ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-7605-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICASSP39728.2021.9414135

ATTACKING AND DEFENDING BEHIND A PSYCHOACOUSTICS-BASED CAPTCHA

Chih-Hsiang Huang, Po-Hao Wu, Yi-Wen Liu, Shan-Hung Wu

Department of Electrical Engineering, National Tsing Hua University, Taiwan
Department of Computer Science, National Tsing Hua University, Taiwan

ABSTRACT

This paper proposes a novel audio CAPTCHA system that requires a user to respond immediately after hearing a short and easy-to-remember cue in its mixture with background music. Potential attacking paths based on cross correlation (CC) and sound event detection (SED) are implemented to test the security of the system. Then, two defending measures based on phase-modification of the cue and audio watermarking are established against CC and SED-based attacks, respectively. Human subjects were recruited to test the system and the results indicated that the subjects can pass the proposed audio CAPTCHA >90% of the times. In contrast, the proposed defending measures suppress the passing rate of both attacks to 8.2% and 0.4%, respectively.

Index Terms— Audio CAPTCHA, Watermarking, Signal Decorrelation, Sound Event Detection

1. INTRODUCTION

In order to prevent malicious accessing to websites from bots or attackers, Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) is often implemented at the entry. Although World Wide Web Consortium (W3C) has announced Web Content Accessibility Guidelines (WCAG), most of the websites lack of alternative ways for those who are vision-impaired, deaf or dyslexic to textual tests to complete the CAPTCHA. In this research, we develop a novel audio CAPTCHA system.

Existing audio-based CAPTCHA could be classified into two categories: speech CAPTCHA and acoustic CAPTCHA [1]. Speech CAPTCHA is designed based on the human ability to produce or recognize speech. By requiring the user to read out a given sentence, Gao et al. [2] developed a system that checked whether the sentence was spoken or synthesized. Other systems require the user to listen. Kochanski et al. [3] added noise or distorted the signal to make it hard for bots to recognize the contents automatically. A quite peculiar system was designed to utilize sounds that are only meaningful for machines but nonsense for human [4, 5]; therefore, if the answer is correctly entered, the "user" must be a bot.

Acoustic CAPTCHA relies on the human ability to detect and recognize sound events. Meutzner et al. [6] proposed a CAPTCHA based on sound event and scene classification. Three or four kinds of sound events were mixed with a background scene. Users were asked to pick up the correct sound events and scene after listening to a sound clip. Lazar et al. [7] designed CAPTCHA only based on sound event recognition. One challenge was composed of two to four target sounds with remaining decoy sounds. Target and decoy sounds were lined up in a series with spoken delimiters in between. When the challenge was played, the user should press the space bar intuitively immediately after hearing the target.

The main difference between these two works is the allowed response time when users interact with the CAPTCHA. The work in [7] required the user to give real-time interaction, while the work in [6] allowed the user to replay the audio clip with unlimited chances. There are pros and cons in both practices. Unlimited times of replay increases the confidence of the users to finish the CAPTCHA. However, although the users are given plenty of time to resolve the CAPTCHA, attackers or bots also take advantage of it. In contrast, the work in [7] required the users to interact with the CAPTCHA in real time, which built the first defensive gate against offline attack. However, this kind of real-time interactive CAPTCHA should be carefully designed to avoid frustrating the users while still maintaining the security to a certain extent.

2. THE PROPOSED SYSTEM

The proposed audio CAPTCHA is inspired by the work mentioned above [6, 7]. It consists of two stages: the instruction stage and the answering stage. The users are allowed to check a short-duration target sound several times in the instruction stage before turning to real-time interaction in the answering stage. The task for the users is to identify the target sound that is mixed with background music (BGM). We refer to the short-duration target sound as 'CUE'. During the instruction stage, the user is allowed to listen to CUE unlimitedly until becoming confident in identifying it. After the user presses the 'Play' button, our system would move on to the answering stage; the user then to respond immediately upon hearing any instance of the CUE while the BGM is continually played.

We created a database with acoustic variety so a comprehensive analysis of the proposed audio CAPTCHA could be conducted. Different styles of BGMs were collected from

Youtube, and CUEs were obtained from Freesound.org.[1] In total, 54 BGMs and 49 CUEs were collected with a sampling rate of 16kHz. Before mixing each BGM with each CUE, we randomly cropped the original BGMs into 10-second clips and manually cropped 1-second CUEs so as to preserve the most ear-catching part. At every round of mixing, the same BGM was randomly cropped for four times and the number of occurrences of the CUE was 2, 3 or 4. Thus, $54 \times 49 \times 4 = 10584$ audio CAPTCHAs were created.

