Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.26, NO.4, Aug. 2016 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2016.26.4.885

스마트폰으로 촬영된 동영상의 출처 식별에 대한 연구

김 현 승,† 최 종 현, 이 상 진‡
고려대학교 정보보호대학원

A Study on Identification of the Source of Videos Recorded by Smartphones

Hyeon-seung Kim,† Jong-hyun Choi, Sang-jin Lee‡
Center for Information Security Technologies, Korea university

요 약

스마트폰이 널리 보급됨에 따라 누구나 쉽게 사진과 동영상을 촬영하고 배포할 수 있는 시대가 되었다. 개인이 스마트폰으로 촬영한 동영상은 중요한 수사 단서나 증거로 활용되고 이때 동영상이 특정 스마트폰으로 촬영되었음을 입증해야 하는 상황이 발생한다. 이를 위해 기존 연구들에서 제시한 다양한 방식의 fingerprint 기법을 활용할 수 있다. 하지만 fingerprint 기법을 사용한 결과의 신빙성을 보강해야 하거나 그 기법을 활용할 수 없는 상황들이 존재한다. 따라서 fingerprint 기법의 사용 이전에 스마트폰 포렌식 조사가 선행되어야 하고, 동영상 파일의 메타데이터 정보를 정리한 데이터베이스를 구축할 필요가 있다. 본 논문에서는 동영상 촬영이 스마트폰에 남기는 아티팩트와 상기한 데이터베이스에 대해 설명하고자 한다.

ABSTRACT

As smartphones become more common, anybody can take pictures and record videos easily nowadays. Video files taken from smartphones can be used as important clues and evidence. While you analyze video files taken from smartphones, there are some occasions where you need to prove that a video file was recorded by a specific smartphone. To do this, you can utilize various fingerprint techniques mentioned in existing research. But you might face the situation where you have to strengthen the result of fingerprinting or fingerprint technique can't be used. Therefore forensic investigation of the smartphone must be done before fingerprinting and the database of metadata of video files should be established. The artifacts in a smartphone after video recording and the database mentioned above are discussed in this paper.

Keywords: martphone, video file, timestamp, log, avcC box

I. 서 론

오늘날 스마트폰이 널리 보급됨에 따라 누구나 쉽게 동영상을 촬영하고 배포할 수 있는 시대가 되었다. 그러나 이와 함께 기밀 유출, 아동 포르노, 저작권 침해 동영상 등의 불법 영상들이 늘어날 수 있는 환경이 조성되었다. 특히 성적 욕망을 불러일으키는 타인의 신체부위를 촬영하는 이른바 '몰래카메라' 범죄(「성폭력범죄의 처벌 등에 관한 특례법」 제14조 제1항 위반)가 기승을 부리고 있다[1]. 이러한 범죄들은 스마트폰에 남아있는 동영상 파일이 결정적인 증거가 된다.

이뿐만 아니라 동영상의 출처 식별은 해당 동영상의 변조 여부를 판단할 때 참고 기준으로 활용할 수 있다. 보통 동영상의 변조 여부를 판단할 때는 의심되는 변조 기법을 발견할 수 있는 수치적 분석이 수

Received(11. 03. 2015), Modified(1st: 03. 11. 2016, 2nd: 06. 17. 2016), Accepted(07. 06. 2016)
† 주저자, dongkid321@naver.com
‡ 교신저자, sangjin@korea.ac.kr(Corresponding author)

행된다. 그러나 만약 특정 기기에서 촬영되었다고 주장되는 동영상의 경우 그 기기로부터 결코 도출될 수 없는 메타데이터를 갖고 있는지를 조사하여 변조 여부를 1차적으로 판단할 수 있다.

II. 관련 연구

동영상 파일의 촬영 기기 식별에 관해서는 정지 이미지의 출처 기기 식별에 관한 기존 연구들이 제시한 원리를 활용할 수 있다. 기존 연구들은 촬영 기기가 현실 세계의 장면을 센서로 받아들이고 처리한 후 저장 매체에 이미지 파일로 저장하기까지의 과정에서 생성되는 기기 고유의 흔적들을 활용한다. 이를 'fingerprint 기법'이라 한다. fingerprint 기법에 관한 기존 연구들은 그 연구 대상에 따라 크게 세 가지로 분류할 수 있다.

첫째, 촬영 기기의 Color Filter Array(CFA) 특성을 활용한 연구이다[2][3]. CFA는 빛이 센서에 도달하기 전에 각 픽셀이 하나의 색만 받아들이도록 필터링하는 역할을 한다. 이 때문에 각 픽셀의 나머지 색 채널의 값을 추정(interpolation)하는 절차(demosaicing)를 수행해야 한다. 이때 사용된 CFA 패턴과 추정에 사용되는 알고리즘을 정지 이미지를 가지고 추측함으로써 촬영 기기를 식별하는데 활용할 수 있다. 그러나 이 방법은 동일 모델의 다른 기기를 식별하지 못하고 압축 등의 처리에 취약하다. Demosaicing으로 인해 생성된 픽셀들 간의 공간적 상관관계가 그러한 처리로 인해 사라지기 때문이다.

