8-3-2015

A Method to Detect AAC Audio Forgery

Qingzhong Liu
Sam Houston State University

Andrew Sung
University of Southern Mississippi

Lei Chen
Georgia Southern University, lchen@georgiasouthern.edu

Ming Yang
Kennesaw State University

Zhongxue Chen
Indiana University

See next page for additional authors

Recommended Citation

Liu, Qingzhong, Andrew Sung, Lei Chen, Ming Yang, Zhongxue Chen, Yanxin Liu, Jing Zhang. 2015. "A Method to Detect AAC Audio Forgery." *EAI Endorsed Transactions on Security and Safety*, 15 (6): 1-7. doi: 10.4108/icst.mobimedia.2015.259141 source: https://eudl.eu/doi/10.4108/icst.mobimedia.2015.259141
https://digitalcommons.georgiasouthern.edu/information-tech-facpubs/39

Authors

Qingzhong Liu, Andrew Sung, Lei Chen, Ming Yang, Zhongxue Chen, Yanxin Liu, and Jing Zhang

A Method to Detect AAC Audio Forgery

Qingzhong Liu
Department of Computer Science
Sam Houston State University
Huntsville, TX 77341, USA
liu@shsu.edu

Andrew H. Sung
School of Computing
University of Southern Mississippi
Hattiesburg, MS 39406, USA
andrew.sung@usm.edu

Lei Chen
Department of Computer Science
Sam Houston State University
Huntsville, TX 77341, USA
chen@shsu.edu

Ming Yang
Department of Information Technology
Kennesaw State University
Marietta, GA 30060, USA
myang8@kennesaw.edu

Yanxin Liu
Department of Computer Science
Sam Houston State University
Huntsville, TX 77341, USA
yanxin@shsu.edu

Zhongxue Chen
School of Public Health
Indiana University Bloomington
Bloomington, IN 47045, USA
zc3@indiana.edu

Jing Zhang
School of Electronic Information
Engineering, Tianjin University
Tianjin, 200072, China
zhangjing@tju.edu.cn

ABSTRACT

Advanced Audio Coding (AAC), a standardized lossy compression scheme for digital audio, which was designed to be the successor of the MP3 format, generally achieves better sound quality than MP3 at similar bit rates. While AAC is also the default or standard audio format for many devices and AAC audio files may be presented as important digital evidences, the authentication of the audio files is highly needed but relatively missing. In this paper, we propose a scheme to expose tampered AAC audio streams that are encoded at the same encoding bit-rate. Specifically, we design a shift-recompression based method to retrieve the differential features between the re-encoded audio stream at each shifting and original audio stream, learning classifier is employed to recognize different patterns of differential features of the doctored forgery files and original (untouched) audio files. Experimental results show that our approach is very promising and effective to detect the forgery of the same encoding bit-rate on AAC audio streams. Our study also shows that shift recompression-based differential analysis is very effective for detection of the MP3 forgery at the same bit rate.

Categories and Subject Descriptors

H.5.1 [Multimedia Information Systems]: Audio input/output; K.6.m [Miscellaneous]: Insurance and Security

General Terms

Algorithms and Security.

MOBIMEDIA 2015, May 25-27, Chengdu, People's Republic of China

DOI 10.4108/icst.mobimedia.2015.259141

Keywords

Forgery detection; audio forensics; AAC; same bitrate.

1. INTRODUCTION

In multimedia forensics, steganalysis and forgery detection are two interesting areas with broad impact to each other. While multiple promising and well-designed steganalysis methods have been proposed and several steganographic systems have been successfully steg-analyzed [3, 5, 6, 7, 10, 11, 12, 15], it seems that the advance in forgery detection falls behind.

While we enjoy huge volumes of digital multimedia, our traditional confidence in the integrity via our eyes and ears has also been undermined since doctored pictures, video clips, and audio streams are easily manipulated. Generally, tampering manipulation in digital media involves several basic operations and the detection of these fundamental manipulations and relevant forgery has been well studied [1, 2, 4, 8, 9, 13, 14, 16]. To our knowledge, most study of multimedia forgery detection is focused on digital images.

