A SPIT Detection Mechanism Based on Audio Analysis

Yacine Rebahi
Fraunhofer Fokus
Kaiserin Augusta Allee 31
10589 Berlin, Germany
+49 30 34637141
yacine.rebahi@fokus.fraunhofer.de

Sven Ehlert
Fraunhofer Fokus
Kaiserin Augusta Allee 31
10589 Berlin, Germany
+49 30 34637378
sven.ehlert@fokus.fraunhofer.de

Andreas Bergmann
Eleven GmbH
Hardenbergplatz 2
10623 Berlin, Germany
+49 30 520056159
Andreas.Bergmann@eleven.de

ABSTRACT

The Session Initiation Protocol (SIP) provides new means for establishing and maintaining IP based multimedia services. Similar to most of the Internet protocols, SIP might suffer from the spam problem. The latter refers to sending unsolicited information in bulk. In this paper, we describe a way of how analysis of the audio content can help to detect spam calls. After identifying the requirements for the audio analysis to be successful in detecting spam, an architecture suited for VoIP media is suggested. To prove the feasibility of our solution, the audio analysis technique is validated through various testing methods.

Categories and Subject Descriptors

I.5 [**Pattern Recognition**]

General Terms

Security, Performance, Reliability.

Keywords

Spam, SPIT, audio analysis, speech processing, signal processing

1. INTRODUCTION

The session Initiation Protocol (SIP) [1] is becoming the primary standard for managing IP multimedia sessions. Similar to most of the email protocols currently in use, SIP can suffer from the spam problem. The latter refers in general to any unsolicited information sent to any recipient without his consent. SIP is still an emerging technology - however, it is more reasonable to address the spam problem right now than wait until this problem becomes a serious threat.

This paper is organised as follows: Section 2 describes the state of the art in the use of audio analysis methods in order to face the spam problem. While section 3 gives a brief summary of our audio analysis technique, section 4 describes in details the corresponding architectural components. Section 5 shows the feasibility of our solution through various tests and scenarios. Section 6 concludes the paper.

2. RELATED WORK

The fight against VoIP spam, termed SPam over Internet Telephony (SPIT), has taken different shapes - however, all of the proposed anti-SPIT solutions fit in one of two categories: signalling based detection and content based detection. Although a lot of work has been dedicated to the development of mechanisms for mitigating SPIT at the signalling level (the SIP protocol), no real anti-SPIT activities can be mentioned when the media itself is concerned. The signalling based anti-SPIT solutions are described and assessed in [2] and they are not going to be discussed again in this paper. However, we will focus on the ones analysing the media content for detecting SPIT.

The company Empirix [3] is a vendor of several different network analysis products with the focus on voice traffic analysis. They announced in the late of 2006 a module for their product Hammer XPS that would be capable of blocking SPIT messages on the network level. Empirix' idea to prevent SPIT calls is to monitor every incoming VoIP call and check if the caller fits a "normal" caller profile. It is assumed that a SPITter places much more calls than a human caller. This way SPIT is identified by the amount of calls over a certain amount of time. In addition it was mentioned that the content of a call is also analyzed for "suspicious" patterns. Unfortunately no more information was given on this aspect. It seems that Empirix stopped the development and sales of the SPIT module as all references have been removed from their webpage.

Researchers at NEC Europe developed an anti-SPIT solution called VoIP Seal, which tries to find out whether a communication partner is a human being or a machine [4]. The assumption is that a human caller follows a so-called "conversation pattern", which consists of three phases: ringing, greeting and question and answer. A human caller normally waits before he starts to talk until the callee has answered the call and has finished his greeting. This solution assumes that SPIT software is not able to differentiate between a human callee and software answering the call. So the prediction is that SPIT software does not follow the communication pattern and plays the SPIT message during the "greeting & question" phase. Furthermore, VoIP Seal is a system which answers each call automatically and plays back a welcome message and a request to wait for a short time while the call is being "transferred". During the playback and a certain amount of time past the announcement, the system is recording the stream from the caller and analyses whether the caller is breaking the communication pattern due to the fact that he generates any kind of noise in that time frame. So if the caller transfers any kind of sound which is louder than a predefined threshold the call is identified as SPIT. If not and the caller keeps silent for a certain time frame the call will be transferred to the callee. To get better filtering results, a second step is proposed. If a call has not been clearly identified as SPIT or not, the software should playback another audio message which asks the caller a question that could be answered very briefly. The reply of the caller would then be analyzed whether it was "brief

Mobimedia 2008 July 7-9, 2008, Oulu, Finland.
Copyright 2008 ICST ISBN 978-963-9799-25-7
DOI 10.4108/ICST.MOBIMEDIA2008.3850

enough”. If an answer would take too long the call would be identified as SPIT.