둘째, 촬영 기기의 PRNU(Photo Response Non-Uniformity)를 활용한 연구가 있다[4][5][6][7]. PRNU는 이미지 상에서 발견되는 센서 패턴 노이즈 중의 하나로 그것이 발생하는 주된 원인은 PNU(Pixel Non-Uniformity)이다. 이는 각 픽셀의 빛에 대한 민감도 차이를 의미하는 것으로 빛을 감지하여 전기적 신호로 변환하는 각 CCD/CMOS 센서들이 제조 과정에서 갖게되는 비동질성으로 인해 발생한다.

PRNU를 활용하는 방법은 다음과 같다. 일단 동일한 기기에서 촬영된 여러 이미지들의 PRNU를 추출하고 이것들을 평균화한 값을 해당 기기의 참조 패턴으로 지정한다. 이후 출처가 문제되는 이미지의 PRNU와 여러 참조 패턴들을 비교하여 가장 상관관계가 높은 참조 패턴을 갖고 있는 기기를 촬영 기기로 추정한다. 이 방법에는 다음과 같은 한계점들이 존재한다. 일단 이 방법 또한 이미지에 대한 특정한 처리(리사이징, 자르기, 회전 등)에 의해 신뢰도가 감소한다. 그리고 문제 이미지의 PRNU를 제거하여 출처 식별을 방해하거나 문제 이미지의 PRNU를 제거한 후 다른 기기의 이미지의 PRNU를 덧씌워 출처 식별에 오인을 일으킬 가능성 또한 존재한다[4].

또 PRNU를 이미지의 어떤 부분에서 추출하느냐에 따라 출처 식별 시 신뢰도 차이가 존재하고[8], [9]에서 제시된 바와 같이 촬영 설정(노출 시각, 초점 거리)에 따라 같은 기기라도 다른 PRNU가 도출될 수 있어 PRNU 추출 시 주의가 필요하다.

셋째, 촬영 기기 내 렌즈의 수차(lens aberration)를 활용한 연구이다. 이는 크게 구면 수차[10]를 이용한 것과 색 수차[11]를 이용한 것으로 나누어진다. 전자는 제조과정에서 발생하는 기기 내 렌즈 간의 차이로 인해 직선 이미지에 다르게 발생하는 왜곡을 기기의 fingerprint로 이용하는 것이다. 후자는 서로 다른 색의 빛이 초점면(센서)의 같은 곳에 도달하지 못하는 점을 fingerprint로 이용하는 것이다. 그러나 전자는 이미지에 직선 모양의 물체가 반드시 존재해야 하고 후자는 압축, 자르기 등의 이미지 처리에 취약하다는 단점이 있다. 또한 이미 렌즈 왜곡의 보정 패턴을 추측하여 출처 기기를 판단하는 [12]가 등장할 정도로 촬영 기기 자체적으로 이러한 수차들을 보정하는 기능이 정교화, 보편화되고 있기 때문에 더 이상 활용하기 힘든 방법이다.

이같이 기존 연구들은 정지 이미지에 대해 연구하였지만 [13]이 제시한 바와 같이 동영상 또한 각 프레임들을 대상으로 fingerprint를 추출한다면 그 촬영 기기를 식별할 수 있을 것이다.

그러나 이러한 fingerprint 기법은 그 실험 대상이 되는 데이터 셋에 따라 오차율이 변한다는 한계가 있고 동영상 내의 이러한 fingerprint들 또한 변조 가능성이 전혀 없다고 할 수 없다. 또 굳이 fingerprint 기법을 사용하지 않고 스마트폰의 포렌식 조사만으로 그 출처 식별을 신속하게 종결할 수도 있고 수사실무상 fingerprint 기법을 적용할 수 없는 상황들에 대한 대책도 필요하다.

이하에서는 스마트폰으로 동영상을 촬영할 경우 남는 아티팩트를 실험을 통하여 정리하였다. 그리고 fingerprint 기법을 사용할 수 없을 때 차선책으로 활용할 수 있는 동영상 메타데이터 데이터베이스를 제시하였다.

III. 동영상 파일 관련 아티팩트 분석

동영상 파일은 그 생성 원인에 따라 스마트폰 내 기본 저장 경로가 다르다. 그러나 경로 정보만으로 해당 동영상이 해당 기기에서 촬영되었음을 확실하게 입증할 수는 없으므로 다음과 같은 것들이 필요하다. 만약 동영상 파일이 삭제된 경우라면 해당 동영상의 한 장면이 저장되는 썸네일 파일의 존재 여부를 확인한 후, fingerprint 기법을 활용해야 한다.

