Mel-Cepstrum Based Steganalysis for VoIP-Steganography

Christian Kraetzer[a] and Jana Dittmann[a]

[a]Research Group Multimedia and Security, Department of Computer Science,
Otto-von-Guericke-University of Magdeburg, Germany

ABSTRACT

Steganography and steganalysis in VoIP applications are important research topics as speech data is an appropriate cover to hide messages or comprehensive documents. In our paper we introduce a Mel-cepstrum based analysis known from speaker and speech recognition to perform a detection of embedded hidden messages. In particular we combine known and established audio steganalysis features with the features derived from Mel-cepstrum based analysis for an investigation on the improvement of the detection performance. Our main focus considers the application environment of VoIP-steganography scenarios.
The evaluation of the enhanced feature space is performed for classical steganographic as well as for watermarking algorithms. With this strategy we show how general forensic approaches can detect information hiding techniques in the field of hidden communication as well as for DRM applications. For the later the detection of the presence of a potential watermark in a specific feature space can lead to new attacks or to a better design of the watermarking pattern. Following that the usefulness of Mel-cepstrum domain based features for detection is discussed in detail.

Keywords: Steganography, speech steganalysis, audio steganalysis

1. MOTIVATION AND THE APPLICATION SCENARIO OF VOIP STEGANOGRAPHY

Digital audio signals are, due to their stream-like composition and the high data rate, appropriate covers for a steganographic method, especially if they are used in communication applications. Dittmann[1] et. al and Kraetzer[2] et. al describe for example the design and implementation of a VoIP based steganography scenario, indicating possible threats resulting from the embedding of hidden communication channels into such a widely used communication protocol. When comparing the research in image and audio steganalysis it is obvious that the second one is mostly neglected by the information hiding community so far. While advanced universal steganalysis approaches exist for the image domain (e.g. by Ismail Avcibas[3] et. al, Siwei Lyu[4] et. al, Yoan Miche[5] et. al, Mehmet U. Celik[6] et. al or Jessica Fridrich[7]) only few approaches exist in the audio domain. This fact is quite remarkable for two reasons. The first one is the existence of advanced audio steganography schemes, like the one demonstrated by Kaliappan Gopalan[8] for example. The second one is the very nature of audio material as a high capacity data stream which allows for scientifically challenging statistical analyses. Especially inter-window analyses (considering the evolvement of the signal over time) which are possible on this continuous media distinguish audio signals from the image domain.
Chosen from the few audio steganalysis approaches the works of Hamza Ozer[9] et. al, Micah K. Johnson[10] et. al, Xue-Min Ru[11] et. al and Ismail Avcibas[12] shall be mentioned here as related work. These approaches can be grouped into two classes:

1. **Tests against a self-generated reference signal:** A classification based on the distances computed between the signal and a self-generated reference signal (e.g. by Xue-Min Ru[11] et. al) via linear predictive coding (LPC), benefiting from the very nature of the continuous wave-based audio signals; or from Hamza Ozer[9] et. al and Ismail Avcibas[12] by using a denoising function).

2. **Classification against a statistical model for normal and "abnormal" behaviour:** Micah K. Johnson[10] et. al show very good results for this technique based on two steganography algorithms by generating a statistical model that consists of the errors in representing audio spectrograms using a linear basis. This basis is constructed from a principal component analysis (PCA) and the classification is done using a non-linear SVM (support vector machine).

In this work we introduce an approach for steganalysis which combines both classes to a framework for reliable steganalysis in a Voice-over-IP (VoIP) application scenario and imply how it can be transferred to the general application field of audio steganography. The VoIP application scenario assumes that while the VoIP partners speak they transfer also a hidden message using a steganographic channel (for a more detailed description of this scenario see Dittmann[1] et. al). It is assumed that this steganographic message is not permanently embedded from start to end of the conversation. In VoIP scenarios we have therefore the advantage to capture voices in such a way that we can assume that: Either the captured voice data is partly an unmarked signal which can be used as training data for un-marked and by specific algorithms marked data, or the stream as input for a stego classifier displays on the time based behaviour differences to determine between marked and un-marked signals as the speech data comes from one speaker and has therefore non-changing speech characteristics. To simulate this VoIP application scenario, we use a set of files which are used for training and analysis. Each file from this set is divided into two parts, a first part for training to build a model and the second for analysis to test for hidden channels. With this set-up we can simulate the streaming behaviour and non-permanent embedding of hidden data.
For our evaluations we furthermore assume that it is possible to train and test models on the appropriate audio material (in our application scenario the speech in VoIP communications as well as marked material for every information hiding algorithm considered) without considering the legal implications such an action might have.

Our introduced framework, named AAST (AMSL Audio Steganalysis Tool Set), allows for SVM based intra-window analysis on audio features as well as χ^2-test based inter-window analysis. In the case of AASTs intra-window analysis a model for each of a number of known information hiding algorithms can be created during the observation of a communication channel or in advance. Based on this trained model a SVM is used to decide whether a signal to be tested was marked with the algorithm for which this model was generated. Focusing on the VoIP steganography scenario and with the goal to improve the security (with regards to integrity) of this communication channel as well as the detection performance of the steganalysis tool used by Kraetzer[2] et. al, new measures (features) were sought for with the assumption that the considered signal is a band limited speech signal (which is the most common payload in VoIP communications). Measures using exactly this assumption were found with the Mel-cepstral based signal analysis in the field of speech and speaker detection.
If the inter-window analysis capability of AAST is used, a feature based statistical model for the behaviour of the channel over time is computed and compared by χ^2-testing against standard distributions. Other innovations (besides the combination of intra- and inter-window steganalysis in one framework) which are introduced in this work are the Mel-cepstrum based features (MFCCs and FMFCCs) for audio steganalysis, the feature fusion as well as initial results for inter-window analysis. These innovations and their impact are reflected in the test objectives and results of this work.

This work has the following structure: An introduction and description of the application scenario is given in section 1. In section 2 the new AAST (AMSL Audio Steganalysis Toolset) is introduced including in subsection 2.2 the set of features which can be computed. Consecutively follows the description of test objectives, test sets and the test set-up as well as the test procedure in section 3. In section 4 the test results are presented and summarised. Section 5 concludes the work by drawing conclusions and deriving ideas for further research in this field.

2. THE PROPOSED STEGANALYSER

Dittmann[1] et. al described in 2005 a basic steganalysis tool which was subsequently enhanced by the research group Multimedia and Security at the Otto-von-Guericke University of Magdeburg, Germany and used in publications concerned with audio steganalysis (e.g. Kraetzer[13,2] et. al). Its functions and measures were derived from image steganalysis and it was shown that the introduced measures had only a limited relevance for the VoIP speech steganography algorithm developed by Kraetzer[2] et. al. As a consequence we introduce new Mel-cepstral analysis based measures, derived from advanced audio signal analysis techniques like speech and speaker detection, for audio steganalysis with the intention to advance the performance of the steganalysis tool introduced by Dittmann[1] et. al.
The improved tool set, referred to as AAST (AMSL Audio Steganalysis Toolset), consists of four modules:

1. pre-processing of the audio/speech data
2. feature extraction from the signal
3. post-processing of the resulting feature vectors (for intra- or inter-window analysis)
4. analysis (classification for steganalysis)

In the following sections these modules are described in more detail.

