Bypassing Audio CAPTCHA with Automatic Speech Recognition Models

Paul Aubry, Juliette Devoivre, Damien Carron, Simon Fernandez, Andrzej Duda, and Maciej Korczyński
Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, F-38000 Grenoble, France
Email: firstname.lastname@univ-grenoble-alpes.fr

***Abstract*—CAPTCHAs are challenges designed to distinguish humans from automated bots. However, with the growing capabilities of Automatic Speech Recognition (ASR) models, these challenges are increasingly vulnerable to automated resolution. In this paper, we evaluate the feasibility of bypassing the audio versions of CAPTCHAs. We automate the collection and transcription of Google audio CAPTCHAs and compare the performance of multiple mmodels including Google Speech-to-text, DeepSpeeach, Whisper, Azure AI Speech, and Deepgram focusing on accuracy, speed, and cost. The performance results highlight how easily audio CAPTCHAs can be bypassed, in one second for the fastest methods. We also discuss possible countermeasures that could be deployed.**

***Index Terms*—audio CAPTCHA, Automatic Speech Recognition models**

1. Introduction

Online platforms increasingly face the challenge of automated abuse, ranging from spamming and scraping to fraudulent account creation. This issue arises from the widespread availability of sophisticated automated tools capable of mimicking human behavior, making it essential to deploy robust mechanisms to distinguish between human users and automated systems.

CAPTCHAs (Completely Automated Public Turing tests to tell Computers and Humans Apart) are widely used to prevent automated abuse online [1]. Visual CAPTCHAs, the most common variant, typically require users to identify objects or text in distorted images. They leverage the cognitive and perceptual abilities of humans, which surpass those of most automated systems, to block bots from gaining unauthorized access to online resources.

Audio CAPTCHAs are alternatives to visual CAPTCHAs that improve accessibility for users with visual impairments. The audio version of CAPTCHAs is still widely adopted, for instance, Google audio reCAPTCHA v2 [2] is currently deployed on more than 200,000 websites globally, which includes 21% of the top one million websites [3].

When cybercriminals manage to bypass different forms of CAPTCHAs, they can deploy a wide range of attacks such as credential stuffing, automated account creation, and web scraping with minimal cost and effort, which enables large-scale fraud, spam dissemination, and data harvesting, undermining the security of online services. As a consequence, the analysis of CAPTCHA robustness becomes crucial for cybersecurity experts.

At the same time, modern AI-powered transcription models have demonstrated remarkable capabilities in recognizing speech under challenging conditions [4], [5]. Tools like Whisper and Google Speech-to-Text API exhibit high accuracy, raising concerns about the resilience of audio CAPTCHA systems. To evaluate this risk, this paper investigates the feasibility of empirically bypassing audio reCAPTCHA v2 with six state of the art AI-powered transcription tools, using recent technologies improvement and new models like Whisper. By leveraging them, we explore the vulnerabilities in current audio CAPTCHA designs and evaluate the potential for automated systems to challenge their robustness.

In this paper, we conduct the first comparative evaluation of multiple cloud-based and local ASR models, focusing on accuracy, speed, and cost. Our findings highlight how easily security measures can be bypassed and how quickly such implementations can be deployed using various tools. We successfully solve CAPTCHAs in nearly one second, while the challenges themselves typically last around four seconds, at a cost of less than $0.01 per attempt.

These results raise serious concerns, showing that an attacker with minimal development skills and low financial investment can effectively bypass CAPTCHAs at scale. They also highlight the importance of countermeasures, like automatic bot recognition CAPTCHAs, such as Google reCAPTCHA v3, which eliminates challenges while still ensuring accessibility.

2. Background and Related Work

CAPTCHAs emerged as a response to the increasing threat of automated bots exploiting online services [6]. The need for their improvement stems from the constant advancement of machine learning and artificial intelligence techniques, which have progressively enhanced the ability of automated systems to bypass traditional CAPTCHAs [1], [7]—early CAPTCHAs relied on simple distortions of text-based images, but adversarial attacks and improvements in optical character recognition (OCR) made them less secure [8]. Several studies explored the security of CAPTCHAs and the evolving landscape of attacks leveraging machine learning and artificial intelligence [9], [10]. Prior research demonstrated the vulnerability of visual CAPTCHAs to deep learning-based OCR systems [11].

