VoIP Forgery Detection

Satish Tummala, Yanxin Liu and Qingzhong Liu
Department of Computer Science
Sam Houston State University
Huntsville, TX, USA
Emails: sct137@shsu.edu; yanxin@shsu.edu; liu@shsu.edu

Abstract- With the rapid increase in low-cost and sophisticated digital technology the need for techniques to authenticate digital material will become more urgent. Inspired by the image forgery detection, we develop a method to detect the forgery in VoIP audio streams by checking the offset differential features with learning machine. Our experimental results on Speex VoIP streams show that our approach is promising to expose the forgery manipulation in VoIP audio streams.

Keywords- Multimedia forensics, forgery detection, SVM classifier, speex, VoIP

1. Introduction

Recently, there has been a tremendous increase in the use of VoIP applications. In some cases, they may be submitted as digital evidence. If these files are being submitted as digital evidence, then there is a need to authenticate such material. In order to authenticate such material, there is an urgent need to develop authentication methods for VoIP files. Digital watermarking and signature are the two typical technologies for digital multimedia authentication. These two techniques need some side information such as a signature or digital watermark at the time of detection. Since, it is impossible to retrieve any available side information from the questionable data; these methods are useless in many real applications.

In the last several years, multimedia forensics has been an emerging research field of information security, because it doesn’t need any side information like digital watermark or signature during detection. What it needs is the features within the multimedia. These features can further be analyzed so as to provide forensics information on how this data is acquired and processed.

The paper is organized as follows. First, a brief overview of VoIP and the tools used in this project has been presented. Next to it, the procedure of creating a forgery database of VoIP has been described. After that, the feature extraction module and the machine learning module have been explained. The experimental results have been discussed. Finally, a conclusion along with future works has been presented.

2. Related Work

There are few works on authentication for digital audio. A technique for detecting digital audio forgeries by checking frame offsets [1, 16] has been proposed, based on the assumption that forgeries lead to the broken frame grids. Hany Farid [2] used bispectral analysis to detect digital forgery in speech signals, based on an idea that forgery in speech would introduce unnatural correlations. Recently, the detection of doubly compressed MP3 audio streams has been in-depth investigated [17, 18].

Grigoras [3] reported that the Electronic Network Frequency Criterion can be used as a means of assessing the integrity of digital audio evidence. It could be used to verify the exact time when a digital audio was created. This could be done by comparing the ENF of audio recording with a reference frequency database from the electric company or the laboratory. Dittmann et al. [4] proposed that the authenticity of the speaker’s environment could be determined by extracting the background features of an audio stream. These features provide information for determining its origin location and the used microphone.

There are still no passive authentication methods focusing especially on VoIP. This issue needs to be addressed, because a lot of applications now-a-days are based on VoIP. The audio forgery detection methods described above cannot be directly applied to VoIP, because there is a vast difference in the process of encoding between audio and VoIP. But the ideas of these methods can be improved so as to apply them to VoIP files.

3. Speex VoIP Codec

Speex is an open source audio compression format, based on CELP and is designed to compress voice at bitrates ranging from 2 to 44 kbps. [9]. It is part of the GNU Project [10] and is available under the revised BSD license [11]. The best part of using speex is that it is free compared to some other expensive proprietary speech codecs. It has a lot of useful features that are not present in many other speech codecs, which makes it well adapted to internet applications especially VoIP. Some of Speex’s features include:

- Narrowband (8 kHz), wideband (16 kHz), and ultra-wideband (32 kHz) compression in the same bitstream
- Intensity stereo encoding
- Packet loss concealment
- Variable bitrate operation (VBR)
- Voice Activity Detection (VAD)
- Discontinuous Transmission (DTX)
- Fixed-point port
- Acoustic echo canceller
- Noise suppression

Speex is based on CELP (Code Excited Linear Prediction) [12]. The techniques are based on the following ideas:

1. Use of a Linear Prediction Model
2. Use of codebook entries as input of the Linear Prediction Model
3. Search performed in a closed loop in a perceptually weighted domain

CELP also utilizes the source-filter model for speech prediction. It assumes that the vocal cords are the source of spectrally flat sound, and that the vocal tract acts as a filter to spectrally shape the various sounds of speech i.e., the source and filter are totally independent of each other. This model is mainly used because of its simplicity. Also, this model is usually tied with the use of linear prediction, illustrated by Figure 1.

