An Efficient Search Method for the Content-Based Identification of Telephone-SPAM

Julian Strobl, Bernhard Mainka, Gary Grutzek and Heiko Knospe
Cologne University of Applied Sciences, Germany
Email: {julian.strobl|bernhard.mainka|gary.grutzek|heiko.knospe}@fh-koeln.de

***Abstract*—With the help of VoIP technology, large numbers of unsolicited calls can be conveniently placed and SPAM over Internet Telephony may become a major nuisance and threat. Various mitigation methods have been proposed which are mostly based on a pattern analysis of the signaling traffic. This contribution shows that an analysis of the audio content is also feasible and can provide protection against replayed calls. In order to identify similar or equal audio data, spectral features are extracted and a short and robust audio fingerprint is computed. The definition of the fingerprint is optimized for a fast index-based search. Then, the matching of telephone speech data is based on the intersection of inverted files of audio fingerprints. Furthermore, the system design of a working prototype is explained and experimental results on the recognition rate and the performance of the system are presented. It can be shown that the search method is suitable for an efficient identification of SPAM calls.**

***Index Terms*—SPAM, VoIP, SPIT, Audio Fingerprinting, Speech Identification, Audio Search.**

I. Introduction

Telephone-SPAM is characterized by bulk unsolicited calls. The spammer attempts to initiate a voice session and relays a prerecorded message if the callee answers [1]. The prevalent Voice over IP (VoIP) technology provides convenient tools and low-priced possibilities to place a large number of SPAM calls.

Various mitigation methods have been proposed over the last years which usually include a call pattern analysis of the signaling traffic, reputation systems and blacklisting of callers. It is often assumed that a content-based call analysis is inadequate due to the synchronous character of telephony. When the beginning of the SPAM audio content is delivered, it is in fact too late to protect the affected callee. But the originating audio data can well be used to identify a replayed call and to anticipate subsequent similar SPAM calls. In addition, the content analysis is able to detect SPAM after few such calls. It has been shown in our previous paper [2] that content-based detection is feasible, as well as that the employed audio comparison nevertheless requires high computational costs. This contribution describes an efficient method to identify SPAM calls with similar media data and presents a system to protect users in carrier networks.

This paper is organized as follows: the following section presents work related to SPIT (SPAM over Internet Telephony) mitigation and efficient audio identification. Afterwards the extraction of robust audio feature vectors and their quantization is explained. As a key contribution, we adapt a method for efficient index-based audio search to the SPIT scenario. The integration of the system in VoIP networks is described afterwards and finally the efficiency and performance of the proposed method is analyzed.

II. Related Work

A. SPIT Mitigation Frameworks

The definition of the problem and some basic approaches for SPIT mitigation are contained in RFC 5039 [1], e.g. black and white lists, reputation systems and Turing tests. In this connection, authenticated caller identities, e.g. as described in RFC 4474 [3], are of particular importance. In [4], a SPIT protection algorithm called *Progressive Multi Gray-Leveling* (PMG) is proposed. The call frequency of a caller is monitored and a call is blocked if the sum of the long-term and short-term gray-level of a caller exceeds a certain threshold. In the SPIDER project [5], SPIT callers shall be discovered using a detection and a decision layer. While the detection layer consists of modules which implement various tests to detect SPITters, the decision layer controls the selection of the modules and combines their results to make a decision. The project also proposes the use of audio signatures [6]. This is similar to the approach in this contribution, but the construction of the audio fingerprint and the integration of the audio analysis differs nevertheless.

The NEC SEAL system [7], [8] uses different layers, called stages, to identify SPIT. If a user cannot be clearly identified as SPITter or normal caller by the signaling analysis in stage 1, he has to pass a test in stage 2, e.g. to answer a CAPTCHA. In the following, some more recent approaches for SPIT mitigation are presented.

In [9] two methods are proposed, based on the detection of anomalies of selected call features (i.e., day and time of calling, call durations, etc.). The first method uses *Mahalanobis Distance* [10] to detect individual SPIT calls. The second method is designed to detect groups of SPIT calls by computing the *entropy* of call durations. This entropy computation can detect deviations from normal human call behavior that characterizes bulk SPAM.