2.1. Pre-processing of the audio/speech data

The core of AAST, the feature extraction process, assumes audio files as input media. Therefore audio signals in other representations (e.g. the audio stream of a VoIP application) have to be captured into files. This is done by the application of specific hardware or software based capturing modules on the host or in the network. In the case of the VoIP application considered, a modified version of the IDS/IPS (Intrusion Detection/ Intrusion Prevention System) described by Dittmann[14] et. al is used as capturing device.
Additional pre-processing of the audio data (in our application scenario the speech data) handles the input and provides basic functions for data filtering (bit-plane filtering, silence detection), windowing and media specific operations like channel-interleaving/demerging.

2.2. Feature extraction from the signal

The core part of the steganalysis tool set is a sensor computing first order statistical features (sf_i; $sf_i \in \mathbb{SF}$; $\mathbb{SF}$ = set of features in the steganalysis framework) for an audio signal. Based on the initial idea of an universal blind steganalysis tool for multimedia steganalysis a set of statistical features used in image steganalysis was transferred to the audio domain. Originally the set of statistical features ($\mathbb{SF}$) computed for windows of the signal (intra-window) consisted of: sf_{ev} empirical variance, sf_{cv} covariance, $sf_{entropy}$ entropy, $sf_{LSB_{rat}}$ LSB ratio, $sf_{LSB_{flip}}$ LSB flipping rate, sf_{mean} mean of samples in time domain and sf_{median} median of samples in time domain. This set is enhanced in this work by:

- sf_{mel_1}, ..., sf_{mel_C} with C = number of MFCCs which is depending on the sampling rate of the audio signal; for a signal with a sampling rate of 44.1 kHz $C = 29$) computed Mel-frequency cepstral coefficients (MFCCs) describing the rate of change in the different spectrum bands
- sf_{melf_1}, ..., sf_{melf_C} with C = number of FMFCCs with the same dependency on the sampling rate like the MFCCs) computed filtered Mel-frequency cepstral coefficients (FMFCCs) describing the rate of change in the different spectrum bands after applying a filtering function to remove the frequency bands carrying speech relevant components in the frequency domain

The cepstrum (an anagram of the word spectrum) was defined by B. P. Bogert, M. J. R. Healy and J. W. Tukey[15] in 1963. Basically a cepstrum is the result of taking the Fourier transform (FT) or short-time Fourier analysis[16] of the decibel spectrum as if it were a signal. The cepstrum can be interpreted as information about the rate of power change in different spectrum bands. It was originally invented for characterising seismic echoes resulting from earthquakes and bomb explosions. It has also been used to analyse radar signal returns. Generally a cepstrum $\tilde{S}$ can be computed from the input signal S (usually a time domain signal) as:

$$\tilde{S} = FT(\log(FT(S))) \quad (1)$$

Besides its usage in the analysis of reflected signals mentioned above, the cepstrum has found its application in another field of research. As was shown by Douglas A. Reynolds[17] and Robert H. McEachern[18] a modified cepstrum called Mel-cepstrum can be used in speaker identification and the general description of the HAS (Human Auditory System). McEachern models the human hearing based on banks of band-pass filters (the ear is known to use sensitive hairs placed along a resonant structure, providing multiple-tuned band-pass characteristics; see Hugo Fastl and Eberhard Zwicker[19] or David J. M. Robinson and Malcolm O. J. Hawksford[20]) by comparing the ratios of the log-magnitude of energy detected in two such adjacent band-pass structures. The Mel-cepstrum is

considered by him an excellent feature vector for representing the human voice and musical signals. This insight led to the idea pursued in this work to use the Mel-cepstrum in speech steganalysis.
For all applications which are computing the cepstrum of acoustical signals, the spectrum is usually first transformed using the Mel frequency bands. The result of this transformation is called the Mel-spectrum and is used as the input of the second FT computing the Mel-cepstrum represented by the Mel frequency cepstral coefficients (MFCCs) which are used as sf_{mel_1}, ..., sf_{mel_C} in AAST. The complete transformation for the input signal S is described in equation 2.

$$MelCepstrum = FT(MelScaleTransformation(FT(S))) = \begin{pmatrix} sf_{mel_1} \\ sf_{mel_2} \\ \cdots \\ sf_{mel_C} \end{pmatrix} \quad (2)$$

Figure 1 shows the complete transformation procedure for a FFT based Mel-cepstrum computation as introduced by T. Thrasyvoulou and S. Benton[21] in 2003. Other approaches found in literature use LPC based Mel-cepstrum computation. A detailed discussion about which transformation should be used in which case is given by Thrasyvoulou[21] et. al. From these discussion it is obvious that the FT based approach suffices the means of this paper (since no inversion of the transformation is required in any of the analyses).

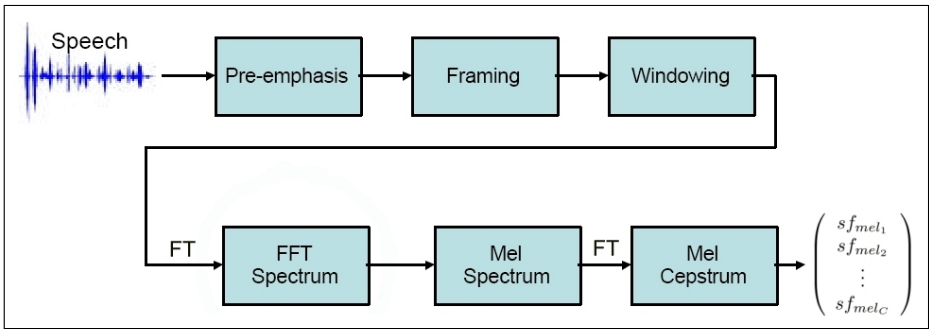

Figure 1: FFT based Mel-cepstrum computation as introduced by Thrasyvoulou[21] et. al

In the implementation of the AAST the pre-emphasis step is done by boosting the digitalised input signal by approximately $20dB$/decade. The window size *window_size* for the framing step in AAST is an application parameter and set in the tests for this work to 1024 samples for the intra-window tests and to 32768 for the inter-window analysis. Windowing is done using non-overlapping Hamming windows. For the computation of the Fourier transforms the AAST uses functions from the *libgsl*[22] package. The implementation of the consecutive filtering steps is based on the description by Thrasyvoulou[21] et. al.

