Video Surveillance Authentication: Real-Time ENF Signal Hiding at the Edge

Antonios Lykourinas and Athanassios Skodras
Department of Electrical and Computer Engineering
Faculty of Engineering
University of Patras
Patras, Greece
email: up1059356@upnet.gr, skodras@upatras.gr

***Abstract*—Due to the significance of the visual information exchanged in Internet of Video Things (IoVT) networks, attackers are constantly launching new attacks and attempt to exploit new vulnerabilities. One of the most common and difficult-to-prevent attacks on the Visual Layer is the Frame Duplication Attack (FDA). Recently, two techniques were proposed for FDA detection at the edge by using the embedded Electrical Network Frequency (ENF) signals in an effort to surpass limitations of conventional passive methods. In this paper, a Real-Time ENF signal hiding technique at the edge is proposed. Our motivation is to examine the possibility of authenticating the surveillance feed by hiding the ENF signal. Experiments are conducted, including an extensive performance comparison between the proposed and reference encoder, a feasibility study for the proposed encoder's integration to a Raspberry Pi for video streaming purposes and finally the implementation of a proof-of-concept prototype. According to the findings, the proposed approach provides real-time FDA detection at reduced computational complexity and hardware requirements, thus rendering this method appropriate for applications at the edge.**

I. Introduction

VSSs have evolved rapidly in recent years thanks to Internet of Things (IoT) whose benefits have broadened their range of applications. With the integration of technologies, such as machine learning and computer vision, modern surveillance systems are capable of self-detecting abnormal events and relay them to other systems so that immediate actions can be taken [1]. A VSS malfunction could lead to missing a fast-paced crime so it is crucial to maintain their uninterrupted and unaffected operation [2]. Furthermore, VSS's are used in important places where only authorized agents should have access to monitor and control them; thus privacy and security should be given top priority when designing them [1].

VSS's are widely deployed because they offer flexibility, affordability, remote monitoring and ease of use. While several IoVT devices are offered in the market, their manufacturers are not obliged to comply with requirements established by IoT security standards, resulting in devices with outdated communication protocols, firmware with known vulnerabilities that is never patched and backdoor accounts that the user is unaware of. A lot of times these backdoor accounts cannot even be deleted [3]. Security or privacy breaches on VSSs can occur either in the edge devices, on the cloud or by the devices used for accessing their resources. VSS's are prone to Visual Layer, covert channel [4], [5], Steganography, Pan-Tilt-Zoom, Denial of Service, Malware and Privilege Escalation Attacks. A very common attack which targets the Visual Layer is the Frame Duplication Attack (FDA).

By compromising a surveillance camera, an attacker can manipulate the output of the surveillance feed by injecting previously recorded frames or by repeating a single static frame thus avoiding detection of any on-going suspicious activity. While several frameworks have been proposed for IoVT networks, including a cloud-based framework [6], a system based on a Raspberry Pi and a passive infrared sensor [7] and a secure method for safe data transfer over the internet by combining steganography with cryptography [8], they do not have a built-in feature for the prevention of FDA's. FDA's are the most difficult to prevent because their detection involves checking for spatial and temporal domain similarities between consecutive frames by extracting features and analysing them, which is computationally intensive. Methods achieving high detection rates while maintaining low computational complexity have been proposed, including one based on standard deviation of residual frames [9] and another one based on extracting binary features of the frame [10]. However, both operate only on previously stored media.

Two effective techniques were proposed for real-time detection of injected audio and video frames in the surveillance feed, using the Electrical Network Frequency (ENF) signals [11], [12]. ENF, which fluctuates around the power grid's nominal frequency (50Hz/60Hz) [13], was utilized for video surveillance feed authentication at the edge. In both cases, the ENF signal was extracted directly from the power grid, creating a reference database, and from the surveillance feed at the edge. In [11], ENF was extracted from the audio feed using a short-time Fourier Transform based method at the edge and a more robust spectrum-combining method on fog nodes. In [12], the ENF traces, created by indoor light sources and captured by the imaging sensors of the camera in the form of illumination frequency, were extracted from the video feed. In both cases, the Pearson Correlation Coefficient of the two ENF signals was obtained regularly using a sliding window approach and a certain threshold was determined through experimental studies for FDA self-detection at the edge.

