A Non-speech Audio CAPTCHA Based on Acoustic Event Detection and Classification

Hendrik Meutzner and Dorothea Kolossa
Ruhr-University Bochum
Institute of Communication Acoustics
Cognitive Signal Processing Group
Universitätsstrasse 150, 44801 Bochum, Germany
Email: {hendrik.meutzner,dorothea.kolossa}@rub.de

***Abstract*—The completely automated public Turing test to tell computers and humans apart (CAPTCHA) represents an established method to prevent automated abuse of web services. Most websites provide an audio CAPTCHA—in addition to a conventional visual scheme—to facilitate access for a wider range of users. These audio CAPTCHAs are generally based on distorted speech, rendering the task difficult for untrained or non-native listeners, while still being vulnerable against attacks that make use of automatic speech recognition techniques.**

In this work, we propose a novel and universally usable type of audio CAPTCHA that is solely based on the classification of acoustic sound events. We show that the proposed CAPTCHA leads to satisfactorily high human success rates, while being robust against recently proposed attacks, more than currently available speech-based CAPTCHAs.

I. Introduction

A widely used approach to protect computer systems and web services from automated abuse is an interactive test that can distinguish between human users and computer programs. These tests are usually referred to as CAPTCHAs and they should represent a task that is easy for human users but difficult for computer algorithms. CAPTCHAs have now been employed for more than one decade [1]–[4] where research and development in this field has mainly focused on the design of visual tasks, e.g., the recognition of distorted characters or the classification of images.

In order to support visually impaired users and to enable the usage of devices with limited display capabilities, an acoustic verification scheme, i.e., an audio CAPTCHA, is typically provided in addition to the visual scheme. The majority of currently available audio CAPTCHAs is based on non-continuously spoken words, often limited to a small vocabulary (e.g., digits), that have been artificially distorted to harden the task for automatic speech recognition (ASR) systems.

A weakness of most available audio CAPTCHAs is that they exhibit a relatively disappointing trade-off between human usability and robustness against automated attacks [5]. This is due to the fact that ASR systems can achieve a very high performance—even under severe noise conditions—when the underlying vocabulary is small, whereas speech intelligibility for humans decreases when the signal-to-noise ratio is low. Thus, extending the vocabulary of the CAPTCHA can be beneficial to impede automated attacks. However, using large-vocabulary distorted speech not only creates a more challenging task for ASR but may also overtax untrained listeners or non-native speakers.

To overcome these problems, we propose an audio CAPTCHA that is based on the detection and classification of non-speech sounds, i.e., acoustic events. The advantage of using non-speech sounds is that it enables us to create a vast number of different acoustic scenarios that exhibit highly diverse spectro-temporal characteristics, rendering machine-driven attacks more difficult. Another benefit of using non-speech sounds is that the CAPTCHA becomes independent of language skills[1], making the CAPTCHA suitable for a broader group of users.

II. Related Work

Several audio CAPTCHAs have been proposed that are based on recognizing a sequence of distorted digits, utilizing linear or non-linear signal distortions. For example, the authors of [7] analyze 18 different types of signal distortions (e.g., additive white noise, echo, and signal bursts) and show that most of them can be used to increase the performance gap between humans and ASR systems. In [8], the authors investigate the differences between mixing-based and deletion-based methods, and find that the latter is more suitable for controlling the degree of difficulty. However, recent research has shown that especially digit-based audio CAPTCHAs can be easily broken at critically high success rates [5], [9]–[11] between 50 %–90 %.

In [12], the authors propose an audio CAPTCHA that uses an extended vocabulary. The advantage of this CAPTCHA is that it does not require conventional kinds of signal distortions in order to be secure, as it includes additional non-sense speech sounds that are highly confusing for ASR but not so much for human listeners.

Lazer et al. [13] design a CAPTCHA that is based on a series of environmental sounds that must be identified in real-time by the user, each time they occur in the signal. The CAPTCHA was found to be very usable, as the human success rate was measured to be above 90 %. However, the CAPTCHA

[1]Note that the CAPTCHA instructions could be automatically translated into the user's preferred language, e.g., using Google translate [6].

