Analysis of the HIKVISION DVR File System

Jaehyeok Han, Doowon Jeong, and Sangjin Lee(✉)

Center for Information Security Technologies (CIST), Korea University, Anam-Dong,
Seoungbuk-Gu, Seoul, Republic of Korea
{one01h,dwjung77,sangjin}@korea.ac.kr

Abstract. The video security market has recently seen a great expansion in addition to an increasing usage of the Digital Video Recorder (DVR), a device for storing and managing video data on a hard disk under file systems. This study first analyzes its file system for evaluating the DVR and examines the HIKVISION, a video surveillance product supplier, and its proprietary file system on the DVR that has yet been widely recognized by the market. Thus, this paper comprehensively analyzes the HIKVISION DVR file system and proposes a reliable method for digital forensic analyses.

Keywords: Digital forensic · File system · DVR · HIKVISION

1 Introduction

In recent years, video surveillance systems are widely used for various purposes. An embedded DVR (Digital Video Recorder), one of video surveillance systems, is used to monitor behavior, activities, or other changing information. In particular, general DVRs record video data in a digital format on a mass storage device.

Numerous DVRs use well-known file systems, such as FAT [1] and XFS [2]. However, the HIKVISION [3] products use its own file system, assumedly for increasing the efficiency of video management and copy protection. This study temporary named the system as a HIKVISION file system because it does not have an official name. The HIKVISION file system is a simple file system compare to other file systems; the HIKVISION file system excludes some of file operations and includes only the necessary functions. For example, in the HIKVISION file system, it is not possible to delete a file or change a filename. Despite its high market shares, the HIKVISION has yet conducted any related digital forensic analysis. Reference [4] tried to obtain video data in the hard disk using keywords of frame, however, there is no discussion about the file system and storage mechanism.

Therefore, this study identifies the structure of a HIKVISION file system. This study uses a video file format [5] and the reverse engineer manufacturer's application software [6] to analyze the DVR hard disk storage system. The video compression format also comprises a mechanism to extract meaningful information from video data fragments [7]. After identifying the file system, this study conducts an operation test on the file system to analyze the system in detail. This study demonstrates the structure of a HIKVISION file system and proposes a reliable method to access video data and counter anti-forensic activities, such as system initialization or data overwriting.

J.I. James and F. Breitinger (Eds.): ICDF2C 2015, LNICST 157, pp. 189–199, 2015.
DOI: 10.1007/978-3-319-25512-5_13

2 The HIKVISION File System Structure

The basic logic of a HIKVISION file system is that each video data is allocated in the data structure, called a data block entry, which contains the time records, channels, and starting locations of the data block. A video data area is placed in data units called data blocks. If video data were to be allocated in more than one data block, other data blocks can be found by using a structure called the HIKBTREE. The HIKBTREE is used to identify the data block in a video data area, and it is also used to identify the allocation status of the data block.

The layout of a HIKVISION file system consists of four physical sections as shown in Fig. 1. The first section, the *Master Sector* has the information about the overall structure of the file system. The second section, the *System Logs* store the data regarding the events and condition of a DVR. The third section, the *Video Data Area* has numerous data blocks for storing video data. The fourth section, the *HIKBTREE* contains the metadata of video data, including the time records and others.