3.1 동영상 파일 관련 데이터베이스

안드로이드 기반의 스마트폰으로 동영상을 촬영하면 "/data/data/com.android.providers.media/databases/external.db"의 video 테이블에 동영상 파일들의 이름, 경로, 용량, 재생시각, 해상도, 촬영 위치 좌표값, 생성 및 수정 시각 등의 정보가 담긴다. 특히, 생성 및 수정 시각의 정보는 후술할 파일시스템 상 동영상 파일의 생성 시각, 동영상 파일 내 생성 시각과 일치하는지 검사하여야 한다.

아이폰으로 동영상 촬영 후 그 백업파일을 살펴보면 "/Media/PhotoData/Photos.sqlite" 파일의 ZADDITIONALASSETATTRIBUTES 테이블에 동영상 파일의 생성 관련 정보가 담겨 있다. 이 테이블에는 동영상 생성 프로그램의 Bundle ID, 동영상 파일 이름, 생성 시각, 동영상 파일의 경로가 남는다. iphone5(iOS 8.2)에서 유명 앱으로 동영상을 실제 촬영하거나 다운로드하여 확인한 결과는 Table 1.과 같다.

Table 1. Apps which generated a video file

App name	ZCREATORBUNDLEID value
Default camera	null
SMS	com.apple.MobileSMS
Kakaotalk	com.iwilab.KakaoTalk
Nateon	com.nate.nateon
3rd party recording app	com.burbn.hyperlapse
Gmail	com.google.Gmail

3.2 동영상 파일의 생성 시각

동영상 파일의 생성 시각은 3.1에서 제시한 테이블 내의 시각 정보와 파일 시스템 단의 시각 정보, 동영상 파일의 내부에 기록된 시각 정보 이 3가지로 알 수 있는데 이 값들이 서로 유사한 경우 동영상이 해당 스마트폰으로 촬영되었음을 입증할 수 있다. 직접 촬영한 것이 아닌 다른 경로(웹 다운로드나 메신저 앱을 통한 전송 등)로 생성된 동영상은 일반적으로 동영상 파일 내부의 시각 값들이 나머지 두 값과 큰 차이를 보이거나 현실적으로 불가능한 시각 값이 기록되는 경우(시각 값이 0으로 채워진 경우 등)가 많기 때문이다.

스마트폰으로 촬영한 MP4, MOV, 3GP 포맷의 동영상 파일은 박스라고 하는 단위로 구성되어 있다[14]. 박스들은 각각의 명칭에 대응하는 동영상 파일에 관한 정보를 담고 있고 동영상 파일 내에서 이러한 박스들이 트리 구조로 구성되어 있다. 이 중에서 생성 시각 정보는 "mvhd" 박스, 오디오 트랙과 비디오 트랙 내 2개의 "tkhd" 박스와 2개의 "mdhd" 박스에 Creation Time, Modification Time(Mac Timestamp 형식)으로 총 10번 기록된다. 기록된 값에 우리나라 시간대에 맞춰 9시간(UTC+9)을 더해야 실제 촬영 시각과 일치한다.

특히 아이폰은 이와 별도로 ".day" 라는 박스에 위 10가지 시각과 동일한 시각 값을 따로 기록하고 "meta"라는 박스에는 시각 뿐만 아니라 촬영 아이폰의 모델명, OS 버전을 기록한다.

3.3 스마트폰의 운영체제 로그 정보

3.3.1 로그의 의의

스마트폰의 로그는 사용자 행위를 파악하는데 활용할 수 있다[15]. 만약 동영상 파일의 생성시각과 근접한 이전 시간대에 카메라 관련 기능을 사용한 로그가 남아있다면 해당 스마트폰으로 특정 동영상이 촬영됐음을 입증할 수 있다.

3.3.2 안드로이드 운영체제 로그

안드로이드에서는 로그가 커널 버퍼에 저장되는데 시스템의 서로 다른 부분에 대한 로깅을 위해 다음과 같은 4가지 버퍼가 존재한다. 안드로이드 로그는 루팅(rooting)을 하지 않아도 안드로이드 스마트폰에 기본으로 내장된 logcat 프로그램을 사용하여 각 버퍼의 내용을 출력할 수 있다.

Table 2. Logs after video recording(Android)

Log buffer	Logs
events	(1) com.sec.android.app.camera (2).receiver.VideoRecordingReceiver (3) TranscodeService
main	(1) startrecording (2) handleVideoRecordingStarted (3) onVideoRecordingStart (4) doPrepareVideoRecordingAsync (5) initializeRecorder (6) OMXCodec: Successfully allocated OMX node 'OMX.SEC.avc.enc'
system	com.sec.android.app.camera

본 연구에서는 갤럭시 3(SHV-E210K, Android 4.1.2)에서 ADB(Android Debug Bridge)를 활용하여 동영상 촬영 시 남는 로그를 확인하였다. Table 2.는 그 실험결과이다.

스마트폰의 종류에 따라 기록되는 로그들이 다르므로 로그 추출 후 'video, record, enc, codec' 등의 동영상 촬영과 관련된 키워드로 동영상 촬영 행위를 탐색해야 한다. 써드파티 동영상 촬영 앱을 활용하여 촬영한 경우 해당 앱의 이름과 저장된 동영상 파일명이 로그에 남는 경우도 발견하였다. 따라서 획득된 스마트폰에 써드파티 동영상 촬영 앱이 설치되어 있다면 그 앱의 패키지 이름을 로그 내에서 검색해볼 필요가 있다.