The basis of VoIP Seal is similar to the one described later in this paper. VoIP Seal is a software solution, which answers calls and records the caller audio while playing back an audio message which asks human callers to wait for a short time. Recorded audio of the beginning of the incoming call is the basis for both solutions but the difference lies in the analysis techniques of the audio data.

VoIP Seal analyses the sound level of the caller audio (for being above a predefined level). If a caller is calling from a noisy environment he will probably identified as a SPITter. The second possible problem lies in SPITters being able to track the end of the callee welcome message and start their message afterwards such that the communication pattern would not be broken by the SPIT software. NEC wants to close that gap with their second described technique. This technique is a mixture of a challenge response technique and the assumption of how a human caller answers to question with a short answer. If a human caller is not able to understand the question or if he does not answer as expected he would be identified as a SPITter.

The audio analysis described in this paper eliminates the potential risks of VoIP Seal. It identifies a call only as SPIT if the same or mostly the same audio data is recorded more than once rapidly reducing the possibility of false positives.

3. THE AUDIO ANALYSIS TECHNIQUE

The audio analysis technique is intended to be used for fighting SPIT generated by automats. This implies that the SPITters place a large amount of calls with the same media content towards a destination network. We assume that individual, non-SPIT calls are (a) always coming from different callers, have (b) always different content and timing of the conversation (though somewhat similar greeting phrases) and (c) never occur in really large amounts within a defined time frame of a few hours.

The key requirement is to obtain a content-based identification code or signature per call which is small enough to be conveniently stored away and retrieved again from a database, yet contains enough information to identify two or more calls having the same content. If this identification code is seen more than once over a defined time span, the probability of it being a pre-recorded SPIT message is very high.

3.1 SIP-Signalling and data flow

The audio signature shall be calculated before the call reaches the receiver and while an announcement is played back to the caller. For calls to Interactive Voice Response (IVR) systems or voicemail its best to start the analysis as soon as the media channels are set up to make use of the announcements already played back by such systems. In a “live” environment, the flow diagram for an announcement plus call-transfer using the SIP REFER method [5] is depicted in Figure 1:

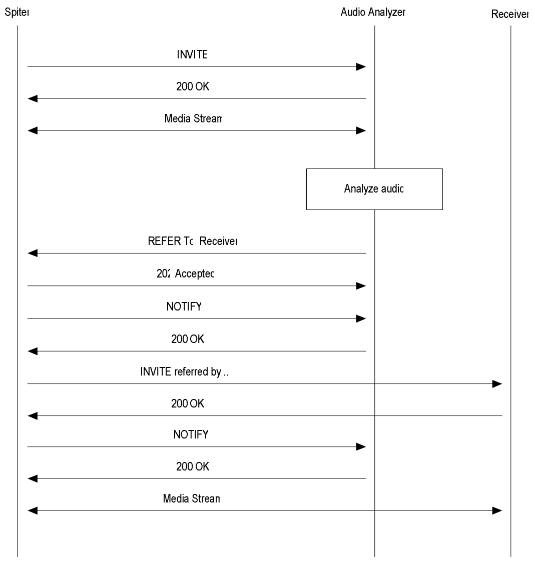


Figure 1: Message flow for the audio analysis technique

The analysis module answers a call and plays a pre-recorded message to the human caller. This announcement informs the caller of a short delay while his call is being processed. This does not only inform human callers, it also simulates a human receiver for SPITters, waiting for some kind of “answer” before sending their messages. While playing the information the analyser starts recording the caller audio. The recorded data will be fed to an algorithm calculating the calls signature which is checked against a database of all signatures seen over a recent time period. If the signature is unknown, this call is definitely individual, otherwise the call is likely to be SPIT. The algorithm describing this solution is depicted in Figure 2.