Some works have been presented to detect the forgery or related manipulation in audio streams, including MPEG-1 Audio Layer 3, also known as MP3. For example, if two MP3 audio streams encoded at different bit-rates are selected in part and composited together and encoded in MP3 format, such forgery manipulation undergoes double MP3 compression. While we will be able to reveal the behavior of double MP3 compression, we may catch the forged part in MP3 audio streams. However, if two MP3 audio streams encoded at the same bit-rate and composited together and encoded in MP3 format with the same bit-rate, the method of detecting double MP3 compression does not work [15].

Advanced Audio Coding (AAC), a lossy audio compression scheme, standardized by ISO and IEC, which was designed to be the successor of the MP3 format, generally obtains better sound quality than MP3 at similar bit rates. AAC is supported on iPhone, iPod, iPad, Nintendo DSi, iTunes, DivX Plus Web Player, PlayStation 3, PlayStateion Portable, Wii, Sony Walkman MP3, Sony Ericsson, Nokia, Android, Blackberry, and webOS-based mobile phones [18]. While AAC audio files widely spread, to our knowledge, the literature of the forgery detection of AAC audio files is still missing to this date. Inspired by the method to detect MP3 forgery by checking offset [16] and the method to detect misaligned recompression-based forgery in JPEG images [13], in this paper, we propose a scheme to detect the forgery with the same bit-rate in AAC audio streams, by designing a shift-recompression-based differential analysis with learning classifier. We describe background and relevant work in section 2, and propose our detection method in section 3. Experiments are presented in section 4, followed by our conclusion in section 5.

2. BACKGROUND AND AAC TAMPERING

AAC is a standardized, lossy digital audio compression scheme. It was developed with the cooperation and contributions of companies mainly including Dolby, Fraunhofer (FhG), AT&T, Sony and Nokia, and was officially declared an international standard by the Moving Pictures Experts Group in April of 1997. AAC was promoted as the successor to MP3 for audio coding at medium to high bitrates. It follows the same basic coding paradigm as Layer-3 including high frequency resolution filter bank, non-uniform quantization, Huffman coding, iteration loop structure using analysis by-synthesis but improves on Layer-3 in a lot of details and uses new coding tools for improved quality at low bit-rates. Its popularity is currently maintained by it being the default iTunes codec, the media player which powers iPod, the most popular digital audio player on the market. Furthermore, the iTunes Music Store, whose sales account for 85% of the market for legal online downloads, sells AAC-encoded songs (encapsulated with FairPlay Digital Rights Management)

Compared to MP3, AAC improves the following aspects:

1) More sample frequencies (from 8 kHz to 96 kHz) than MP3 (16 kHz to 48 kHz);

2) Up to 48 channels (MP3 supports up to two channels in MPEG-1 mode and up to 5.1 channels in MPEG-2 mode);

3) Arbitrary bit-rates and variable frame length. Standardized constant bit rate with bit reservoir;

4) Higher efficiency and simpler filter bank (rather than MP3's hybrid coding, AAC utilizes a pure MDCT);

5) Higher coding efficiency for stationary signals (AAC uses a block size of 1024 or 960 samples, allowing more efficient coding than MP3's 576 sample blocks);

6) Higher coding accuracy for transient signals (AAC uses a block size of 128 or 120 samples, allowing more accurate coding than MP3's 192 sample blocks);

7) Can use Kaiser-Bessel derived window function to eliminate spectral leakage at the expense of widening the main lobe;

8) Much better handling of audio frequencies above 16 kHz;

9) More flexible joint stereo (different methods can be used in different frequency ranges);

10) Adds additional modules (tools) to increase compression efficiency: temporal noise shaping (TNS) [20], Backwards Prediction, Perceptual Noise Substitution (PNS) [21], etc.

11) Improved Huffman Coding: In AAC, coding by quadruples of frequency lines applied more often. In addition, the assignment of Huffman code tables to coder partitions can be much more flexible.

Figure 1 shows the general processing flow in MPEG-2 AAC encoding.