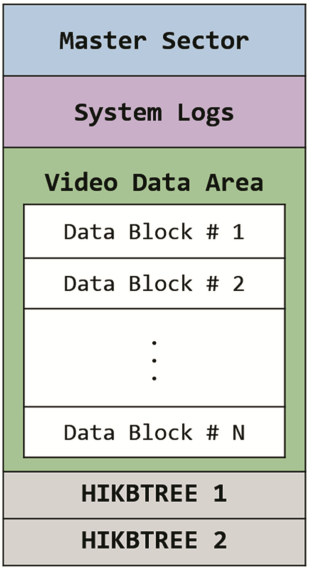
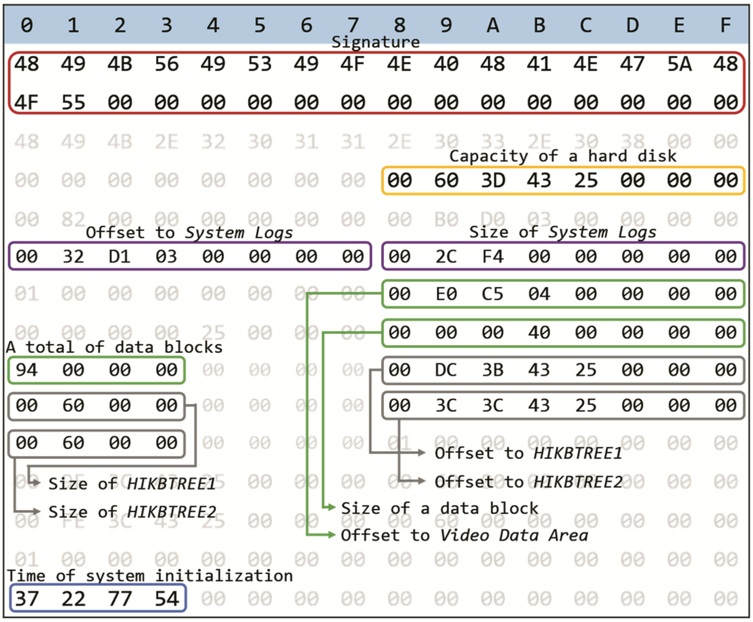


Fig. 1. The physical layout of the HIKVISION File System.

2.1 Master Sector

The Master Sector has the information about the overall structure of the file system. This area starts from the offset 0x200 and the size of master sector is 256 bytes. The signature values of the file system is 'HIKVISION@HANGZHOU (0x48 49 4B 56 49 53 49 4F 4E 40 48 41 4E 47 5A 48 4F 55)' as shown in Fig. 2. The Backup Master Sector is located next to system logs and stores exactly the same data.



```
0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
48 49 4B 56 49 53 49 4F 4E 40 48 41 4E 47 5A 48
4F 55 00 00 00 00 00 00 00 00 00 00 00 00 00 00
48 49 4B 2E 32 30 31 31 2E 30 33 2E 30 38 00 00
00 00 00 00 00 00 00 00 00 60 3D 43 25 00 00 00
00 82 00 00 00 00 00 00 00 B0 D0 03 00 00 00 00
00 32 D1 03 00 00 00 00 00 2C F4 00 00 00 00 00
01 00 00 00 00 00 00 00 00 E0 C5 04 00 00 00 00
00 00 00 00 25 00 00 00 00 00 00 40 00 00 00 00
94 00 00 00 00 00 00 00 00 DC 3B 43 25 00 00 00
00 60 00 00 00 00 00 00 00 3C 3C 43 25 00 00 00
00 60 00 00 00 00 00 00 01 00 00 00 00 00 00 00
00 [illegible] 3C 43 25 00 00 00 00 [illegible] 00 00 00
00 FE 3C 43 25 00 00 00 00 00 60 00 00 00 00 00
01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
37 22 77 54 00 00 00 00 00 00 00 00 00 00 00 00
```

Fig. 2. An example of Master Sector.

The data in each field is stored by Little-endian systems. The Master Sector defines the followings: the capacity of a hard disk (0x25433D6000), offset and size of the system logs (0x3D13200 and 0xF42C00), offset to the video data area (0x4C5E000), size of a data block (0x400000), total number of data blocks (0x94), offset of the HIKBTREE (0x25433BDC00), size of the HIKBTREE (0x6000), time of system initialization (0x37227754), and others. The hexadecimal values in the brackets are the values of samples as shown in Fig. 2.

The *time of system initialization* in the Master Sector is referred as the last system initializes the UNIX time in the UTC. Since the DVR does not provide a delete function, new video data can only be recorded after initializing or overwriting of the system. Thus, this value could be used as an important factor as previously identified by investigators for anti-forensic usage such as in the case, where the time records are fabricated and misused as digital evidence.

