Received September 23, 2019, accepted October 9, 2019, date of publication October 28, 2019, date of current version November 8, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2949839

Digital Video Source Acquisition Forgery Technique Based on Pattern Sensor Noise Extraction

ANA LUCILA SANDOVAL OROZCO, CARLOS QUINTO HUAMÁN, JENNY ALEXANDRA CIFUENTES QUINTERO, AND LUIS JAVIER GARCÍA VILLALBA

Group of Analysis, Security and Systems (GASS), Department of Software Engineering and Artificial Intelligence (DISIA), Faculty of Computer Science and Engineering, Universidad Complutense de Madrid (UCM), 28040 Madrid, Spain

Corresponding author: Luis Javier García Villalba (javiergv@fdi.ucm.es)

This work was supported by the European Union's Horizon 2020 Research and Innovation Programme under Grant 700326.

ABSTRACT Digital camera of a smartphone is a component frequently used by people to capture a large number of videos, which can illustrate situations that compromise their presumption of innocence in court cases. Usually, these videos circulate on the Internet, prone to intentional manipulation to prosecute one person or exempt another. In this sense, digital videos are a matter of great importance for forensic science, because they can be useful to verify the authenticity of such evidence in judicial processes, helping to make sound decisions. However, it is possible that criminals or attackers know weaknesses of forensic techniques and use anti-forensic techniques to manipulate videos without leaving any trace of the procedure performed. Forensic science confronts anti-forensic techniques, analyzes them rigorously and applies anti-measures in the development of techniques to detect anti-forensic operations. In this paper, anti-forensic techniques are proposed to perform the source anonymization and forgery in MP4 videos.

INDEX TERMS Anti-forensic analysis, digital video, forensic analysis, PRNU, sensor noise, source identification, video anonymization, video forgery, video metadata, wavelet transform.

I. INTRODUCTION

Mobile devices have become a fundamental element in the daily life of people because they allow the solution to immediate needs in a fast and easy way. Among the key factors that demonstrate the manner in which these devices have changed our lives are: almost universal access to the Internet, the use of applications that offer an infinity of services in a single device, the generation of a large number of multimedia files, such as images and videos, which have caused the reduction of users' communication through text and the increase of the exchange of this kind of files, the trend towards greater preference of the mobile device over traditional television to watch films or to scan the latest news, the device use as a computer to get work done or even to purchase products without being tied to a computer terminal and finally, it is a perfect element to use in spare time. According to the report *The Mobile Economy 2019* [1], in 2025 there will be 5.800 million subscriptions to mobile services, 71% of the world population for that year, compared to 5.100 million, only 60% of the world's population by the end of 2018. Likewise, the number of registered users to mobile internet will increase from 3,600 million to 5,500 million during 2018-2025. Similarly, the global consumption estimation of mobile data in 2024 will reach 24 GB per month in comparison to just 5.3 GB in 2018. According to the report developed by Cisco Systems [2], it is estimated that smartphones, laptops and tablets will generate 98% of global mobile data traffic in 2020, which will represent 366.8 exabytes per year, compared to 2015, where 89% represented 44,2 exabytes. In particular, the generation of mobile videos will be the element with the highest demand compared to other applications, increasing the use of devices with 4G or 5G connections. Therefore, it is estimated that in 2020, 75% of mobile network traffic will be due to the transference of high resolution videos. This means that 80 billion images (28 images per day) and 7 billion videos

The associate editor coordinating the review of this manuscript and approving it for publication was Tai-hoon Kim.

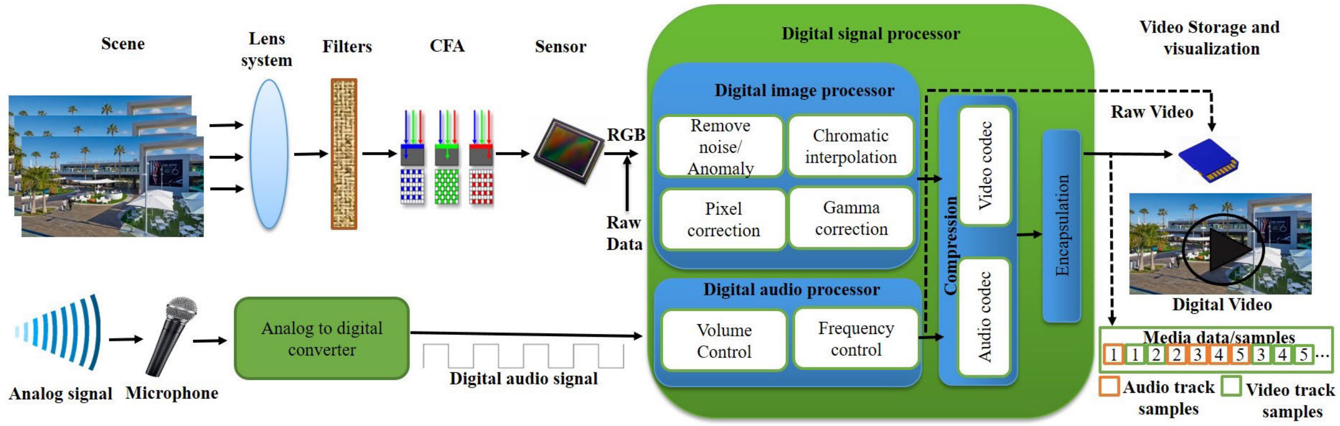

FIGURE 1. Generation process of a digital video.

(more than 2.5 videos per day) per inhabitant of the world will be generated in a single year.

Consequently, mobile devices are used to produce a large number of photographs and videos, which can later be stored on other devices, posted on social networks, sent by email, etc. This means that, from a forensic point of view, criminal actions such as theft of confidential information, creation and distribution of Child Sexual Abuse Material, kidnapping, espionage, etc., are wide-spread everyday uses of this technology.

In this sense, anonymization and forgery detection of video source are necessary procedures to determine responsibilities and present evidence that will later be used as legal evidence. However, these forensic techniques are counteracted by alternative strategies called anti-forensic techniques, responsible for alterations to images and videos to obstruct the respective process of forensic analysis. Specifically, they try to hide, eliminate or falsify digital evidence. Based on this idea, it is necessary to develop forensic techniques capable to identify these methods and face the obstacles and drawbacks derived from their use [3].

Considering the case of designing a highly secure lock, which can not be opened without its original key, an alternative solution for opening the lock without the key, must be contemplated. Thus, a lock can be designed to prevent attempted access without the use of the applicable key. Based on the same idea, the acquired knowledge about anti-forensic techniques that anonymize or forger the identity of a multimedia recording device, lead to the development of forensic techniques to identify the most robust strategies to avoid this type of attack. This study presents anti-forensic techniques aimed at anonymizing and forgering the source of a video, focusing specifically on videos generated by mobile devices.

The paper is structured as follows. Section II briefly describes the generation process of a digital video. Section III presents the state of the art of forensic analysis techniques and anti-forensic techniques for videos recorded by mobile devices. Section IV elaborates on the proposed anti-forensic techniques. In Section V, the experimental results are explained and discussed. Finally, in Section VI conclusions of this study are drawn.

II. GENERATION OF A DIGITAL VIDEO

In the generation process of digital video or *pipeline* [4], [5], image, audio and text features are combined. The structure of the *pipeline* is similar between manufacturers and type of devices, differing in the quality of the camera and some additional features and details of each manufacturer. A digital camera consists of a lens system, a group of filters, a *Color Filter Array* (CFA), an image sensor (*Charge Coupled Device* (CCD)) [6], though currently it is common that every mobile device has a sensor *Complementary Metal Oxide Semiconductor* (CMOS). In addition, it is composed by a *Digital Signal Processor* (DSP) which in turn contains a *Digital Image Processor* (DIP), a digital audio processor, a microphone and an analog/digital converter. This process is shown schematically in Figure 1.