Although the audio-based method is designed to run on a Raspberry Pi 3 Model B, a fog node is required for re-estimating the ENF with a more robust extraction algorithm in order to eliminate the false alarms produced by edge devices [11]. Moreover, the two ENF-based methods, also estimate the ENF signal twice at the edge and every edge requires a

sound card and an external circuit for the ground truth ENF estimation. Additionally, the video-based method is dependent on the presence of indoor light sources, whereas the audio-based method is dependent on the presence of ENF traces within the audio feed, and as more IoVT devices are becoming battery-powered their capture may be affected by a variety of environmental and device-related factors [14].

Both active and passive methods have been proposed for video tampering detection. Active methods rely on known information about the video, such as the existence of a digital watermark or signature [15]. None of the previous works mentioned, utilizes a data hiding scheme for FDA prevention. Many data hiding schemes that embed the data in the encoding process by exploiting redundancy removal mechanisms of a video CODEC such as intra-prediction [16]–[18], motion vectors and DCT coefficients have been proposed. In contrast to other data hiding schemes, they can be implemented by modifying the encoder without affecting its performance and they offer low complexity which makes them suitable for real-time applications [19]. Based on that, we designed an IoVT framework for FDA prevention whose operation relies on embedding the ENF signal into the video stream at the edge, using an expanded version of a data hiding scheme which exploits the IPCM macroblocks of the H.264 standard [17]. This data hiding scheme offers low complexity, considerable data capacity and reusability and thus it is suitable for real-time video streaming on an edge device. Moreover, this scheme was selected because it is a fragile one, which means that absence of the hidden data from the encoded video sequence could indicate that the video content has been tampered. This scheme is blind in the sense that the message can be extracted directly from the encoded stream without the need of the original video [17]. Our intention is to extract the hidden ENF signal at a Remote Control Center (RCC) and compare it to the ground truth signal for the detection of any abnormal activity.

The rest of the paper is organised as follows. In Section II the proposed method is introduced alongside with its enabling technologies that are used for building the framework. In Section III, we present our current implementation followed by an extensive performance comparison between the modified and reference encoder, the determination of the appropriate encoding settings for video streaming using a Raspberry Pi and a proof-of-concept implementation. Finally, in Section IV we present our conclusions followed by a brief discussion about our future work.

II. The Proposed Method

A. Data Hiding scheme

The H.264 standard [20] supports coded sequences that contain I, P and B slices (and other types of slices) used for storing encoded macroblocks. Each slice type supports different types of macroblocks. An I slice contains only intra-coded macroblocks predicted from reconstructed macroblocks from the current slice, whereas a P slice contains both intra- and inter-coded macroblocks predicted from a single reference list. Finally, a B slice contains both intra- and inter-coded macroblocks predicted from two reference lists. Both reference lists contain frames that have been previously encoded, filtered and reconstructed and can occur before and after the current frame in display order.

For encoding a macroblock, the encoder typically chooses the prediction mode with the lowest prediction error, which is the difference between the actual and the predicted macroblock. In some rare instances, predicting, transforming, quantizing and entropy coding the residual may result in data expansion rather than compression. This can be caused by various encoding parameters, for example when encoding a video to near lossless compression. In order to resolve this issue, the H.264 standard provides the IPCM macroblock mode in which the encoder directly entropy encodes the macroblock's pixel values by skipping the prediction, transformation and quantization process. The IPCM macroblock defines an upper limit for the number of bits that can be used to represent an encoded macroblock within a slice. An IPCM macroblock can also be used to store part of a frame without any visual loss.

Given the fact that the IPCM macroblock's compression is lossless, we can use the Least Significant Bits (LSBs) of the pixel values of the Luma and two Chroma components for data hiding purposes. Text information, which is UTF-8 encoded, can be converted to its binary representation and stored in the LSBs of the pixel values of the Luma and Chroma components of the macroblock. Modifying the LSBs does not affect the visual quality of the video. Actually, any change to the four LSBs is not perceived by the Human Visual System (HVS) [17].