2.2 System Logs

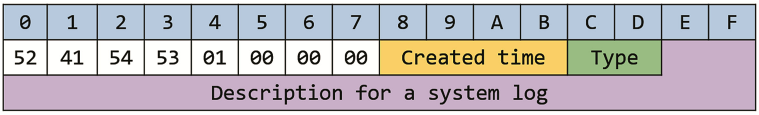
The system logs have system logs information about the events and condition of a DVR. By analyzing this, it is possible for users to discover the operation history and track the DVR performance. Offset to the system logs is defined in the Master Sector.

The system logs are classified into four types in the HIKVISION DVR and each type of the system logs also have several detail logs (Table 1).

Table 1. Types of System Logs.

Type	Value	Description
Alarm	0x01	- Start Motion Detection - Stop Motion Detection, etc.
Exception	0x02	- Video Loss Alarm - Illegal Login - HDD Full, etc.
Operation	0x03	- Power On/Local Operation Shutdown - Local Operation: Login/Logout - Local Operation: Configure Parameters - Abnormal Shutdown, etc.
Information	0x04	- Local HDD Information - HDD S.M.A.R.T - Start Recording/Stop Recording, etc.

The Fig. 3 shows the structure of a system log. Each system log starts with a constant value ‘RATS (0x52 41 54 53 01 00 00 00)’. The value of the *Created time* stores the UNIX time, when a system log is generated. Next to the type, the *description for the system log* is recorded. Because this field is a variable for each system log, the size of this field is also different according to each type of a system log.



0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
52	41	54	53	01	00	00	00	Created time				Type			
Description for a system log															

Fig. 3. Structure of a system log.

2.3 Video Data Area

The video data area stores video data in numerous data blocks. All data blocks and sizes of video data areas are defined in the Master Sector. The size of one data block is generally 1 GB (0x40000000bytes). A data block stores video data areas according to the channels and time records. In order to access the video data, the HIKBTREE should be identified first. This process is covered later in the paper.

A data block is divided into *Video data* and *IDR table*, the former occupies most of the data block and the latter is at the back of a data block (Fig. 4). Video data is encoded to H.264, so each frame is stored in a NAL (Network Abstraction Layer) unit. Each frame can be distinguished by a 1 byte NAL header (Table 2), which is used in combination with 4 bytes sequence ‘0x00 00 00 01’ [8].

Table 2. Types of NAL header.

Type	Value	Type	Value
SEI	0x06	IDR Picture	0x65
Access Unit Delimiter	0x09	SPS	0x67
Non-IDR Picture	0x61	PPS	0x68