In order to generate a standard video, two actions are usually conducted in parallel: The processing of images or video sequences and the audio processing. Image or video sequence processing begins when the lens system captures the light of the scene by controlling exposure, focus and image stabilization. Then, the light enters the camera by the lens system and goes through several filters that improve the visual quality of the image (infrared and anti-aliasing). Further, the light passes the image sensor, through the CFA, to capture the color information. To keep costs low, some mobile devices have a single monochrome image sensor and the CFA precedes it to generate a color image. However, manufacturers are currently incorporating one or more sensors to capture only black and white colors and to manage the optical zoom, the wide-angled shot and the depth of the scene. This signal is converted into a digital signal, which is transmitted to the DSP and specifically to the DIP. The digital signal, generated by one or more sensors, is captured by the image processor, where it is subject to different treatments (camera processes) to stabilize the signal and correct alterations (*artifacts*) such as the elimination of noise and other existing anomalies [7], [8].

Subsequently, the stabilized signal is processed in the compression stage by means of a video codec after synchronization. There are different video compression standards, but currently the most used by mobile devices are: 1) H264/AVC or MPEG-4 Part 10 [9] and 2) H265/HEVC or MPEG-H Part 2 [10], both developed by the ITU-T and ISO/IEC.

Finally, the encapsulation process is performed in a multimedia container and its result is stored in an internal storage medium of the device. Nowadays, the most used multimedia containers are the MP4, which is part of the standard *Moving Picture Experts Group* (MPEG)-4 part 14. Usually, it is used by manufacturers that introduce the Android operating system in mobile devices and the MOV container of the standard *QuickTime*, developed by *Apple* [11] that has iOS operating system in its devices.

The audio processing starts when the sound signal, transmitted by air, is captured through the microphone, which works as an electro-acoustic sensor. The microphone transforms the sound waves into an electrical signal in order to increase its intensity and transmit it to *Analog Digital Conversion* (ADC), which in turn, converts it to a digital signal. The respective digital audio signal is captured by the DSP, in particular by the digital audio processor, which manipulates it to improve the audio quality before compression (volume and frequency control). Finally, the signal is compressed by means of a coding algorithm, encapsulated in a container and stored. In general, cameras of mobile devices use the compression algorithm with loss *Advanced Audio Coding* (AAC), established in the MPEG-4 standard part 3.

A. MP4 MULTIMEDIA CONTAINER

The MP4 multimedia container is designed to contain multimedia information (video, audio, metadata, subtitles) [11] [12]. After data encapsulation, the multimedia container is organized hierarchically, through objects called atoms that are intrinsic elements of each video. One atom contains additional atoms and in turn labels with their respective values. The MP4 multimedia container structure is shown in Figure 2. Atoms are grouped into two types: a) interpretation and metadata atoms, which contains 5% of data and provides the information required to extract the audio and frames. Usually they have one audio track, one video track, and two metadata tracks; b) storage atom; it is a single atom called media data (mdat), which contains 95% of the video information.

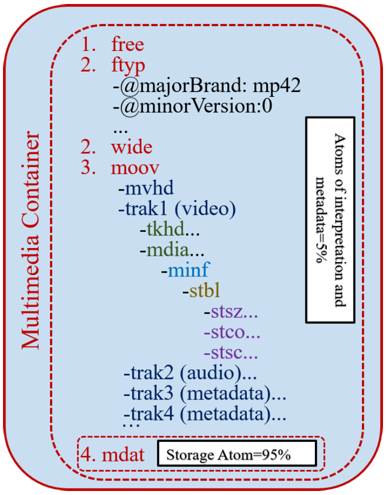

FIGURE 2. MP4 multimedia container structure.

III. FORENSIC AND ANTI-FORENSIC TECHNIQUES FOR VIDEOS

A. FORENSIC ANALYSIS TECHNIQUES

Forensic techniques in videos are grouped in forensic tools based on the objective to be fulfilled: source identification, compression or manipulation analysis of a video [13]. Forensic tools for source identification analysis recognize brand and model information from different devices. These researches have been performed usually on images, however the respective analysis on digital videos remains very limited. In this case, most techniques that have been applied to images, are used on video frames. In [14], a detailed comparison is carried out, focusing on the main groups of source identification techniques. These groups are divided into five, based on these features: metadata, image characteristics, defects of the CFA matrix and chromatic interpolation, sensor imperfections and wavelet transformations. Metadata-based strategies are easy to analyze, but the results depend on the data that manufacturers include. Techniques based on image characteristics can be divided into color characteristics, image quality metrics (*Image Quality Metrics* (IQM)) and wavelet domain statistics. In this case, images are classified through a *Support Vector Machine* (SVM) approach [15].

Approaches based on CFA matrix defects and chromatic interpolation induce certain marked differences among different camera models [6], [16]–[18]. In techniques based on sensor imperfections, two relevant key features have been identified: pixel defects or sensor noise patterns [19]–[22]. Regarding the strategies based on the wavelet transform described in [23], it has been proved that the use of sensor noise patterns in conjunction with the wavelet transform, is an effective method for source identification, reaching an average accuracy of 87.21%. Forensic tools for compression analysis use video coding architectures, more complex than those implemented for images. Most widely used coding standards, such as MPEG or H.26x families, inherit the use of encoding block systems by the JPEG standard transform. In [24], a method to detect double compression in MPEG, based on block artifacts, is proposed. Likewise, a measurement system is defined to calculate the *Block Artifact Strength* (BAS) of each frame. The mean BAS is calculated to eliminate the sequences obtained from 1 to 11 frames, obtaining a feature vector of BAS values. If the sequence has been previously

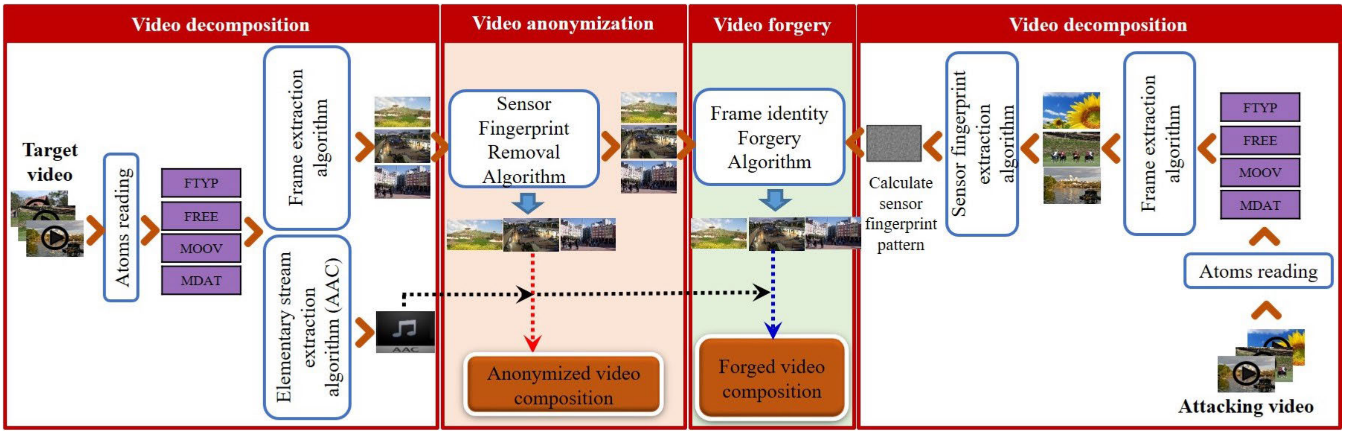
Video decomposition
Video anonymization
Video forgery
Video decomposition
Target video
Atoms reading
FTYP
FREE
MOOV
MDAT
Frame extraction algorithm
Elementary stream extraction algorithm (AAC)
AAC
Sensor Fingerprint Removal Algorithm
Anonymized video composition
Frame identity Forgery Algorithm
Forged video composition
Calculate sensor fingerprint pattern
Sensor fingerprint extraction algorithm
Frame extraction algorithm
FTYP
FREE
MOOV
MDAT
Atoms reading
Attacking video

FIGURE 3. General process of source anonymization and forgery in MP4 videos.

manipulated, whether it is frame deletion or re-compression, the feature vector exhibits a characteristic structure.