In our case, we obtain an ENF stamp containing the current ENF estimation every second and we choose to hide every stamp twice per second in the video stream, captured at 30 fps. The ENF stamp is a string formatted as 'date time estimation', estimation being a 4-digit float, with a total of 27 characters required for its representation. Two extra encoder variables are declared. A variable, allocated with memory for 32 characters, used for storing the ENF stamp and a variable, initialized with zeros, used for storing the 256 bits of its corresponding binary representation. The variable holding the binary representation of the ENF stamp is used to modify the LSB of each pixel of the Luma component. The Luma component of an IPCM macroblock consists of 16x16 samples. Thus, by modifying only the LSB of every Luma pixel an embedding capacity of 256 bits can be achieved, which is the same as the size of the variable reserved for the binary representation. Finally, there was no need to use the two Chroma components, proving that the embedding capacity of the data hiding scheme proposed in [17], with a theoretical limit of 1,536 bits per macroblock can easily fulfill this application's requirements.

Since the proposed method requires control over IPCM macroblock production as well as independence from encoding parameters, the encoder was forced to treat specific macroblocks as IPCM. For our convenience, the third macroblock of the frame was selected every 15 frames. A simplified flowchart of the ENF-based data hiding scheme is depicted in Figure 1.

When a macroblock is forced to be treated as IPCM, the encoder makes an Application Programming Interface (API) call in a web application and receives the latest ENF stamp in JSON format. It parses the JSON file and stores the stamp in a stack. Prior to entropy coding, the encoder pops the stamp out of the stack and performs the embedding process. Using an API service is advantageous because only one edge device is needed to extract the ENF signal and update a reference

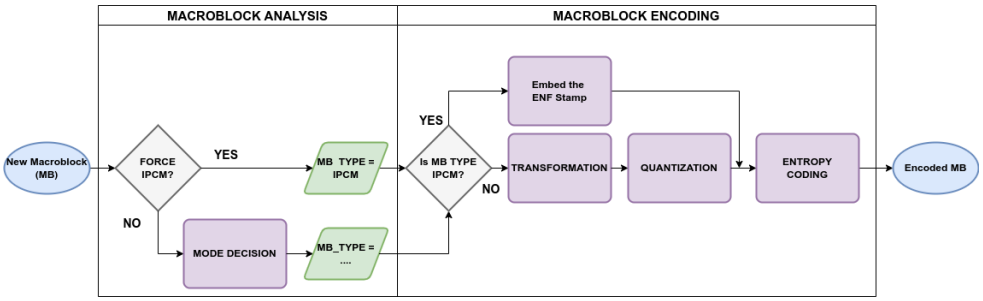

Fig. 1. A simplified flowchart of the proposed ENF-based data hiding scheme.

database, whereas all the other edge devices can simply request the most recent ENF stamp from the web application.

The encoded video stream now contains an ENF stamp which can be extracted while decoding the H.264 bitstream by reading the LSB of pixel values of the Luma component of the IPCM macroblock and converting them to their corresponding UTF-8 characters.

B. Supporting Framework

A Raspberry Pi 3 Model B, with the help of a sound card and an external power signal capture circuit, provides the one-second ENF estimations from the power grid and updates our reference database with the corresponding ENF stamps. A Google Drive folder is updated on midnight with the previous day's (24 hours) ENF estimations [21].

A python-based web application was developed with the Flask Framework [22] in order to visualise ENF estimations from the reference database in real-time and for implementing the API service.

III. Experiments and Results

A. Implementation

This data hiding scheme was implemented on the open source x264 software library. The software was compiled on both a Computer and a Raspberry Pi 3 Model B. The x264 library is used in many popular video editing and streaming tools. Source code can be found online on GitLab [23].

Three different sets of experiments were conducted, as detailed below. The meanings of Constant Rate Factor (CRF) mode, H.264 profiles and x264 presets are necessary for following the experiments. CRF is a constant quality mode that has direct relation to the HVS. HVS perceives more details from the static parts of a frame. By varying Quantization Parameter, the CRF discards less information for motionless parts of a video sequence resulting in a more stable bitrate for a certain amount of perceived quality. The higher the CRF, the lower the quality, ranging from 0 to 51. The profiles are defined by the H.264 standard; they define a set of coding functions that can be used by the encoder [20]. Finally, to control the trade-off between quality and encoding time, the x264 encoder introduces several presets.