![Figure 1. Illustration of CELP](image)

In what follows we briefly describe the techniques used in CELP.

**Linear Prediction (LPC)**

It is at the base of CELP. The basic idea behind it is that it predicts the signal \( x[n] \) using a linear combination of its past samples.

\[
y[n] = \sum_{i=1}^{N} a_i x[n - i]
\]

where \( y[n] \) is the linear prediction of \( x[n] \). The prediction error is thus given by:

\[
e[n] = x[n] - y[n] = x[n] - \sum_{i=1}^{N} a_i x[n - i]
\]

The goal of the LPC analysis is to find the best prediction coefficients \( a_i \) which minimizes the quadratic error function.

\[
E = \sum_{n=0}^{L-1} [e[n]]^2 = \sum_{n=0}^{L-1} [x[n] - \sum_{i=1}^{N} a_i x[n - i]]^2
\]

That can be done by making all derivatives equal to zero

\[
\frac{\partial E}{\partial a_i} = \frac{\partial}{\partial a_i} \sum_{n=0}^{L-1} [x[n] - \sum_{i=1}^{N} a_i x[n - i]]^2 = 0
\]

For an order N filter, the filter coefficients \( a_i \) are found by solving the system \( N \times N \) linear system \( Ra = r \), where

\[
R = \begin{bmatrix}
R(0) & R(1) & \cdots & R(N-1) \\
R(1) & R(0) & \cdots & R(N-2) \\
\vdots & \vdots & \ddots & \vdots \\
R(N-1) & R(N-2) & \cdots & R(0)
\end{bmatrix}
\]

with \( R(m) \), the auto-correlation of the signal \( x[n] \), computed as

\[
R(m) = \sum_{i=0}^{N-1} x[i] x[i - m]
\]

**Pitch Prediction**

Pitch prediction is used in most speech codecs. Here, we find a period (because the signal is periodic) that looks similar to the current frame i.e., the pitch predictor look for similar patterns outside the current frame. The pitch period is encoded along with a prediction gain.

\[
e[n] \equiv p[n] = \beta e[n - T]
\]

where \( T \) is the pitch period, \( \beta \) is the pitch gain.

**Innovation Codebook**

The final excitation will be the sum of the pitch prediction and an innovation signal taken from a fixed codebook, hence the name Code Excited Linear Prediction.
Noise Weighting

Most of the modern speech codecs shape the noise so that it appears mostly in the frequency regions where the ear cannot detect it. In order to maximize the speech quality, CELP codecs minimize the mean square error in the perceptually weighted domain.

\[ e[n] = p[n] + c[n] = \beta e[n-1] + c[n] \]

Analysis-by-Synthesis

This principle means that the encoding is performed by perceptually optimizing the decoded signal in a closed loop. The best CELP stream would be produced by trying all possible bit combinations and selecting the one that produces the best-sounded decoding signal.

4. Speex VoIP Tampering

We have created a VoIP audio database containing 1000 files by using Audacity [13] and speex [9]. While we create the forgery, we decode the VoIP file into temporal domain, and manipulate the file in temporal domain, and then encoded the doctored file to speex format. Here we show an original file and the tampering, shown in Figures 3 and 4.

The original audio file contains the voice, “A Rose is a woody perennial of the genus Rosa, within the family Rosaceae. There are over 100 species. They form a group of erect shrubs, and climbing or trailing plants, with stems that are often armed with sharp prickles. Flowers are large and showy, in colors ranging from white through yellows and reds.”

The doctored audio file means: “A Rose is a woody perennial of the genus Rosa, within the family Rosaceae. There are over 100 species. They form a group of erect shrubs or trailing plants, with stems that are often armed with sharp prickles. Flowers are large and showy, in colors ranging from white through yellows and reds.”