Forensic tools for manipulation analysis face different kinds of manipulation in videos, such as the substitution and elimination of frames, the replication of a set of frames and the insertion or elimination of objects in the scene. Mobile devices usually leave a unique digital footprint in recorded videos. Although these fingerprints are frequently considered only for device identification, some researches such as [25]–[27] have been developed to detect manipulations.

B. ANTI-FORENSIC ANALYSIS TECHNIQUES

The attacker can have more knowledge about forensic techniques, which could help him to develop sophisticated anti-forensic techniques to manipulate videos without leaving any trace. This action is mainly focused on hiding any vestige of the procedure performed and thereby, deceiving the forensic analyst, which could lead to inaccurate decisions. In order to implement this technique, the authors identify the properties of the temporal fingerprint. These properties were used to model the effect of deletion/addition of frames. With regard to static images, in [28], [29], an attacks classification is performed for the forensic image analysis, according to their main objectives: (1) camouflage of malicious post-processing on the image, (2) destruction of the correct identification of the image source and (3) forgery of the image source.

IV. PROPOSED ANTI-FORENSIC TECHNIQUES

This work aims to deceive or circumvent forensic source identification methods in MP4 format videos.

In this sense, two anti-forensic techniques to anonymize and forger MP4 videos source are proposed. The techniques are based on the decomposition of the video, the sensor imperfections and the wavelet transform. Figure 3 shows a scheme of the general process for video source anonymization and forgery in MP4 format.

For the decomposition of videos, the reading, and analysis of the atoms is carried out previously, followed by the extraction of the elementary stream of audio (AAC) and video H.264.This to avoid manipulating (re-compression) the generated file and thus compose a manipulated video with characteristics similar to the original.

Currently, there is little research that uses atoms for these purposes. This is because manufacturers have a high level of confidentiality with information related to the container structure and because it is a relatively new and complex topic.

As mentioned in section II-A the multimedia container is divided into two parts: a) interpretation and metadata atoms; b) storage atom. Therefore, Figure 4 shows how to interpret the content of atoms to extract samples from the audio and video track. First, we identify the audio or video track. Second, we obtain the atoms contained in the sample table atom (stbl) with location information and description of the media data atom (mdat).Finally, we obtain the samples from mdat atom using information from the stco, stsc, stsz, stsd atoms.

A. MP4 VIDEO ANONYMIZATION ALGORITHM

The main aim of the anonymization algorithm is to eliminate the entire existing sensor information of the video frames from the mobile device camera. The objective of this task is to prevent the mobile device identification used to record a specific video. The algorithm is composed by three procedures: (1) an algorithm for the elementary stream AAC (audio) extraction, (2) an algorithm for the frames extraction, and (3) an algorithm for the noise elimination from each video frame.

Algorithm 1 graphically represents the anonymization of a video. It starts with the reading of the video data to verify if the structure meets the specifications. The required data are obtained from the container atoms *mdat*, *moov*, *trak* and their respective daughter atoms. Then Algorithm 2 extracts the elementary stream AAC (*Audio*) from the original video ($Video_{original}$), keeping the raw data to avoid recompression. Then, the extraction of the objective frames ($Frames_{target}$ from the original video ($Video_{original}$) is carried out by means of the Algorithm 3.

Subsequently, the existing noise fingerprint *Photo Response Non Uniformity* (PRNU) in each frame, extracted from the video ($Frames_{target}$), is removed. As a result, a set

Algorithm 1 Anonymization of an MP4 Video

```
Input: $Video_{target}$ : It is the target MP4 video
Result: $Video_{anonymized}$ : This is the anonymized
        MP4 video
procedure AnonymizeVideo($Video_{original}$)
    Read $Video_{original}$;
    $Audio \leftarrow$ ExtractAccStream($Video_{original}$);
    $Frames_{target} \leftarrow$
    ExtractFrames$Video_{original}$;
    foreach $frame$ in $Frames_{target}$ do
        $Frames_{nonoise} \leftarrow$
        RemovePRNU($frame$);
    $Video_{anonymized} = Audio + Frames_{nonoise}$;
end procedure
```

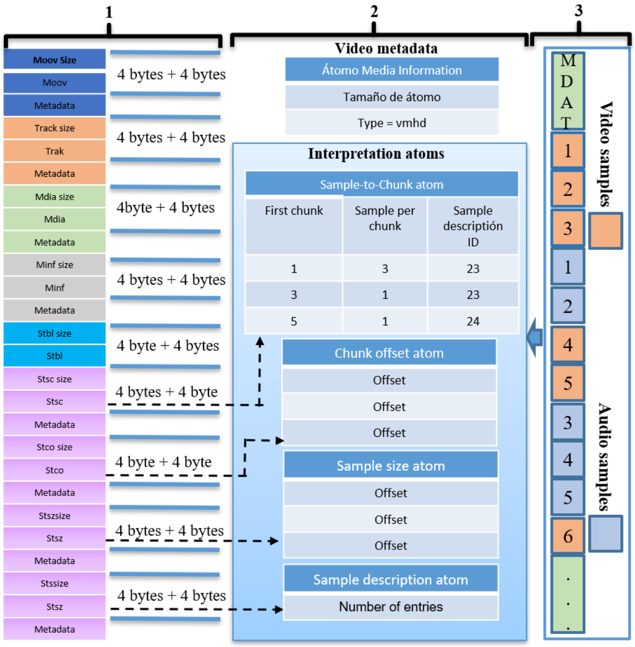

FIGURE 4. Interpretation of atom content.

of noise-free frames is obtained ($Frames_{nonoise}$). This process is based on the work presented in [28]. Finally, the recomposition of the video is performed using the noiseless frames ($Frames_{nonoise}$), together with the extracted audio (*Audio*) in order to generate an anonymized video ($Video_{anonymized}$).

1) AUDIO EXTRACTION

Currently, some tools extract the audio from a multimedia file, but when executing the procedure the data is forcibly re-compressed, and significantly modified to the elementary stream of audio, inserting new features for the forensic analyst to detect a manipulation. In this sense, the Algorithm 2 is proposed that extracts the audio elementary stream transparently and naturally, without re-compressing and using the information of the atoms.