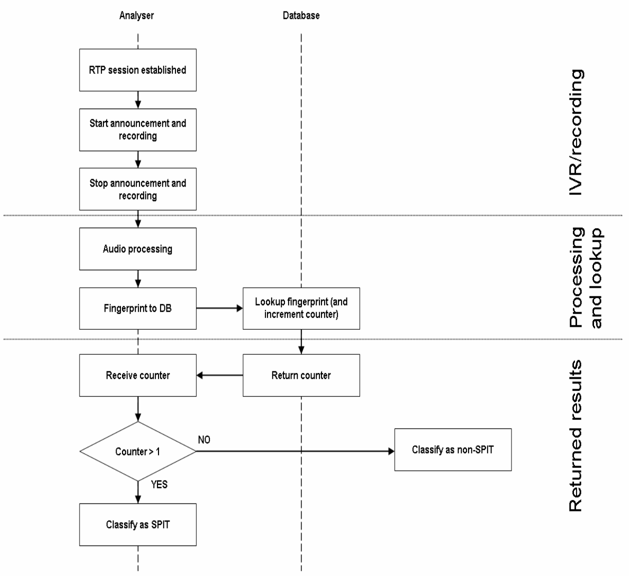


Figure 2: Analysis timeline

3.2 Problem domain and real-time media specific requirements

Automated SPIT calls will surely include a unique "link" of some sort - the recognition of such a link is beyond the scope of this software. The design paradigm the audio analysis module yet has in common with speech processing is feature extraction and separation, but not speech recognition on the level of the "textual" content. Similar greeting phrases for the majority of calls would render a purely speech recognition based approach inappropriate as well. However, the analysis shares the building blocks of speech processing technology.

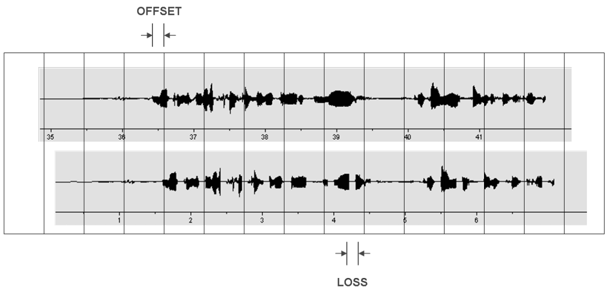


Figure 3: Comparison of a message subject to loss/offset, grid representing frame durations/packetisation

On the media layer all VoIP-calls are subject to (a) packet loss at the transport level, (b) packet loss concealment (PLC) algorithms used by most current CODECs and (c) offsets or alterations introduced through various processing stages or eventually added at will by the SPITter as a counter-measure to such an analysis solution. Therefore, the generation of a "signature" for each call must be to some extent robust against loss of information.

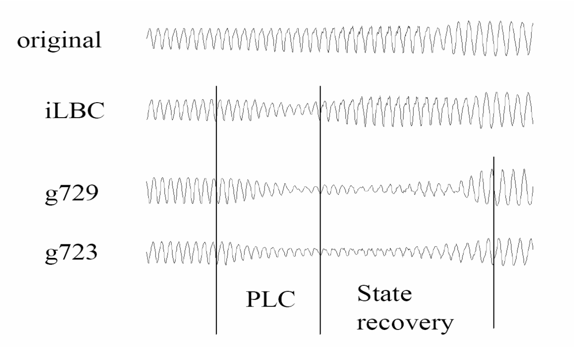


Figure 4: PLC comparison of three CODECs [6]

When addressing packet loss a loss rate of 5 to 10% seems a reasonable estimation for real-life networks with dialup users. Additionally, packets not lost but arriving too late to be processed (as determined by the RTP jitter buffer) are dropped. Dependent on the behaviour of the IVR/media server they are nonetheless not guaranteed to be recorded as a "gap". Second, CODECs feature different methods to conceal packet loss dependent on the technology they are based upon, e.g. lookahead requirements. The Asterisk VoIP PBX features a CODEC-independent PLC module, filling gaps by synthesising the last received frame's detected pitch. Therefore, pitch detection is a first candidate for designing the presented analysis solution.