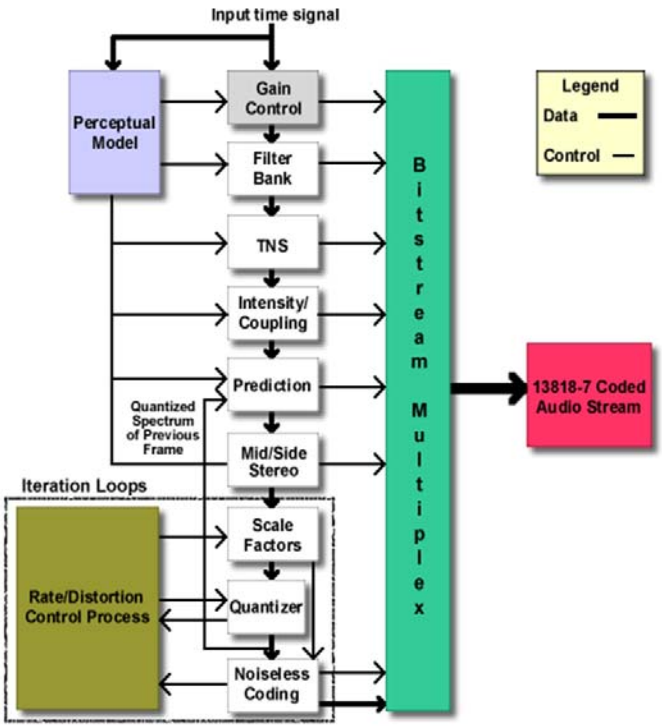


Figure 1. AAC flowing chart [22]

Regarding AAC audio tampering, generally an original AAC audio signal is decoded into temporal domain, the doctoring occurs at the temporal domain by inserting or removing some voice signals, and the modified audio signal is encoded to AAC audio format. If the bitrate of inserted audio signal fragment is different from the bitrate of original AAC audio signal; or the bitrate used for encoding modified audio signal is different from the bitrate of original AAC audio signal, the tampering may be exposed by revealing the different bitrates once used in the control process. However, if the bitrates are the same, the detection becomes difficult.

Our task targets on the detection of the AAC audio forgery encoded at the same bitrate used for the original AAC audio stream.

3. PROPOSED DETECTION METHOD

3.1 Related Study in Image Forgery Detection

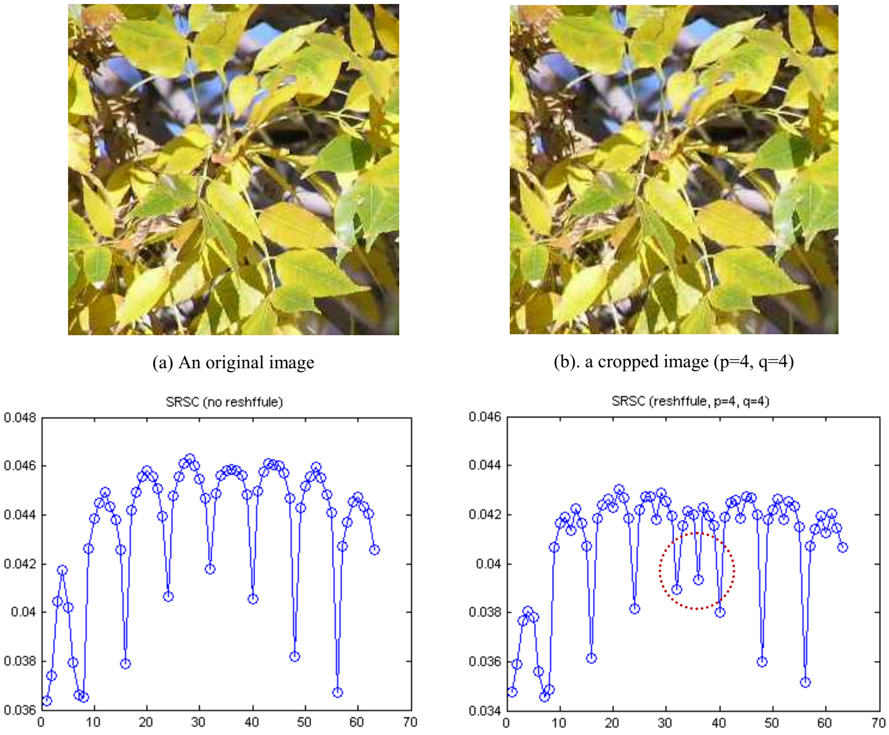
Our proposed approach is based on our previous work in detecting image forgery, therefore, we described the feature extraction for image forgery detection below [13].