Algorithm 2 Extraction of the Elementary Stream AAC (Audio)

```
Input: $Video_{original}$ : It is the target video MP4
Result: $Stream_{elementalaac}$ : This is the elementary audio
        with AAC encoding
procedure ExtractAccStream($Video_{original}$)
    Read $Atom_{stbl}$ from audio track;
    Get data from Atoms $std$, $stsc$, $stco$ and $stsz$;
    Get $Freq_{audio}$ from $Atom_{std}$;
    Get $Channel_{audio}$ from $Atom_{std}$;
    Set extension of $Stream_{elementalaac}$;
    Calculate $Num_{entries}$ from $Atom_{std}$;
    foreach $chunk$ in $Number_{entries}$ do
        Get $Num_{chunk}$;
        Get $Num_{samplexChunk}$;
        Get $offset_{chunk}$;
        foreach $sample$ in $Num_{samplexChunk}$ do
            Get $Size_{sample}$ ($sample$);
            Write $Cab_{sample} = Size_{sample} +$
                    $Freq_{audio} +$
                    $Channel_{audio}$;
            Get data block ($sample$);
            Dump raw data of ($sample$);
end procedure
```

It starts with a recursive reading of every atom in the video, focusing on the atom *trak* of audio: *stbl* (data), *stsd* (channels, frequency), *stsc* (samples number in a *chunk*), *stco* (sample location), *stsz*(size of each sample in a *chunk*), *mdat* (audio samples). Then, the number of entries of the atom *stsc*, the number of *chunk* and the number of samples contained in each *chunk* are computed. The next step is to obtain the shifting for each sample. Subsequently, the size of each sample of the *chunk* is obtained in order to write the header of each one with the following elements: Sample size, and audio and channel frequency. The location of each sample is moved according to the information of the atom *stsc*, to obtain the corresponding data block *mdat* of the MP4 video. Finally, this information is saved in a file with extension AAC.

2) FRAMES EXTRACTION

Most frame extraction methods run directly on videos. This procedure inserts increased noise over the extracted frames, causing subsequent procedures to be adversely affected. Therefore, a novel method of frame extraction is proposed and is composed of two phases: a) Extract the elementary video stream (H.264) from the media container; b) Extract frames from the H.264 file. This method of extraction allows obtaining frames with more real and precise characteristics.

The algorithm 3 represents the frames extraction procedure. It starts with a recursive reading of every existing atom in the video, focusing on the atom *trak* of the video:

Algorithm 3 Frames Extraction

```
Input: $Video_{original}$ : It is an MP4 video, whether it is
       target or attacking
Result: $Group_{frames}$ : Frames extracted from an
        MP4 Video
procedure ExtractFrames($Video_{original}$)
    Read $Atom_{stbl}$ of the video track of $Video_{mp4}$;
    Get data from $Atom_{std}$; $Atom_{stsc}$; $Atom_{stco}$;
  $Atom_{stsz}$;
    Set extension of $Stream_{elementalh264}$;
    Calculate $Number_{entries}$ of $Atom_{std}$;
    foreach chunk in $Number_{entries}$ do
        Get $Number_{chunk}$;
        Get $Number_{samplesperChunk}$;
        Get $offset_{chunk}$;
        foreach sample in $Number_{samplesperChunk}$
  do
            Get $Size_{sample}$;
            Get data block of sample;
            Verify length Nalu size of sample;
            Include the first part of the sample
      header in the file;
            Include the second part of the sample
      header in the file;
            Include delimiter file in sample;
            Dump raw data (samples);
    $Group_{frames}$ ← Extract (H.264 file);
end procedure
```

stbl (data), *stsd* (log data of decoder configuration), *stsc* (number of samples in a *chunk*), *stco* (location of the *sample*), *stsz* (size of each sample of a *chunk*), *mdat* (video samples). Then, the number of entries of the atom *stsc* is obtained. The following steps are similar to those used for the audio extraction until the calculation of the size of each sample of *chunk*. From each of these samples, the data block is obtained and the length field (*nalu size*), usually composed of 4 bytes, is verified. The first part of the access unit header (*access unit*) is added to the H.264 file. In this header, the following data are inserted: a start prefix of the access unit (*start of access unit*), represented by (00 00 00 01), the header of the drive *Network Abstraction Layer* (NAL), which contains the fields *forbidden_zero_bit*, *nal_ref_idc* (0) and *nal*_unit_type (9), identified as *access unit delimiter*, the *slice types*: (E0) that represents any of the following types of *frames* (I, P, B, SI and SP) and a delimiter (00 00 00 01). This header is included at the beginning of each access unit. From position 7 of the *decoderConfigurationRecord*, the size in bytes of the sequence is extracted with the data that will be read from the decoder. This information is incorporated to the file .H264 followed by the delimiter (00 00 00 01). Likewise, in the second part of the header, it is required to obtain the size in bytes of the following data sequence (from position 26) from the decoder. These data are included, succeeded by the delimiter (00 00 01). The data representing the video information is written to the H.264 file. Then, frames are extracted in JPEG format from H.264 file using the maximum quality allowed (JPEG_QUALITY = 100). The frames were extracted using the following settings: a) Extraction of 100% of frames (I and P frames), which are used to generate anonymized and forged videos, which are very similar to the originals; b) Extraction of a certain number of keyframes, which are used to measure the effectiveness of the proposed techniques.

3) VIDEO COMPOSITION

This process requires two elements: a) extracted audio; b) set of frames without noise. For this, the library Libx264 (H.264) of the free software FFmpeg is used, which helps substantially in the treatment of adjustment, conversion, and creation of audio and video. To make the anonymized video as similar as possible to the original, the following aspects must be taken into account: the number of frames per second or frame rate, the type of codec, the bit rate of video and audio, the video profile, the type of pixel format YUV420P, metadata information such as the date of creation of the video and finally the most important, the synchronization between audio and video.

To synchronize audio and video, FFmpeg (itsoffset) is used, which allows you to move back or forward the start time of both the audio data stream or the video data stream. The following shows how to synchronize an anonymized video, which produces a delay of 2 seconds and affects the audio stream.

B. MP4 VIDEO FORGERY

In order to perform this procedure, two elements are involved: an attacking video and a target video, as illustrated in Figure 5. The Algorithm 4 represents this process, starting with the data and information extraction from the atoms of the target video, *mdat moov*, *trak* and their respective children atoms, from the audio and video tracks. Subsequently, the extraction of the elementary stream AAC is performed with the Algorithm 2 and the frames extraction with the Algorithm 3. Each frame of the target video contains the intrinsic fingerprint traces left by the mobile device's sensor. Therefore, the next step is to eliminate the fingerprint of the sensor in each of the frames of the original video with the algorithm proposed in the work [28]. In this case, the attacking video is processed as previously described, until the extraction of frames and audio of the video is completed. The extraction of the sensor noise pattern is then performed by processing the frames extracted from the attacking video [28]. As such, the falsification of each frame is carried out without any noise from the target video, embedding the attacker pattern to each frame. As a result, a set of false frames $Frames_{false}$ is generated. This procedure is performed using the algorithm proposed in [30]. Finally, the recomposition of