978-0-9928-6265-7/16/$31.00 ©2016 IEEE

TABLE I
OVERVIEW OF THE SYNTHESIS DATABASE, SHOWING THE SCENE AND EVENT CATEGORIES, THE SOURCE OF ORIGIN (CORPUS), AND THE NUMBER OF AVAILABLE EXAMPLES (EXAMPLES).

Type	Category	Corpus	Examples
Scene	Office	IEEE	2
Scene	Park/Nature	IEEE	2
Scene	Restaurant/Cafe	IEEE	2
Scene	Street/Bus/Car	IEEE	2
Scene	Shop/Supermarket	IEEE	1
Event	Alarm/Phonering	IEEE	11
Event	Barking dog	Two!Ears	20
Event	Crying baby	Two!Ears	20
Event	Cough/Clear throat	IEEE	45
Event	Doorclose/Doorknock	IEEE	33
Event	Laughter/Giggle	IEEE	34
Event	Dropping keys/coins	IEEE/freesound	14

security was not analyzed in this work but only discussed theoretically.

III. CREATING CAPTCHAS

We adopt the general idea of [13] and propose a non-speech audio CAPTCHA, posing the task of detecting and classifying a series of isolated acoustic events, embedded in a continuous environmental scene.

In contrast to [13], the proposed CAPTCHA does not require real-time interaction during playback and allows an unlimited number of replays. This has the advantage that the requirements for the user back-end are lower[2], which increases the compatibility of the proposed CAPTCHA for various devices and browsers. Furthermore, we assume that prohibiting replay might lead to increased user frustration upon mistakes, especially when dealing with relatively long CAPTCHA signals (e.g., durations around 45 s as in [13]).

Our proposed CAPTCHA is designed such that the average signal duration is relatively short at approximately 8.5 s , to create a task that is comparatively time-saving and competitive to most commercially available audio CAPTCHAs.

A. Synthesis Database

For synthesizing audio CAPTCHAs, we start by creating a database that is manually compiled from several corpora, i.e., the dataset of the IEEE AASP Challenge [14], the Two!Ears sound database [15] and a small number of sounds taken from freesound.org [16], [17]. The database is designed such that ambiguities between individual events are reduced or even avoided. This means that similar perceived sounds are grouped into the same category (e.g., clear throat and cough are grouped into one category). Table I provides an overview of the synthesis database, showing the scenes and events that are used for our approach. Note that the synthesis database could be arbitrarily adjusted, e.g., to make the CAPTCHA more robust against machine-learning-based attacks.

[2]Real-time interaction generally requires certain web technologies such as JavaScript, which are not necessarily enabled or available.

TABLE II
EVENT GUESSING PROBABILITIES FOR $N_E = 7$.

K_E	1	2	**3**	**4**	5	6	7
$P_E(N_E, K_E)$	$\frac{1}{7}$	$\frac{1}{21}$	$\mathbf{\frac{1}{35}}$	$\mathbf{\frac{1}{35}}$	$\frac{1}{21}$	$\frac{1}{7}$	1

B. Composition and Mixing

Each CAPTCHA consists of a sequence of events that are mixed with an environmental background scene. The individual events and scenes are sampled uniformly at random from the synthesis database by taking their individual frequency of occurrence into account. All events are clearly separated in time, using random offsets that are chosen from the interval $[0.5, 1.0]$ s. The events are mixed with the environmental background scene at a predefined event-to-scene ratio (ESR), which is defined as the ratio of the power of the respective event signal and the power of the corresponding part of the scene signal[3].