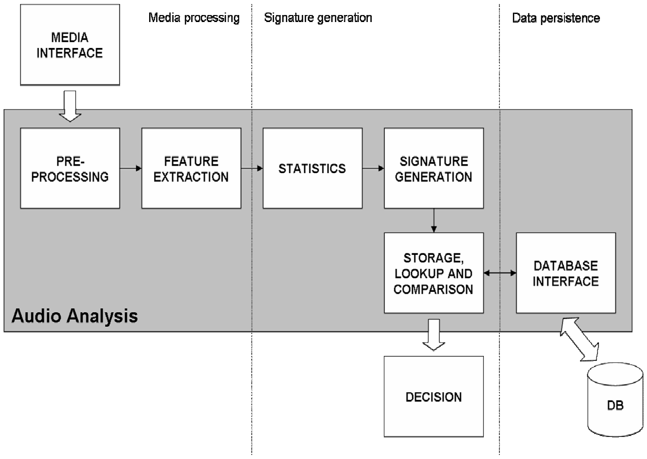


Figure 5: Audio analyser architecture

4. ARCHITECTURE

The different components of the audio analyser are depicted in Figure 5. We will give a detailed explanation of an analysis algorithm addressing above requirements along the description of the architectures' components.

4.1 Media interface and processing

The reception of an incoming call's media must result in a defined audio format, for telephony PCM (signed) 16-bit samples at a sampling rate of 8 kHz are a common starting point. Whereas on-disk storage is not necessarily required by the architecture, we record the media and pass audio file to the analyser. Though it is advantageous to capture VoIP audio directly at the RTP stack, which knows about aforementioned packet loss, we decide to choose this universal approach with a uniform "interface" over the implementation specifics of such a tightly-coupled solution.

The recording shall have a fixed length of about 10 seconds, let the assumptions that (a) successful SPIT needs to state its message quickly to grab the attention of the receiver, (b) announcements should not delay "real" calls too much, and (c) a minimum amount of data is required for the robustness of the analysis. Through experimentation this has proven to be a suitable duration.

As a primary feature set the pitch is derived from short-time spectra acquired over overlapping frames using a window of 30/32 ms. The maximum value in the spectrum depicts the most significant frequency. The obtained frequency is put on a scale of

16 units weighted to adapt to human perception using the MEL (melodic) scale.

To evaluate an additional set of data and as a cross reference, 16 cepstral coefficients are calculated. The cepstral analysis introduces additional computational costs but is able to give more differentiated results in detection of the pitch. Last, the energy level average of the call is monitored to sort out whether the average volume is at least above a minimal level, yielding a third, independent criterion.

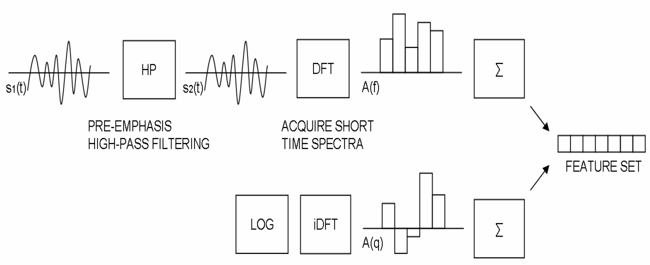


Figure 6: Signal processing building blocks

Again, all techniques are taken from common speech recognition methods and adapted to this solution via experimentation. The building blocks of the detector were shown in Figure 6 to give an overview whereas the details are out of the scope of this paper, for those building blocks, the feature extraction tool [7] was modified for our specific needs to build the resulting software solution. However, the bottom line is: The large amount of media data is reduced to few key features and shifted out of a time domain towards a more robust analysis level.

4.2 Signature generation and comparison

In the next stage we need to obtain a signature that allows for minimum-size storage and quick lookup because we are not going to store or compare any part of the original media in later steps. To be able to do quick lookups in the SPIT database, a “code” or “index” must be generated out of the results and stored along with the extracted data, timestamps and information about the caller.