1) Decode an JPEG image under examination to spatial domain, which is denoted by matrix S(i,j) (i=1, 2, …, M; j = 1, 2 …, N);
2) Shift the matrix S(i,j) by d_1 rows and d_2 columns in the spatial domain, $(d_1, d_2) \in \{(0,1), \ldots, (0,7), (1,0), \ldots, (7,7)\}$ and generate a shifted spatial image S′(d_1, d_2), S′(d_1, d_2) = S(i- d_1, j- d_2)(i= d_1+1, d_1+2, …, M; j= d_2+1, q+ d_2, …, N), then compress the shifted spatial image S′(d_1, d_2) to JPEG format at the same quantization matrix;
3) Decode the shifted JPEG image to spatial domain, denoted by a matrix S′′(d_1, d_2);
4) Calculate the difference D (d_1, d_2)= S′(d_1, d_2) - S′′(d_1, d_2);
5) **S**hift-recompression based **ReS**huffle **C**haracteristic features (**SRSC**) on the region of interest R, **SRSC***R* are defined by:

$$SRSC_R(d_1,d_2) = \frac{\sum \left| D_R(d_1,d_2) \right|}{\sum \left| S'_R(d_1,d_2) \right|}, \qquad (1)$$

Where $(d_1,d_2) \in \{(0,1),\ldots,(0,7),(1,0),\ldots,(7,7)\}$.There are a total of 63 features for each R.

If an image was cropped under the misalignment by p rows and q columns, mod (p, 8) ≠ 0 or mod (q, 8) ≠0, $0 \le p \le 8$, $0 \le q \le 8$, and then recompressed at the same quantization matrix to the original JPEG image, we expect that the SRSC features will be distinct due to the misalignment, and the values of p and q can be determined by the SRSC features. Figure 2 shows an original JPEG image (a) and a cropped image with misalignment p=4 and q=4 (b). The SRSC features from original image and the misaligned recompressed image (p=4, q=4), respectively, are shown in (c), and (d). The circle highlights the major differences of SRSC features between the original and manipulated image. It shows that SRSC features may be effective in exposing the misaligned operations that are encoded with the same quantization matrix.



(a) An original image

(b). a cropped image (p=4, q=4)

(c) SRSC (original image)

(e) SRSC(cropped, p=4, q=4)

Figure 2. A comparison of SRSC features from an original JPEG image and three cropped JPEG images. X-label shows the SRSC feature index and y-label indicates the value [13].

3.2 Proposed Approach to Detecting AAC Audio Forgery

As pointed out in [13], JPEG compression generally generates block artifacts. We surmise that similar to JPEG compression, AAC audio compression also introduces block (frame) artifacts. While two AAC audio files are manipulated together, or some part is removed from an AAC audio file in temporal domain, and doctored audio data are re-encoded in AAC format at the same bit rate, the original block artifacts are generally undermined, in other words, original block switching structure will be reshuffled with a part of the neighbor blocks. By revealing such reshuffling manipulation, we may locate the doctored areas in the AAC audio forgery that was encoded at the same bit rate. According to our previous work in image forgery detection [13], we propose a shift-recompression-based differential analysis to detect the forgery in AAC audio streams with the same compression bit rate, described as follows.

Shift-Recompression-based Differential Analysis Algorithm

1. *Decode the examined AAC audio stream to temporal domain, denoted by a matrix S(i,j) (i=0,1, 2, …, M; j indicates the number of channel of the audio signal);*
2. *For AAC stationary audio signal, each frame contains 1024 time-domain samples. Allegedly, each frame stands alone and does not depend on previous frames (whereas many perceptual audio codecs such MP3 overlap data with the previous frame).*
3. *For t=1:1023*
    - a) *Shift the matrix S(i,j) by t samples in the temporal domain (t=1,2,…1023). A shifted temporal WAV signal S′(i, j, t) is produced. S′(i, j, t) = S(i-t, j), i= t, t+1, t+2, …, M;*
    - b) *Encode the shifted temporal signal S′(i, j, t) to AAC format at the same bit rate;*
    - c) *Decode the encoded audio signal from the above step to temporal domain, denoted by S′′(i, j, t);*
    - d) *Calculate the difference D(i, j, t)= S′(i, j, t) - S′′(i, j, t);*
    - e) *Shift-recompression based reshuffle characteristic features (SRSC) are given by:*

$$SRSC(t) = \frac{\sum_{(i,j)} \left| D(i,j,t) \right|}{\sum_{(i,j)} \left| s'(i,j,t) \right|} \tag{2}$$