The approach [11] uses three techniques to analyze original call records from one of the largest phone providers in North America: They use a new technique called *Loose Tie Detection (LTD)* to identify outliers based on social ties. SPITters cannot avoid making a large number of calls and most likely the calls will be short, because the callee will hang up very quickly. The

second technique is called *Enhanced Progressive Multi Gray-Leveling (EPMG)* to identify outliers based on call density and reciprocity. Reciprocal calls are characteristic for normal telephone traffic. The last technique is called *SymRank*, which is an adaption of the PageRank [12] algorithm. *SymRank* calculates a ranking of callers considering incoming and outgoing calls. This ranking is used as a measure of caller reputation. All three techniques compute overlapping sets of suspicious callers, but *LTD* seems to be the most feasible.

In [13], a three-layer approach is presented which involves signaling analysis, Text-To-Speech (TTS) and voice activity detection.

B. Audio Identification

A robust audio fingerprint for music identification has been developed by *Shazam* [14], [15] using spectral peaks. Some of these spectral peaks are regarded as anchor points whereby each anchor point has a target zone associated with it, in which other spectral peaks are located. Each anchor point is iteratively paired with the spectral peaks in its target zone and at each step a hash over the current pair and a time difference is computed. These hashes are used as feature classes.

The audio identification system *AudioID* [16] uses Low Level Descriptors (LLD) from the MPEG-7 audio standard. The audio features SFM (Spectral Flatness Measure) and SCF (Spectral Crest Factor) describe the flatness resp. the tonality of the signal's spectrum. Both features prove to be robust against common alterations. The feature classes are computed using vector quantization and form the audio fingerprint accordingly.

Another music identification system is *audentify!* [17], which was developed at University Bonn and continuously improved at Fraunhofer FKIE. As a result of a cooperation with FKIE it was possible to adapt the search method for the identification of telephone-SPAM (see section IV).

III. Audio Fingerprint

A. Feature Extraction

For a reliable identification of replayed calls, characteristic and robust audio features are extracted from the spectrum of the recorded audio data. The extracted P-dimensional feature vectors are not immediately suitable for an index-based search method. For indexing, a sequence V of feature vectors $(v(1), v(2), \ldots, v(N))$, where $v(i) \in \mathbb{R}^P$, is quantized into a feature document D, which is a series of feature classes.

To quantize P-dimensional spectral feature vectors, combinations of peak values are used. The quantization assigns each feature vector $v(i) = (v_1, v_2, \ldots, v_P)$ the position $p \in [1, P]$ with the maximum absolute value $|v_p|$. These positions are highly robust against disturbances and can be used for indexation. To minimize the probability of random matches and to enlarge the number of different feature classes, a set of n peak positions is combined and forms a specific pattern. There are P^n combination of peaks and the feature classes can hence be represented by the set $\mathcal{R} = \{1, 2, \ldots, P^n\}$.

B. Implementation

The window size for spectral feature extraction is 128 ms with an overlap of 96 ms. For six seconds of recorded audio, around $N = 175$ vectors are extracted. The resulting spectrum is band-limited from 330 Hz to 1.8 kHz and decomposed into $P = 21$ subbands using a Mel filter bank. For each vector the subband with maximum energy is determined. A combination of three maxima, each with a distance of five windows, forms a pattern that represents one of $P^3 = 9{,}261$ feature classes. Only energy-rich windows are significant and their associated feature vectors are used for peak combination. The resulting fingerprint is a set of up to N tuples (t, r) where $t \in \{1, 2, \ldots, N\}$ denotes the time parameter and $r \in \{1, 2, \ldots, P^3\} = \mathcal{R}$ the feature class.

IV. Searching in audio data

Generally speaking, searching in audio material is a non-trivial problem. As mentioned in section II-B, there are several approaches to efficiently identify music, but not specifically telephone speech. We subsequently summarize parts of [18] as required in our setting, focusing on the modeling of feature documents and the index-based matching strategy.

A. Feature Documents

The extracted audio features are represented as feature documents (audio fingerprints). A feature document D is a finite subset of $U := \mathbb{Z} \times \mathcal{R}$. The first component describes the time of occurrence of a peak pattern and the second component denotes the feature class. Hence one particular audio feature is a tuple $(t, r) \in U$.

