Type	Component classification, RIFF | LIST | CHUNK | BOX
ID	Component identifier, FOURCCs code
Size	Component size, 32 bit unsigned integer value
Data	Data in Component, byte array

In this paper, we designed a component model of video file structure which based on the table 1 and [6]. Figure 2 show a component model for video container formats.

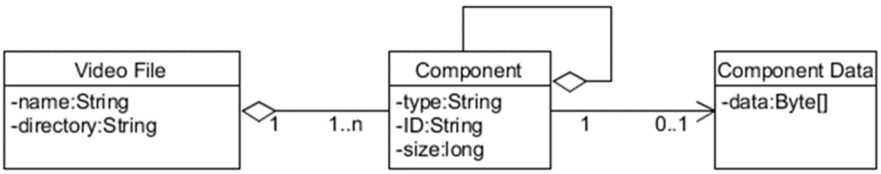


Figure 2. Component model of video container formats

3.2 System architecture

The video file structure analysis tool parse the video file structure and identify it as a set of components. The components are parsed into a set of properties or data chunks. So, the main parts of the tool's architecture are the structure parser and the structure property reader. Figure 3 shows the overall architecture of the video file structure analysis tool.

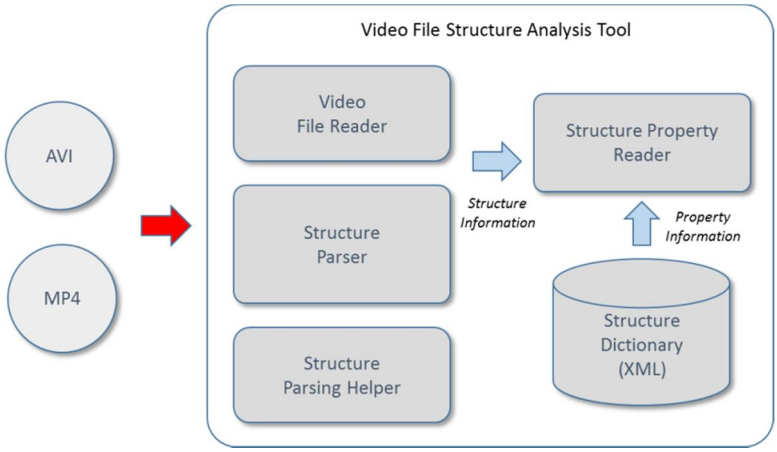


Figure 3. Overall system architecture

The Video File Reader reads a video file in binary form and provides the data to the Structure Parser after scanning the basic information of the file such as file location, physical file name, and file size. The Structure Parser constructs a set of components based on the component model by parsing the received video structure data from the Video File Reader. The Structure Parsing Helper supports parsing functions such as binary file reading and type casting to the Structure Parser. The Structure Property Reader analyzes the results of the Structure Parser and extracts the properties of each component. The Structure Property Reader identifies the components by referring to the Structure Dictionary and extracts interested properties and outputs them in various display formats. The Structure Dictionary stores the information of the container format in the form of XML. Figure 5 show XML structure for interested components of container format.

```
<containers>
    <container type="avi | mp4">
        <component id="container_format_item_ID" read_type="table|string|hex" >
            <property name="property_name" offset="relative_offset"
                      size="size_in_bits" data_type="string|long|hex"/>
            ...
        </component>
        ...
    </container>
    ...
</containers>
```

Figure 4. XML structure for interested components of container format

<containers> can contain one or more <container>. <container> represents the container format of the video file and contains one or more <components>. <component> represents the component of interest in container formats. The id attribute is the identifier of the component, which records the FOURCCs code of the AVI or the identifier of the MP4 components. The read_type attribute records in which form the value of the component is to be showed. If the value is 'table', the component data is parsed as properties and the data can be showed in each of property. If the value is 'string', the binary value is converted into a string and if the value is 'hex', it is shown in binary form. <property> means the detail attribute of <component> and may appear one or more. The <property> is consists of the property name and relative position from start of the component, the bit size of the property, and the data type of the property.