We obtain 1023 SRSC features for a stationary AAC audio file since t = 1, 2, ..., 1023.

While AAC audio stream are tampered in temporal domain and original frame structures are generally broken, by checking the SRSC feature under each different shift-recompression, we surmise that untouched SRSC features and tampered SRSC features are different, especially at the corresponding shift. As a result, the manipulation may be exposed.

4. EXPERIMENTS

4.1 Detection of Cropping and Recompression

To verify our proposed shift-recompression-based differential analysis, we select 1000 never compressed WAV files; each file is in the length of 20 seconds. These WAV files are compressed in AAC format by using FAAC encoder, which is based on the original ISO MPEG reference code [19]. To simulate the shift-recompression of AAC audio forgery manipulation, AAC audio files are decoded into temporal domain and cropped by different samples at the beginning of the audio signals, then re-encoded in AAC format at the same bit rate. In our study, we tried to produce the cropping database at each possible cropping, or the number of samples removed is set from 1 to 1023, however, the time-consuming is too high to complete. Therefore, in our experiments, the numbers of the samples cropped are only set as 5, 50, 200, 400, 480, 512, 750, 900, and 1000, respectively. 1023 SRSC features are extracted from 1000 untouched AAC audio files and from the nine categories of doctored AAC audio files. Figure 3 shows the SRSC features extracted from an untouched AAC audio file and from the cropping by 50 samples and the cropping by 900 samples individually and recompressed versions.

We apply a popular SVM technique LibSVM [17] with a linear kernel for training and testing. One hundred experiments are performed for training and testing. In each experiment, 60% feature sets from each category are randomly selected for training and the remainders are used for testing. The mean testing of the confusion matrix over 100 experiments are shown by Table 1.

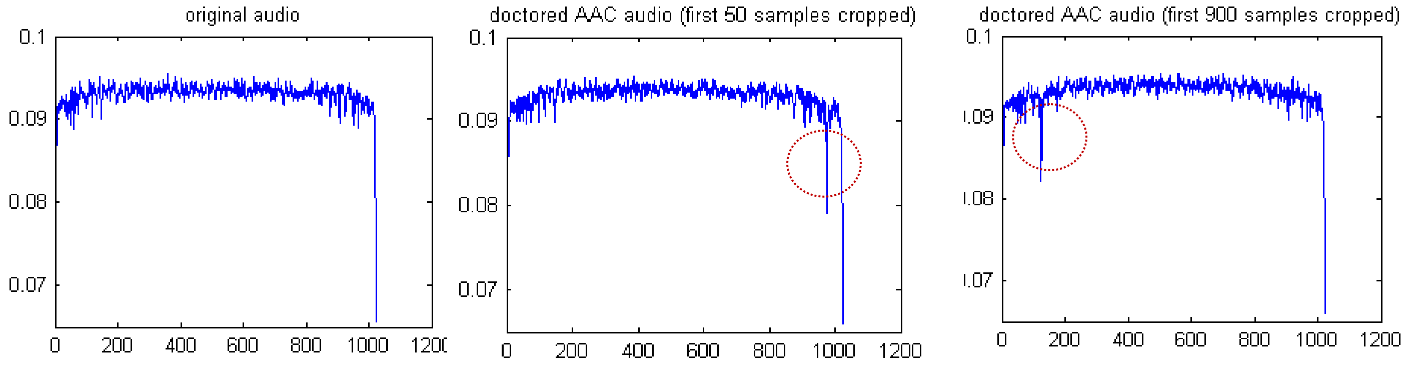


Figure 3. SRSC features of original AAC audio (a) and the AAC audio once cropped by 50 samples (b) and by 900 samples (c)

Table 1. Confusion matrix on testing sets (mean values, %) by using LibSVM with linear kernel over 100 experiments.