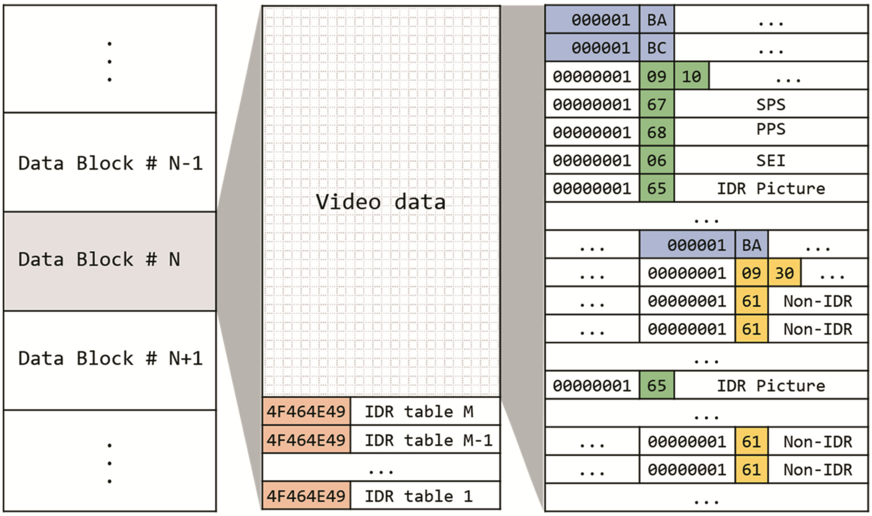


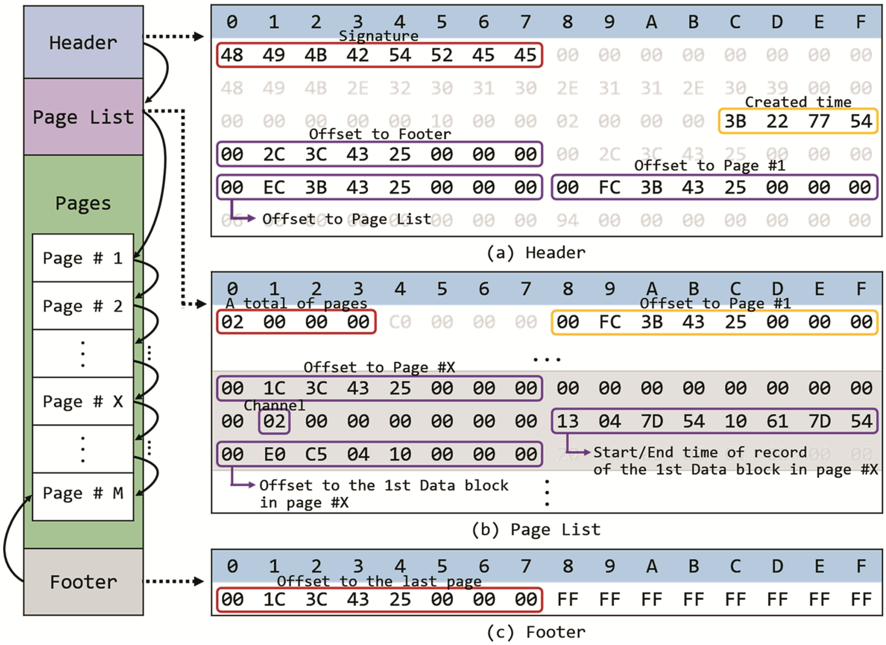
Fig. 4. Video data encoded to H.264.

In front of the NAL unit, the index of a picture is stored with one byte header ‘0xBA’ or ‘0xBC’, which is also used as a combination with three bytes sequence ‘0x00 00 01’. Due to this parts, noise occurs on the screen when playing with other players except the time when ‘player.exe’ is downloaded from the HIKVISION DVR.

The IDR table is created to store metadata, which contains index, channel, and timestamp of an IDR (Instantaneous Decoding Refresh) picture. Each record of the IDR table is recorded in the direction to decrease offset from the end of a data block. It starts with a signature ‘OFNI (0x4F 46 4E 49)’ and is fixed to 56 bytes for each record. Through the comparison of the IDR table’s timestamps as it stored in data block entries, it can be verified the time of the IDR pictures and channel. In general, a series of video data is being stored in the same data block when it is continued to record. If the DVR was being paused or the channel was being changed, other video data will be stored in one data block regardless of the condition. In this case, different video data can be extracted by comparing the timestamps of the IDR table with the recorded timestamps in data block entries.

2.4 HIKBTREE

The HIKBTREE contains the metadata of each video data in data blocks. The HIKBTREE is a fundamental area to discover the offset to a data block, existence of video data, and other additional information of recordings. Since it has a signature value 'HIKBTREE (0x48 49 4B 42 54 52 45 45)' as shown in Fig. 5(a), the term HIKBTREE will be used. The backup HIKBTREE is located after the former one.



```
0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
48 49 4B 42 54 52 45 45 00 00 00 00 00 00 00 00
48 49 4B 2E 32 30 31 30 2E 31 31 2E 30 39 00 00
00 00 00 00 00 10 00 00 02 00 00 00 3B 22 77 54
00 2C 3C 43 25 00 00 00 00 2C 3C 43 25 00 00 00
00 EC 3B 43 25 00 00 00 00 FC 3B 43 25 00 00 00
06                00 00 00 94 00 00 00 00 00 00 00
```