3.3.3 iOS 로그

아이폰은 운영체제 버전별로 동영상 촬영 관련 로그를 저장하는 방식이 다르다. 실험을 통해 확인한 로그는 Table 3.과 같다.

이처럼 동영상 파일 내부의 생성 시각이 상기한 로그 기록과 근접한 이전 시간대에 속해 있다면 신뢰도가 높은 정황증거로 활용할 수 있다. 다만, 로그는 그 용량 한계로 인해 휘발성이 높은 아티팩트라는 점에서 현행범 사건이 아닌 경우 활용하기 힘들고 수사관의 신속한 수집작업이 필요하다는 단점이 있다.

이처럼 스마트폰에서 동영상 파일이 삭제되지 않고 동영상 파일에 별다른 조작이 가해지지 않은 상황에서는 상기한 포렌식 조사로 조기에 동영상 출처 식별을 종결할 수 있다.

IV. 동영상 메타데이터의 의의

4.1 문제 상황

상기한 스마트폰 포렌식 조사나 기존 연구에서 제시하는 fingerprint 기법을 수행한다면 동영상을 촬

Table 3. Logs after video recording(iOS)

iphone	Logs
iphone5 (MF333KH/A) iOS 8.2	- Database path : /private/var/mobile/Library/Logs/CurrentPowerlog.PLSQL - Tables with logs(record time in Unix timestamp format in each table) (1) PLCameraAgent_EventForward_BackCamera (2) PLCameraAgent_EventForward_FrontCamera (3) PLAudioAgent_EventPoint_SpeakerAmp (4) PLAudioAgent_EventPoint_AudioApp

iPad Mini
(MD543KH/A)
iOS 7.1.2
	- Log file path : /var/mobile/Library/Logs/CurrentPowerlog.powerlog
	- logs(record time in Unix timestamp format prior to each log)
	(1) 〔Application〕 id=com.apple.camera; pid=4146.00;
	mode=Foreground Running; reason=〈unknown〉;
	display_name=Camera; executable=Camera; version=1.0.0;
	(2) 〔CoreLocation Client〕 id=com.apple.camera; location=active;
	(3) 〔Audio〕 active=YES; route=Speaker;
	volume=25.0; output_category=Audio/Video; muted=NO;

영한 스마트폰을 식별할 수 있다. 그러나 수사실무상 다음과 같은 경우들에는 이 2가지 방법으로 스마트폰 소유자의 범죄 행위를 입증하는 데 공백이 생긴다.

① 스마트폰이 없고 해당 스마트폰으로 촬영되었다고 확신할 다른 사진, 동영상도 없는 경우

(ex. 용의자가 현재는 갖고 있지 않은, 과거에 사용하던 스마트폰으로 촬영한 것으로 의심되는 동영상이 있는 경우)

② 스마트폰은 있으나 해당 스마트폰으로 촬영되었다고 확신할 다른 사진, 동영상이 없고 스마트폰 자체가 하나의 증거물이라 fingerprint를 도출하기 위해 더 이상 사진이나 동영상 등을 찍는 것이 증거의 무결성을 침해한다고 판단되는 경우

③ 스마트폰은 있으나 해당 스마트폰으로 촬영되었다고 확신할 다른 사진, 동영상이 없고 렌즈 부분이 손상되어 fingerprint를 도출하기 위해 더 이상 사진이나 동영상 등을 찍을 수 없는 경우

④ 용의자가 다수인 상황에서 모두의 스마트폰에 fingerprint 기법을 적용하기 곤란하여 용의자를 1차적으로 선별할 필요가 있는 경우

(ex. 이메일로 전송된 기밀 유출 동영상이 적발되어 사내 직원들을 조사해야 하는 경우)

이런 상황에서는 동영상 파일의 메타데이터로 그 촬영 스마트폰의 종류를 추론하는 것이 차선의 해결책이 될 수밖에 없다. 그리고 메타데이터는 fingerprint 기법의 활용 결과, 촬영 기기를 식별할 만큼의 높은 상관 관계 지수가 도출되지 않을 때 그 결과의 신빙성을 보강하기 위해, 그리고 동영상 변조 여부의 1차적 판단 기준으로 활용하기 위해 필요하다. 후자는 특히 [16]에서 제시된 바와 같이 동영상 편집 툴이 동영상 파일 내에 특유의 박스를 추가하는 경우들이 있기 때문이다.