Truth \ Prediction		untouched	Manipulation (cropped by)								
			5	50	200	400	480	512	750	900	1000
untouched		**99.4**	0.2	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0
Manipulation (Cropped by)	5	3.4	**96.6**	0	0	0	0	0	0	0	0
	50	1.8	0	**98.2**	0	0	0	0	0	0	0
	200	2.1	0	0	**97.9**	0	0	0	0	0	0
	400	1.5	0	0	0	**98.6**	0	0	0	0	0
	480	1.2	0	0	0	0	**98.8**	0	0	0	0
	512	2.3	0	0	0	0	0	**97.7**	0	0	0
	750	2.0	0	0	0	0	0	0	**98.1**	0	0
	900	2.4	0	0	0	0	0	0	0	**97.7**	0
	1000	2.2	0	0	0	0	0	0	0	0	**97.9**

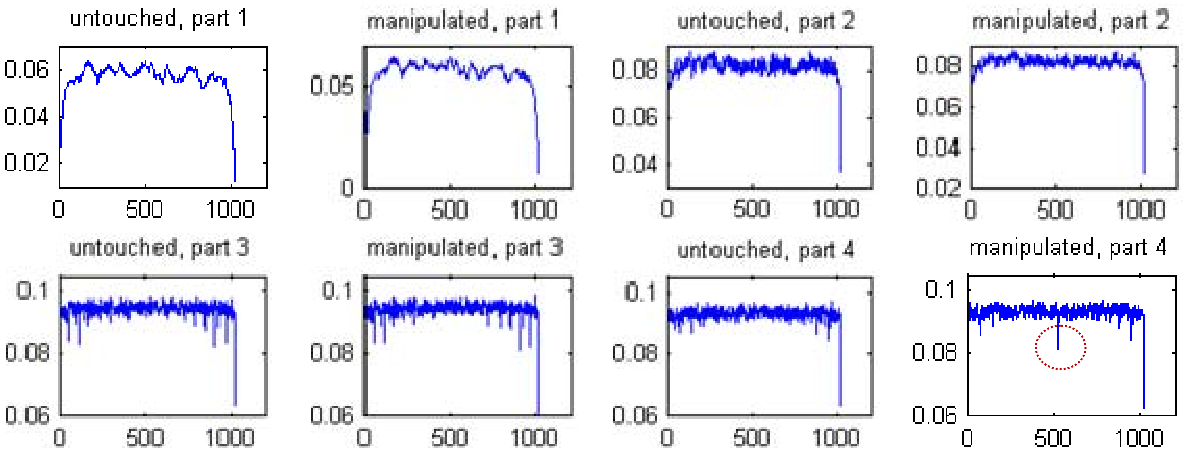
4.2 Detection of AAC Forgery

In this type of experiments, we randomly select 200 AAC audio files, and remove a few audio samples in the middle, with the block switch offset by 100, 300, 500, 700, and 900 samples, then encode the doctored audio signals into AAC format at the same bit rate. There are total of 1000 doctored AAC audio files. We apply shift-recompression-based differential analysis to each audio file (including untouched and doctored audio files), each audio file is equally divided into six segments, as a result, 1200 untouched segments and 3000 touched segments are obtained. SRSC features are extracted from each segment, in order to discriminate the doctored audio files from untouched files, and identify the doctored areas. The experimental design is the same to the process described in III.A. Table 2 shows the confusion matrix with the experimental results over 100 experiments.

Figure 4 shows the SRSC features extracted from the six parts of an original AAC audio file (first row) and from the six part of the doctored AAC audio file with the forgery taking place on the middle. The comparison show that the first three parts of the original audio and doctored audio are similar, but the pattern of the SRSC features from the last three parts are different, which approximately reveals the forged area in doctored AAC audio stream in the middle.

Table 2. Confusion matrix on testing sets (mean values, %) by using LibSVM with linear kernel over 100 experiments.