In this paper **a Modification of the Mel-cepstral based signal analysis** is introduced. It is based on the application scenario of VoIP telephony and the basic assumption which was already indicated in section 1: a VoIP communication consists mostly of speech communication between human speakers. This, in conjunction with the knowledge about the frequency limitations of human speech (see e.g. Fastl[19] et. al), led to the idea of removing the speech relevant frequency bands (the spectrum components between 200 and 6819.59 Hz) in the spectral representation of a signal before computing the cepstrum. This procedure, which enhances the computation described by equation 2 by a filter step, returns the FMFCCs (filtered Mel frequency cepstral coefficients; sf_{melf_1}, ..., sf_{melf_C} in AAST) and is expressed in equation 3.

$$FilteredMelCepstrum = FT(SpeechBandFiltering(MelScaleTransformation(FT(S)))) = \begin{pmatrix} sf_{melf_1} \\ sf_{melf_2} \\ \cdots \\ sf_{melf_C} \end{pmatrix} \quad (3)$$

2.3. Post-processing of the resulting feature vectors

In the steganalysis tool set the post-processing of the resulting feature vectors is responsible for preparing the following analysis by providing normalisation and weighting functions as well as format conversions on the feature vectors. This module was introduced to make the approach more flexible and allow for different analysis or classification approaches. Besides the operations (subset generation, normalisation, SVM training, etc) on the vector of intra-window features computed in the second module, a second feature vector can be provided by applying statistical operations like χ^2 testing to the intra-window features, thereby deriving inter-window characteristics describing the evolution of the signal over time.

2.4. Analysis

The subsequent analysis as the final step in the steganalysis process is either done using a SVM (Support Vector Machine) for classification of the signals (in the case of intra-window analysis) or by χ^2 (for inter-window analysis). The SVM technique is based on Vapnik's[23] statistical learning theory and was used as a classification device in different steganalysis related publications (e.g. by Johnson[10] et. al, Ru[11] et. al or Miche[5] et. al). For more details on SVM classification see for example Chih-Chung Chang and Chih-Jen Lin[24] or the section concerned with SVM classification in steganography by Johnson[10] et. al.

3. TEST SCENARIO

Two test goals are to be defined for this work: The primary goal is to reliably detect the presence of a given hidden channel within the defined application scenario of VoIP steganography. The secondary goal is to show the general applicability of our approach and the Mel-cepstral based features in speech and audio steganalysis. In the following the defined sets, set-up, procedure and objectives for the tests necessary for the evaluation of these goals are described.

3.1. Test sets and test set-up

This section describes the set of algorithms A, sets of test files $TestFiles$ and the classification device used in the evaluations.

3.1.1. Information hiding algorithms used

For the evaluations in this work the set of algorithms A from Kraetzer[25] et. al was reused and enhanced by one new algorithm. For this work A_i, $A_i \in A$ denotes a specific information hiding algorithm with a fixed parameter set. The same algorithm with a different parameter set (e.g. lowered embedding strength) would be identified as A_j with $j \neq i$. The set of A is considered in this work to consist of the subsets A_S (audio steganography algorithms) and A_W (audio watermarking algorithms) with $A = A_S \cup A_W$.

A_S **chosen:** the following A_S are used for testing:

- A_{S_1} - LSB (version Heutling051208): This is the algorithm used in the implementation of the VoIP steganography application described by Vogel[26] et. al and Kraetzer[2] et. al, for a detailed description of the algorithm see these publications; parameter set: $silence\_detection = 1$, $embedding\_strength = 100$
- A_{S_2} - Publimark (version 0.1.2): for detailed descriptions see the Publimark website[27] and Lang[28] et. al; parameter set: $none$ $(default)$
- A_{S_3} - WaSpStego: A spread spectrum, wavelet domain algorithm, embedding ECC secured messages into PCM coded audio files. The embedding is done by the modification of the signum of the lower third of wavelet coefficients of each block. Detection is done by correlating the signums of these coefficients with the output of the PSNR initialised with the same key as in the embedding case. Parameter set: $block\_width = 256$, $embedding\_strength = 0.01$
- A_{S_4} - Steghide (version 0.4.3): for detailed descriptions see the Steghide website[29] and Kraetzer[25] et. al; parameter set: $default$

- A_{S_5} - Steghide (version 0.5.1): see A_{S4} above; parameter set: $default$

A_W **chosen:** For evaluating digital audio watermarking algorithms we use the same four A_W already considered by Kraetzer[25] et. al:

- A_{W_1} - Spread Spectrum; parameter set: $ECC = on$, $l = 2000$, $h = 17000$, $a = 50000$
- A_{W_2} - 2A2W (AMSL Audio Water Wavelet); parameter set: $encoding = binary$, $method = ZeroTree$
- A_{W_3} - Least Significant Bit; parameter set: $ECC = on$
- A_{W_4} - VAWW (Viper Audio Water Wavelet); parameter set: $threshold = 40$, $scalar = 0.1$

Those four A_W are also described in detail in Lang and Dittmann.[28]

3.1.2. Test files

Following the two test goals identified above, two different sets of test files ($TestFiles$) are defined: Based on the assumption, that a VoIP communication can be generally modelled as a two channel, speech communication with one non-changing speaker per channel, one of the channels was simulated by using a long audio file (characteristics: duration 27 min 24 sec, sampling rate 44.1 kHz, stereo, 16 bit quantisation in an uncompressed, PCM coded WAV-file) containing only speech signals of one speaker. The signal (set of test files) used was recorded for this purpose at the AMSL (Advanced Multimedia and Security Lab, Otto-von-Guericke University Magdeburg, Germany). This set of test files is in the following denoted with $TestFiles = longfile$.
For the evaluation of the second test goal (the general applicability of the AAST in audio steganalysis) the same set of 389 audio files (classified by context into 4 classes with 25 subclasses like female and male speech, jazz, blues, etc.; characteristics: average duration 28.55 seconds, sampling rate 44.1 kHz, stereo, 16 bit quantisation in uncompressed, PCM coded WAV-files) is used as described by Kraetzer[2] et. al to provide for comparability of the results in regard to the detection performance. This set of test files is in the following denoted with $TestFiles = 389files$.
As shown in figure 2 from both sets of test files modified sets $TestFiles^{\star} = TestFiles \cup TestFiles^M$ (where $TestFiles^M$ is the result of completely marking $TestFiles$ with A_i) are generated for each A_i. This results in one $longfile^{\star}$ and one $389files^{\star}$ for each A_i. For each $TestFiles^{\star}$ the output of AAST's feature extraction process is divided by the user defined ratio s_{tr}:s_{te} (the ratios 64:16, 400:2200 and 2200:400 are chosen for the tests in this work) into two disjoint subsets set_train and set_test (with $s_{tr} = sizeof(set\_train)$ and $s_{te} = sizeof(set\_test)$). The subset set_train (which contains an equal number of feature vectors originating from original and marked audio material as well as a number of s_{tr} vectors from each file in $TestFiles$) is then used to train the classification device used for the classification of the subset set_test.