A query $\mathcal{Q}$ is also a feature document and therefore a finite subset $\mathcal{Q} \subset U$, i.e. a set of audio features:

$$\mathcal{Q} = \{(t_1, r_1), \ldots, (t_m, r_m)\}. \tag{1}$$

B. Matching

First, the time shift of feature documents has to be defined:

$$D + t := \{(t + T, r) | (T, r) \in D\}. \tag{2}$$

Then a ($\mathcal{Q}$-)match is described as follows:

$$H(\mathcal{Q}) := \{(t, j) \,|\mathcal{Q} + t \subseteq D_j\}. \tag{3}$$

This means that $\mathcal{Q}$ is up to a time shift t a subset of the feature document D_j. For an elementary query $q = (t, r)$, we define $H(t, r) := H(\{q\})$. The following relation then holds:

$$H(t, r) = H(0, r) - t. \tag{4}$$

C. Index

Feature documents are stored in an index, which consists of *inverted files* $H(0, r) := \{(t, j) \,|(t, r) \in D_j\}$. The collection of inverted files constitutes the index $\mathcal{I}$:

$$\mathcal{I} := (H(0, r))_{r \in \mathcal{R}}\,. \tag{5}$$

For each feature class r, the index $\mathcal{I}$ gives the time positions t and the document numbers j, where the feature class occurs.

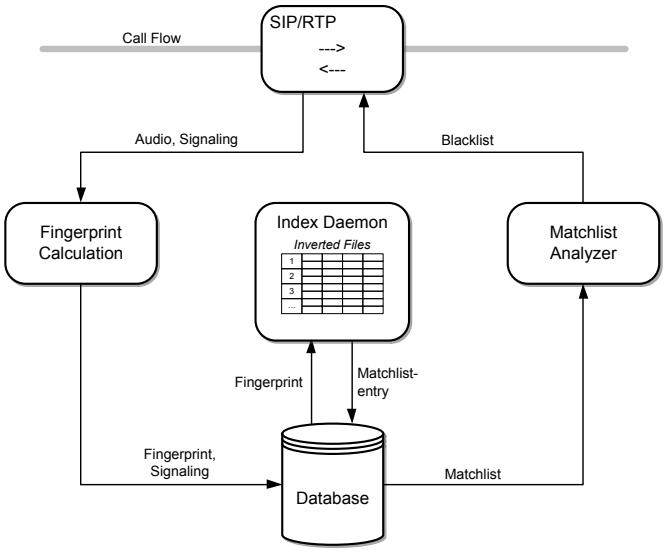

Fig. 1: Data flow diagram of VIAT.

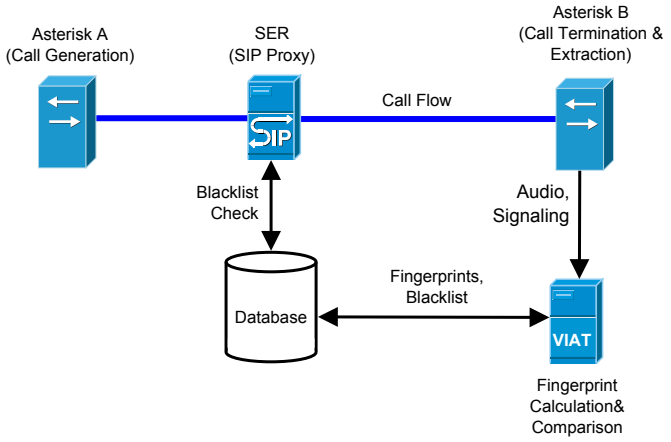

Fig. 2: Prototype II implementation.

D. Efficient Matching

Efficient, fault tolerant matching can then be performed in a dynamic programming setting, using a suitable combination of intersections and unions of inverted files, see [18] for details.

V. System Integration

A. Description

The data flow of the VIAT (Method for the Identification and Blocking of Telephone-SPAM) system is shown in Fig. 1. A Session Initiation Protocol (SIP) and Real-Time Transport Protocol (RTP) processing component extracts a copy of the first seconds of the caller's unencrypted audio data together with some signaling metadata. This component could be a Session Border Controller (SBC) or any other component that is directly involved in the handling of SIP and RTP data. However, extracting the data from a copy of the network stream would be more flexible as described in section V-C. The latter has the advantage that it does not affect the call flow in case of overload or any errors during the extraction.

The audio fingerprint computation works as described in section III-B. The fingerprints are stored in a database. The *Index Daemon* takes new fingerprints from the database and extends the inverted files. They are used for an efficient comparison of the fingerprints as described in section IV-D. If the new fingerprint matches with existing fingerprints, which means that the corresponding audio data is equal or similar, a database matchlist entry is produced. It consists of the fingerprint size, the actual number of mismatches, which do not exceed a defined threshold (see tables I and II), and the call identifier of the matching calls.

Another module, the *Matchlist Analyzer*, evaluates the results of the *Index Daemon*. Depending on the policy, e.g. based on the similarity and number of matching calls, it may put the conspicuous caller URI on the blacklist. The SIP and RTP processing component is then able to block new calls that originate from a blacklisted caller URI. The enforcement of a blacklist (and a whitelist) can also be done by a SIP proxy server.