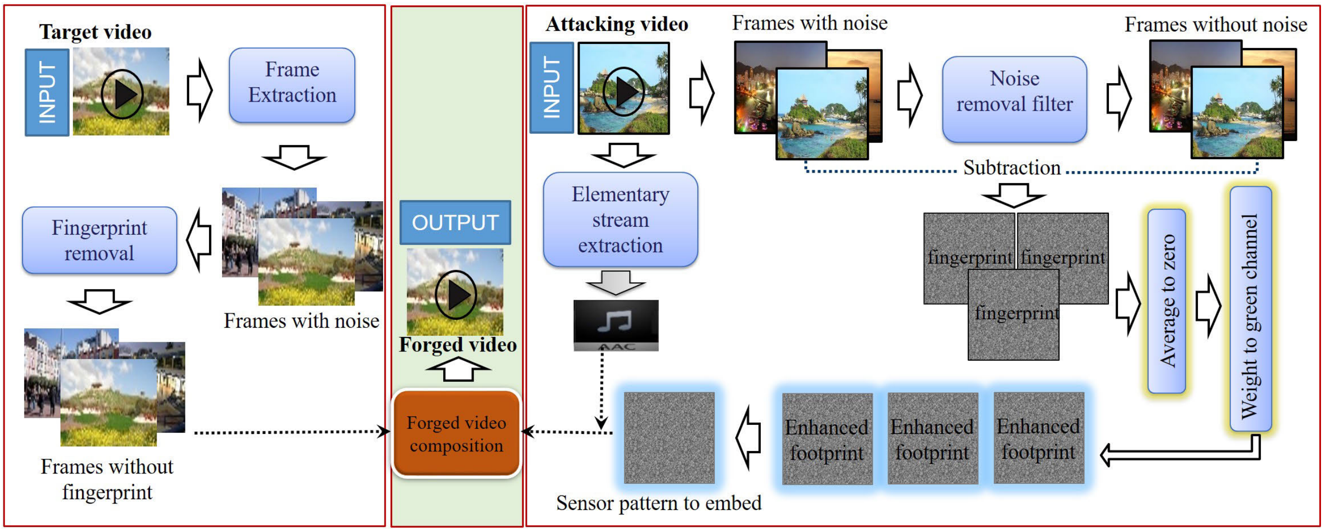

FIGURE 5. Video forgery of an MP4 Video.

Algorithm 4 MP4 Video Forgery

```
Input: $Video_{original}$ : It is the target MP4 video
       $Video_{attacking}$ : This is the attacking MP4 video
Result: $Video_{false}$ : It is the forged MP4 video
procedure ForgeMP4($Video_{original}$ , $Video_{attacking}$)
    Read $Video_{original}$;
    $Frames_{target} \leftarrow$ ExtractFrames$Video_{original}$;
    $Audio \leftarrow$ ExtractAccStream$Video_{original}$;
    foreach $frame0$ in $Frames_{target}$ do
        $Frames_{nonoise} \leftarrow$ RemovePrnu($frameO$);
    Read $Video_{attacking}$;
    $Frames_{attacking} \leftarrow$ ExtractFrames($Video_{attacking}$);
    foreach $frameA$ in $Frames_{attacking}$ do
        $Fingerprint_{attacking} \leftarrow$ ExtractFingerprint($frameA$);
    Calculate the noise pattern $Pattern_{attacking}$= average ($Fingerprint_{attacking}$);
    foreach $frameH$ in $Frames_{nonoise}$ do
        $Frames_{false} \leftarrow$ ForgerFrame($frameH$);
    $Video_{false} = Audio + Frames_{false}$;
end procedure
```

the $Video_{false}$, using the audio extracted *Audio* from the original video, and the set of false frames $Frames_{false}$ previously obtained, is completed.

V. EXPERIMENTS AND RESULTS

In this section, we evaluate the effectiveness of the proposed techniques through the execution of experiments. These experiments are distributed as follow: a) MP4 multimedia container structure analysis; b) Source identification for videos without post-processing, to verify the behavior of the method that will be in charge of measuring the efficiency of the proposed techniques; c) Source identification for videos with post-processing, to verify if the re-compression process affects the classification; d) Evaluation of the technique of anonymization of videos with equal conditions of creation; e) Evaluation of the effectiveness of the video forgery technique.

TABLE 1. Mobile devices classified by brand and model.

Brand	Huawei	Motorola	Samsung	Sony	Xiomi
Model	Y635-L01	Nexus 6	Galaxy S5 Galaxy S6	Xperia M2	Mi3

A. MP4 MULTIMEDIA CONTAINER STRUCTURE ANALYSIS

The aim of this analysis is to describe the intrinsic structure of MP4 multimedia containers, namely to detail the hierarchical organization of the atoms, labels with their respective values and to examine the metadata.In this way, we verify whether manufacturers of mobile devices meet the specifications of the standard.

The set of videos is composed by 60 recorded videos of 6 models from 5 different manufacturers. Table 1 shows in detail the mobile devices involved in this classification.

The first atom found in all the videos is the fytp as the standard reports. When the video contains the free atom, the next atom found is the moov, followed by its child atoms; the last one is the mdat atom. When the video does not have the free atom, the next observed atom is the mdat, followed by the moov atom with its respective child atoms.

In this regard, the specification does not contemplate the specific order of atoms, nor the ocurrence of a particular atom; this feature will depend on each manufacturer or model. However, after the analysis on several videos, it is noted that they generally follow the same pattern in terms of order and types of atoms.

In the case of moov atom, although it has the same structure, the child atoms model may vary. Examples of this particularity are the mobiles Motorola Nexus 6, which have internal atoms that in turn contain child atoms hdlr, keys and ilst or the Samsung S5, Samsung S6, and Sony Xperia M2 mobiles, which have the atom udta. The first atom trak is associated to the video track and the second one to the audio track. This structure is determined by the hdlr atom associated to the first trak, which incorporates the fields component type and subtype with values vide and videohandle and the hdlr atom associated to the second track, which considers the fields component type and subtype with values soun and soundhandle. After analyzing the first atom trak (video), it is observed that the majority of atoms are the same, except for the child atom minf, which contains the atoms vmhd and dinf. The atom dinf, in turn, always has the child atom dref, which has the atom url, as indicated in the specification. Regarding to the stbl atom, it comprises the stsd, stts, stss, stsz, stsc and stco atoms. The atom avc1, included in the atom stsd, indicates the type of codec, and contains an atom avcC and in some instances, an atom called pasp. The atom track, associated to the audio track, is composed by a child atom minf. The latter has child atoms smhd, in substitution to the atoms vmhd and dinf, which is the same as the first atom trak previously described. The atom stbl, on the other hand, is equal to the first trak atom, with the exception of the inexistence of an atom stss and the contents of the stsd atom.For the latter, it contains the atom mp4a, which indicates the type of codec and is composed by the atom esds. Therefore, upon completion of the analysis of the atoms structure and information of the MP4 videos, it can be concluded that the same mobile phone always has the same video atoms, keeping their respective order. Nearly all manufacturers follow the same specifications with some exceptions, such as the preferred format, the list of compatible formats, the data contents (user data or metadata), etc.

B. SOURCE IDENTIFICATION FOR VIDEOS WITHOUT POST-PROCESSING

The main objective of the techniques of anonymization and forgery of videos proposed is to prevent the correct identification of the source of videos in MP4 format. To evaluate the efficiency of both techniques is used the method of source identification based on clustering [31]; that the hereafter will be called *validation method*. This method focuses on the combination of hierarchical and flat clustering and the use of Sensor Pattern Noise (SPN).

The experiments were conducted using 20 videos generated by 5 models of mobile device (Samsung Brand). These devices use the H.264 video encoder, AAC audio encoder and Android operating system. Table 2 details the dataset used in the experiments with their respective characteristics. Table 3 shows the parameters used in the *validation method*.

First, the tolerance of the *validation method* was analyzed using different keyframe crop sizes. For this, two experiments with two crop sizes were carried out: 640 × 480 and 1280 × 720 pixels. The 20 videos from the dataset in Table 2 were used, with no additional post-processing to those of their capture. Fifty keyframes were extracted from each video, for a total of 200 keyframes per model, following the parameters shown in Table 3. The clustering results with each crop size are presented in Tables 4 and 5.

TABLE 2. Set of videos used in the experiments.

Model	FPS	Video Resolution	#Videos	Terms Of Capture
Galaxy A3	30	1920x1080	4	Scene Type: Any
Galaxy Ace Style	30	1280x720	4	Flash:Disabled
Galaxy S5 Neo	30	1920x1080	4	Light: Natural
Galaxy S6	30	1920x1080	4	White balance: Auto
Galaxy GT-I9000	30	1280x720	4	Capture time: 2 minutes

TABLE 3. Parameters of the evaluation method.