4. Implementation and discussion

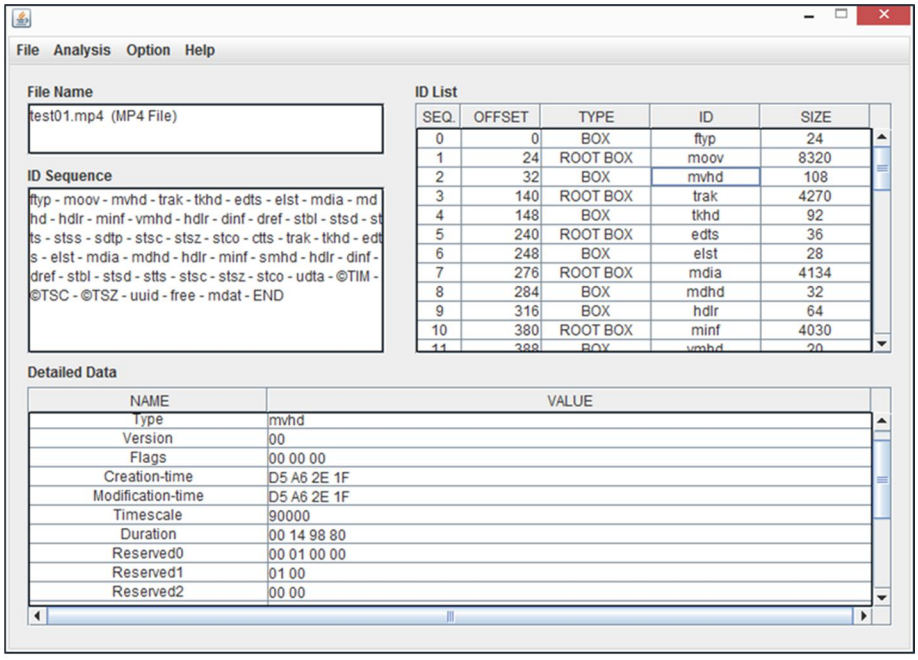
We implemented the video structure analysis tool based on the proposed system architecture with Java. The implemented tool parses and shows detailed information about AVI Headers for AVI file and each of property fields of BOX for MP4 file. The information can be displayed in the form of a processed form, a string, and a binary form. Figure 5 shows XML structure of interested components and the video structure analysis tool.

The tool shows the component id sequence and basic data of the components after parsing and analyzing. The component ID Sequence can be used to find what device or editing software is used to create the video file. The ID List displays the component IDs in the order in which they appear in the file structure, the absolute offset value of each component, the type of component, id, and size. When a component is selected in the ID List, properties and its value of the component is displayed. If the component is defined in the Structure Dictionary, the detailed property data can be displayed in the form of table. Therefore, when detecting the video manipulation based on the change of the property value, just specify the interested component in the Structure Dictionary, the detailed component data can be displayed.

In order to verify the usefulness of the implemented video structure analysis tool, a file structure analysis was performed on a total of 50 AVI and MP4 videos. As a result of analyzing video structure for each file, we confirmed that structure analysis is possible regardless of the type of the video container format. Figure 6 show the example of the analysis result about an AVI file and a MP4 file.

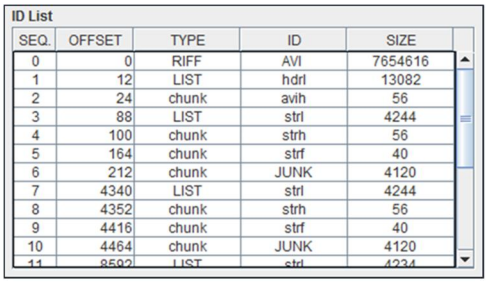
```
<container type="mp4">
    <item id="mvhd" read_type="table">
        <property name="Version" offset="64" size="8" data_type="hex" />
        <property name="Flags" offset="64" size="24" data_type="hex" />
        <property name="Creation-time" offset="96" size="32" data_type="hex" />
        <property name="Modification-time" offset="128" size="32" data_type="hex" />
        <property name="Timescale" offset="160" size="32" data_type="hex" />
        <property name="Duration" offset="192" size="32" data_type="hex" />
        <property name="Reserved0" offset="224" size="32" data_type="hex" />
        <property name="Reserved1" offset="256" size="16" data_type="hex" />
        <property name="Reserved2" offset="272" size="16" data_type="hex" />
        <property name="Reserved3" offset="288" size="64" data_type="hex" />
        <property name="Reserved4" offset="352" size="288" data_type="hex" />
```