(a) Header

```
0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
02 00 00 00 C0 00 00 00 00 FC 3B 43 25 00 00 00
                           ...
00 1C 3C 43 25 00 00 00 00 00 00 00 00 00 00 00
00 02 00 00 00 00 00 00 13 04 7D 54 10 61 7D 54
00 E0 C5 04 10 00 00 00 20                00 00
                        :
```

(b) Page List

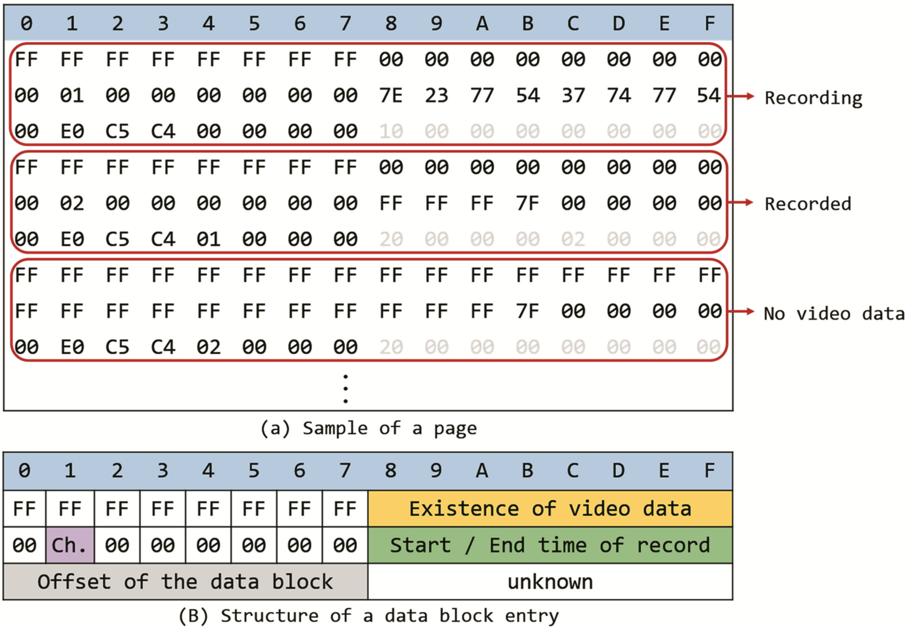
```
0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00 1C 3C 43 25 00 00 00 FF FF FF FF FF FF FF FF
```

(c) Footer

Fig. 5. An example of HIKBTREE.

The HIKBTREE consists of a number of sections including a header, page list, page number, and footer. The *Header* lists the created time, offset to page list, the first page, and footer. The *Page list* contains a total number of pages, offset to each page, which have information for the connection between video data and metadata, including storage location. Each *page* has numerous data block entries and the size of the page is 4 KB. Every page contains an offset to the next page. Using the offset to next page, one may move to the next page. But if the page is the last page, that field is written by '0xFF' hexadecimal values. The *Footer* is located in the last of the HIKBTREE and contains an offset to the last page.