Recent progress in deep learning AI models further highlighted the potential for automating CAPTCHA-solving tasks. Plesner et al. showed that deep learning models trained on large datasets can transcribe and interpret complex image patterns, effectively solving visual reCAPTCHAv2 challenges [12].

While effective, visual CAPTCHAs raise accessibility challenges for users with visual impairments, which led to the development of audio CAPTCHAs [13]. They consist of distorted audio files containing sequences of numbers, letters, or words, often embedded in background noise or with altered pitch and tempo. Audio CAPTCHAs are designed to be challenging for automated recognition systems while remaining comprehensible to human users, thereby providing an accessible alternative for individuals who cannot solve visual CAPTCHAs.

Audio CAPTCHAs also evolved in response to advances in Automatic Speech Recognition (ASR) that improved the possibility to decipher distorted audio challenges [14], [15]. To improve the robustness of audio CAPTCHAs against automated abuses, Hossen and Hei designed an audio adversarial CAPTCHA (aae-CAPTCHA) system [16]. Abdullah et al. analyzed multiple attacks against audio CAPTCHAs to propose a new mechanism that is both intelligible and hard to automatically transcribe [17]. As a result, modern CAPTCHAs incorporate more sophisticated noise patterns, adversarial distortions, and time-based challenge-response mechanisms to maintain their efficacy against automated attacks [18].

Several research efforts specifically targeted Google reCAPTCHA, demonstrating its vulnerabilities to both traditional ASR models and modern deep learning frameworks [11], [12]. They provided critical insights into the ongoing arms race between CAPTCHA designers and attackers, emphasizing the need for more resilient verification techniques.

Solanki et al. [19] highlighted that Google reCAPTCHA v2 audio challenges evolve over time as Google adapts its challenges to counter new attacker tools. Consequently, techniques developed in earlier research may no longer achieve the same accuracy on current audio CAPTCHAs.

Our study *empirically* evaluates and compares the effectiveness of various state-of-the-art ASR models in bypassing the latest versions of Google reCAPTCHA v2 audio challenges.

3. Methodology

This section presents our methodology to bypass Google v2 audio reCAPTCHA. We selected this CAPTCHA system for testing because of its widespread adoption, which makes it a relevant target for evaluating the vulnerabilities in current audio CAPTCHA systems. We can also apply the methodology to other systems if audio files are available. The bypass pipeline consists of several steps: 1) setting up a controlled testing environment, 2) collecting the CAPTCHA audio files, 3) transcribing the audio challenge using ASR models, and 4) validating the transcriptions by submitting them to the CAPTCHA system in real time.

3.1. Local Test Environment

To systematically test the bypass approach, we developed a local web page integrating Google reCAPTCHA v2 [2]. This setup facilitates extensive testing without triggering alarms on external websites due to automated attempts, while ensuring that the CAPTCHA operates in a standard manner. The page is served via a local Python HTTP server.

Even when operating in a local environment, our server forwards all verification requests to Google, ensuring that the CAPTCHA difficulty and countermeasures (e.g., bot detection, rate limiting) are consistent with those encountered in real-world scenarios. The traffic sent to Google's servers contributes only marginally to the overall load on their infrastructure.

3.2. Retrieving CAPTCHA Audio Files

To automate CAPTCHA resolution, we extract the audio files directly from the CAPTCHA challenge. Our network traffic analysis revealed that the browser downloads an MP3 file with each call to the Google API, storing it in the cache. To streamline this process, we developed a Python script that identifies recently downloaded files by detecting the MP3 signature—since most cache files lack an extension—and extracts the MP3 file for further processing with the selected models. The collected audio files have a length between 4 and 5.5 seconds, with an average of 4.5 seconds.