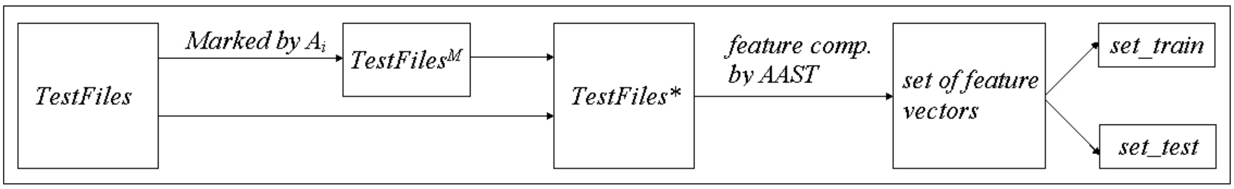

Figure 2: Generation of the two sets for training and testing

3.1.3. Classification Devices

For the classification in the intra-window evaluations the *libsvm* SVM (support vector machine) package by Chih-Chung Chang and Chih-Jen Lin[24] was used. Due to reasons of computational complexity we decided not to change the SVM parameters (γ and c as well as the SVM kernel chosen (RBF) are left to $default$) for the tests performed. This set of SVM parameters as well as the SVM chosen (*libsvm*) is denoted in the following by $SVMmode = default$.
For the inter-window evaluations the χ^2 test included into AAST's post-processing module was used. Its results are subsequently analysed manually.

3.2. Test procedure

As an initial step all required sets of test files ($TestFiles^{\star}$) are generated as described in section 3.1.2. After this step the four modules of the AAST described in section 2 are used to generate the statistical data and classifications required for the evaluation of the test goals.

Pre-processing of the audio/speech data
For the intra-window evaluation the steganalyzer parameters *sp* are set to $sp = (window\_size = 1024, overlap = none)$. In the inter-window evaluations the window size for the steganalysis process had to be increased to $sp = (window\_size = 32768, overlap = none)$. In preliminary test smaller window sizes did not lead to useful results for the χ^2 analysis.

Feature extraction from the signal
By using this module the feature vectors are computed from the audio material. For this work we use additionally to the single features sf, $sf \in \mathbb{SF}$ the sets of features SF ($SF \subseteq \mathbb{SF}$) defined in table 1.

feature set (SF)	**sf or SF in the set**
SF_{std}	$\{sf_{ev}, sf_{cv}, sf_{entropy}, sf_{LSB_{rat}}, sf_{LSB_{flip}}, sf_{mean}, sf_{median}\}$
SF_{MFCC}	$\{sf_{mel_1}, ..., sf_{mel_C}\}$
SF_{FMFCC}	$\{sf_{melf_1}, ..., sf_{melf_C}\}$
$SF_{std \cup MFCC}$	$SF_{std} \cup SF_{MFCC}$
$SF_{std \cup FMFCC}$	$SF_{std} \cup SF_{FMFCC}$

Table 1: Definition of feature sets for evaluation

The maximum possible number of MFCCs and FMFCCs to be computed for audio material with 44.1 kHz sampling rate is $C = 29$.

Post-processing of the resulting feature vectors
For the intra-window evaluations in this step a pre-processing for the SVM application has to be done for each A. After the feature vectors are computed each is identified as belonging to a original or marked file and the complete vector field is normalised using the normalisation function of *libsvm*. By dividing for each file in $TestFiles^{\star}$ the output of AAST's feature extraction process by the user defined ratio s_{tr}:s_{te} with $s_{tr} = sizeof(set\_train)$ and $s_{te} = sizeof(set\_test)$ two disjoint subsets of feature vectors (*set_train* and *set_test*) are generated. This guarantees that *set_train* and *set_test* contain the same number of feature vectors from original and marked files. The subset *set_train* is then used to train with the SVM the model M_{A_i} for each A_i. This M_{A_i} will be used in the analysis to perform the classification. In the training and testing for this work the SVM parameters are set as described in section 3.1.3 ($SVMmode = default$).
For the inter-window evaluation no SVM classification is required. Instead, a inter-window analysis by a χ^2 test for all $sf \in \mathbb{SF}$ against three standard distributions (equal, normal and exponential distribution) is performed here. For this the corresponding post-processing function of AAST is used.

Analysis (classification)
For inter-window analyses the models M_{A_i} generated in the previous step are applied to the subset *set_test*, returning the detection probability $p_{D_{A_i}}$ for $A_i \in A$ and the parameterisations used. For inter-window test the output of the χ^2 test is returned.

3.3. Test objectives

From the goals stated above (first: reliable detection of the presence of a given hidden channel constructed with A_{S_1} within the defined application scenario of VoIP steganography and second: proving the general applicability of the presented approach and the Mel-cepstral based features in speech and audio steganalysis) the following test objectives are derived (the basic assumptions, parameters and feature sets are summarised in tables 2 and 3 below):

- O_1 optimising the detection probability $p_{D_{S_1}}$ for the algorithm used in the VoIP application scenario (A_{S_1}), assuming the fact that a VoIP communication can be generally modelled as a two channel speech communication with one non-changing speaker per channel

O_2 analysing the inter-window characteristics describing the evolving of the signal marked by A_{S_1} over time by applying χ^2 testing to the fs ($fs \in \mathbb{FS}$)

O_3 determining the relevance (for $p_{D_{A_i}}$) of all features fs ($fs \in \mathbb{FS}$) for all selected A and fixed sp, $SVMmode$ and $TestFiles^\star$

O_4 determining the influence of the size of the model M_{A_i} on $p_{D_{A_i}}$ for signals marked by the selected A

O_5 determining the gain in $p_{D_{A_i}}$ by fusioning selected fs or FS ($fs \in \mathbb{FS}$; $FS \subseteq \mathbb{FS}$) in the classification process

The test objective O_1 is the obvious test goal within the focus of this work. A high $p_{D_{S_1}}$ is proving the usefulness of applying steganalysis to VoIP channels.
The second test objective briefly evaluates the possibilities for inter-window analysis on A_{S_1} using the features $sf \in \mathbb{SF}$. Test objectives O_3, O_4 and O_5 are aimed at determining the overall quality of our steganalysis approach and the features used on a larger set of algorithms A. The fitness in steganalysis for all features as well as the statistical transparency of the considered watermarking algorithms with regards to these features is observed. Special attention is paid in these evaluations to the quality of the MFCCs and FMFCCs as features for steganalysis.
In particular the test objectives O_4 and O_5 are formulated to address the impact of the size of the model (in feature vector computed per file in $TestFiles^\star$) on the classification and the gain on $p_{D_{A_i}}$ by feature fusion.
To provide a reasonable sequence for the presentation of the research results, the test objectives derived from the goals are ordered in a way to move from the most specific to a more general case. In the tests performed the class of audio material used as a cover and the kind of energy spreading used by the steganographic algorithm is first considered according to the application scenario identified in section 1 and then in a larger scope to identify possible constraints to the applicability of this method.

Summarising sections 2 and 3, tables 2 and 3 list the basic assumptions, parameters and feature sets used in the evaluation of the test objectives O_1 to O_5 .