물론 메타데이터를 기준으로 한 출처 식별은 동일 모델 내의 다른 기기를 식별하지 못하고 기존 연구들에 제시된 fingerprint들에 비해 메타데이터는 그 변조가 용이한 것이 사실이다. 그러나 상기한 것처럼 메타데이터는 fingerprint 기법만으로는 온전히 담당할 수 없는 수사 실무상의 문제를 해결하는데 도움이 될 가능성이 있기 때문에 동영상의 출처 식별 시 배제할 수 없고 동영상 파일에 대한 변조가 시도되지 않는 일반적인 사건들의 경우에는 신속한 사건 해결에 기여할 것으로 기대된다.

4.2 메타데이터 데이터베이스의 구축

이와 같은 필요성에 의해 스마트폰으로 촬영한 동영상의 메타데이터 데이터베이스를 구축해야 한다. 동영상의 경우 사진에 비해 그 구조가 더욱 복잡하고 그 구성요소인 각 박스의 존재나 순서가 선택적인 사항이 많다는 점에서 이러한 데이터베이스에 의해 식별하기가 보다 효과적이다.

데이터베이스를 구축하기 위해 각 스마트폰의 종류에 따라 촬영되는 동영상들 자체를 데이터베이스에 저장하여야 한다. 스마트폰의 종류는 그 모델명, OS 버전, 빌드번호를 조사해야만 그 유일성을 확보할 수 있다. 촬영되는 동영상 자체를 저장하는 이유는 향후 특정 상황에서 메타데이터의 어떤 부분이 식별성을 가져다줄 것인지 미리 판단하기 힘들기 때문에 다시 동영상 파일 내부를 확인할 수 있도록 하기 위함이다. 또 이 데이터베이스가 향후 fingerprint 기법을 대상으로 하는 연구에서 촬영 기기의 종류를 식별하기 위한 실험을 할 때 그 촬영 출처가 검증된 방대한 데이터 셋으로 활용될 수 있도록 하기 위함이다.

물론 사건 발생 시 수사관이 문제 동영상의 출처를 식별하기 위해 데이터베이스에서 검색을 할 때 활용할 수 있는 기준이 되는 메타데이터들은 미리 파악하여 데이터베이스에 저장하여야 한다. 현재까지 발

Table 4. Difference between Galaxy smartphones

Model name	Galaxy 2(SHW-M250K)	Galaxy 3(SHV-E210K)	Galaxy 4(SHV-E330L)
OS version	4.0.3	4.1.2	4.2.2
Configuration of boxes	ftyp mdat moov mvhd trak tkhd	ftyp mdat moov mvhd **udta** smrd	ftyp mdat moov mvhd **udta** **SDLN**

견한 기준들은 다음과 같다.

4.2.1 박스의 구성

어떤 박스들이 있는지를 보면 Samsung 사의 스마트폰, LG 사의 스마트폰, Apple 사의 아이폰 등을 구분할 수 있다.

한 제조사의 스마트폰들도 박스의 구성으로 어느 정도 구별할 수 있다. Table 4.를 보면 Samsung 사의 서로 다른 스마트폰을 박스의 구성만으로 구별할 수 있음을 보여준다.

그러나 같은 스마트폰으로 촬영한 영상이더라도 박스의 구성이 늘 같은 것은 아니다. 예를 들어 ".xyz" 라는 이름의 박스는 GPS 기능을 활성화하고 동영상을 촬영했을 시에 추가되는 박스로 동영상 촬영 지점의 좌표 값을 갖고 있다.

4.2.2 박스 내의 정보

박스의 구성만으로 스마트폰의 종류를 식별할 수는 없다. 여기에 특정 박스의 특정 값을 조합하여 판단해야 한다. 실험을 통해 찾은 값들은 다음과 같다.

첫째, ftyp 박스의 각 필드 값을 활용할 수 있다. ftyp 박스의 Major brand 필드와 Compatible brand 필드는 각각 해당 동영상 파일의 주 형식과 호환 가능한 형식을 나타낸다. 같은 박스의 구성을 보이더라도 이 값들은 다른 경우가 존재한다.

둘째, avcC 박스 내의 특정 필드 값을 기준으로 할 수 있다. avcC 박스는 스마트폰 동영상 촬영 시 H.264(MPEG-4 파트 10 AVC) 표준으로 인코딩했을 때 활용된 옵션 값들을 담고 있는 박스로 동영상을 디코딩하기 위해 고려해야 하는 부분이다. 이 박스에는 동영상 디코딩을 위한 옵션값들을 담고 있는 SPS(sequence parameter set)와 PPS(picture parameter set) NAL Unit이 담겨 있다.

NAL(Network Abstraction Layer) Unit이란 H.264 표준으로 인코딩된 비디오의 비트스트림을 네트워크 상에서 전송하기 용이한 형태로 구성할 때 쓰이는 단위로 이것이 실제로 동영상 디코더에 입력되는 단위이다. H.264가 오늘날 스마트폰 동영상의 비디오 스트림 인코딩에 보편적으로 쓰이고 SPS, PPS의 값이 여러 옵션들로 인해 다양한 값을 가질 수 있기 때문에 출처식별력이 강한 기준이다.