B. Prototypes I and II

The above described procedure has been implemented as a prototype using open source components as depicted in Fig. 2. A VoIP call flow is generated with two Asterisk communication servers, whereby the first Asterisk server generates the calls and the second server answers the calls. The script-based call generator uses the Asterisk call file interface and allows the setup of complex call scenarios which simulate real telephone traffic. The answering communication server extracts the audio data and certain signaling data for the fingerprint generation and other processing of the system. The SIP proxy SER (SIP Express Router) routes SIP messages between both Asterisk servers and sends a negative response (603) to the SIP Invite request, if the caller's URI is on the blacklist. In a real scenario, the proxy could be used as a signaling gateway between local and foreign networks. Then, SPIT calls are already blocked at the network edge which would help to deal efficiently with possible SPIT waves.

In our prototype I [2], the feature vectors consisted of 16 Mel Frequency Cepstrum Coefficients (MFCC) and minimum distances were calculated between all fingerprints. This has been replaced in prototype II by higher-dimensional feature vectors and the index-based search method (see sections III and IV).

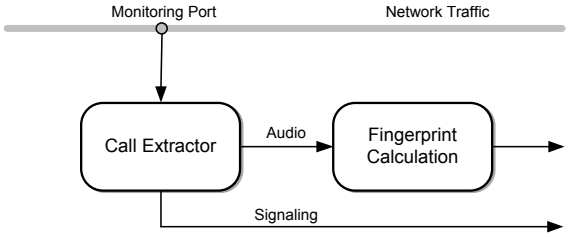

Fig. 3: Passive call extraction.

TABLE I: k-mismatch search in 4,926 audio files.

k [%]	$\bar{T}$ [ms]	RRR [%]	FRR [%]
0	8.07	7.90	0
20	58.78	35.95	0
40	192.61	72.34	0
60	420.56	91.53	$2.89 \cdot 10^{-5}$
80	700.85	98.68	0.01

TABLE II: k-mismatch search combined with fuzzy search in 4,926 audio files.

k [%]	$\bar{T}$ [ms]	RRR [%]	FRR [%]
0	9.60	7.95	0
20	69.45	44.89	0
40	220.20	81.97	0
60	479.92	96.47	$7.71 \cdot 10^{-4}$
80	822.05	99.76	0.13

C. Prototype III

In prototype III, we focus on the system integration and the overall system performance. One of the main improvements is the passive extraction of the signaling and audio data from a copy of the network stream (see Fig. 3). Prototype II however requires interfaces on a system which is actively involved in the processing of signaling and audio data.

The network stream is copied to a monitoring port. The *Call Extractor* processes SIP and RTP packets as a promiscuous network device. It contains a passive SIP stack and analyzes the relationships of all SIP user agents communicating over the network. The passive SIP stack works like a normal user agent SIP stack but does not actively participate in the communication. After decoding the RTP data into audio data it is transferred together with some metadata to the *Fingerprint Calculation* module. This architecture allows a seamless integration into an existing provider's environment. Further improvements concern the data transfer between the systems using network sockets and the optimizations of the data handling by holding fingerprints exclusively in memory.

VI. Test Results

All tests were performed on prototype II. We have analyzed the *hit rate* resp. the *error rate* of our audio fingerprint as well as the performance of our method with two different test scenarios: First, we used real audio material with a test corpus of moderate size to examine the recognition rate and to determine suitable search parameters. Then, we used randomly generated audio fingerprints to analyze the search speed with the previously determined parameters.

The tests were performed on a virtual PostgreSQL database server, running on a Six-Core AMD Opteron Processor 2431 with 16 GB RAM. The *Index Daemon* runs on the same server, using one of the six cores. A performance increase is possible with more processor cores.

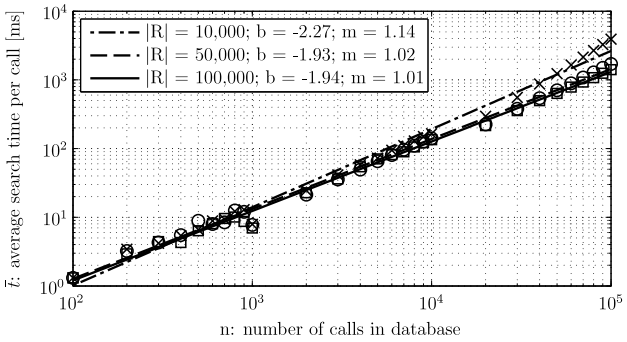

Fig. 4: Measured data of average search times and the corresponding regression lines $g_{\mathcal{R}}(n) = 10^b \cdot n^m$ using random data, 40% allowed mismatches and fuzzy search.