Test objective	**basic assumption**	**algorithms tested**	**type of analysis**
O_1	VoIP steganalysis	S_1	intra-window (SVM)
O_2	VoIP steganalysis	S_1	inter-window (χ^2)
O_3	audio steganalysis	$\forall A_i \in A$	intra-window (SVM)
O_4	audio steganalysis	$\forall A_i \in A$	intra-window (SVM)
O_5	audio steganalysis	$\forall A_i \in A$	intra-window (SVM)

Table 2: Assumptions made in the evaluation of the test objectives O_1 to O_5

Test objective	sp	$TestFiles^\star$	s_{tr}:s_{te}	**feature sets**
O_1	$window\_size = 1024$	$longfile^\star$	400:2200 and 2200:400	$\forall\, SF$ defined in table 1
O_2	$window\_size = 32768$	$longfile^\star$	n.d. (not defined)	$\forall\, sf \in \mathbb{SF}$
O_3	$window\_size = 1024$	$389files^\star$	64:16	$\forall\, sf \in \mathbb{SF}$
O_4	$window\_size = 1024$	$389files^\star$, $longfile^\star$	64:16, 400:2200 and 2200:400	$\forall\, sf \in \mathbb{SF}$
O_5	$window\_size = 1024$	$389files^\star$, $longfile^\star$	64:16, 400:2200 and 2200:400	$\forall\, SF$ defined in table 1

Table 3: Parameters and features used in the evaluation of the test objectives O_1 to O_5

4. TEST RESULTS

This section describes the results for the test objectives O_1 to O_5. The results presented here are summarised from a far larger set of test results, which is provided in full detail as additional material on `http://wwwiti.cs.uni-magdeburg.de/~kraetzer/publications.htm`. For improved readability all lines are removed from the following tables which do not carry at least one result above $p_{D_{A_i}} = 52\%$ (which is considered in this work to be the lower boundary for discriminating features; we assume that detection probabilities above 50 % and below 52 % might still be a result of a random classification on a non-discriminating feature). Additionally all results above $p_{D_{A_i}} = 52\%$ are marked italic.

Test objective O_1 (optimisation of $p_{D_{S_1}}$):
Table 4 shows the relevance of single features on the $p_{D_{S_1}}$ for two different ratios of s_{tr}:s_{te} (400:2200 and 2200:400). The highest result in this test is found with $p_{D_{S_1}} = 74.375\%$ at the shown parameterisation for the feature $sf_{LSB_{rat}}$ and s_{tr}:s_{te} = 2200:400. This table also shows a higher average result for the FMFCCs when comparing them with their MFCC counterparts.

feature	$s_{tr} = 400;\ s_{te} = 2200$	$s_{tr} = 2200;\ s_{te} = 400$	feature	$s_{tr} = 400;\ s_{te} = 2200$	$s_{tr} = 2200;\ s_{te} = 400$
sf_{mel_8}	*53.7955*	*53.375*	$sf_{melf_{11}}$	*52.75*	*52.625*
sf_{mel_9}	51.9091	*52*	$sf_{melf_{13}}$	*52.7273*	*52.375*
$sf_{mel_{12}}$	*52.6136*	51	$sf_{melf_{15}}$	*53.6591*	*57*
$sf_{mel_{13}}$	51.9091	*52.125*	$sf_{melf_{18}}$	*52.4545*	51.875
$sf_{mel_{15}}$	51.4545	*52.25*	$sf_{melf_{20}}$	*54.0227*	*53.5*
$sf_{mel_{16}}$	*52*	51.125	$sf_{melf_{21}}$	*52*	*54.5*
$sf_{mel_{18}}$	*52.8182*	51.75	$sf_{melf_{22}}$	*53.1818*	*53.5*
$sf_{mel_{21}}$	*54.1136*	*54*	$sf_{melf_{23}}$	*57.3864*	*57.125*
$sf_{mel_{22}}$	*56.8864*	*56.125*	$sf_{melf_{24}}$	50.75	*52.625*
$sf_{mel_{23}}$	*58.25*	*58*	$sf_{melf_{25}}$	*58.7273*	*57.875*
$sf_{mel_{24}}$	51.9091	*52.375*	$sf_{melf_{26}}$	*54.7045*	*54.625*
$sf_{mel_{25}}$	*52.4091*	*52.75*	$sf_{melf_{27}}$	*56.8409*	*56.5*
$sf_{mel_{27}}$	*52.4318*	*52.75*	$sf_{melf_{28}}$	51.6364	*52.75*
$sf_{mel_{28}}$	*54.8636*	*56.125*	$sf_{LSB_{flip}}$	*54.9545*	*69.125*
sf_{melf_3}	*52.5227*	*53.125*	$sf_{LSB_{rat}}$	*74.1818*	*74.375*

Table 4: $p_{D_{S_1}}$ for all $sf \in \mathbb{SF}$ where $p_{D_{S_1}} \leq 52\%$

Table 5 shows the impact of selected feature fusions on $p_{D_{S_1}}$ for the same two ratios of s_{tr}:s_{te} used above. Perfect results with $p_{D_{S_1}} = 100\%$ can be found at the shown parameterisation for SF_{FMFCC} and $SF_{std \cup FMFCC}$ at s_{tr}:s_{te} = 2200:400. Since $SF_{FMFCC} \subset SF_{std \cup FMFCC}$ the evaluations could be limited to this feature set.

feature set	$s_{tr} = 400;\ s_{te} = 2200$	$s_{tr} = 2200;\ s_{te} = 400$
SF_{std}	*72.8864*	*77.875*
SF_{MFCC}	*64.1818*	*67*
$SF_{std \cup MFCC}$	*71.7273*	*79*
SF_{FMFCC}	*98.2273*	*100*
$SF_{std \cup FMFCC}$	*96.9318*	*100*

Table 5: $p_{D_{S_1}}$ for selected feature sets $FS \subseteq \mathbb{FS}$

A detection probability $p_{D_{S_1}} = 100\%$ indicates that, by applying the corresponding model to a intra-window based classification of a vector field generated by AAST using the feature set SF_{FMFCC} on audio material of the same type as $longfile^{\star}$ (i.e. speech) and with the same parameterisations as described in section 3, the result would be a perfect classification into marked and un-marked material.

Test objective $\mathbf{O}_2$ (inter-window analysis for A_{S_1}):
By applying the inter-window analysis by a χ^2 test for all $sf \in \mathbb{SF}$ against three standard distributions (equal, normal and exponential distribution), a maximum distance of 3.5596% between un-marked and marked material can be found in $sf_{melf_2 6}$ in the case of an assumed exponential distribution. This result is shown in figure 3.