3.3. Selection and Analysis of the Tested Models

After conducting extensive research, we selected six Automatic Speech Recognition (ASR) models for testing and comparisons:

- **Google Speech-to-Text** [20]: A cloud-based proprietary ASR service developed by Google.
- **Azure AI Speech** (Microsoft) [21]: a distributed computing speech-to-text model with multilingual support.
- **Speech to Text API** Nova-3 (Deepgram) [22]: A popular cloud-based API offering real-time and batch transcription.
- **Whisper** with two different model sizes: **tiny.en** and **base.en** (OpenAI) [23]: state-of-the-art, open-source ASR models capable of local execution.
- **DeepSpeech** (Mozilla) [24]: A free and open-source ASR model trained on large-scale datasets.

We initially selected the Google Speech-to-Text model to demonstrate that Google's own tools can be leveraged to bypass its security measures. Since this cloud-based model requires sending audio files to its API, latency and cost become critical factors. To broaden our comparison, we also included Azure AI Speech from Microsoft and Nova-3, a service known for its efficiency.

For locally run models, we selected Whisper—whose popularity has been boosted by ChatGPT—and evaluated two versions with different model sizes. This approach eliminates the need for external API calls and associated costs, except during initial configuration. We also included DeepSpeech to assess whether a free, open-source model developed by the community could compete with proprietary solutions from major technology companies.

3.4. Audio Format Preprocessing and Transcription

Since some ASR models require a specific format, we transform MP3 files to WAV to ensure compatibility and

then, use either cloud-based ASR processing or locally executed models.

For the Google Speech-to-Text API, Azure AI speech, and Nova-3, the bypassing process involves sending the file to the appropriate API endpoint for processing. We then retrieve the transcription results and promptly submit them to Google in real time to solve the CAPTCHA, while also saving them for further analysis.

Locally executed models such as Whisper and DeepSpeech operate entirely offline, enabling unlimited transcription of audio files without network dependency, which offers enhanced flexibility and efficiency when handling large datasets. Although these models process data locally—eliminating API rate limits and reducing the risk of being blacklisted—the final CAPTCHA verification still requires submission to Google, as is the case with cloud-based models.

After obtaining the transcriptions from all six models, we compare their outputs to assess overall accuracy. This comparative analysis identifies the most reliable models for bypassing audio CAPTCHAs, taking into account factors such as accuracy, speed, cost, and resource usage (CPU and memory).

3.5. Evaluation Setup and Comparison Criteria

We perform all tests on a laptop with 8 GB of RAM and an Intel® Core™ i5-1035G1 CPU at 1.00 GHz. This configuration serves as a baseline for assessing the computation time and model performance.

Each model had to solve one hundred CAPTCHA challenges within the environment described in Section 3.1. Overall, we retrieved six hundred reCaptcha audio files for evaluation.

Note that challenges cannot be replayed for re-evaluation once solved, and the CAPTCHA validation algorithm may not require an exact word-for-word transcription due to undisclosed provider-specific criteria. Consequently, we cannot test all models on identical audio files; instead, our evaluation relies on processing a substantial number of samples (100 per model) to ensure statistically significant and reliable results.

For each challenge, we obtain the transcription results and measure the computation time, as well as CPU and memory usage. Accuracy evaluates the capacity of a model to solve CAPTCHA successfully and is arguably the most critical factor for automation. If a model fails to generate correct transcriptions, the overall approach becomes ineffective. Efficiency assesses the processing speed of audio files. The computation time is particularly relevant for large-scale implementations and the processing speed impacts the ease of use of the model—if obtaining the results takes longer than manually listening to the audio file, the solution loses its appeal.

As noted in Section 3.4, each model has specific audio format requirements, requiring additional processing time for format conversion. To ensure fair computation time comparisons, we only record the time taken to load the model and perform transcription, except for the cloud models, for which computation occurs remotely. For them, the measured time includes the time interval between sending the request to the API and the reception of the response, which depends on network performance. Our tests ran over a high capacity university network.

TABLE 1. TRANSCRIPTION ACCURACY, COMPUTATION TIME, CPU AND MEMORY USAGE FOR LOCAL MODELS.