A. Test Scenario: Recognition Rate

As VoIP calls we used 4,926 speech audio files, whereby 4,726 files can be considered as normal calls and 200 files as SPIT. The SPIT calls derived from 10 audio files with 19 variations and degradations as described in [2, III-B]. The test corpus includes:

- 200 audio files from the test corpus [2, III-B].
- 574 audio files from the Kiel corpus [19].
- 4,152 audio files of German telephone dialogs from VerbMobil II corpus [20].

We varied the rate of allowed mismatches from 0% to 80%. The k-mismatch search and the fuzzy search [18] were evaluated. The tests yielded the search times $\bar{T}$, which were calculated by averaging the search times of all calls. Furthermore the tests gave the Right Rejection Rate (RRR) and the False Rejection Rate (FRR). Right rejection is the correct classification of SPIT, whereas false rejection means that a normal call is incorrectly recognized as SPAM call.

The results are shown in tables I and II. There is a good compromise between *hit rate* and *error rate* when the fuzzy search with 40% of allowed mismatches is employed. A smaller rate of allowed mismatches yields a faster search and still permits the identification of SPIT after a number of replayed calls.

B. Test Scenario: Search Speed

The randomly generated audio fingerprints have the following properties:

- The number of feature classes $|\mathcal{R}|$ varies between 10,000 and 100,000.
- The audio fingerprints have between 30 and 150 audio features.
- There are up to 100,000 audio fingerprints in the database.
- About 1% SPIT (single replay) with 0% to 80% mismatches.

In Fig. 4 the test results using randomly generated data are shown. The search times are significantly lower, if the number of feature classes is increased.

VII. Conclusion and Future Work

A short and robust fingerprint to identify telephone speech has been developed. The comparison of audio fingerprints was used to detect SPAM calls already after a few replays. It has been shown that an index-based search method is suitable for the efficient identification of audio fingerprints.

A working prototype with VoIP components has been implemented and it was shown that SPIT calls can be successfully detected and blocked. The overall system performance is still limited by the search speed. Currently, we can handle 890 calls per hour assuming $100{,}000$ calls in the database, $|\mathcal{R}| = 10{,}000$, 40% allowed mismatches and fuzzy search on the prototype system described above. With an assumed average call duration of two minutes, this yields $890 \text{ calls} \cdot \frac{2 \text{ min}}{60 \text{ min}} = 30$ Erlang.

Although the current system is a significant improvement compared to [2], the performance has to be further increased for an integration into a real carrier's environment. A possible approach would be to preselect suspicious callers before analyzing their audio data.

Further research on the audio fingerprint is sensible in order to increase the search speed. Since the search algorithm is faster when fewer mismatches are allowed, the fingerprint should be more robust for a higher matching rate. Furthermore, a higher number of feature classes would lead to shorter inverted files per feature class and permit a faster intersection. This can be realized with binary audio fingerprints (hashes) as in [21], but they are more susceptible to signal degradations.

For privacy reasons, it has to be verified that it is neither possible to reconstruct the spoken content nor to identify the speaker from the fingerprint. Furthermore, optimization methods from index-based web search engines could be adapted. The search, i.e. determining the intersection of the inverted files, can also be parallelized on more processor cores.

Our system analyzes the first six seconds of the caller's audio data. Using voice activity detection, an even smaller period and hence a shorter fingerprint might be sufficient. Currently, the majority of calls is non-SPIT, so that the search algorithm should be further optimized for a fast detection of regular calls.

Acknowledgments

A part of this work was carried out during an internship of the first author with Fraunhofer Institute for Communication, Information Processing and Ergonomics FKIE, Wachtberg, Germany. The authors would especially like to thank Frank Kurth for his support.

This work has been conducted within the research project VIAT, which is supported by the German Federal Ministry of Education and Research (BMBF), reference 1736X09.

References

[1] J. Rosenberg and C. Jennings, "The Session Initiation Protocol (SIP) and Spam," RFC 5039 (Informational), Internet Engineering Task Force, Jan. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5039.txt

[2] D. Lentzen, G. Grutzek, H. Knospe, and C. Pörschmann, "Content-based Detection and Prevention of Spam over IP Telephony - System Design, Prototype and First Results," *IEEE International Communications Conference (ICC) 2011*, Jun. 2011.