The number of events in each CAPTCHA is varied between 3 and 4 to achieve an acceptable low probability for simply guessing the CAPTCHA solution—assuming that there is no automated CAPTCHA solver available—while keeping the task as easy as possible for humans. The probability for guessing the events of a given CAPTCHA without prior knowledge is defined by the reciprocal of the binomial coefficient

$$P_E(N_E, K_E) = \binom{N_E}{K_E}^{-1} = \left(\frac{N_E!}{K_E!\,(N_E - K_E)!}\right)^{-1}, \quad (1)$$

where N_E is the number of event categories and K_E is the number of distinct events, present in the CAPTCHA.

Table II lists the event guessing probabilities for $N_E = 7$, when the number of events K_E is varied between 1 and 7.

Thus, for our proposed design (i.e., when $N_E = 7$ and $K_E \in \{3, 4\}$), assuming that the attacker knows N_E and K_E, the average probability for guessing the set of distinct events in the CAPTCHA is

$$P_E(7, 3) = P_E(7, 4) = \frac{1}{35} \approx 0.03\,, \quad (2)$$

which is below the often theoretically considered maximum allowed success rate of 5 % for an attack (e.g., [4], [9]).

IV. EVALUATION

The proposed CAPTCHA is evaluated with respect to security and usability aspects. We assess the security of the CAPTCHA, i.e., the robustness against bots, by means of a simulated attack, using common methods that were recently applied for breaking several commercially available audio CAPTCHAs (e.g., [9]–[11]). Furthermore, we assess the human usability by conducting a large-scale listening experiment using crowdsourcing tests, which have proven to represent a suitable alternative to laboratory-based tests when dealing with the evaluation of audio signals [18], [19].

[3]The definition is similar to that of the signal-to-noise ratio (SNR).

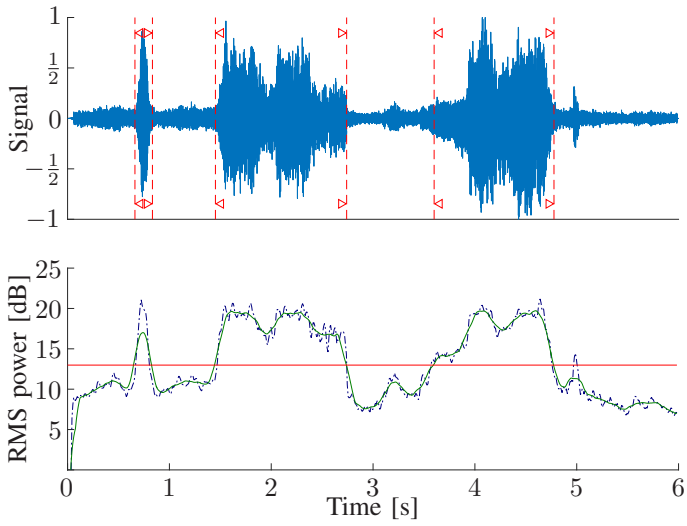

Fig. 1. Example of the proposed audio CAPTCHA showing the waveform (top) and the short-time RMS power (bottom). The segment boundaries of the events are depicted by vertical red lines (dashed). The short-time RMS curve and its smoothed version are shown by a blue semi-dashed and green solid line, respectively. The power threshold γ is given by a horizontal red line. The example contains the event sequence ("Dog", "Alert", "Baby") mixed at 15 dB ESR with the environmental scene "Restaurant".

A. Security

In order to assess the CAPTCHA security, we simulate an attack by using an automated CAPTCHA solver that estimates the respective scene and event labels, based on the audio signal. For this evaluation, we consider an advanced attacker that has fundamental knowledge about signal processing and machine-learning and the resources to create a labeled training data set, comprising a sufficiently large sample size. The CAPTCHA solver is based on a two-stage approach that was—in a similar setting—successfully applied to break a wide range of commercially available audio CAPTCHAs [9]–[11].