To prove the feasibility of the solution a simple code generation method is used combined with basic statistics to achieve (a) robustness towards alterations and transport level loss, (b) separation of individual messages and avoidance of false positives and (c) a basic level of run-length independence to cope with slight offsets. The two feature sets' maxima are summed up over the whole sampling period - let N be the number of scanned audio frames and $\{a_0 \dots a_{15}\}$ the individual frames' maxima yields:

$$A_0 = \sum_{i=0}^{N} a_{0,i} \quad \dots \quad A_{15} = \sum_{i=0}^{N} a_{15,i}$$

To extract a 'signature' code S for vector A, for each element A_n greater than A_{n-1} by a minimum distance of Δ_{min} and if A_n is at least above a certain threshold A_{min} (to avoid counting in elements which would not have a significant influence on the result), assign the value '1', else '0', to S_n. For the starting element A_0, the value of 'zero' is assumed to its predecessor with index '-1'.

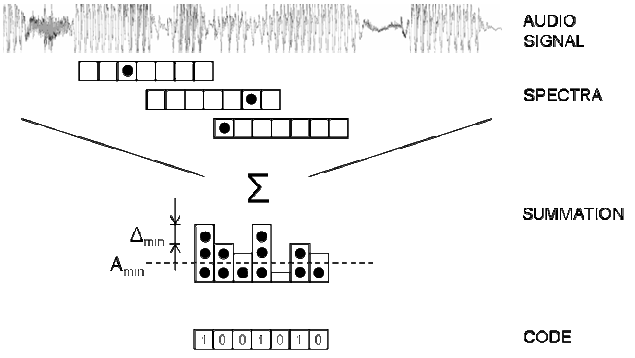


Figure 7: Code calculation

This “code” sets a coarse grid for queries to the database about likely candidates for comparable call contents. Two candidate data sets *x* and *y* having the same index are retrieved, they are matched[1] against each other via their deviation:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - x_i)^2}$$

If $\sigma < \sigma_{Trigger}$ classify the vectors as of the same origin. Again, through experimentation $\sigma_{Trigger}$ gives reasonably robust results at 10%.

4.3 Signature lookup

Using a persistent data store in the form of a relational database allows other analyser instances to share results and to generate reports. The basic table structure therefore consists of:

Unique ID	Code/ Index	Full Signature	Source information	Timestamp
45123	30289	a_0=19.0; a_1=55.2; a_2=45.2; a_3=30.0; …	user@domain.com:6060	2008 Mar 09 18:23:05
…	…	…	…	…

Figure 8: Database example

[1] The field of speech recognition knows of more feature vector comparison (and lookup) methods, as well as there are alternative methods for pitch detection like the "Average Mean Difference Function" etc. - yet above method provides a good pictorial clearness. Architecture-wise the comparison could be done on the server side, e.g. by a user-defined function inside a RDBMS, yet we want to treat the database as a simple data store to include the use of embedded databases in this architecture.

This signature is additionally non-reversible, avoiding aforementioned legal issues over permanent or short-term storage of private data.

5. TEST FRAMEWORK

To prove the feasibility of the technique, the next section presents test methods and results from the solutions’ test framework.

5.1 Audio test framework

Test data is used from the "SpeechDAT II" project [8], an european (fixed or mobile connection telephony) language samples database where standardized words and sentences are stored for different speakers (accent, age, gender, ...) as common test data for research and development. Three female voices' plus one male voice's recordings of a sequence of the same complete sentences are selected and assembled from this database. The test data is then adjusted to an equal starting point and a roughly similar volume and shall prove robustness against false positives.

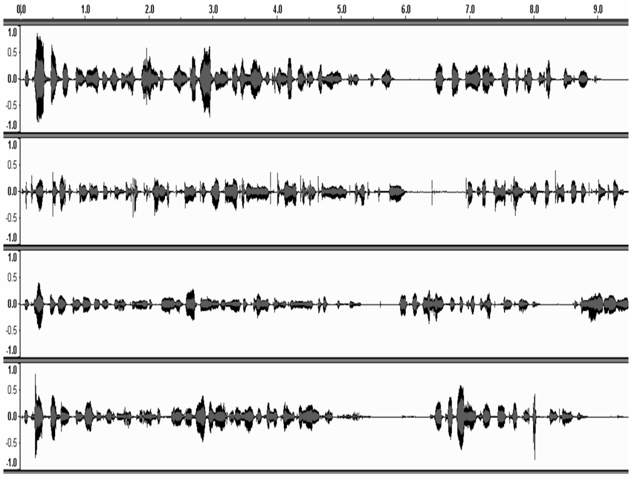


Figure 9: Waveforms of SpeechDAT samples

To illustrate the analysis of aforementioned pitch contour, the fundamental frequency (musical pitch) using enhanced autocorrelation as implemented in the software "audacity" of the same section as in the previous figure is shown below:

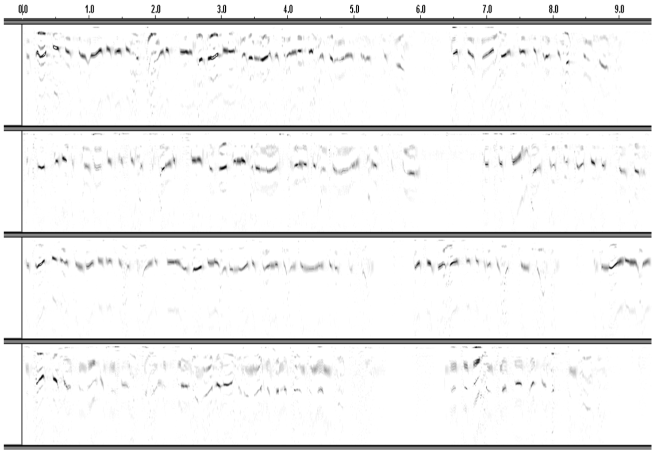


Figure 10: Pitch contour of SpeechDAT samples

The test application is built on top of the reference test program for the Internet Low Bit rate Codec (iLBC) [9] as described in RFC 3951. To simulate network transport predefined (though randomly generated) packet loss patterns which are stored in a so-called "channel" file for reproducibility of the tests are applied to the encoded speech recordings.

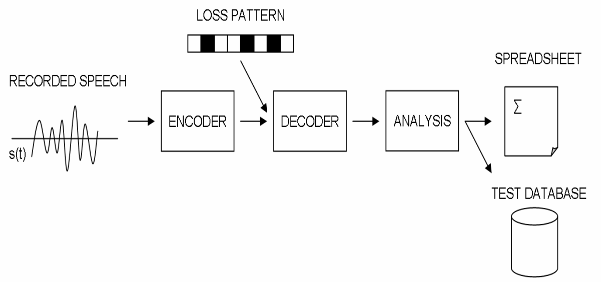
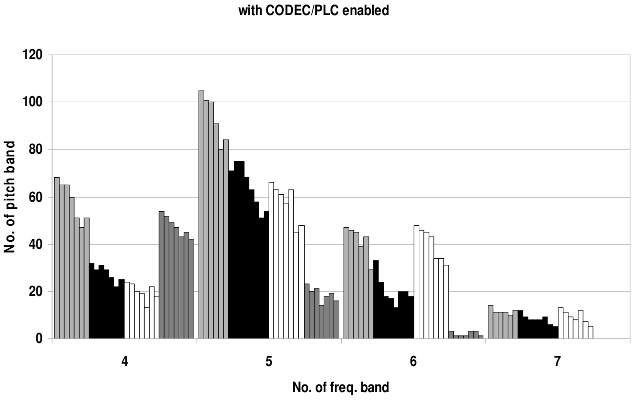


Figure 11: Audio test building blocks

Loss rates are varied from 0% to 30% when running the tests. The data is sampled after the recorded speech has passed the CODEC and its PLC algorithm. Again, to illustrate the tests we will explain a set of exemplary results.

The first five seconds of the same sentence spoken by the four different speakers are compared by the above method. The speakers are distinct by the colours light gray, black, white and dark grey. Seven different packet loss rates (0, 5, 10, 15, 20, 25 and 30%) are denoted left-to-right per speaker as a block of seven adjacent columns, the first columns being the reference the others are compared to.