Parameter	Value
Daubechies wavelet	db8
Descomposition level	4
Crop of the centered frame	Yes
Adaptive variance estimation	Yes
Non-zero mean	Yes
Crop Required	Yes
Keyframes per video	50
Video per Model	4

TABLE 4. Confusion matrix for original video clustering with crop size 640 × 480 pixels.

Mobile Device	Clusters					Accuracy Rate
	1	2	3	4	5	
A3	**159**	18	19	2	2	80%
Ace Style	18	**133**	26	10	13	67%
S5 Neo	21	8	**170**	1	0	85%
S6	2	4	0	**171**	23	86%
GT- I9000	4	30	6	28	**132**	66%

TABLE 5. Confusion matrix for original video clustering with crop size 1280 × 720 pixels.

Mobile Device	Clusters					Accuracy Rate
	1	2	3	4	5	
A3	**192**	8	0	0	0	96%
Ace Style	4	**192**	0	0	4	96%
S5 Neo	0	0	**200**	0	0	100%
S6	0	0	0	**200**	0	100%
GT- I9000	6	4	0	4	**186**	93%

As can be seen in Table 4, clustering with keyframes cropped to 640 × 480 pixels results in 5 clusters, with the same number of existing classes. However, some classes are confused with others, reaching an average accuracy rate of 76.8%. In the case of clustering results with keyframes cropped to 1280 × 720 pixels shown in Table 5, it is observed that 5 clusters were formed, reaching an average accuracy rate of 97%. Being a significantly higher accuracy rate than using a resolution of 640 × 480 pixels.

C. SOURCE IDENTIFICATION FOR VIDEOS WITH POST-PROCESSING

Since the anonymized videos suffer recompression when being reconstructed, we proceeded to extract 100% of frames and audio in AAC format from the 20 videos in Table 2, each video is composed again synchronizing the frames and audio to maintain the same characteristics of the original video. With this procedure, we get a new dataset with

TABLE 6. Confusion matrix for non-anonymized video clustering with crop size 640 × 480 pixels.

Mobile Device	Clusters					Accuracy Rate
	1	**2**	**3**	**4**	**5**	
A3	**158**	18	19	2	3	79%
Ace Style	18	**133**	23	10	16	67%
S5 Neo	21	8	**170**	1	0	85%
S6	1	4	2	**172**	21	86%
GT- I9000	6	29	8	25	**132**	66%

TABLE 7. Confusion matrix for non-anonymized video clustering with crop size 1280 × 720 pixels.

Mobile Device	Clusters					Accuracy Rate
	1	**2**	**3**	**4**	**5**	
A3	**191**	7	2	0	0	96%
Ace Style	5	**190**	2	0	3	95%
S5 Neo	0	0	**200**	0	0	100%
S6	0	0	0	**200**	0	100%
GT- I9000	4	4	2	1	**189**	95%

recompressed videos without any additional modification. The keyframes are extracted and the *validation method* is executed again to evaluate how much the recompression of the videos affects the classification of the source acquisition. For this, 50 keyframes were extracted from each of the recompressed videos in Table 2, for a total of 200 keyframes per model. As in section V-B, two experiments were carried out with 2 crop sizes (640 × 480 and 1280 × 720 pixels). The results of each clustering are shown in Tables 6 and 7 respectively.

As can be seen in Table 6, using a crop size of 640 × 480 5 clusters were obtained, the same number of dataset models. The model that achieved the best accuracy rate is the Galaxy S6 with 172 keyframes correctly classified, which translates into 86% of the total keyframes. However, it can be seen that a small number of keyframes are confused with other models. Overall, the average hit rate was 76.6%. These results are very similar to those obtained in Table 4 (Section V-B), where the Galaxy A3 model managed to group 159 keyframes and the Galaxy S6 171, while in this experiment the number of Galaxy A3 keyframes is reduced by 1 and the Galaxy S6 increases by 1. Likewise, the average success rate decreases by 0.2%.

Analogously, the results from Table 7, show that using crop size of 1280 × 720 pixels, 5 clusters were obtained. The Galaxy S5 Neo and Galaxy S6 models achieved a 100% accuracy rate. The GT-I9000 model grouped fewer keyframes, with a total of 189 (95% correctly grouped). In summary, the accuracy rate achieved in this experiment is 97.2%. As in the previous case, the results of this experiment follow the same behavioral pattern as the results of Table 5 (Section V-B), where it is observed that the Galaxy A3 model managed to group 192 keyframes, the Galaxy Ace Style 192 and the GT-I9000 186; while in Table 7 the number of Keyframes grouped by Galaxy A3 is reduced by 1 and the Galaxy Ace Style decreases by 2 and the GT-I9000 increases by 3. Also, the average accuracy rate increases by 0.2% as in the results in Table 6.

With these results, can be concluded that although the video reconstruction process involves re-compression of the audio stream and frame sequence, this process does not significantly affect the accuracy rate in the classification of original videos (generated by mobile devices) and classification of post-processing videos (generated from original videos).

TABLE 8. Classification of anonimized videos with a crop size of 640 × 480 pixels.

Mobile Device	Clusters											Accurary Rate
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	
A3	15	**29**	22	28	15	24	9	12	18	10	18	14.5%
Ace Style	18	15	16	21	15	19	14	19	**23**	18	22	11.5%
S5 Neo	16	12	21	18	20	**33**	26	14	13	9	18	16.5%
S6	7	21	11	**35**	14	19	13	18	29	17	16	17.5%
GT- I9000	**22**	11	16	21	21	19	17	14	19	21	19	11%

TABLE 9. Classification of anonymous videos of 1280 × 720 pixels.

Mobile Device	Clusters										Accurary Rate
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	
A3	13	10	19	21	16	14	17	**38**	24	28	19%
Ace Style	11	28	15	20	13	**35**	22	19	13	24	17.5%
S5 Neo	17	8	**46**	14	26	15	18	6	17	33	23%
S6	**48**	18	19	28	13	11	7	10	14	22	24%
GT- I9000	23	30	15	22	**32**	16	23	13	15	11	16%

D. MP4 VIDEOS ANONYMIZATION

Once we have examined the correct classification of the videos with the chosen *evaluation method*, we proceed to analyze how the anonymized videos are classified with algorithm 1. The evaluation process used is the follow: First, all videos from Table 2 were anonymized. Then, for each anonymized video 50 keyframes were extracted (200 keyframes by model). Finally, the *validation method* is executed with these keyframes using the settings shown in Table 3. The results obtained with each crop size (640 × 480 and 1280 × 720 pixels), shown in Tables 8 and 9, were compared with the results of the experiment V-C.

As can be seen in Table 8, the *evaluation method* obtained 11 clusters. Correctly grouped keyframes are highlighted in bold for each model and do not exceed 18% at best. Comparing these results with those shown in Table 6, it is clear that the following differences exist: 11 clusters were created instead of 5, and the distribution of keyframes in clusters does not follow a pattern that easily identifies the source of the video. For example, the Galaxy S6 model was the model that grouped more keyframes correctly, reaching 17.5% (only 35 in cluster 4) versus 86% (172 keyframes) obtained by this model in Table 6. The same way, the model with less keyframes correctly grouped was GT-I9000, only 11% (22 in cluster 1) versus of the 66% (132 keyframes) previously obtained.

Analogously, as in the previous experiments and as can be seen in Table 9, the results show that although a larger crop size (1280 × 720 pixels) was used, 10 clusters were formed for the 1000 keyframes analyzed. The maximum success rate was 24% in the Galaxy S6 model, with 48 keyframes grouped in cluster 1.