Figure 2 shows the block diagram of the implemented CAPTCHA solver. The CAPTCHA signal $x(t)$ is first divided into disjoint signal segments $s_i(t)$ that contain the individual acoustic events, where i denotes the i-th event in the CAPTCHA. The segmentation is based on a smoothed version of the short-time root mean square (RMS) power of $x(t)$

$$\tilde{p}(k) = \frac{1}{2M+1} \sum_{m=-M}^{M} p(k-m) \tag{3}$$

where k represents the frame index and M is the width of the moving average filter. The RMS power is given by

$$p(k) = 10 \log_{10} \sqrt{\frac{1}{L} \sum_{l=0}^{L-1} x^2(lR+L)}\,, \tag{4}$$

with L, R representing the frame length and the frame shift, respectively.

A segment $s_i(t)$ is then defined by those positions of $\tilde{p}(k)$ that exceed a certain power threshold γ. Such a threshold can either be defined manually by inspecting a set of audio

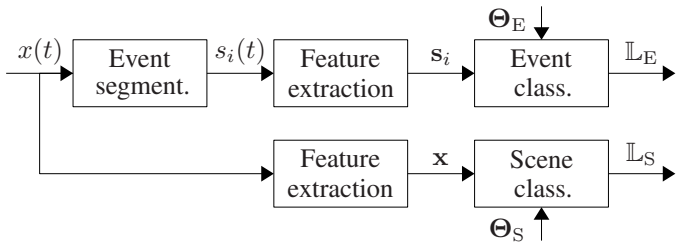

Fig. 2. Block diagram of the CAPTCHA solver.

signals or automatically, by using some heuristic. For the given CAPTCHAs, we found that using the mean value of $\tilde{p}(k)$ yields stable segmentation results. Furthermore, the parameter M has been optimized by maximizing the segmentation accuracy (cf. Eq. 7) on the training set. The segmentation process is visualized by Fig. 1.

After segmentation, a feature vector $\mathbf{s}_i$ is computed for each event signal. In addition, a global feature vector $\mathbf{x}$ is computed for the entire signal $x(t)$ to represent the environmental scene.

We compare the performance of two different feature types, namely temporally averaged Mel frequency Cepstral coefficients (MFCCs) and Cepstral modulation ratio regression (CMRARE) parameters, where the latter has the advantage of being independent of the long-term signal power, which is an important property for scene and event classification [20]. For MFCC features, we compute the first 13 static coefficients using a window length of 25 ms and a frame shift of 10 ms. For computing the CMRARE features we follow the approach in [20]. Using a polynomial order of 3 and concatenating the regression coefficients of the first 3 modulation bands then results in a 12-dimensional vector for the CMRARE features.

Both feature vectors, i.e., $\mathbf{s}_i$ and $\mathbf{x}$, are then classified by means of a linear discriminant analysis using an individual model Θ, to estimate the set of events $\mathbb{L}_\mathrm{E}$ and scenes $\mathbb{L}_\mathrm{S}$.

All experiments are conducted under matched conditions, i.e., using the same event-to-scene ratios for training and classification. Furthermore, the training is based on ideally segmented events, using the oracle information of the segment boundaries[4].

B. Usability

We assess the human usability via crowdsourcing tests at CrowdFlower [21]. The test participants[5] were asked to listen to the audio CAPTCHAs and to select those events and scenes that they had perceived. Figure 3 depicts the test interface as it was used for solving a single audio CAPTCHA. Each test participant provided a response for 10 audio CAPTCHAs and the overall number of participants was 800.

C. Metrics

For our evaluation, we consider the classification performance for events and scenes individually. It is important to note that the event classification performance does not depend

[4]This information would be very expensive to obtain for a real attacker.
[5]Referred to as contributors on CrowdFlower.