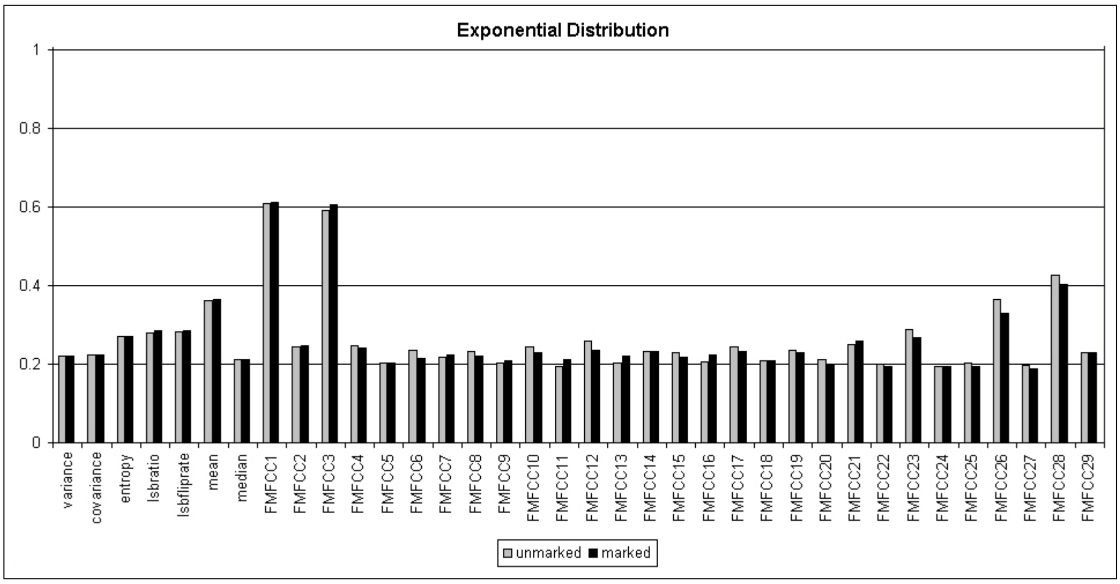

Figure 3: Normalised distances of all elements of $SF_{std \cup FMFCC}$ in a χ^2 test against an assumed exponential distribution

Generally a larger distance in between un-marked and marked material can be seen in the FMFCCs than in MFCCs. The average distances computed are 0.88% and of 0.74%.

Test objective $\mathbf{O}_3$ (feature relevance for all $sf \in \mathbb{SF}$ for all A):
As already stated above, $p_{D_{A_i}} = 52\%$ is considered in this work to be the lower boundary for discriminating features. Table 6 shows the $p_{D_{A_i}}$ for each single feature $sf \in \mathbb{SF}$ for each A.

	A_{S_1}	A_{S_2}	A_{S_3}	A_{S_4}	A_{S_5}	A_{W_1}	A_{W_2}	A_{W_3}	A_{W_4}	**rel. feat.**
sf_{mel_1}	50.3615	51.842	*52.5466*	*52.3297*	*52.635*	*55.6716*	50.371	*52.3458*	50.233	5
sf_{mel_2}	49.9197	51.1583	50.7471	50.6507	51.2516	*56.8204*	*52.75*	50.5141	50.2651	2
sf_{mel_3}	50.3856	50.37	50.5302	50.3374	51.0046	*54.9325*	51.4597	50.4659	50.3856	1
sf_{mel_6}	49.9759	51.0296	50.9078	50.9801	51.2681	*53.0045*	51.2903	51.1729	50.0964	1
sf_{mel_7}	50.1928	50.4987	50.3133	50.49	51.2516	*52.3136*	51.0806	50.6587	50.6105	1
$sf_{mel_{14}}$	50.008	50.0724	50.715	50.5382	51.2516	*52.1369*	51.0161	50.6025	50.0643	1
sf_{melf_1}	50.0482	50.5872	51.8959	51.1889	51.6222	*74.7349*	*54.379*	50.6507	51.1247	2
sf_{melf_2}	50.0482	51.3755	51.1327	51.0684	51.5316	*68.7179*	*56.9032*	51.1086	50.482	2
sf_{melf_3}	49.9839	50.5068	50.6507	50.6186	50.6094	*62.6767*	*52.1613*	50.5864	50.3213	2
sf_{melf_4}	50.3374	51.295	51.1648	51.0122	51.3834	*53.9765*	50.3871	51.0925	50.233	1
sf_{melf_5}	50.2892	51.5927	*54.8924*	*53.125*	*52.8574*	*56.74*	51.5323	*52.2735*	50.8435	5
sf_{melf_6}	50.6186	*52.9038*	50.49	*52.3297*	*53.2609*	50.4579	*53.9435*	*53.2214*	50.5463	5
sf_{melf_7}	50.0321	51.3514	*54.8924*	51.8557	51.4575	*52.0967*	*52.3468*	51.3817	50.8917	3
sf_{melf_8}	49.8313	*53.0647*	*54.1934*	*53.8239*	*53.6644*	*54.5549*	*53.2177*	*53.9123*	49.7831	7
$sf_{melf_{10}}$	49.9679	50.925	*52.0485*	*52.2413*	*52.1657*	*60.0096*	51.0645	51.5183	50.3213	4
$sf_{melf_{11}}$	50.2008	51.4559	51.4139	*52.1208*	51.1117	50.9158	*54.0645*	51.7915	50.5784	2
$sf_{melf_{12}}$	50.1687	51.1583	*52.1771*	51.6067	51.5399	*59.5839*	*52.0161*	51.5103	50.3535	3
$sf_{melf_{13}}$	49.8634	*52.204*	*52.884*	*53.5106*	*53.2362*	*65.866*	*52.621*	*52.5868*	50.3695	7
$sf_{melf_{14}}$	50.4258	50.555	50.8114	50.6266	51.0375	*56.2982*	50.6048	50.5945	49.6064	1
$sf_{melf_{15}}$	49.8634	51.4559	*52.9483*	*52.6751*	*52.4292*	*69.0071*	51.4516	*52.1449*	50.9399	5
$sf_{melf_{16}}$	49.9036	*52.5901*	51.8718	*52.7796*	*52.7668*	*54.5469*	51.0645	*52.8438*	50.1205	5
$sf_{melf_{17}}$	49.9197	50.5309	51.2612	51.4219	51.2269	*59.8329*	51.7097	50.5463	49.7269	1
$sf_{melf_{18}}$	50.2892	*53.0808*	*53.2857*	*53.1491*	*52.9233*	*52.3377*	50.7097	*53.2616*	50.4097	6
$sf_{melf_{19}}$	50.1044	50.6194	50.5382	50.6909	51.0952	50.5222	*53.2177*	50.5784	49.7188	1
$sf_{melf_{20}}$	50.482	50.5792	*52.7715*	*52.6912*	51.0952	*63.1828*	*52.0565*	*52.394*	50.3294	5
$sf_{melf_{21}}$	50.1526	*53.0084*	51.4862	*53.3821*	*53.2773*	51.9682	*52.2258*	*53.117*	50.3936	5
$sf_{melf_{22}}$	50.4017	50.4826	50.5784	50.6346	51.2105	*55.2378*	51.7258	50.5945	51.0363	1
$sf_{melf_{23}}$	50.6105	51.4318	50.964	*52.9643*	*52.141*	*55.9929*	50.7258	51.7674	50.5623	3
$sf_{melf_{24}}$	50.1767	50.6998	50.8435	50.8515	50.6588	50.8033	*52.4194*	50.6828	50.474	1
$sf_{melf_{26}}$	50.2651	*52.936*	*52.6751*	*53.3017*	51.2516	*53.8641*	49.9758	*52.1771*	50.6587	5
$sf_{melf_{28}}$	49.992	51.2066	51.8075	51.9441	50.3953	*55.1655*	49.7177	51.1488	50.3695	1
sf_{cv}	51.1086	50.9009	*52.0807*	51.0765	51.2516	*87.1144*	50.9758	51.7915	51.4058	2
$sf_{entropy}$	50.1848	51.5042	50.4097	51.1648	51.3916	*63.7371*	50.7581	51.687	50.241	1
$sf_{LSB_{flip}}$	51.5263	*52.4051*	51.8638	*52.2253*	*52.1574*	*53.3178*	51.5806	*52.2092*	51.446	5
$sf_{LSB_{rat}}$	*55.4627*	*57.5129*	*59.8329*	*57.6317*	*60.9848*	*64.2433*	*60.7339*	*57.7121*	*52.402*	9
sf_{ev}	50	51.0135	50.49	50.8596	51.474	*57.1417*	50.75	51.0202	50.1526	1