4.2.3 동영상 촬영 앱 사용 시

기본 카메라 앱이 아닌 별도의 동영상 촬영 앱을 활용하는 경우 사용한 앱이 무엇인지, 그 앱의 버전은 무엇인지, 어떤 스마트폰에 다운로드하여 사용하였는지, 동영상 촬영 설정을 어떻게 하고 촬영하였는지(컬러 필터 사용 등) 등에 따라 상기한 식별 기준들을 포함한 다양한 메타데이터들이 바뀐다. 특히 어떤 스마트폰에 다운로드하여 사용하느냐에 따라 메타데이터가 다른 것은 해당 앱이 그것이 설치된 스마트폰의 소프트웨어, 하드웨어의 기능에 종속되어 동작하기 때문이다.

실험 결과 아이폰에서와는 달리 안드로이드 스마트폰에서는 Fig. 1과 같이 해당 앱이 사용한 것으로 추정되는 libavformat 라이브러리의 버전이 동영상 파일의 "meta" 박스에 남는 경우들이 발생함을 확인하였다. libavformat은 비디오 재생, 녹화, 변환 등에 관한 오픈 소스 프로젝트인 FFmpeg에 포함된 라이브러리로 오디오와 비디오 데이터를 분리하거나 합쳐주는 역할(Muxer/Demuxer)을 한다. 앞으로 특정 촬영 앱마다 고유하게 남기는 메타데이터가 추가적으로 발견되고 문제의 스마트폰에 해당 앱이 설치되어 있는 것이 밝혀진다면 이는 출처 식별 시 강력한 정황 증거가 될 것이다.

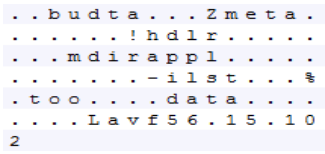

Fig. 1. libavformat library version in 'meta' box

4.2.4 데이터베이스 설계

데이터베이스 설계는 다음과 같이 한다. 먼저 사용자가 카메라를 찍는 모든 상황을 포함할 수 있도록 다음과 같은 컬럼들이 있는 테이블 1을 구축한다.

- Model : 스마트폰의 모델명
- OS version : 스마트폰의 OS 버전
- Build number : 스마트폰의 빌드 번호
- App name(version) : 사용한 앱 이름(기본 카메라일 경우 default, 구글 마켓/앱스토어 앱일 경우 그 이름과 버전을 병기)
- app effect : 앱에서 설정한 효과

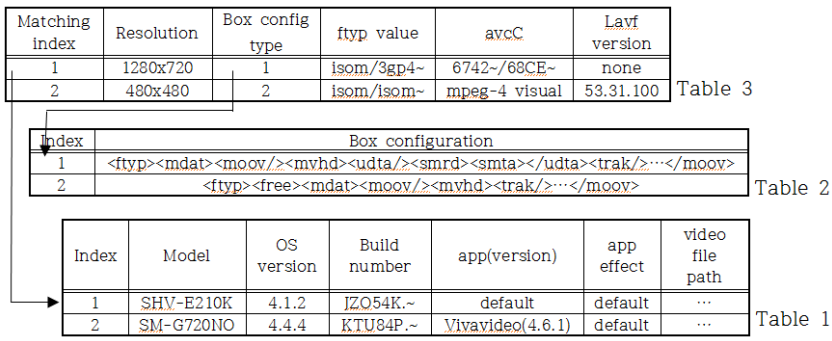

Fig. 2. Relationship between three tables related to metadata of video files

(없으면 default)
- video file path : 서버 내 동영상 파일 경로

그리고 박스의 구성 타입을 정리한 테이블 2를 만든다. 이때 Box configuration 컬럼을 두어 여기에 해당 타입의 박스 구성을 XML 형식으로 저장한다. 독립적인 박스는 〈박스명〉, 하위 박스들을 포함한 박스는 〈박스명/〉~〈/박스명〉 형식으로 저장한다.

세 번째로 지금까지 정리된 출처 식별성이 있는 메타데이터의 컬럼들이 있는 테이블 3을 만든다.

- Resolution : 동영상의 해상도
- Box config type : 테이블 2 내 해당 인덱스
- ftyp value : ftyp 박스 내 major/compatible brand 값
- avcC(SPS/PPS NAL Unit) : avcC 박스 내의 SPS/PPS 값(H.264 이외의 표준으로 인코딩된 경우 해당 표준명을 대신 기록)
- libavformat version : libavformat 버전
- matching index : 테이블 1 내 당시 촬영환경에 해당하는 인덱스

4.3 메타데이터 방식의 한계

어떤 종류의 스마트폰으로 촬영된 동영상인지를 조사할 때 메타데이터가 갖는 식별력을 확인하고자 직접 다양한 스마트폰으로 동영상을 촬영해 데이터베이스를 구축하였다. 아이폰의 경우 메타데이터에 기기 종류가 기록되어 당연히 스마트폰의 종류를 알 수 있기 때문에 일단 제외하였다.