Truth \ Prediction		untouched	forgery (shifted by)				
			100	300	500	700	900
untouched		**98.8**	0.2	0.1	0.2	0.3	0.3
forgery (shifted by)	100	1.2	**98.7**	0	0	0	0.0
	300	0.8	0.1	**99.1**	0	0	0.0
	500	0.6	0	0.0	**99.4**	0.0	0.0
	700	0.8	0	0.0	0	**99.2**	0.0
	900	1.2	0.2	0.0	0.0	0	**98.6**



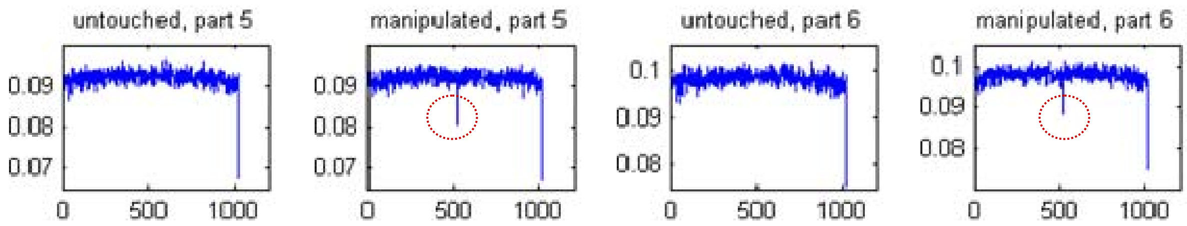
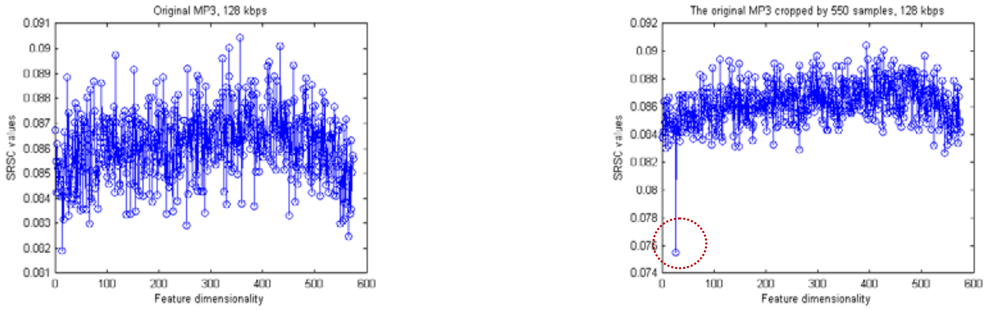


Figure 4. The SRSC features extracted from an original AAC audio file and from manipulated AAC audio file (doctoring in the middle).



(a) Original MP3 (b). MP3 cropped by 550 samples and encoded at the same bit rate

Figure 5. SRSC features of original (a) and manipulated MP3 files (b)

4.3 Discussion and Future Study

Although our experiments presented above do not examine all possibility of AAC audio forgery due to the very high computational cost (it is very time-consuming to examine of all possible forgery shifted by 1023 positions), by simulating part cases of AAC audio forgery at a reasonable computational cost, our experimental results do verify the effectiveness of our proposed shift-recompression-based approach to detecting AAC audio forgery of the same bit rate. The detection accuracy is very promising.

It is worth noting that shift-recompression-based method is effective not only for detecting AAC audio forgery, but also for detecting MP3 audio forgery on the same bit rate. Figure 5 shows an example of SRSC features extracted from an untouched MP3 audio file and a doctored MP3 file. The difference is that each MP3 frame contains 576 time-domain samples, therefore, the feature set only consists of 575 features, not 1023 features.

Our previous studys shows that image complexity and audio signal complexy [11, 15, 23] may play important role in evaluating the detection performance. In the future, we will exam the detection peformance under different audio signal complexity, and explore new audio forgery detection methods at a lower computational cost.

5. CONCLUSIONS

There was no literature before targeting on the forgery detection of AAC audio streams on the same compression bit-rate. In this paper, we propose a shift-recompression-based differential analysis to detecting AAC audio forgery with learning classifiers. Although our method is pretty straightforward, experimental results show that our approach is quite promising and effective.