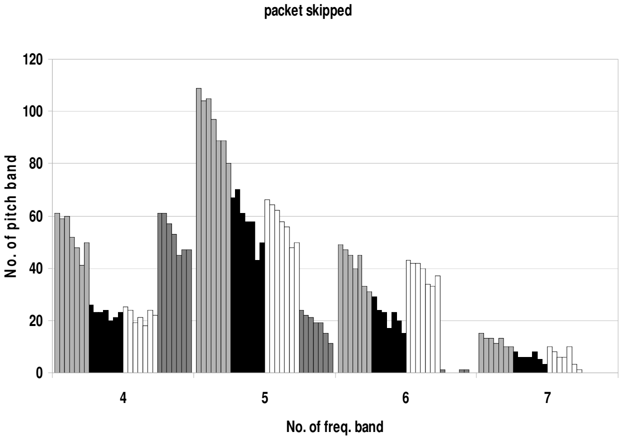


Figure 12: Speech sample comparison

The feature set above two figures is made up of the 16 different frequency bands – on the x-axis four consecutive frequency bands are zoomed out. The y-axis denotes the sum of the number of times a speakers pitch was inside a frequency band.

In the upper diagram iLBC's PLC algorithm is active, in the next figure packets lost according to the same pattern are skipped - modeling "dropped" packets. We can see the PLC's effectiveness in concealing the lost audio: without the loss concealment the columns heights are way more rapidly decreasing (left-to-right).

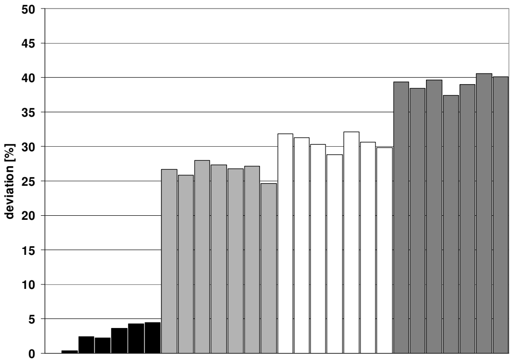


Figure 13: Standard deviation against first sample (with PLC)

Looking at the deviation of all samples against the first sample, every speakers "recording" has a distinct profile. Note that this is one of more extractable vectors to support the final result.

5.2 VoIP framework

To test the capabilities of the audio analysis technique embedded into an online environment a test bed is set up using the de-facto standard SIPp software [10]. SIPp allows acting as a user agent client and server respectively, the test configuration specific behaviour can be defined in a XML "scenario" files. The test specifications are:

- Test data is sent via pre-recorded RTP streams via a SIPp "media" scenario.
- SIPp has been extended for these tests to take a "loss rate" parameter for the RTP playback - a configurable percentage of packets is just not sent.

A SIPp client then sets up calls and plays back the various audio messages against the media server's announcement at different loss rates. Note that this is basically the same test scenario as in above section, only with a "real" RTP stack and the server load taken into account.

5.3 Real-life test results

To evaluate the technique with real-life data, a snapshot of a voicemail repository of 8545 messages is scanned. Pre-sorting via the index and restricting the search to complete recordings results in 246 candidates. The deep comparison of the candidates results in 53 messages classified as being of the same content, grouped by featuring 7 different index values without false positives.

Tracking the samples we found a repeated message from a (non-SPIT) announcement service, all other cases were "telephony" sounds, i.e. sequential beeps (Fig. 14, 15).