총 39종의 안드로이드 스마트폰의 기본 카메라로, 한 스마트폰에서 중복되는 해상도가 없도록 촬영하여 총 107개의 동영상이 있는 데이터베이스를 구축한 후 서로 다른 종류의 스마트폰이 동일한 메타데이터를 갖는 경우 즉, 동영상과 스마트폰 종류의 1대1 대응이 불가능한 경우를 Table 5,에 정리하였다. 이를 통해 메타데이터만으로 스마트폰 종류를 1대1 대응할 수 없는 경우도 발생함을 알 수 있다.

V. 결 론

이때까지 스마트폰 동영상의 출처를 식별하기 위해 활용할 수 있는 방안들에 대해 살펴보았다. 먼저 fingerprint 기법을 활용하기 전에 스마트폰 포렌식을 통해 동영상이 해당 기기에서 촬영되었음을 신속히 입증하는 방안을 정리하였고 fingerprint 기법을 활용한 결과의 신빙성을 보강하거나 fingerprint 기법만으로 해결할 수 없는 문제 발생 시 차선책으로 활용할 수 있는 동영상 메타데이터 데이터베이스를 제안하였다.

Table 5,를 보면 동영상의 메타데이터만으로 스마트폰 종류를 1대1 대응할 수 없는 경우가 발생함을 알 수 있다. 그럼에도 불구하고 현재 수사실무상 스마트폰을 증거로 입수한 경우 무결성을 보장하기 위해 그 스마트폰으로 새롭게 동영상을 촬영할 수 없는 즉, fingerprint 기법을 활용할 수 없는 상황에서 보다 현실적인, 메타데이터를 통한 간접적인 방식을 활용할 필요가 있다. 1대1 대응이 불가능한 상황이라도 증거인 스마트폰의 종류가 메타데이터를 통한 조회 결과에 포함되어 있는 경우라면 증명에 도움이 될 것이다. 또한 "4.1 문제상황"에서 제시한 ④의 상황에는 메타데이터 방식만이 용의자 추출을 위해 사용할 수 있는 수단이다.

현재 스마트폰으로 촬영한 동영상을 업로드할 수

Table 5. Groups of smartphone types which generated the same video metadata

video resolution	model name	OS version	build number
1920X1080	SHV-E330L	5.0.1	LRX22C.E330LKLUDOL4
	SHV-E330S	4.4.2	KOT49H.E330SKSUCNJ1
1280X720	SHV-E210K	4.1.2	JZO54K.E210KKKJMK1
	SHV-E270S	4.1.2	JZO54K.E270SKSUAOH1
1280X720	SHV-E330S	4.4.2	KOT49H.E330SKSUCNJ1
	SM-G720N0	4.4.4	KTU84P.G720N0U1APC1
1280X720	LG-F300K	4.4.2	KOT49I.F300K20j
	LG-F410S	4.4.2	KOT49I.F410S10i
	LG-F320K	4.4.2	KOT49I.F320K22h
1920X1080	LG-F300K	4.4.2	KOT49I.F300K20j
	LG-F320K	4.4.2	KOT49I.F320K22h
1920X1080	SM-G906K	5.0.1	LRX22C.G906KKTU1BOL1
	SM-A700S	5.0.2	LRX22G.A700SKSU1BPB4
1920X1080	SM-G935K	6.0.1	MMB29K.G935KKKU1APDA
	SM-G920L	6.0.1	MMB29K.G920LKLU3DPD2
	SM-G930S	6.0.1	MMB29K.G930SKSU1APDA
640X480	SM-G906K	5.0.1	LRX22C.G906KKTU1BOL1
	SM-A700S	5.0.2	LRX22G.A700SKSU1BPB4
1920X1080	SM-N920K	6.0.1	MMB29K.N920KKKU2BPB2
	SM-N920S	6.0.1	MMB29K.N920SKSU2BPE3
	SM-G920S	6.0.1	MMB29K.G920SKSU3DPAC
1920X1080	SM-G935K	6.0.1	MMB29K.G935KKKU1APDA
	SM-N920S	6.0.1	MMB29K.N920SKSU2BPE3

있도록 데이터베이스 서버 구축, 안드로이드 스마트폰용 앱과 아이폰용 앱 제작이 완료된 상태이다. 앱 배포를 통해 동영상 수집 작업을 계속 진행할 것이고 좀더 많은 데이터를 축적한 후, 동영상을 업로드하면 그 메타데이터를 기준으로 해당하는 촬영 환경들을 보여주는 조회용 사이트를 개발할 예정이다. 또 동영상 처리 기술이 발전함에 따라 새로운 포맷과 코덱이 적용되는 스마트폰 동영상들이 등장하거나 특정 동영상 촬영 앱들이 남기는 또다른 공통 메타데이터가 있다면 그 메타데이터들도 컬럼으로 추가할 것이다. 이러한 데이터베이스는 그 규모가 방대해질수록 수사 실무 상 효용성이 커질 것이고 이는 포렌식 수사관에게 하나의 자산이 될 것이다.