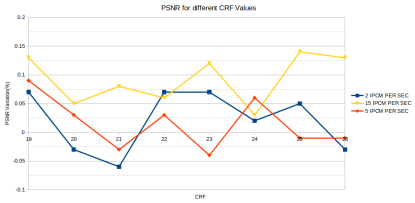

Fig. 2. PSNR variation between the reference and modified encoder for different CRF values and different watermark embedding rates.

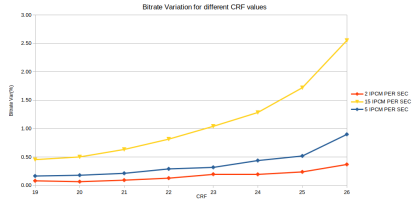

Fig. 3. Bitrate variation between the reference and modified encoder for different CRF values and different watermark embedding rates.

B. Performance Comparison

In the first set of experiments, we re-encoded one minute long video segments, e.g. VIRAT_S_000001.mp4 of the VIRAT dataset [24], in the default high profile in order to conduct an extensive comparison of the proposed and reference encoders' performance. The PSNR and bitrate variation between the video sequences produced by the two encoders for CRF values ranging from 19 to 26 while forcing 2, 5 and 15 IPCM macroblocks per second were calculated. PSNR remained almost unaffected from the use of the proposed encoder, according to Figure 2, while the maximum bitrate variation was $\approx 2.5\%$, (Figure 3), which occurs when we embed the ENF stamp 15 times per second with a CRF value of 26. As expected, these results indicate that the proposed and the original encoder perform identical in any given circumstances.

C. Integration to a Raspberry Pi

In order to use the proposed encoder on the Raspberry Pi we had to determine the appropriate encoding settings. The Raspberry Pi does not possess the computational power of a high-end computer, thus the *constrained baseline* profile was selected. We used the same one-minute long videos from the previous experiments, selected *superfast* and *ultrafast* presets and forced the encoder's output bitrate to a constant 1000, 1500 and 2000 kb/s through the whole re-encoding process.

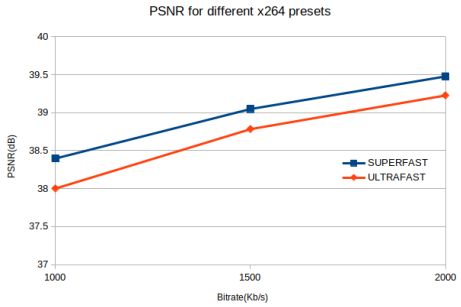

Fig. 4. PSNR calculation for fixed bitrates using the superfast and ultrafast presets.

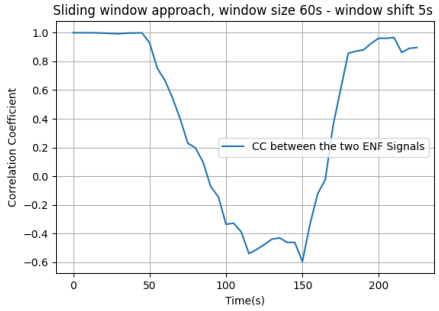

Fig. 5. CC obtained with the sliding window approach for a window size of 60s and a window shift of 5s

As can be seen in Figure 4, the PSNR value is in the range of 38 to 39.5 dB and can be nicely used for live video streaming. Actually, any value above 35 dB is considered visually very acceptable in watermarking / data hiding applications.

D. Proof-of-Concept Prototype

To test our assumptions, a proof-of-concept prototype was implemented. In our prototype, surveillance feed, captured by a USB webcam connected to a Raspberry Pi, was streamed to a computer on our LAN acting as the RCC, where both ENF signals from the video and our reference database were obtained in real time. For that purpose, the ffmpeg library [25] was modified for the watermark's extraction and also provided the local streaming solution. We implemented the scenario in which the surveillance feed of a fully compromised camera was interrupted and a previously recorded stream was inserted. The attack, which lasted 60 seconds, was launched after 120 seconds of normal surveillance operation and it was followed by another 120 seconds of normal surveillance operation.