2.1. Attacking paths

In the following discussion, we define the person with malicious intention to illegally access the CAPTCHA as an 'attacker'. According to the design of our CAPTCHA, two intuitively attacking paths are studied. The first one is a signal processing-based attack which assumes that the attacker will fetch the target CUE in the instruction stage. Once the attacker has this reference CUE, the cross-correlation between the mixed data and the reference can be computed. The attacker could detect the peak value to determine where the CUEs occur. This type of attack is called *cross correlation attack* (CCA) in this paper. The other attacking approach is deep learning-based. Identifying the target CUE in the mixed data could be viewed as a sound event detection (SED) task.

The reasons why we aim to study these two attacking approaches are because (i) data could be easily and exhaustively used, (ii) for CCA, the computation cost is rather low, and (iii) for SED attack, numerous models and papers are openly accessible (such as via Github and arxiv).

2.1.1. Cross correlation attack (CCA)

The cross correlation function (CCF) $c[k]$ with time lag k is defined as follows:

$$c_{xy}[k] = \sum_{n=0}^{M-1} x[n]y[n+k], 0 \leq k < N - M + 1, \quad (1)$$

where x and y denote the zero-padded signal of target CUE and the mixed signal of CAPTCHA, and M and N are the length of each sequence, respectively.

With the function $c[k]$, the attacker can set a threshold to detect where along the time axis the CUE happens by identifying the peaks. An example of $y[n]$, its spectrogram, and $c_{xy}[k]$ are shown in Fig. 1. Note that peaks can be easily identified in $c_{xy}[k]$ at the location of the CUEs.

2.1.2. SED-based Attack

Lately, many SED systems use variants of convolutional recurrent neural network (CRNN). In this research, a CRNN model proposed by Adavanne et al. [8, 9] was adopted and modified for the purpose of attacking our CAPTCHA; the architecture of the model is shown in Fig. 2. The shape of input Mel-spectrogram is 32 frames $\times$ 40 filter banks. As previously reported [8, 9], CNN layers are used to learn local shift-invariant features and gate recurrent unit (GRU) is used to learn temporal patterns. Max pooling is only performed along the frequency axis to keep the time information intact for SED. Finally, with time-distributed dense layer, the model outputs the prediction at the frame level.

[1] Only the sound clips with non-profit usage were utilized in this research.

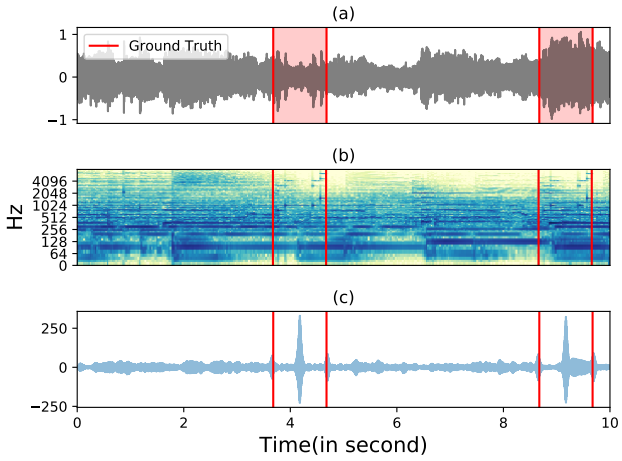

Fig. 1. Demonstration of the defending problems. **(a)** the CAPTCHA waveform with the red-shaded range marking the location of the CUEs. **(b)** the spectrogram of the CAPTCHA. **(c)** the CCF between the CAPTCHA and the CUE.

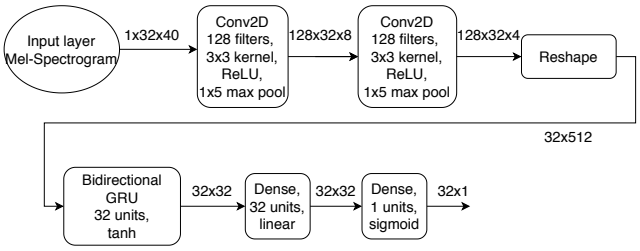

Fig. 2. Architecture of the modified CRNN model. Conv2D: two dimensional convolution; ReLU: rectified linear unit.

2.2. Defending measures

In this research, counter-measures against CCA and SED-based attacks are first considered separately.