Table 6: $p_{D_{A_i}}$ for all $sf \in \mathbb{SF}$ where $p_{D_{A_i}} \leq 52\%$ (s_{tr}:s_{te}=64:16). Additionally for each line the number of $p_{D_{A_i}} \leq 52\%$ is given.

Table 6 shows the 36 (out of 65) features sf, $sf \in \mathbb{SF}$ which are relevant for at least one A_i. If a $p_{D_{A_i}}$ is larger than 52% it is printed italic to improve readability. The last column of table 6 indicates that out of these 36 features 22 have relevance for 1 to 4 A_i, 13 have relevance for 5 to 8 A_i and only one ($sf_{LSB_{rat}}$) is relevant for all A.

Test objective $\mathbf{O}_4$ (influence model size):
When comparing the $p_{D_{S_1}}$ in tables 4 and 6 it is obvious that the models applied to obtain the results for table 4 ($sizeof(set\_train) = 400$ and 2200) are better fitting for A_{S_1} than the models derived with fewer feature vectors ($sizeof(set\_train) = 64$). Generally the results imply that a larger model (in terms of feature vectors computed per file) is better than a smaller model.

Test objective $\mathbf{O}_5$ (feature fusion):
The results already seen for the feature fusion for A_{S_1} are confirmed by the results for the fusions on all A displayed in table 7. For the highest fusion result achieved for every A_i is generally better than the best $p_{D_{A_i}}$ for any single feature sf, $sf \in \mathbb{SF}$.

	A_{S_1}	A_{S_2}	A_{S_3}	A_{S_4}	A_{S_5}	A_{W_1}	A_{W_2}	A_{W_3}	A_{W_4}
SF_{std}	*57.1015*	*54.5447*	*61.0138*	*60.4193*	*61.1989*	*88.8496*	*61.8468*	*59.3107*	*54.9004*
SF_{MFCC}	51.1086	*53.5634*	*56.0733*	*53.3901*	*52.9397*	*75.6668*	*57.9597*	*53.7034*	*52.8358*
$SF_{std \cup MFCC}$	*54.6674*	*55.9041*	*60.8451*	*59.383*	*59.7579*	*91.0427*	*63.9194*	*58.0334*	*55.4868*
SF_{FMFCC}	*52.9884*	*58.832*	*64.4441*	*59.3429*	*58.7698*	*95.0755*	*67.6935*	*59.0215*	*57.5594*
$SF_{std \cup FMFCC}$	*56.4508*	*59.9743*	*67.2156*	*60.6523*	*60.8696*	*97.5177*	*71.629*	*60.5559*	*59.5035*

Table 7: $p_{D_{A_i}}$ for selected feature sets $FS \subseteq \mathbb{FS}$ (s_{tr}:s_{te}=64:16)

5. SUMMARY

The results for the five test objectives defined in section 3.3 show the following: in the intra-window tests for test objective O_1 a prediction rate of $p_{D_{S_1}} = 100\%$ could be reached for A_{S_1}, even if the intra-window tests for objective O_2 do not lead to useful results for this algorithm. The feature relevance tests for all $sf \in \mathbb{SF}$ for all A show that for different A different sf are relevant. Only one feature ($sf_{LSB_{rat}}$) is relevant for all A with the given parameterisations. Regarding the model size (which is equal to the size of *set_train*) it is implied in the results from O_4 that increasing the number of vectors computed per audio signal might increase the quality of the model and therefore p_{D_A} too. More tests are necessary to substantiate this implication. From the feature fusion tests for O_5 it can be seen that the fusion has a positive impact on the detection probability. To reach optimal results it might be useful to apply a fusion only to SF where each $sf \in SF$ is considered relevant for the A under observation.

Test objective	A_{S_1}	A_{S_2}	A_{S_3}	A_{S_4}	A_{S_5}	A_{W_1}	A_{W_2}	A_{W_3}	A_{W_4}
O_1	100%	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	
O_2	3.56%	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	
O_3	55.4627%	57.5129%	59.8329%	57.6317%	60.9848%	87.1144%	60.7339%	57.7121%	52.402%
O_4	100%	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.
O_5	57.1015%	59.9743%	67.2156%	60.6523%	61.1989%	97.5177%	71.629%	60.5559%	59.5035 %

Table 8: $\max(p_{D_{A_i}})$ computed in the evaluation of test objectives O_1 to O_5

The maximum values for all $p_{D_{A_i}}$ computed in the evaluation of test objectives O_1 to O_5 are summarised in table 8. Concluding these figures and the knowledge gained from the tests it can be said that the two test goals described in section 3: first a reliable detection of a hidden channel constructed using A_{S_1} within the defined application scenario of VoIP steganography, and second the demonstration of the general applicability of our approach and the Mel-cepstral based features in speech and audio steganalysis have been successfully reached.

From the findings presented here room for further research can be found considering the following aspects: The tests from O_1 and O_2 should be applied as well to all other A_i, first to review results from $longfile$ on a larger scale (as already mentioned above) and second to further evaluate our approach for inter-window statistical detection. Furthermore the number of algorithms evaluated should be increased, either by varying the parameters for the A already considered or by adding new algorithms to the test set. From this we hope to gain information whether classes of algorithms can be identified. This step would also generate more M_{A_i} which would be a necessary input for a intra-window based, automatic audio steganalysis tool. For this also more evaluations on model quality determination are necessary.
Changes on the global AAST parameters (*window_size*, *overlap*, etc) should be evaluated to find for each A_i a M_{A_i} with a $p_{D_{A_i}} = 100\%$ and the smallest *set_train* required to maximise the performance of our intra-window analysis approach. Further research should also be focused on the classification technique used. Other classification techniques (e.g. kNN-classification) might lead to a easier discrimination approach for different A_i.