A sliding window based approach, as described in [11], [12], with window sizes of 30 and 60 seconds and a window shift of 5 seconds, was used for evaluating the correlation coefficient (CC) between the two ENF signals. The CC has been a widely adopted similarity metric for ENF signals because the vertical offset that could occur in ENF estimations by different devices is neglected [26]. An initialization time equal to the length of the window and an additional 10 to 15 seconds for connection establishment between the two participants, are required.

In [11], the authors propose using a CC as high as 0.8

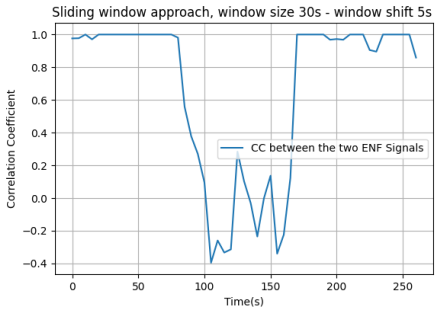

Fig. 6. CC obtained with the sliding window approach for a window size of 30s and a window shift of 5s.

as a threshold for FDA self-detection by the edge devices. According to the results we obtained, as seen in Figures 5 and 6, a higher threshold than 0.8 could be set. Additionally, due to the low complexity of the proposed approach, a fog node could be used for authenticating the surveillance feed from the cameras rather than consuming cloud's resources.

IV. Conclusions and Future Work

In this paper, a new method has been proposed for detecting FDA's on VSS's. The ENF signal, which is time-varying (stochastic) in nature and thus unique, is embedded to live video feeds at the edge, i.e. while video is coded and before it is stored or transmitted. The IPCM macroblocks of H.264 standard are exploited for hiding the ENF signal [17]. This kind of video protection gives the possibility of detecting whether a video stream has been compromised or not. In case that such an attack occurs, immediate actions can be taken. A number of expirementations have been performed in order to find the extend of the capacity of the proposed data hiding method. In the first set of experiments, an extensive comparison of the PSNR and bitrate variations for different data payloads, has been performed. It was observed that in all cases, the reference and the modified encoders performed similarly. The second set of experiments deals with video streaming based on a Raspberry Pi 3 Model B. By means of these tests the appropriate encoding settings for this particular edge device, were determined. Finally, the full prototype for streaming video using a Raspberry Pi across a LAN was tested. It was seen that the proposed method offers real-time detection of FDA's at a lower computational cost and with less hardware than the existing approaches, without depending on environmental or device-related factors.

Our future work will focus on incorporating a public key cryptographic scheme in various stages of the data hiding process, namely, the choice of the frame and the IPCM macroblock, as well as the ENF value itself. This will provide additional security to the proposed approach. We also consider introducing an IPCM macroblock selection mechanism so the encoding of a macroblock as IPCM will be based on a meaningful set of criteria. Finally, we would like to test our concept on a greater scale by incorporating more surveillance cameras and framework features such as auto-detection, phone notifications when abnormal activity is detected and a database for storing timestamps of frame intervals with a high possibility of being forged, for off line inspection.

REFERENCES

[1] P. Vennam, P. T. C., T. B. M., Y.-G. Kim, and P. K. B. N., "Attacks and preventive measures on video surveillance systems: A review," *Applied Sciences*, vol. 11, no. 12, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/12/5571

[2] M. Senthil, "CCTV surveillance system, attacks and design goals," *International Journal of Electrical and Computer Engineering*, vol. 8, 06 2018.

[3] J. Liranzo and T. Hayajneh, "Security and privacy issues affecting cloud-based ip camera," in *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, 2017, pp. 458–465.

[4] M. Guri, D. Bykhovsky, and Y. Elovici, "air-jumper: Covert air-gap exfiltration/infiltration via security cameras & infrared (ir)," *Computers & Security*, vol. 82, 09 2017.

[5] M. Guri, O. Hasson, G. Kedma, and Y. Elovici, "Visisploit: An optical covert-channel to leak data through an air-gap," 2016. [Online]. Available: https://arxiv.org/abs/1607.03946

[6] M. Hossain, "Framework for a cloud-based multimedia surveillance system," *International Journal of Distributed Sensor Networks*, vol. 2014, pp. 1–11, 05 2014.

[7] S. Prasad, P. Mahalakshmi, A. John, C. Sunder, and R. Swathi, "Smart surveillance monitoring system using Raspberry Pi and PIR sensor," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 6, pp. 7107–7109, 2014.