2.2.1. Phase-modified CUE

To act against CCA attacks, we aim to modify the CUE before mixing with BGM so that the max absolute value (MAV) of the CCF between the CUE (available during the instruction

stage) and the modified CUE is decreased. In the meantime, the modified CUE remains perceptually similar to the original CUE at least when mixed with BGM.

Empirically, we found that existing all-pass filtering-based methods for signal decorrelation such as [10, 11] still resulted in spikes in the CCF. To suppress the peaks, we aggressively perform randomization in the phase spectrum. The following notations are used for denoting the MAV of the CCF,

$$J(x, y') = \max_{0 \le k < N} |c_{xy'}[k]|, \tag{2}$$

where x and y' denotes the original and the modified CUE, respectively.

The following procedure heuristically reduces J while keeping the magnitude spectrum of y' equal to that of x. It adopts the idea from Griffin-Lim algorithm [12] except for the objective function. First, the magnitude spectrograms $S_x(\omega)$ and phase spectrograms $P_x(\omega)$ of x are calculated through short-time Fourier transform (STFT). The desired signal y' is initially set equal to x. Then, each frequency bin of $P_x(\omega)$ is replaced randomly with a uniform distribution over the interval $(0, 2\pi)$ and a new y' is synthesized. If $J(x, y')$ is smaller than that of the previous round, the modified $P(\omega)$ is kept; otherwise, $P(\omega)$ is abandoned and the procedure restarts. The stopping criterion is that $J(x, y')$ does not improve for five consecutive rounds. At the end, the unchanged $S_x(\omega)$ and the modified $P_{y'}(\omega)$ are combined to synthesize y' through inverse STFT.

2.2.2. *Backdoor and watermarks*

Though not obvious immediately, a closer inspection of Fig. 1(b) may reveal repeating patterns anywhere the target CUEs are added. To defend against SED-based attacks, we resort to creating *backdoor* [13] in the released audio CAPTCHAs to lead the attacker's model into false detection.

The idea of backdoor originates from that someone wants to hack in the authentication system by leveraging the injected poisoning samples into the training set [13]. However, in this paper, the defender utilizes the backdoor in a reverse way; it is because the attackers would be the data receiver and the defenders would be the data distributor. Poisoned CAPTCHA data can be leveraged to mislead the attackers' models to make wrong predictions. In the mean time, the backdoor should not be perceptible by genuine human, neither for good person nor malicious attacker. Hence, it is suitable for an imperceptible dummy signal to do this job. We refer to this signal as watermarks.

Following the generic design in perceptual audio codecs [14], the creation of watermarks involves (i) for each frame, dividing the frequency axis into 21 Barks; (ii) in each Bark, picking up the tone with the highest sound level as the masker; (iii) For each masker, calculating spreading functions so a global masking curves could be obtained [15], and finally (iv) in each Bark, adding watermarks under the global masking curve. These steps are illustrated in Fig. 3.

Fig. 4 shows how the watermarks are deployed to be the backdoor in this research. We distribute the CAPTCHAs in the manner of Fig. 4a: the watermarks poison where the target CUEs are. We assume that the attackers would collect the poisoned CAPTCHAs for training the SED models. If the attackers finish training in a reasonable period (e.g., several days), the defender can then embed the watermarks in complimentary positions as shown in Fig. 4b as a countermeasure.

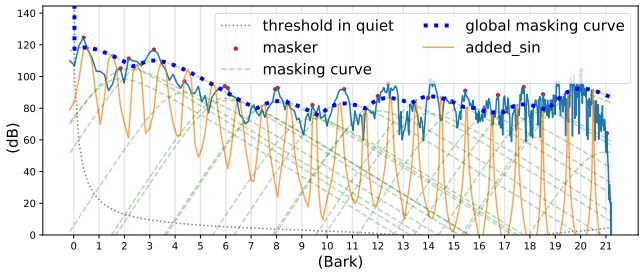

Fig. 3. Demonstration of the maskers, masking curves and the watermark. The red dots represent the maskers, and spreading functions are placed under them (dashed green lines). Psychoacoustic principles suggest that any signal beneath the global masking curve (the blue dotted line) would be inaudible to human ears. Therefore, a dummy signal can be added under the curve as watermarks (the orange line).