Each page has numerous data block entries. The *Data block entry* has information about the existence of video data, channel, start/end time records, and offset to the data, as shown in Fig. 6. The field *Existence of video data* has '0x00' hexadecimal values if the data block becomes full of video data or '0xFF' hexadecimal values under the condition that the data block has no video data nor recording. The field *Ch. (channel)* identifies the number of connected cameras which are assigned by the DVR. For example, the value ‘0x01’ means that the video data had recorded from camera #1. The field *Start/End time* records identify the start and end of the UNIX time records in UTC only when the data block is full of video data, otherwise ‘0xFF FF FF 7F 00 00 00 00’. The field *Offset to the data block* identifies an offset to the data block which have the video data for user to play back.



```
0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
FF FF FF FF FF FF FF FF 00 00 00 00 00 00 00 00
00 01 00 00 00 00 00 00 7E 23 77 54 37 74 77 54   Recording
00 E0 C5 C4 00 00 00 00 10 00 00 00 00 00 00 00
FF FF FF FF FF FF FF FF 00 00 00 00 00 00 00 00
00 02 00 00 00 00 00 00 FF FF FF 7F 00 00 00 00   Recorded
00 E0 C5 C4 01 00 00 00 20 00 00 00 02 00 00 00
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF 7F 00 00 00 00   No video data
00 E0 C5 C4 02 00 00 00 20 00 00 00 00 00 00 00
                      ⋮
```

(a) Sample of a page

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
FF	FF	FF	FF	FF	FF	FF	FF	Existence of video data							
00	Ch.	00	00	00	00	00	00	Start / End time of record							
Offset of the data block								unknown							

(B) Structure of a data block entry

Fig. 6. An example of a page and Structure of a data block entry.

In general, different data block entries have different values of the offset to the data block. However, this hexadecimal values of a number of data block entries are sometimes the same. It means that the DVR had been paused or channel had been changed during recording. In this case, the video data can be extracted through the comparison of the IDR table timestamps with the timestamps stored in data block entries.

3 Experiments for DVR System Operation

This study demonstrates ways to access video data in addition to introducing digital forensic analyses after the experiments in initiating or overwriting the system. We conducted a test with the HIKVISION DS-7204HVI-SV (DVR), HIKVISION DS-2CE5582 N (camera), and Seagate Barracuda 7200.9 ST3160811AS SATA 3.5” 160 GB (HDD).

3.1 Video Data Access

The HIKVISION's DVR provides 'Playback' and 'Export' functions. The 'Playback' function allows to play video and the 'Export' function allows to download video file from the DVR to an external device. Using these functions, it is allowed to play video if a hard disk was collected as digital evidence. Once the hard disk becomes connected with DVR, its integrity becomes damaged. Thus, the access method of video data as digital evidence with integrity is necessary.

To access the video data upon user's requests, it is necessary to answer the questions, which of the data blocks store the video data and where the data blocks are. Thus, it is important to inspect the Master Sector prior to reading the offset to the HIKBTREE. After moving to the HIKBTREE, it is also important to verify and scan all pages including the headers. While reading the data block entries on the page, it is essential to find the data block entry that users prefer. If no data block entry is found, it means that there is no video data in the hard disk. Figure 7 shows the procedure for accessing the video data in a HIKVISION file system.

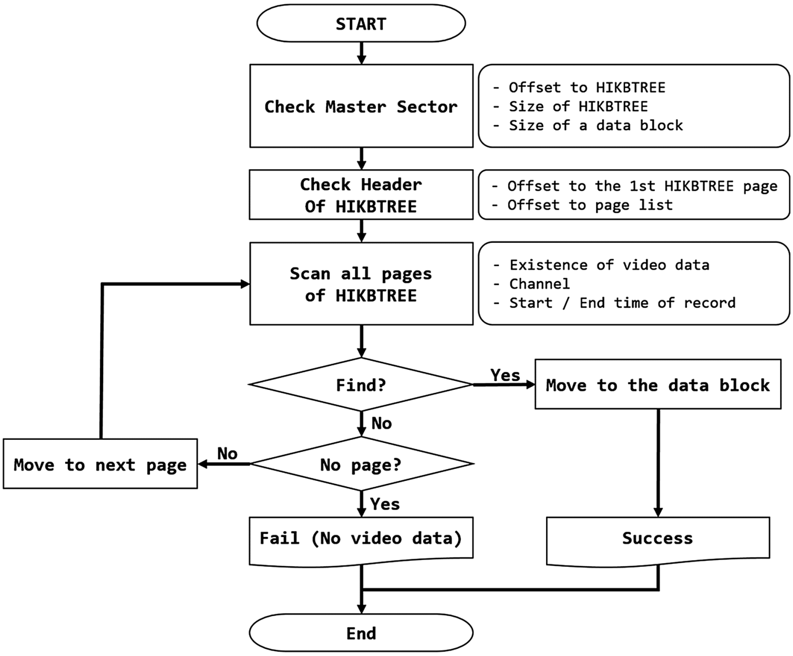


Fig. 7. Procedure for accessing to video data in the HIKVISION file system.