Acknowledgements

We wish to thank Claus Vielhauer for suggesting to transfer the Mel-cepstral based signal analysis from biometric speaker verification to the domain of steganalysis and Stefan Kiltz for his help in processing the mathematical and signal theoretic backgrounds. We also wish to express our thanks to Sebastian Heutling for improving the implementation of AAST and Jan Leif Hoffmann for providing his algorithm WaSpStego for the tests.
The work about MFCC and FMFCC features described in this paper has been supported in part by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT. The information in this document is provided as is, and no guarantee or warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.
Effort for implementing the steganalysis tool described in this paper was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number FA8655-04-1-3010. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government.

REFERENCES

1. J. Dittmann, D. Hesse, and R. Hillert, "Steganography and steganalysis in Voice-over IP scenarios: operational aspects and first experiences with a new steganalysis tool set," in *Security, Steganography, and Watermarking of Multimedia Contents VII, SPIE Vol. 5681*, P. W. Wong and E. J. Delp, eds., *SPIE and IS&T Proceedings*, pp. 607–618, (San Jose, California USA), Jan. 2005.
2. C. Kraetzer, J. Dittmann, T. Vogel, and R. Hillert, "Design and Evaluation of Steganography for Voice-over-IP," in *Proceedings of the IEEE International Symposium on Circuits and Systems, Kos, Greece, 21-24th May*, 2006.
3. I. Avcibas, M. Kharrazi, N. Memon, and B. Sankur, "Image steganalysis with binary similarity measures," in *EURASIP Journal on Applied Signal Processing Volume 2005 Issue 17*, pp. 2749–2757, 2005.
4. S. Lyu and H. Farid, "Detecting hidden messages using higher-order statistics and support vector machines," in *Proc. 5th Int'l Workshop on Information Hiding, SpringerVerlag*, 2002.
5. Y. Miche, B. Roue, A. Lendasse, and P. Bas, "A feature selection methodology for steganalysis," in *Proceedings of the International Workshop on Multimedia Content Representation, Classification and Security, Istanbul (Turkey), September 11-13, 2006*, Springer Berlin / Heidelberg, 2006.
6. M. Celik, G. Sharma, and A. M. Tekalp, "Universal image steganalysis using rate-distortion curves," in *Proceedings of SPIE: Security and Watermarking of Multimedia Contents VI, vol. 5306, San Jose, CA, Jan.*, 2004.
7. J. Fridrich, "Feature-based steganalysis for jpeg images and its implications for future design of steganographic schemes.," in *Proceedings of the Information Hiding Workshop*, pp. 67–81, 2004.
8. K. Gopalan, "Cepstral domain modification of audio signals for data embedding: preliminary results," *Security, Steganography, and Watermarking of Multimedia Contents VI* **5306**(1), pp. 151–161, SPIE, 2004.
9. H. Ozer, I. Avcibas, B. Sankur, and N. Memon, "Steganalysis of audio based on audio quality metrics," in *SPIE Electronic Imaging Conf. On Security and Watermarking of Multimedia Contents, Jan. 20-24, Santa Clara*, 2003.
10. M. K. Johnson, S. Lyu, and H. Farid, "Steganalysis of recorded speech," in *in Proc. SPIE, vol. 5681, Mar. 2005*, pp. 664–672, 2005.
11. X.-M. Ru, H.-J. Zhang, and X. Huang, "Steganalysis of audio: Attacking the steghide," in *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, China, 18-21 August*, pp. 3937–3942, 2005.
12. I. Avcibas, "Audio steganalysis with content-independent distortion measures," in *IEEE Signal Processing Letters, Vol. 13, No. 2, February 2006*, pp. 92–95, 2006.
13. C. Kraetzer and J. Dittmann, "Früherkennung von verdeckten Kanälen in VoIP-Kommunikation," in *Proceedings of the BSI-Workshop IT-Frühwarnsysteme, Bonn, Germany, July 12th*, pp. 207–214, 2006.
14. J. Dittmann and D. Hesse, "Network based intrusion detection to detect steganographic communications channels - on the example of audio data," in *Proceedings of IEEE 6th Workshop on Multimedia Signal Processing, Sep. 29th - Oct. 1st 2004, Siena, Italy, ISBN 0-7803-8579-9*, 2004.
15. B. P. Bogert, M. J. R. Healy, and J. W. Tukey, "The frequency analysis of time series for echoes: cepstrum, pseudo-autocovariance, cross-cepstrum, and saphe cracking," in *Proceedings of the Symposium on Time Series Analysis*, M. Rosenblatt, ed., (Wiley New York, USA), Feb. 1963.
16. J. B. Allenand and L. R. Rabiner, "A unified approach to short-time Fourier analysis, synthesis," in *Proc. IEEE*, pp. 1558–1564, Nov. 1977. Published as Proc. IEEE, volume 65, number 11.
17. D. A. Reynolds, *A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification.* Phd thesis, Department of Electrical Engineering, Georgia Institute of technology, USA, 1992.
18. R. H. McEachern, "Hearing it like it is: Audio signal processing the way the ear does it," in *DSP Applications*, February 1994.
19. H. Fastl and E. Zwicker, *Psychoacoustics. Facts and Models.*, Springer, Berlin, second ed., 1999. ISBN 3-540-65063-6.
20. D. J. M. Robinson and M. O. J. Hawksford, "Psychoacoustic models and non-linear human hearing," in *Proceedings of the AES Convention*, (109), AES, (Los Angeles), 2000.
21. T. Thrasyvoulou and S. Benton, *Speech parameterization using the Mel scale Part II*, 2003.
22. GNU, *libgsl*, 2006. Available at `http://www.gnu.org/software/gsl`.
23. V.N.Vapnik, *The nature of statistical learning theory*, Springer Verlag, New York, 1995.
24. C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001. Available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.
25. C. Kraetzer, J. Dittmann, and A. Lang, "Transparency benchmarking on audio watermarks and steganography," in *SPIE conference, at the Security, Steganography, and Watermarking of Multimedia Contents VIII, IS&T/SPIE Symposium on Electronic Imaging, 15-19th January, 2006, San Jose, USA*, 2006.
26. T. Vogel, J. Dittmann, R. Hillert, and C. Kraetzer, "Design und Evaluierung von Steganographie für Voice-over-IP," in *Sicherheit 2006 GI FB Sicherheit, GI Proceedings*, (Magdeburg, Germany), Feb. 2006.
27. G. L. Guelvouit, *Publimark*, 2004. Available at `http://perso.wanadoo.fr/gleguelv/soft/publimark`.
28. A. Lang and J. Dittmann, "Profiles for evaluation and their usage in audio wet," in *IS&T/SPIE's 18th Annual Symposium, Electronic Imaging 2006: Security and Watermarking of Multimedia Content VIII, Vol. 6072*, P. W. Wong and E. J. Delp, eds., *SPIE Proceedings*, (San Jose, California USA), Jan. 2006.
29. S. Hetzl, *Steghide*, 2003. Available at `http://steghide.sourceforge.net`.