Again, if we compare the results shown in Tables 7 and 9, we find the following differences: the number of clusters doubles (from 5 to 10 clusters, the average accuracy rate decreases considerably (from 97.2% to 19.9%). In summary, the keyframe distribution of anonymized videos does not

TABLE 10. Devices involved during the forgery experiment.

Target Video			Attacking Video		
Brand	Model	Resolution	Brand	Model	Dimensions
Samsung	Galaxy Ace Style	1280x720	Samsung	Galaxy GT-I9000	1280x720

TABLE 11. Confusion matrix for the forgery classification results.

Mobile Device	Clusters					Accuracy Rate
	1	2	3	4	5	
A3	**191**	6	0	0	3	**96%**
Ace Style	4	**184**	0	0	12	**92%**
S5 Neo	0	0	**200**	0	0	**100%**
S6	0	0	0	**200**	0	**100%**
GT- I9000	4	27	1	2	**166**	**83%**

follow a specific pattern, making it difficult to classify them correctly.

These results confirm that the videos resulting from applying the proposed video anonymization technique (Algorithm 1) efficiently circumvent or mislead the video classification algorithm. This method was tested with videos with the same creation conditions to avoid possible deviations in the results.

E. EFFECTIVENESS OF THE MP4 VIDEO SOURCE FORGERY

In order to evaluate the effectiveness of the source forgery for MP4 videos generated by mobile devices, the falsification of a video (target video) was performed to make it seem recorded by another model of mobile device (attacking video). For this purpose, the same models of experiment 1 were used from Section V-B. The characteristics associated to both videos are detailed in Table 10. No video of the remaining 4 models was forged.

Similarly to the previous experiments, 50 keyframes were extracted from each video (200 keyframes for each model). In this way, 50 keyframes were extracted from the forged video, classified as Galaxy GT-I9000, and 150 keyframes were extracted from the remaining three videos of the same model to complete a total of 200 keyframes for the model. The default keyframe size was used (1280 × 720 pixels).The classification results after the forgery process are detailed in Table 11.

Based on the results shown in Table 11, the following observation can be made: (i) in comparison with the classification results summarized in Table 5, which describe videos without anonymization and falsification, accuracy rates have only changed for GT I9000 and Galaxy Ace Style models, (ii) in the case of the Galaxy Ace Style model, the accuracy rate decreased from 96% to 92%; this result is obtained because after frames anonymization to the Galaxy Ace Style models and forgery process to relate them with the GT I9000 model, the algorithm has more difficulties to perform an accurate classification, (iii) accuracy rate for the GT I9000 (83%), decreased 10% compared to the experiments without any alteration; 166 from 200 keyframes (150 originals and 50 with alterations) were correctly classified, in comparison to the classification results of Table 5, where 186 out of 200 had been correctly classified. It can be concluded that 30 out of 50 falsified keyframes achieved the goal.

VI. CONCLUSION

In the present study, two anti-forensic techniques for digital videos are proposed. First, an anti-forensic technique to anonymize an MP4 video is presented and then, an anti-forensic technique to forger the source of an MP4 video is developed. Both techniques are composed of a series of algorithms that are based on the video decomposition, the sensor noise characterization and the wavelet transform. The proposed algorithms are aimed at the extraction of frames and audio from the video, the elimination of the sensor footprint of each frame, the extraction of residual noise or the sensor footprint of each frame, the calculation of the noise pattern of the sensor and, finally, the forgery of a video source identification by adding the noise pattern of a different camera. The proposed video anonymization technique efficiently circumvents the video classification algorithm used as the evaluation method. This is clearly seen in the results obtained in the classification of anonymized videos with both crop size of 640 × 480 pixels and crop size of 1280 × 720 pixels. As can be seen in Tables 8 and 9, the 1000 keyframes were grouped into 11 and 10 clusters respectively; when they were expected to be grouped into 5 clusters in both cases. Additionally, it has been proven that the higher the resolution of the video the evaluation method increases its success rate considerably, from 76.8% to 97% in Section V-B experiments with videos without post-processing and from 76.6% to 97.2% in videos that have been affected by the re-compression produced by being reconstructed with the process presented in Section IV-A.3 without being anonymized. In relation to the acquisition source forgery, the classification algorithm had more difficulty to properly classify the keyframes, showing that 30 out of 50 forged keyframes achieved their objective. The evaluation of both anti-forensic techniques has been carried out by means of different experiments, reaching sufficient accuracy rates. Based on those results, it can be concluded that the proposed anti-forensic techniques were successful.

REFERENCES

[1] GSM Association. (Feb. 2019). *The Mobile Economy 2019*. [Online]. Available: https://www.gsma.com/r/mobileeconomy/

[2] Cisco Systems. (Feb. 2016). *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015–2020*. [Online]. Available: https://www.cisco.com/c/dam/m/en_in/innovation/enterprise/assets/mobile-white-paper-c11-520862.pdf

[3] M. R. Chourasiya and A. P. Wadhe, "Video anti forensics a review," *Proc. IJRITCC*, vol. 3, no. 4, pp. 2110–2114, Apr. 2015.

[4] M. Tennøe, E. Helgedagsrud, M. Næss, H. K. Alstad, H. K. Stensland, V. R. Gaddam, D. Johansen, C. Griwodz, and P. Halvorsen, "Efficient implementation and processing of a real-time panorama video pipeline," in *Proc. IEEE Int. Symp. Multimedia*, Washington, DC, USA, Dec. 2013, pp. 76–83.

[5] Texas Instruments Incorporated. (2012). *Digital Still Camera*. [Online]. Available: http://www.ti.com/solution/digital_still_camera

[6] S. Bayram, H. T. Sencar, and N. Memon, "Classification of digital camera-models based on demosaicing artifacts," *Digit. Invest.*, vol. 5, nos. 1–2, pp. 49–59, Sep. 2008.

[7] Panasonic Corporation. (2019). *Lumix Digital Camera Know-Hows*. [Online]. Available: http://av.jpn.support.panasonic.com/support/global/cs/dsc/knowhow/index.html

[8] J. Nakamura, *Image Sensors and Signal Processing for Digital Still Cameras*. Boca Raton, FL, USA: CRC Press, 2005.

[9] International Telecommunication Union. (2016). *Advanced Video Coding for Generic Audiovisual Services H.264*. [Online]. Available: https://www.itu.int/rec/T-REC-H.264/

[10] International Telecommunication Union. (2018). *High Efficiency Video Coding*. [Online]. Available: https://www.itu.int/rec/T-REC-H.265

[11] Apple Inc. (2019). *QuickTime File Format Specification*. [Online]. Available: https://tinyurl.com/yylr6mtb

[12] International Telecommunication Union. (2003). *Information Technology Coding of Audio-Visual Objects Part 14: MP4 File Format*. [Online]. Available: https://www.iso.org/standard/38538.html

[13] P. Bestagini, M. Fontani, S. Milani, M. Barni, A. Piva, M. Tagliasacchi, and S. Tubaro, "An overview on video forensics," in *Proc. 20th Eur. Signal Process. Conf.*, Bucharest, Romania, Aug. 2012, pp. 1229–1233.

[14] A. L. Sandoval Orozco, D. M. Arenas González, J. Rosales Corripio, L. J. García Villalba, and J. C. Hernandez-Castro, "Techniques for source camera identification," in *Proc. 6th Int. Conf. Inf. Technol.*, Amman, Jordan, May 2013, pp. 1–9.