3.2 System Initialization

The HIKVISION DVRs have no delete function. System initialization is the only way to delete video data on a hard disk. When the hard disk is being initialized, the following will be changed: the time of system initialization in the Master Sector, restoration of the system logs to zero, and initialization of the HIKBTREE. After the system initialization, all video data in data blocks will remain. Regardless of system initialization, it is possible to extract video data by scanning all data blocks, for instance, by the carving technique.

In order to determine whether system initialization had been performed (Fig. 8), the value 'Time of system initialization' can be checked in the Master Sector and also be verified by reading the offset to video data and the HIKBTREE. Users can read recording times from the IDR table in the video data and 'Start/End time of record' of data block entries in the HIKBTREE. By comparing the 'Time of system initialization' values with the values of the IDR table and data block entries, users can determine whether system initialization had performed. If any time-reversal happens — the new time is created prior to the old time creation, thereby, users can conclude that this DVR was initialized at that time.

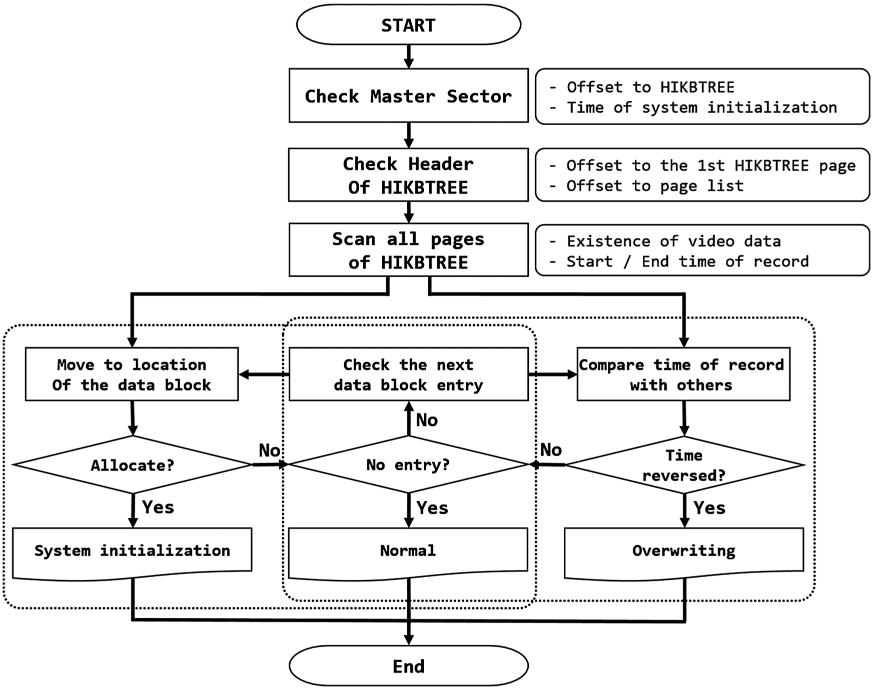
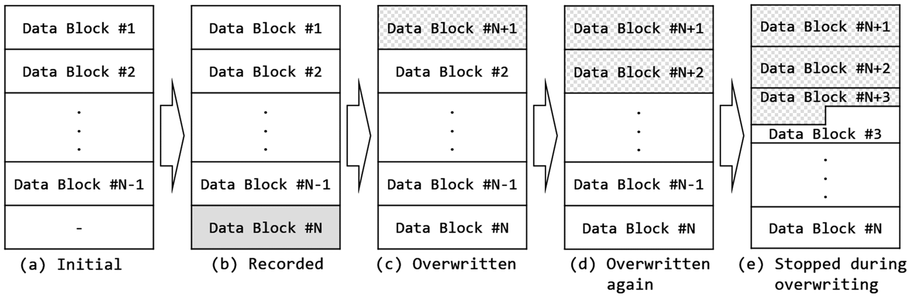