(a) A part of interested component definition in XML

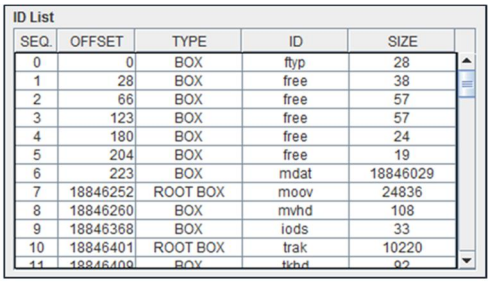


(b) Implementation of the video structure analysis tool

Figure 5. Component definition and video structure analysis tool



(a) Item list of AVI file



(b) Item list of MP4 file

Figure 6. Examples of file structure analysis result

It is possible to check the value of each component by selecting an item of the ID List. In the case of the component specified as a target in the Structure Dictionary, the detailed information of properties constituting the component in the form of a table could be displayed. In other cases, the value of the component could be displayed in the form of string or hex. Figure 7 shows the detailed data forms of a

component.

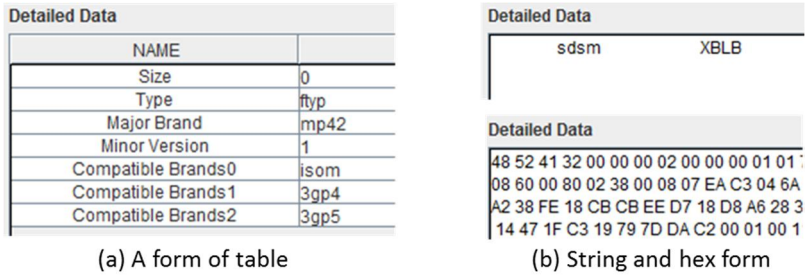


Figure 7. Examples of detailed component data

Therefore, we can confirm the desired property in various form with only modify the XML when we analyze the manipulation of a video content by checking a specific property.

5. Conclusions

As the use of video data increases, the integrity of video content becomes more important. There are methods for verification of video content through video structure analysis in researches to verify video content. Tool support is essential for video structure analysis, so we proposed the design and implementation of a tool for video structure analysis. The proposed tool can analyze the structure regardless of the type of containers by using the structural model reflecting various video container formats, since the components of interest in the video structure are designated to XML, it is easy to modify the setting of the tool to change the target component or to analyze video contents having the new file structure. Therefore, it is possible to perform various types of video content analysis using the tool, and it is possible to easily select and analyze the information of interest components. In future studies, we will develop automated tools that verifies whether video contents is manipulated by applying rules which are created through video file structure analysis.

Acknowledgement

This research was supported by the Dongduk Women's University Grant, 2016

References

[1] S.O Choi, Y.P Kim, Y.S Im, Y.J Kim, and E.Y Kang, "Smart Mobile Blackbox DVR in Car Environment," The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 13, No. 5, 2013
DOI: http://dx.doi.org/10.7236/JIIBC.2013.13.5.9

[2] J.Y Choi and N.S Chang, "Integrity Verification in Vehicle Black Box Video Files with Hashing Method," The Journal of Korean Institute of Communications and Information Sciences, Vol. 42, No. 1, pp. 241-249, 2017.
DOI: 10.7840/kics.2017.42.1.241

[3] B.G Jeong, Y.I Kim, and I.K Eom, "Blurred Edge Detection Method for Forged Image Decision," Journal of Korean Institute of Information Technology, Vol. 11, No. 8, pp. 93-100, 2013.

[4] J.E Song, K.Y Lee, W.Y Lee, and H.J Lee, "Integrity Verification of the Ordered Data Structures in Manipulated Video Content," Digital Investigation, Vol. 18, pp. 1-7, 2016.
DOI: https://doi.org/10.1016/j.diin.2016.06.001

[5] T. Gloe, A. Fischer, and M. Kirchner, "Forensic Analysis of Video File Formats," Digital Investigation, Vol. 11, pp. 68-76, 2014.
DOI: https://doi.org/10.1016/j.diin.2014.03.009
[6] Y.S Choi, "A Reference Model for Representation of Video File Container Structure," Dongduk Journal of Natural Science and Computer Science, Vol. 2, pp. 71-78, 2017
[7] S.G Kim, Technology and Status of Video Compressions, Conservation of the Archives, Vol. 5, pp. 47-62, 2012
[8] Microsoft Docs, AVI Riff file reference, https://docs.microsoft.com/en-us/windows/desktop/directshow/avi-file-format
[9] Information Technology –Coding of Audio-Visual Object– Part 14: MP4 file format, International Standard ISO/IEC 14496-14:2003(E), https://www.iso.org/standard/38538.html
[10] Apple Developer, Quick Time File Format, https://developer.apple.com/standards/qtff-2001.pdf