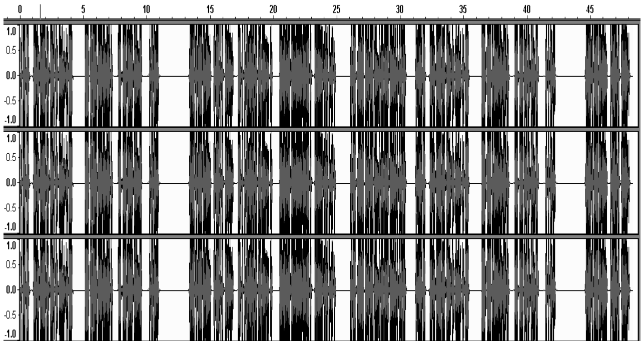


Figure 14: Announcements found in a voicemail repository

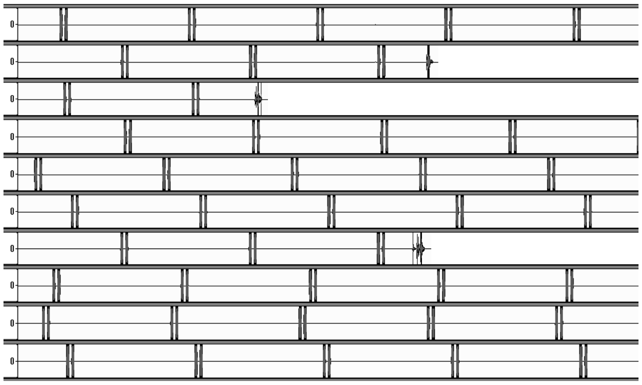


Figure 15: Telephony beeps

Above test results were validated through manually probing messages for being false positives. Choosing less tight parameters increases the set of candidates, but, again, starts to introduce false positives. However, the feasibility of the technique of an analysis of audio content in combination with a quick lookup method could be demonstrated.

6. ACKNOWLEDGMENTS

This work has been conducted within the EU-SME “Spam over Internet Telephony Detection sERvice (SPIDER)” project and was extensively discussed in a series of meetings and extensive exchanges over a period of more than one year. Special thanks to Dr. Carsten Olbert who helped in achieving the results under consideration.

7. CONCLUSION

This technique assumes that a SPITter generates a lot of automated messages in a given amount of time, yet through being content-based it is independent of testing for the same caller ID or the same IP address and works on a per-call basis (except for the first call, that is). As we see today, spammers control millions of computers in so called bot-networks, so we can assume that SPITters will as well be able to generate SPIT calls from thousands of different IP addresses and caller IDs. Though optimisation of parameters is a topic of further research, we present a content-based anti-SPIT technique tailored to the specifics of VoIP-calls.

8. REFERENCES

[1] J. Rosenberg et al, ”SIP: Session Initiation Protocol”, RFC 3261, June 2002

[2] S. Dritsas, J. Mallios, M. Theoharidou, G. F. Marias, and D. Gritzalis, “SPIT Identification Criteria and Anti-SPIT Mechanisms Evaluation Framework”, submitted on IEEE Global Telecommunications Conference (IEEE GLOBECOM 2007), Washington, D.C., U.S., Nov. 2007

[3] Empirix, link: http://www.empirix.com

[4] NEC, link: http://www.nec.de

[5] R. Sparks, “The Session Initiation Protocol (SIP) Refer Method”, RFC 3515, April 2003

[6] S. V. Andersen et Al, “iLBC Speech Codec and Payload Format (draft-andersen-ilbc-00.txt)”; PROCEEDINGS OF THE FIFTY-THIRD INTERNET ENGINEERING TASK FORCE, March 17-22, 2002, http://www.ietf.org/-proceedings/02mar/slides/avt-6.pdf.

[7] P. Fousek, P. Pollak, CTU Prague Speech Processing Group, "Additive Noise and Channel Distortion-Robust Parametrization Tool", Proceedings of the EUROSPEECH 2003, GENEVA; link: http://noel.feld.cvut.cz/speechlab/

[8] SpeechDAT II, link: http://www.speechdat.org/

[9] S. Andersen et Al, “Internet Low Bit Rate Codec (iLBC)”, RFC 3951, 2004

[10] SIPp, link: http://sipp.sourceforge.net

[11] F. D. Garcia et Al, ”Spam Filter Analysis” link: http://www.cs.ru.nl/flaviog/publications/spam-filter.pdf

[12] B. Mathieu et Al, ”SPIT Mitigation by a Network-Level Anti-Spit Entity”, In the proceedings of the 3nd Workshop on Securing Voice over IP, June 2006

[13] Y. Rebahi et Al, ”SIP Service Providers and the Spam Problem”, In Proceedings of the 2nd Workshop on Securing Voice over IP, June 2005

[14] Y. Rebahi et Al, ”SIP Spam Detection”, In the Proceedings of the IEEE International Conference on Digital Telecommunications (ICDT06), Cap Esterel, France, August 21-31, 2006

[15] Rosenberg et Al, ”The Session Initiation Protocol (SIP) and Spam”, draft-ietf-sipping-spam-02, March 6, 2006