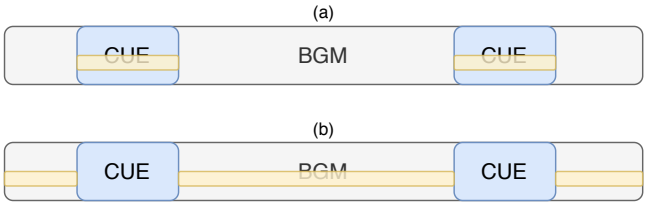

Fig. 4. Watermarks as a backdoor. Gray area means the span of BGM. Blue ones are the target CUEs mixed into the BGM. Yellow and transparent bands stand for the watermarks.

3. EXPERIMENTS AND RESULTS

To evaluate the successful attacking rate and the defending effectiveness, a metric compatible with our proposed CAPTCHA is needed. We define the region where the target CUEs are mixed as *ground truth* (GT), i.e., the region bounded by the onset and offset of the CUE. An extra tolerance region of 0.5 seconds is allowed after the offset of the CUE. Once a user presses the button outside these regions, the response is judged as a failure. If no mistake and no slow response are made, the attempt is called *completely correct*.

3.1. CCA analysis

We assume that the attackers' goal is to set a threshold to maximize its "completely correct" rate. Two factors would affect the results — the threshold set by the attacker, and the volume of the target CUE set by the defender. To analyze the attacker and the defender's performance in this game, four different situations are simulated: (1) the defender applies no counterattack, (2) the defender mixes the BGM with phase-modified CUEs, (3) the defender watermarks everywhere except at where the CUE appears [as in Fig. 4(b)], and (4) the defender combines (2) and (3).

The performance of CCA is summarized in Fig. 5. From the defender's perspectives, method (4) appears to reduce the attacker's completely-correct rate ($R_{\rm cc}$) effectively; however, when the volume of the target CUE is set equal to that of the BGM, $R_{\rm cc}$ could be as high as 29% when the attacker selects an appropriate threshold [Fig. 5(a), threshold = 1/4]. We then decrease the volume of the target CUE by 10 dB and re-analyze the performance. Now, the defender successfully suppresses the attacker's $R_{\rm cc}$ to 8.2% [Fig. 5(b), threshold = 1/8]. In Sec. 3.3, we will report the corresponding performance by human subjects.

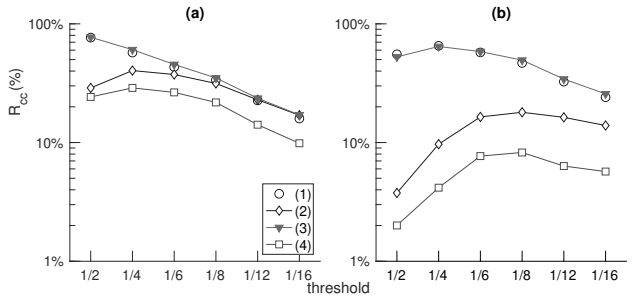

Fig. 5. The completely correct rate $R_{\rm cc}$ achieved by CCA when adjusting the detection threshold. **(a)** and **(b)** show the results of the defender when the CUE volume are 0 dB and -10 dB, respectively. The unit of the threshold is relative to the total energy of the CUE.

3.2. SED attack analysis

The number of CAPTCHAs in the training/validation/ testing set is 5684/1568/3332, respectively; all the 49 CUEs occur in training, validation, or testing, while the BGM clips in the three sets do not overlap. The number of BGM clips that are used in training, validation, and testing is 29/8/17, respectively. Training and validation are completed with the data in the form Fig. 4(a), while testing is done with the complimentary form in Fig. 4(b). Results show that the bot detects the CUEs completely correctly only in 15 CAPTCHAs (0.4%); it produces false alarms in 3315 CAPTCHAs and misses the CUEs in 194 CAPTCHAs. For comparison purposes, the SED attack achieves an $R_{\rm cc} = 59\%$ if the defender does not deploy any counter measure.

3.3. Listening test

To ensure that the defending measures mentioned above would not annoy the genuine human users, 14 participants, consisting 6 females and 8 males, were invited to evaluate our audio CAPTCHAs. All of them were between 20 and 40 of age. During the listening tests, the CAPTCHAs were created with two different intensities of phase-modified CUEs (0 dB and -10 dB, same as in Fig. 5); all of the CUEs were added with watermarks. Each kind of tests repeated 11 to 14 times for each participants. We successfully collected 177 and 180 responses, respectively, from CAPTCHAs with 0 dB and -10 dB CUEs. The users were completely correct 170 vs. 157 times, Thus the passing rates were 96.05% and 92.25%, respectively. Further analysis shows that the participants produced wrong detection 6 vs. 20 times, and reacted too slowly only in 1 vs. 3 times when the target volume was 0 dB vs. -10 dB. Though we are not certain if the participants noticed, none of them complained about the sound quality of the phase-modified CUEs being any different from the CUE they heard during the instruction stage.