[15] M.-J. Tsai, C.-L. Lai, and J. Liu, "Camera/mobile phone source identification for digital forensics," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Honolulu, HI, USA, Apr. 2007, pp. 221–224.

[16] S. Bayram, H. T. Sencar, and N. Memon, "Improvements on source camera-model identification based on CFA interpolation," in *Proc. Int. Conf. Digit. Forensics*, Jan. 2006, pp. 24–27.

[17] O. Celiktutan, I. Avcibas, B. Sankur, N. P. Ayerden, and C. Capar, "Source cell-phone identification," in *Proc. IEEE 14th Signal Process. Commun. Appl.*, Antalya, Turkey, Apr. 2006, pp. 1–3.

[18] Y. Long and Y. Huang, "Image based source camera identification using demosaicking," in *Proc. IEEE 8th Workshop Multimedia Signal Process.*, Victoria, BC, Canada, Oct. 2006, pp. 419–424.

[19] J. Lukäš, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 205–214, Jun. 2006.

[20] Z. J. Geradts, J. Bijhold, M. Kieft, K. Kurosawa, K. Kuroki, and N. Saitoh, "Methods for identification of images acquired with digital cameras," in *Proc. Enabling Technol. Law Enforcement Secur.*, Boston, MA, USA, Feb. 2001, pp. 505–512.

[21] T. Van Lanh, K.-S. Chong, S. Emmanuel, and M. S. Kankanhalli, "A survey on digital camera image forensic methods," in *Proc. IEEE Int. Conf. Multimedia Expo*, Beijing, China, Jul. 2007, pp. 16–19.

[22] F. de O. Costa, M. Eckmann, W. J. Scheirer, and A. Rocha, "Open set source camera attribution," in *Proc. 25th SIBGRAPI Conf. Graph., Patterns Images*, Ouro Preto, Brazil, Aug. 2012, pp. 71–78.

[23] J. Rosales Corripio, D. M. Arenas González, A. L. Sandoval Orozco, L. J. García Villalba, J. C. Hernandez-Castro, and S. J. Gibson, "Source smartphone identification using sensor pattern noise and wavelet transform," in *Proc. 5th Int. Conf. Imag. Crime Detection Prevention*, London, U.K., Dec. 2013, pp. 1–6.

[24] W. Luo, M. Wu, and J. Huang, "MPEG recompression detection based on block artifacts," *Proc. SPIE*, vol. 6819, Mar. 2008, Art. no. 68190X.

[25] N. Mondaini, R. Caldelli, A. Piva, M. Barni, and V. Cappellini, "Detection of malevolent changes in digital video for forensic applications," *Proc. SPIE*, vol. 6505, Feb. 2007, Art. no. 65050T.

[26] C.-C. Hsu, T.-Y. Hung, C.-W. Lin, and C.-T. Hsu, "Video forgery detection using correlation of noise residue," in *Proc. IEEE 10th Workshop Multimedia Signal Process.*, Cairns, QLD, Australia, Oct. 2008, pp. 170–174.

[27] M. Kobayashi, T. Okabe, and Y. Sato, "Detecting forgery from static-scene video based on inconsistency in noise level functions," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 883–892, Dec. 2010.

[28] A. L. Sandoval Orozco, L. J. García Villalba, D. M. Arenas González, J. Rosales Corripio, J. C. Hernandez-Castro, and S. J. Gibson, "Smartphone image acquisition forensics using sensor fingerprint," *IET Comput. Vis.*, vol. 9, no. 5, pp. 723–731, Sep. 2015.

[29] T. Gloe, M. Kirchner, A. Winkler, and R. Bohme, "Can we trust digital image forensics?" in *Proc. 15th Int. Conf. Multimedia*, Augsburg, Germany, Sep. 2007, pp. 78–86.

[30] L. J. García Villalba, A. L. Sandoval Orozco, J. Rosales Corripio, and J. C. Hernandez-Castro, "A PRNU-based counter-forensic method to manipulate smartphone image source identification techniques," *Future Gener. Comput. Syst.*, vol. 76, pp. 418–427, Nov. 2017.

[31] L. J. García Villalba, A. L. Sandoval Orozco, and J. Rosales Corripio, "Smartphone image clustering," *Expert Syst. Appl.*, vol. 42, pp. 1927–1940, Sep. 2015.

ANA LUCILA SANDOVAL OROZCO was born in Chivolo, Magdalena, Colombia, in 1976. She received the degree in computer science engineering from the Universidad Autónoma del Caribe, Colombia, in 2001, the Specialization Course degree in computer networks from the Universidad del Norte, Colombia, in 2006, and the M.Sc. degree in research in computer science and the Ph.D. degree in computer science from the Universidad Complutense de Madrid, Spain, in 2014 and 2009, respectively. She is currently a Postdoctoral Researcher with the Universidad Complutense de Madrid. Her current research interests include coding theory, information security, and its applications.

CARLOS QUINTO HUAMÁN received the degree from the Universidad Inca Garcilaso de la Vega in Lima, Perú, in 2012, and the M.Sc. degree from the Universidad Complutense de Madrid (UCM), Spain, in 2016, all in computer science, where he is currently pursuing the Ph.D. degree with the Department of Software Engineering and Artificial Intelligence, Faculty of Computer Science and Engineering. He is a member of the Complutense Research Group, Group of Analysis, Security and Systems (GASS, http://gass.ucm.es). His current research interests include computer forensics, cybersecurity, electronic warfare, and cyberdefense.

JENNY ALEXANDRA CIFUENTES QUINTERO received the B.Sc. degree in mechatronics engineering, the M.Sc. degree in industrial automation, and the Ph.D. degree in mechanical and mechatronics engineering from the Universidad Nacional de Colombia, Bogota, in 2008 and 2010, respectively, and the Ph.D. degree in industrial automation from INSA de Lyon (INSA), France, in 2015. She has been a Visiting Researcher with the University of Alberta, Canada, and the Universidad Complutense de Madrid. Her current research interests include modeling and analysis of dynamic systems, signal processing, and pattern recognition.

LUIS JAVIER GARCÍA VILLALBA received the degree in telecommunication engineering from the Universidad de Málaga, Spain, in 1993, and the Ph.D. degree in computer science from the Universidad Politécnica de Madrid, Spain, in 1999. He was a Visiting Scholar with the Computer Security and Industrial Cryptography (COSIC), Department of Electrical Engineering, Faculty of Engineering, Katholieke Universiteit Leuven, Belgium, in 2000, and a Visiting Scientist with IBM Research Division, IBM Almaden Research Center, San Jose, CA, USA, in 2001 and 2002. He is currently an Associate Professor with the Department of Software Engineering and Artificial Intelligence, Universidad Complutense de Madrid (UCM) and the Head of Complutense Research Group, Group of Analysis, Security and Systems (GASS), Faculty of Computer Science and Engineering, UCM Campus. His professional experience includes the management of both national and international research projects (Spanish Ministry of Research and Development, Spanish Ministry of Defense, Horizon 2020 - European Commission) and both public and private financing (Hitachi, IBM, Nokia, Safelayer Secure Communications, and TB Solutions Security). He has authored or coauthored numerous international publications. He is an Editor or Guest Editor of numerous journals, such as *Entropy* (MDPI), *Future Generation Computer Systems* (FGCS), *Future Internet* (MDPI), the IEEE Latin America Transactions, *IET Communications* (IET-COM), IET *Networks* (IET-NET), IET *Wireless Sensor Systems* (IET-WSS), the *International Journal of Ad Hoc and Ubiquitous Computing* (IJAHUC), the *International Journal of Multimedia and Ubiquitous Engineering* (IJMUE), and the *Journal of Supercomputing*, and *Sensors* (MDPI).