[3] J. Peterson and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)," RFC 4474 (Proposed Standard), Internet Engineering Task Force, Aug. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4474.txt

[4] D. Shin, J. Ahn, and C. Shim, "Progressive multi gray-leveling: a voice spam protection algorithm," *IEEE Network*, vol. 20, no. 5, pp. 18–24, 2006.

[5] Y. Rebahi, S. Dritsas, T. Golubenco, B. Pannier, and J. F. Juell, "A Conceptual Architecture for SPIT Mitigation," in *SIP Handbook: Services, Technologies, and Security of Session Initiation Protocol*, S. A. Ahson and M. Ilyas, Eds. Boca Raton, FL, USA: CRC Press, Inc., 2009, ch. 23, pp. 563–582.

[6] Y. Rebahi, S. Ehlert, and A. Bergmann, "A SPIT detection mechanism based on audio analysis," in *Proceedings of 4th International Mobile Multimedia Communications Conference MobiMedia 2008: July 7-8, 2008, Oulu, Finland*. ICST; ACM, 2008.

[7] J. Seedorf, N. d'Heureuse, S. Niccolini, and T. Ewald, "VoIP SEAL: A Research Prototype for Protecting Voice-over-IP Networks and Users," in *Konferenzband der 4. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V.(GI)*, A. Alkassar and J. Siekmann, Eds., Apr. 2008.

[8] N. d'Heureuse, J. Seedorf, and S. Niccolini, "A policy framework for personalized and role-based SPIT prevention," in *Proceedings of the 3rd International Conference on Principles, Systems and Applications of IP Telecommunications*, ser. IPTComm '09. New York, NY, USA: ACM, 2009, pp. 12:1–12:11. [Online]. Available: http://doi.acm.org/10.1145/1595637.1595653

[9] H. Sengar, X. Wang, and A. Nichols, "Thwarting Spam over Internet Telephony (SPIT) attacks on VoIP networks," *2011 IEEE Nineteenth IEEE International Workshop on Quality of Service*, pp. 1–3, Jun. 2011.

[10] P. C. Mahalanobis, "On the generalised distance in statistics," in *Proceedings National Institute of Science, India*, vol. 2, no. 1, Apr. 1936, pp. 49–55.

[11] H. K. Bokharaei, A. Sahraei, Y. Ganjali, R. Keralapura, and A. Nucci, "You can SPIT, but you can't hide: Spammer identification in telephony networks," *2011 Proceedings IEEE INFOCOM*, pp. 41–45, Apr. 2011.

[12] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine," *Comput. Netw. ISDN Syst.*, vol. 30, pp. 107–117, April 1998. [Online]. Available: http://dx.doi.org/10.1016/S0169-7552(98)00110-X

[13] C. Hongchang, C. Fucai, and L. Shaomei, "A Multilayered Fusion Method for SPITs Detection," *2011 Fourth International Conference on Intelligent Computation Technology and Automation*, pp. 30–33, Mar. 2011.

[14] A. L.-C. Wang, "An Industrial-Strength Audio Search Algorithm," *ISMIR 2003, 4th Symposium Conference on Music Information Retrieval*, pp. 7–13, 2003.

[15] A. L.-C. Wang and J. O. Smith III, "Methods for recognizing unknown media samples using characteristics of known media samples," 03 2008. [Online]. Available: http://www.patentlens.net/patentlens/patent/US_7346512/

[16] M. Cremer, B. Froba, O. Hellmuth, J. Herre, and E. Allamanche, "AudioID: Towards Content-Based Identification of Audio Material," in *Audio Engineering Society Convention 110*, May 2001.

[17] M. Clausen and F. Kurth, "A unified approach to content-based and fault-tolerant music recognition," *IEEE Transactions on Multimedia*, vol. 6, no. 5, pp. 717 – 731, Oct. 2004.

[18] F. Kurth and M. Müller, "Efficient Index-Based Audio Matching," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 382–395, Feb. 2008.

[19] Kiel University, "The Kiel Corpus of Read Speech, Vol. I." [Online]. Available: http://www.ipds.uni-kiel.de/publikationen/kcrsp.de.html

[20] Bavarian Archive for Speech Signals, "Verbmobil II - VM2." [Online]. Available: http://www.phonetik.uni-muenchen.de/Bas/BasVM2eng.html

[21] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," in *Proc. ISMIR*, vol. 2002, 2002, pp. 144–148.