START
Check Master Sector
- Offset to HIKBTREE
- Time of system initialization
Check Header Of HIKBTREE
- Offset to the 1st HIKBTREE page
- Offset to page list
Scan all pages of HIKBTREE
- Existence of video data
- Start / End time of record
Move to location Of the data block
Check the next data block entry
Compare time of record with others
Allocate?
No entry?
Time reversed?
No
No
No
Yes
Yes
Yes
System initialization
Normal
Overwriting
End

Fig. 8. Procedure of detection for system initialization and overwriting.

3.3 Overwriting

When the data exceeds the capacity of the hard disk, old video data is replaced with new data under the procedure shown in Fig. 9. The old video data could be sometimes stored with the new data in a data block, if device was suddenly stopped or turned off. When the video data become overwritten, the following will change: the values of 'Channel' and 'Start/End time of record' in the data block entry. 'Channel' is updated as the present channel and the 'Start/End time of record' is changed to '0xFF FF FF 7F 00 00 00 00'.



(a) Initial (b) Recorded (c) Overwritten (d) Overwritten again (e) Stopped during overwriting

Fig. 9. Process of overwriting in a HIKVISION file system.

In order to determine whether overwriting had been performed (Fig. 8), users can read the recording time from the IDR table in the video database and the 'Start/End time of record' of data block entries from the HIKBTREE. If any recording time from the IDR tables in the video data predates the 'Start/End time of record' of data block entries, it can be understood that the hard disk had been full at least one time and has previously been overwritten.

4 Conclusion

Video recordings of video surveillance systems are the most useful evidence in forensic activities. However, it is difficult to analyze the hard disk using a HIKVISION file system, because there is no related research that are currently available. It is important to evaluate proprietary file systems, otherwise valuable digital evidence can lose its usefulness. Therefore, it is necessary to identify unknown file systems.

This study identifies the structure and mechanism of a HIKVISION file system which is not well-known. Using the result of this study, investigators can analyze hard disks from the HIKVISION products with integrity of the digital evidence. Furthermore, the procedure in the case analysis can be useful to counter anti-forensic activities, such as system initialization or data overwriting. This paper is conducted to provide useful analysis results regarding a HIKVISION file system for investigators in analyzing digital evidence relating video surveillance.

Acknowledgement. This research was supported by the Public Welfare & Safety Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, IT & Future Planning (2012M3A2A1051106)

References

1. Carrier, B.: File System Forensic Analysis, vol. 3, pp. 156–198. Addison-Wesley, Reading (2005)
2. Hellwig, C.: XFS: the big storage file system for Linux. Mag. USENIX SAG **34**(5), 10–18 (2009)

3. Hikvision Digital Technology Co. http://overseas.hikvision.com/en/
4. Yang, F., Li, R., Wu, C.: Basic principle and application of video recovery software for "Dahua" and "Hikvision" brand. In: SHS Web of Conferences, vol. 14, EDP Sciences (2015)
5. Poole, N.R., Zhou, Q., Abatis, P.: Analysis of CCTV digital video recorder hard disk storage system. Digit. Invest. **5**(3), 85–92 (2009)
6. Tobin, L., Shosha, A., Gladyshev, P.: Reverse engineering a CCTV system, a case study. Digit. Invest. **11**(3), 179–186 (2014)
7. Park, J., Lee, S.: Data fragment forensics for embedded DVR systems. Digit. Invest. **11**(3), 187–200 (2014)
8. ITU-T. H.264, advanced video coding for generic audiovisual services (2004). http://www.itu.int/