5. VoIP Forgery Detection

5.1. Feature Mining Based on Shift-Recompression

To detect the VoIP forgery, inspired by the previous work in detecting image forgery and image steganography in JPEG format [5, 6, 7, 8], we design an algorithm to extract the differential features based on shift recompression, described as follows:

**Shift-Recompression-based Differential Feature Extraction**

1. Decode the examined VoIP audio stream to temporal domain, denoted by a vector \( S(i) \) \( (i=0, 1, 2, ..., M) \);

2. Shift the matrix \( S(i) \) by \( t \) samples in the temporal domain, \( t \in \{1, 2, ..., N-1\} \), here \( N \) stands for the number of samples in a frame/block. For speex VoIP audio signal, a block consists of 160 samples \( (N = 160) \). A shifted temporal WAV signal \( S'(i, t) \) is produced. \( S'(i, t) = S(i-t), i=t, t+1, t+2, ..., M \);

3. For \( t = 1: 159 \)
3.1 Encode the shifted temporal signal \( S'(i, t) \) to speex VoIP audio stream at the same bit rate;

3.2. Decode the encoded audio signal from the above step to temporal domain, denoted by \( S''(i, t) \);

3.3. Calculate the difference \( D(i, t) = S'(i, t) - S''(i, t) \);

3.4. Shift-recompression based reshuffle characteristic features are given by:

\[
SRSC(t) = \frac{\sum_i |D(i,t)|}{\sum_i |S'(i,t)|} \\
\text{Where } t = 1, 2, ..., 159. \text{ There are 159 features for a speex VoIP audio file.}
\]

5.2. SVM

LibSVM [14] is being used for support vector machine classification in this project. Support vector machines are a relatively new learning method used for binary classification [15]. The basic idea is to find a hyperplane which separates the d-dimensional data perfectly into its two classes. However, since example data is not often linearly separable, SVM’s introduce the notion of a “kernel induced feature space” which casts the data into a higher dimensional space where the data is separable. Typically casting into such a space would cause problems computationally, and with overfitting. The key insight used in SVM’s is that the higher dimensional space doesn’t need to be dealt with directly, which eliminates the above concerns.

Suppose we are given data points each of which belong to one of two classes,

\[
D = \{(x_i, c_i) | X_i \in \mathbb{R}^p, c_i \in \{-1, 1\}\}_{i=1}^n
\]

In this method, the differential values are considered as vector \( x_i \), \( c_i \) represents whether the given file is forged or not. Under this environment, the SVM classifier attempts to maximize the geometric margin which is the distance from the hyperplane to the closest instances on either side. The hyperplane can be written as the set of points \( x \) satisfying \( w \cdot x - b = 0 \). To maximize the margin of separation, two hyperplanes are represented by the following equations

\[
W \cdot X_i - b \geq 1, \text{ for } X_i \text{ of the first class}
\]

\[
W \cdot X_i - b \leq 1, \text{ for } X_i \text{ of the second class}
\]

Or equivalently

\[
c_i(W \cdot X_i - b) \geq 1, \text{ for all } 1 \leq i \leq n
\]

The figure below demonstrates the SVM classifier briefly. The points on the line \( w \cdot x - b = -1 \) and \( w \cdot x - b = 1 \) are called the support vectors.

6. Experiments

In each experiments we randomly select 500 untouched VoIP audio files and 500 tampered VoIP files for training, and other 500 untouched and 500 tampered VoIP files for testing. The experiment has been repeated for 100 times and the detection accuracy is the mean of 100 experiments. Due to the computational cost, we only test the forgery manipulations misplaced by 30, 70, and 125 samples.

The results to demonstrate the performance of this forgery detection method based on the SVM classifier and the differential values are given in Table 1.

<table>
<thead>
<tr>
<th>OFFSET (in samples)</th>
<th>DETECTION RATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>82.1%</td>
</tr>
<tr>
<td>70</td>
<td>79.8%</td>
</tr>
<tr>
<td>125</td>
<td>82.1%</td>
</tr>
</tbody>
</table>
7. Conclusion

While VoIP-based services are favorably disseminated in our real-life, there is an increasing challenge to detect the tampering in VoIP audio streams. To this date, there is no such an effective detection of the tampering. In this paper, inspired by the success in image forgery detection and steganalysis, a shift-recompression differential feature analysis is designed to detect VoIP audio forgery with the aid of learning machine. Our preliminary experimental results demonstrate the effectiveness of proposed method.

The future study will be focused on the improvement of proposed method and applied the improvement to other formats of VoIP forgery detection.

REFERENCES