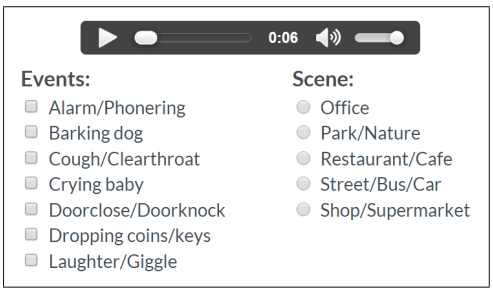

Fig. 3. Test interface of the listening experiment.

on the order or the frequency of individual events, as it would be the case for most conventional audio CAPTCHA that are based on word sequences.

a) Event classification accuracy: The event classification accuracy is given by

$$A_{\mathrm{E}} = 1 - \max\left(|\mathbb{T}_{\mathrm{E}} \setminus \mathbb{E}_{\mathrm{E}}|, |\mathbb{E}_{\mathrm{E}} \setminus \mathbb{T}_{\mathrm{E}}|\right) , \tag{5}$$

where $\mathbb{T}_{\mathrm{E}}$, $\mathbb{E}_{\mathrm{E}}$ are the sets of true and estimated events, respectively. $|\cdot|$ denotes the set cardinality and $\setminus$ is the set difference. Note that the $\max(\cdot)$ operation is performed, as it is not possible to distinguish between substitution errors or the simultaneous occurrence of the same number of deletion and an insertion errors.

b) Scene classification accuracy: The scene classification accuracy is simply computed by comparing the estimated scene $\mathbb{E}_{\mathrm{S}}$ with the true scene $\mathbb{T}_{\mathrm{S}}$[6]

$$A_{\mathrm{S}} = \begin{cases} 1 & \text{if } \mathbb{T}_{\mathrm{S}} = \mathbb{E}_{\mathrm{S}} , \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

c) Event segmentation accuracy: We compute the segmentation accuracy by comparing the number of estimated segments $\hat{N}_{\mathrm{E}}$ with the true number of segments N_{E}

$$A_{\Lambda_{\mathrm{E}}} = \frac{\hat{N}_{\mathrm{E}}}{N_{\mathrm{E}}} . \tag{7}$$

V. Results

We start by presenting the results of the security analysis. Table III shows the classification performance for a varying number of training examples, where we can see that the classifier achieves a very high accuracy on the scenes ($\geq$ 98 %), even when using a relatively small number of training examples. The event classification accuracy increases about linearly between 50 and 800 training examples, where the maximum is given by 36.10 %.

Table IV shows the performance of the segmentation stage. The results indicate that the segmentation performance decreases about linearly with decreasing ESR.

A comparison between different features and ESR conditions is given by Tab. V. The results show that the MFCC features lead to higher scene classification accuracies as compared to the CMRARE features, where the latter perform better

[6]There is only one choice for each scene, by what deletion and insertion errors can not occur.

TABLE III
Average classification accuracy (in percent) for a varying number of training examples, computed using a fixed test set of 1000 CAPTCHAs, each consisting of 3 events. The scores are shown separately for both classifiers (Scenes, Events) and they are based on 10 dB ESR when using oracle segmentation.

# training	50	100	200	400	800	1600
Scenes	98.00	98.80	99.10	99.10	99.10	**99.20**
Events	30.60	32.40	33.00	35.00	**36.10**	34.90

TABLE IV
Performance of the segmentation stage. The scores are based on a fixed training set, comprising 800 examples and a fixed test set, comprising 1000 examples.

ESR [dB]	0	5	10	15
Segmentation accuracy [%]	39.40	57.00	67.50	71.90

for event classification. In addition, the event classification accuracy can be improved by approximately 8.5 % (averaged over all features and ESRs) when using oracle segmentation instead of the power-based segmentation approach.

The results of the listening experiment are given by Tab. VI. It can be seen that the scene classification accuracy improves for lower ESR conditions whereas the event classification accuracy improves for higher ESR conditions. The maximum scene classification accuracy was found to be 54.70 % and the maximum event classification accuracy is 82.60 %.

A. Summary and Comparison

Our results show that the human listeners perform better in classifying noisy sound events than in classifying environmental scenes, whereas a machine-learning-based attack shows superior performance on the latter task. As a result, the CAPTCHA can be simplified to only asking for the sound events and not for the environmental scene. However, our results show that mixing the sound events with the environmental

TABLE V
Results of the simulated attack for MFCC (a) and CMRARE (b) features, showing the classification accuracy in percent. The scores are shown separately for both classifiers (Scenes, Events). (Events*) shows the results when using oracle segmentation. The training set comprises 800 examples and the test set consists of 1000 examples.