4. DISCUSSION AND CONCLUSIONS

Driven by the potentially high value guarded by CAPTCHA systems, the economy of CAPTCHA solving based on *real humans* has emerged recently. In the future, we will study how the proposed system can defend human attackers. One possible direction would be to dynamically adjust the difficulty of an audio CAPTCHA so that an adversary has to take a long time to solve, which offsets the gained value.

In sum, we proposed a new kind of audio CAPTCHA composed of two stages. The first stage allows the users to check the target CUE unlimitedly, while the second stage require real-time interaction. This implementation combines the merits of the work proposed previously. For security analysis, we proposed two kinds of potential attacking paths based on the ease of implementation and then design defending measures based on psychoacoustics. For usability study, 14 participants were invited to evaluate the CAPTCHA. The final results indicates that the proposed audio CAPTCHA works in that the users can easily accomplish the task with $>$90% passing rate. Meanwhile, the passing rates from the automatic attacks (CCA and SED attack) can be kept comparatively low (8% and 0.4%, respectively). We thus conclude that the proposed system meets the design goal of CAPTCHA: hard for bots, but friendly for genuine people.

Acknowledgement: This research is supported by the Ministry of Science and Technology of Taiwan under grant No. 108-2634-F-007-003.

5. REFERENCES

[1] Sushama Kulkarni and Hanumant Fadewar, "Audio captcha techniques: A review," in *Proceedings of the Second International Conference on Computational Intelligence and Informatics*. Springer, 2018, pp. 359–368.

[2] Haichang Gao, Honggang Liu, Dan Yao, Xiyang Liu, and Uwe Aickelin, "An audio captcha to distinguish humans from computers," in *2010 Third International Symposium on Electronic Commerce and Security*. IEEE, 2010, pp. 265–269.

[3] Greg Kochanski, Daniel Lopresti, and Chilin Shih, "A reverse turing test using speech," in *Seventh International Conference on Spoken Language Processing*, Jan. 2002, pp. 1357–1360.

[4] Jusop Choi, Taekkyung Oh, William Aiken, Simon Woo, and Hyoungshick Kim, "Poster: I can't hear this because i am human: A novel design of audio captcha system," in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, 05 2018, pp. 833–835.

[5] Hendrik Meutzner, Santosh Gupta, and Dorothea Kolossa, "Constructing secure audio captchas by exploiting differences between humans and machines," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, New York, NY, USA, 2015, CHI '15, p. 2335–2338, Association for Computing Machinery.

[6] Hendrik Meutzner and Dorothea Kolossa, "A non-speech audio captcha based on acoustic event detection and classification," in *2016 24th European Signal Processing Conference (EUSIPCO)*, 2016, pp. 2250–2254.

[7] Jonathan Lazar, Jinjuan Feng, Tim Brooks, Genna Melamed, Brian Wentz, Jon Holman, Abiodun Olalere, and Nnanna Ekedebe, "The soundsright captcha: an improved approach to audio human interaction proofs for blind users," in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2012, pp. 2267–2276.

[8] Sharath Adavanne, Pasi Pertilä, and Tuomas Virtanen, "Sound event detection using spatial features and convolutional recurrent neural network," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 771–775.

[9] Sharath Adavanne, Archontis Politis, and Tuomas Virtanen, "Multichannel sound event detection using 3d convolutional neural networks for learning inter-channel features," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–7.

[10] Gary S Kendall, "The decorrelation of audio signals and its impact on spatial imagery," *Computer Music Journal*, vol. 19, no. 4, pp. 71–87, 1995.

[11] Yi-Wen Liu and Julius O Smith III, "Perceptually similar orthogonal sounds and applications to multichannel acoustic echo canceling," in *Audio Engineering Society Conference: 22nd International Conference: Virtual, Synthetic, and Entertainment Audio*. Audio Engineering Society, 2002.

[12] Daniel Griffin and Jae Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[13] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv preprint arXiv:1712.05526*, 2017.

[14] Marina Bosi and Richard E. Goldberg, *Introduction to Digital Audio Coding and Standards*, Kluwer Academic Publishers, USA, 2002.

[15] M. R. Schroeder, B. S. Atal, and J. L. Hall, "Optimizing digital speech coders by exploiting masking properties of the human ear," *The Journal of the Acoustical Society of America*, vol. 66, no. 6, pp. 1647–1652, 1979.