ACKNOWLEDGMENTS

The support for this study from the US National Institute of Justice under the award No. 2010-DN-BX-K223 and from the US National Science Foundation under the award CCF-1318688 is greatly appreciated.

REFERENCES

[1] Chen, Y. H. and Huang, H. C. 2015. Coevolutionary genetic watermarking for owner identification. *Neural Computing and Applications*, 26(12):291-298.

[2] Farid, H. 2009. Image forgery detection, a survey. *IEEE Signal Processing Magazine*, 2(26):16-25.

[3] Fridrich, J. and Kodovsky, J. 2012. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3): 868-882.

[4] Li, L., Li, S., Zhu, H., Chu, S. C., Roddick, J. F. and Pan, J. S. 2013. An efficient scheme for detecting copy-move forged images by local binary patterns. *Journal of Information Hiding and Multimedia Signal Processing*, 4(1):46-56

[5] Liu, Q., Sung, A. H. and Qiao, M. 2009. Improved detection and evaluation for JPEG steganalysis. *Proc. 17th ACM Multimedia*, pages 873--876.

[6] Liu, Q., Sung, A. H. and Qiao, M. 2009. Novel stream mining for audio steganalysis. *Proc. 17th ACM Multimedia*, pages 95-104.

[7] Liu, Q., Sung, A. H. and Qiao, M. 2009. Temporal derivative based spectrum and mel-cepstrum audio steganalysis. *IEEE Transactions on Information Forensics and Security*, 4(3): 359-368.

[8] Liu, Q. and A. H. Sung 2009. A new approach for JPEG resize and image splicing detection. *Proc. ACM Multimedia Workshop on Multimedia in Forensics 2009*, pp. 43-47.

[9] Liu, Q., Sung, A. H. and Qiao, M. 2011. A method to detect JPEG-based double compression. In *Proc. of 8th International Symposium on Neural Networks*, pp 466-476.

[10] Liu, Q., Sung, A. H. and Qiao, M. 2011. Neighboring joint density based JPEG steganalysis. *ACM Transactions on Intelligent Systems and Technology*, 2(2), 16:1-16.

[11] Liu, Q., Sung, A. H. and Qiao, M. 2011. Derivative based audio steganalysis. *ACM Transactions on Multimedia Computing, Communications and Application*, 7(3), 18:1-19.

[12] Liu, Q. 2011. Steganalysis of DCT-embedding-based Adaptive Steganography and YASS. *Proc. 13th ACM Workshop on Multimedia and Security*, pp. 76-85.

[13] Liu, Q. 2011. Detection of misaligned cropping and recompression with the same quantization matrix and relevant forgery. *Proc. 3rd ACM Workshop on Multimedia in Forensics and Intelligence*, pp. 25-30.

[14] Qiao, M., Sung, A. H. and Liu, Q. 2010. Revealing real quality of double compressed MP3 audio. *Proc. 18th ACM Multimedia*, pp. 1011-1014.

[15] Qiao, M., Sung, A. H. and Liu, Q. 2013. MP3 audio steganalysis. *Information Sciences*, vol. 231, pages 123-134.

[16] Yang, R., Qu, Z. and Huang, J. 2008. Detecting digital audio forgeries by checking frame offsets. *Proc. 10th ACM Workshop on Multimedia and Security*, pages: 21-26.

[17] Chang, C-C and Lin, C-J. 2011. LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[18] http://en.wikipedia.org/wiki/Advanced_Audio_Coding

[19] http://www.audiocoding.com/faac.html

[20] Herre, J. and Johnston, J. D. 1996. Enhancing the performance of perceptual audio coders by using temporal noise shaping. AES 101st Convention, no. preprint 4384.

[21] Herre, J. and Schulz, D. 1998. Extending the MPEG-4 AAC codec by perceptual noise substitution. AES preprint 4720.

[22] http://www.omninerd.com/articles/How_Audio_CompressioC_Works/print_friendly, accessed on 12 April 2015.

[23] Liu, Q., Sung, A. H., Chen, Z. and Xu, J. 2008. Feature mining and pattern classification for steganalysis of LSB matching steganography in grayscale images. *Pattern Recognition*, 41(1): 56-66.