ESR [dB]	0	5	10	15	Avg.
Scenes	**99.95**	99.90	99.30	97.40	99.14
Events	10.40	16.40	22.70	**27.00**	19.13
Events*	18.75	26.55	33.35	**35.85**	28.63

(a) MFCC features

ESR [dB]	0	5	10	15	Avg.
Scenes	**97.10**	94.70	92.10	88.70	93.15
Events	10.05	17.70	27.10	**36.95**	22.95
Events*	16.55	25.00	35.15	**45.50**	30.55

(b) CMRARE features

TABLE VI
RESULTS OF THE LISTENING EXPERIMENT, SHOWING THE CLASSIFICATION ACCURACY FOR SCENES AND EVENTS IN PERCENT. EACH SCORE IS BASED ON A FIXED TEST SET OF 2000 CAPTCHAS.

ESR [dB]	0	5	10	15	Avg.
Scenes	**54.70**	52.25	51.15	47.95	51.51
Events	56.45	70.45	78.45	**82.60**	71.99

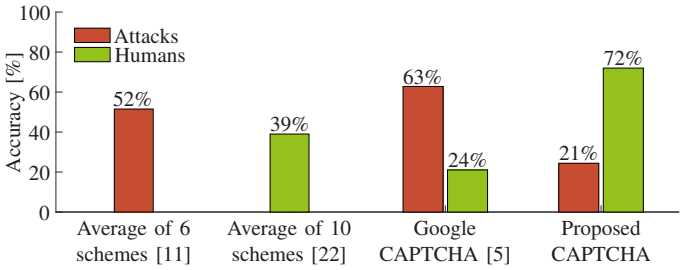

Fig. 4. Performance comparison between recently investigated commercially available audio CAPTCHAs and our proposed design.

scene is highly beneficial for lowering the performance of the automated CAPTCHA solver, especially when it makes use of a signal segmentation stage. The task in the proposed CAPTCHA is thus to detect and classify the set of events that are mixed into an environmental background scene.

We compare the performance of our proposed CAPTCHA with some commercially available schemes in Fig. 4. The first bar shows the average accuracy of 6 popular speech-based CAPTCHAs based on the attack of Bursztein et al. [11]. The second bar shows the human accuracy averaged over 10 popular speech-based CAPTCHAs analyzed by Bigham et al. [22]. The third bar group corresponds to the security and usability study of Google's quasi-standard "reCAPTCHA" conducted by Meutzner et al [5]. It can bee seen that the reported human accuracy is between 24 % and 39 %, which is below the accuracy of 52 % and 63 % achieved by recent attacks. The scores for our proposed CAPTCHA show the event classification accuracy, averaged over all conditions of Tab. V. It can be seen that this new CAPTCHA yields a clearly better trade-off between usability and security (72 % for humans vs. 21 % for the attack) than the commercially available CAPTCHAs.

VI. CONCLUSION

We have proposed a non-speech audio CAPTCHA based on the detection and classification of acoustic sound events that are mixed with an environmental scene. The CAPTCHA has been evaluated with respect to security, using a simulated attack, and regarding usability, using a large-scale listening experiment.

We can conclude that the proposed non-speech CAPTCHA represents a suitable alternative to conventional speech-based schemes in that it is independent of language skills and yields a good trade-off between human usability and robustness against automated attacks. The human success rate for our proposed design shows a relative improvement of 85 % when comparing with the average results that were reported for a wide range of commercially available schemes. Furthermore, a popular attack strategy that has been applied to break a large number of different audio CAPTCHAs, achieving more than 50 % on average, was only able to achieve a moderate success rate of 21 % for our proposed design.

ACKNOWLEDGMENT

This research was supported by the DFG Research Training Group GRK 1817/1.