[8] F. Osuolale, "Secure data transfer over the internet using image cryptosteganography," *International Journal of Scientific and Engineering Research*, vol. 8, p. 1115, 12 2017.

[9] S. Fadl, Q. Han, and Q. Li, "Authentication of surveillance videos: Detecting frame duplication based on residual frame," *Journal of Forensic Sciences*, vol. 63, 10 2017.

[10] G. Ulutas, B. Ustubioglu, M. Ulutas, and V. Nabiyev, "Frame duplication/mirroring detection method with binary features," *IET Image Processing*, vol. 11, no. 5, pp. 333–342, 2017. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-ipr.2016.0321

[11] D. Nagothu, Y. Chen, E. Blasch, A. Aved, and S. Zhu, "Detecting malicious false frame injection attacks on surveillance systems at the edge using electrical network frequency signals," *Sensors*, vol. 19, no. 11, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/11/2424

[12] D. Nagothu, Y. Chen, A. Aved, and E. Blasch, "Authenticating video feeds using electric network frequency estimation at the edge," *EAI Endorsed Transactions on Security and Safety*, vol. 7, no. 24, 2 2021.

[13] C. Grigoras, "Digital audio recording analysis: the electric network frequency (enf) criterion," *International Journal of Speech, Language and the Law*, vol. 12, no. 1, p. 63–76, Feb. 2005. [Online]. Available: https://journal.equinoxpub.com/IJSLL/article/view/9977

[14] A. Hajj-Ahmad, C.-W. Wong, S. Gambino, Q. Zhu, M. Yu, and M. Wu, "Factors affecting enf capture in audio," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 277–288, 2019.

[15] Y. Shi, M. Qi, Y. Yi, M. Zhang, and J. Kong, "Object based dual watermarking for video authentication," *Optik*, vol. 124, no. 19, pp. 3827–3834, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0030402613001113

[16] S. Bouchama, L. Hamami, and H. Aliane, "AVC data hiding based on intra prediction modes for real-time applications," in *Proceedings of The World Congress on Engineering and Computer Science 2012*, vol. 1, 10 2012, pp. 655–659.

[17] S. K. Kapotas and A. N. Skodras, "Real time data hiding by exploiting the ipcm macroblocks in H.264/AVC streams," *Journal of Real-Time Image Processing*, vol. 4, no. 1, pp. 33–41, Mar 2009. [Online]. Available: https://doi.org/10.1007/s11554-008-0100-2

[18] Y. Raskar, "Secure data hiding in H.264/AVC10 video using intra pulse code modulation macro block," *International Journal of Computer Applications*, vol. 86, pp. 14–18, 01 2014.

[19] X. Yu, C. Wang, and X. Zhou, "A survey on robust video watermarking algorithms for copyright protection," *Applied Sciences*, vol. 8, no. 10, 2018. [Online]. Available: https://www.mdpi.com/2076-3417/8/10/1891

[20] I. E. Richardson, *The H.264 Advanced Video Compression Standard*, 2nd ed. Wiley Publishing, 2010.

[21] "DSIP Lab (Digital Signal and Image Processing Laboratory): Reference ENF Recordings for Mainland Greece." [Online]. Available: https://drive.google.com/drive/folders/1EzklfJ_NdvOnKQpK3Q6ez9bgiwWpPKoq?usp=sharing

[22] "Flask micro web framework." [Online]. Available: https://palletsprojects.com/p/flask/

[23] VideoLAN, "x264 - official page." [Online]. Available: https://www.videolan.org/developers/x264.html

[24] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. K. Aggarwal, H. Lee, L. Davis, E. Swears, X. Wang, Q. Ji, K. Reddy, M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, B. Song, A. Fong, A. Roy-Chowdhury, and M. Desai, "A large-scale benchmark dataset for event recognition in surveillance video," in *CVPR 2011*, 2011, pp. 3153–3160.

[25] "FFmpeg official website." [Online]. Available: https://ffmpeg.org/

[26] M. Huijbregtse and Z. Geradts, "Using the enf criterion for determining the time of recording of short digital audio recordings," in *Computational Forensics*, Z. J. M. H. Geradts, K. Y. Franke, and C. J. Veenman, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 116–124.