References

[1] "Hidden camera crime in the Waterpark", http://daily.hankooki.com/lpage/society/201508/dh20150829095323137790.htm

[2] Ashwin Swaminathan, Min Wu and K. J. Ray Liu, "Non-intrusive forensic analysis of visual sensors using output images," Information Forensics and Security, IEEE Transactions, vol. 2, no. 1, pp. 91-106, Mar. 2007.

[3] Sevinc Bayrama, Husrev T. Sencarb and Nasir Memonb, "Classification of digital camera-models based on demosaicing artifacts," Digital Investigation, vol. 5, no. 1, pp. 49-59, Sep. 2008.

[4] Jan Lukas, Jessica Fridrich, and Miroslav Goljan, "Determining digital image origin using sensor imperfections," Proceedings of the SPIE, pp. 249-260, Mar. 2005.
[5] Mo Chen, Jessica Fridrich, Miroslav Goljan, and Jan Lukas, "Determining image origin and integrity using sensor noise," Information Forensics and Security, IEEE Transactions, vol. 3, no. 1, pp. 74-90, Mar. 2008.
[6] Chang-Tsun Li, "Source camera identification using enhanced sensor pattern noise," Information Forensics and Security, IEEE Transactions, vol. 5, no. 2, pp. 280-287, Mar. 2010.
[7] J.R. Corripio, A.L. Sandoval Orozco, L.J. Garcia Villalba, Julio Hernandez-Castro, Stuart Jason Gibson, "Source smartphone identification using sensor pattern noise and wavelet transform," The 5th International Conference on Imaging for Crime Detection and Prevention, pp. 1.16(1), Dec. 2013.
[8] Chang-Tsun Li, Riccardo Satta, "Empirical investigation into the correlation between vignetting effect and the quality of sensor pattern noise," Computer Vision, IET, vol. 6, no. 6, pp. 560-566, Nov. 2012.
[9] Thomas Gloe, Stefan Pfennig, Matthias Kirchner, "Unexpected Artefacts in PRNU-Based Camera Identification: A 'Dresden Image Database' Case-Study," Conference: 14th ACM Multimedia and Security Workshop, pp. 109-114, Sep. 2012.
[10] Kai San Choi, Edmund Y. Lam, and Kenneth K. Y. Wong, "Automatic source camera identification using the intrinsic lens radial distortion," Opt Express, vol. 14, no. 24, pp. 11551-11565, Nov. 2006.
[11] Lanh Tran Van, Emmanuel, S, Kankanhalli, M.S., "Identifying Source Cell Phone using Chromatic Aberration," Multimedia and Expo, 2007 IEEE International Conference, pp. 883-886, Jul. 2007.
[12] Min-gu Hwang, Dong-min Kim, Dong-hwan Har, "Digital Camera Identification Based on Interpolation Pattern Used Lens Distortion Correction," Korea Society for Internet Information, 13(3), pp. 49-59, Jun. 2012.
[13] Mo Chen, Jessica Fridrich, Miroslav Goljan, Jan Lukas, "Source Digital Camcorder Identification Using Sensor Photo Response Non-Uniformity," Proc. SPIE 6505, Security, Steganography, and Watermarking of Multimedia Contents IX, 65051G, Forensics Ⅲ, pp. 65051G, Feb. 2007.
[14] ISO/IEC JTC 1/SC 29 Coding of audio, picture, multimedia and hypermedia information, "ISO Base Media File Format," ISO/IEC 14496-12:2004, Feb. 2004.
[15] Ilyoung Hong, Sangjin Lee, "Research on Efficient Live Evidence Analysis System Based on User Activity Using Android Logging System," Korea Institute of Information Security and Cryptology, 22(1), pp. 67-80, Feb. 2012.
[16] Thomas Gloe, Andre Fischer, Matthias Kirchner, "Forensic analysis of video file formats," Digital Investigation, vol. 11, supplement 1, pp. S68 - S76, May. 2014.

〈저자소개〉

김 현 승 (Hyeon-seung Kim) 정회원
2015년 2월: 경찰대학 법학과 학사
2015년 3월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 디지털 포렌식

최 종 현 (Jong-Hyun Choi) 정회원
2012년 2월: 경희대학교 전자정보대학 컴퓨터공학과 공학사
2014년 8월: 고려대학교 정보보호대학원 정보보호학과 석사
2014년 9월~현재: 고려대학교 정보보호대학원 정보보호학과 박사과정
〈관심분야〉 디지털 포렌식

이 상 진 (Sang-jin Lee) 종신회원
1987년 2월: 고려대학교 수학과 학사
1989년 2월: 고려대학교 수학과 석사
1994년 8월: 고려대학교 수학과 박사
1989년 10월~1999년 2월: ETRI 선임연구원
1999년 3월~2001년 8월: 고려대학교자연과학대학조교수
2001년 9월~현재: 고려대학교정보보호대학원교수
2008년 3월~현재: 고려대학교디지털포렌식연구센터센터장
〈관심분야〉 디지털포렌식, 심층암호, 해쉬함수