Some examples of the proposed CAPTCHA are provided at: http://www2.ika.rub.de/KSV/captchas/eccaptcha.zip. The complete set of CAPTCHAs used for our experiments can be made available upon request.

REFERENCES

[1] L. von Ahn, M. Blum, N. J. Hopper, and J. Langford, "CAPTCHA: Using Hard AI Problems for Security," in *Proc. EUROCRYPT*, 2003.
[2] L. von Ahn, M. Blum, and J. Langford, "Telling Humans and Computers Apart Automatically," *CACM*, 2004.
[3] K. Chellapilla, K. Larson, P. Simard, and M. Czerwinski, "Computers beat humans at single character recognition in reading based human interaction proofs (HIPs)," in *Proc. 2nd Conference on Email and Anti-Spam*, 2005.
[4] K. Chellapilla, K. Larson, P. Y. Simard, and M. Czerwinski, "Building Segmentation Based Human-Friendly Human Interaction Proofs (HIPs)," in *Human Interactive Proofs*. Springer, 2005.
[5] H. Meutzner, V.-H. Nguyen, T. Holz, and D. Kolossa, "Using Automatic Speech Recognition for Attacking Acoustic CAPTCHAs: The Trade-off between Usability and Security," in *Proc. ACSAC*, 2014.
[6] Google Inc., "Google Translate," https://translate.google.com.
[7] G. Kochanski, D. P. Lopresti, and C. Shih, "A reverse turing test using speech," in *Proc. INTERSPEECH*, 2002.
[8] T. Nishimoto and T. Watanabe, "The comparison between the deletion-based methods and the mixing-based methods for audio CAPTCHA systems," in *Proc. INTERSPEECH*, 2010.
[9] J. Tam, J. Simsa, S. Hyde, and L. von Ahn, "Breaking Audio CAPTCHAs," in *Proc. NIPS*, 2008.
[10] E. Bursztein and S. Bethard, "Decaptcha Breaking 75% of eBay Audio CAPTCHAs," in *Proc. WOOT*, 2009.
[11] E. Bursztein, R. Bauxis, H. Paskov, D. Perito, C. Fabry, and J. C. Mitchell, "The Failure of Noise-Based Non-Continuous Audio Captchas," in *Proc. IEEE Symposium on Security and Privacy*, 2011.
[12] H. Meutzner, S. Gupta, and D. Kolossa, "Constructing secure audio captchas by exploiting differences between humans and machines," in *Proc. CHI*, 2015.
[13] J. Lazar, J. Feng, T. Brooks, G. Melamed, B. Wentz, J. Holman, A. Olalere, and N. Ekedebe, "The SoundsRight CAPTCHA: An Improved Approach to Audio Human Interaction Proofs for Blind Users," in *Proc. CHI*, 2012.
[14] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: An ieee aasp challenge," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*, 2013.
[15] Two!Ears, "The Two!Ears Project," http://www.twoears.eu.
[16] F. Font, G. Roma, and X. Serra, "Freesound Technical Demo," in *ACM International Conference on Multimedia*, ACM. ACM, 10 2013.
[17] http://www.freesound.org, "freesound.org (as of December 2015)."
[18] M. Eskenazi, G.-A. Levow, H. Meng, G. Parent, and D. Suendermann, *Crowdsourcing for Speech Processing: Applications to Data Collection, Transcription and Assessment*, 1st ed. Wiley Publishing, 2013.
[19] M. K. Wolters, K. Isaac, and S. Renals, "Evaluating speech synthesis intelligibility using Amazon Mechanical Turk," in *SSW*, 2010.
[20] R. Martin and A. Nagathil, "Cepstral modulation ratio regression (CMRARE) parameters for audio signal analysis and classification," in *Proc. ICASSP*, April 2009.
[21] CrowdFlower, Inc, "CrowdFlower," 2015, http://www.crowdflower.com.
[22] J. P. Bigham and A. C. Cavender, "Evaluating Existing Audio CAPTCHAs and an Interface Optimized for Non-visual Use," in *Proc. CHI*, 2009.