	Whisper		Deepspeech
	tiny.en	base.en	
Accuracy	97%	93%	59%
Time	1.16 s	2.29 s	2.26 s
CPU usage	28.41%	30.86%	14.13%
Memory usage	152.50 MB	238.47 MB	25.36 MB

TABLE 2. TRANSCRIPTION ACCURACY, COMPUTATION TIME, AND OPERATIONAL COST OF CLOUD-BASED MODELS.

	Speech-to-text	Deepgram	Azure AI Speech
Accuracy	99%	99%	99%
Time	1.85 s	2.57 s	1.29 s
Cost for 1000 attempts	$0.50	$0.18	$0.28

4. Evaluation Results

This section presents the evaluation results of the chosen models based on the proposed methodology and discusses the differences observed in the performance of the models.

We separated the models into two categories: local and cloud-based models. For local models, we evaluate their accuracy, efficiency, CPU and memory usage, whereas the cloud-based models are assessed according to accuracy, efficiency, and operational costs. Tables 1 and 2 present the overall results of the evaluation.

4.1. Local Models

Whisper models (tiny.en and base.en) obtain high accuracy (97% and 93%, respectively), but they demand considerable CPU and memory resources. Although base.en has more parameters than tiny.en—leading one to expect better performance—it does not outperform tiny.en. One possible explanation is that the short duration of the audio files (around 4 seconds) may not fully leverage the increased capacity of the base model. Instead, the simpler architecture of the tiny model could be better suited for capturing the essential features needed for accurate transcription in this specific task.

For local models, CPU and memory usage might be important considerations for large-scale deployments, particularly for attackers who need to efficiently process numerous challenges in real time.

DeepSpeech exhibits a notably lower accuracy of 59%, which is its primary limitation, although its lower resource requirements may enable more scalable deployments. This difference in accuracy can be explained by the disparities in the audio training datasets: Whisper models were trained on 680,000 hours of data,[1] while DeepSpeech on only 2,500 hours [25].

1. https://openai.com/index/whisper/

In terms of efficiency, Whisper models process audio faster than DeepSpeech, making them preferable for real-time applications. Despite its lower accuracy, the processing time of DeepSpeech is comparable to Whisper base.en but lacks the same performance benefits, limiting its practicality for CAPTCHA solving.

4.2. Cloud-Based Models

Google Speech-to-Text, Deepgram, and Azure AI Speech achieve high accuracy (99%) with Deepgram standing out as the most cost-effective for large-scale use. Azure AI Speech offers the fastest processing time, making it ideal for speed-critical tasks, while Google Speech-to-Text remains competitive with a slightly higher cost. As explained in Section 3.1, since the audio challenge lasts 4.5 seconds on average, our results show that automatic CAPTCHA solving with ASR is still faster than completing the task manually.

For cloud-based models, the operational cost might be an important consideration. Unlike local models, which primarily rely on available system resources, cloud-based processing incurs API calls, data processing time, and bandwidth usage. A model with high accuracy but an excessive cost may not be viable for large-scale deployments.

Deepgram, although slower, remains a strong contender because of its low operational cost and high accuracy ($0.18 for 1000 API calls). The choice between these models depends on the trade-off involving the processing time and the cost.

4.3. Implications

The results of this study demonstrate that current audio CAPTCHA systems, designed to differentiate between human users and automated systems, can be effectively bypassed using advanced Automatic Speech Recognition (ASR) models. There are several important implications of the results:

Security Vulnerabilities: The high accuracy rates achieved by both local and cloud-based ASR models indicate that these systems can reliably transcribe audio CAPTCHAs, undermining their intended purpose. The result exposes a significant security vulnerability, as malicious actors could exploit the models to automate CAPTCHA solving, thereby bypassing security measures designed to protect online services from automated abuse.

Potential for Abuse: The ability to bypass audio CAPTCHAs using ASR models could be misused in various ways. For instance, attackers could automate the creation of fake accounts, perform credential stuffing attacks, or scrape protected content from websites, or identify websites with specific vulnerabilities. This could lead to increased spam, fraud, exploitation of vulnerable websites for phishing or malware distribution, and other malicious activities, thereby threatening the integrity and security of online platforms.

Some solutions, like automated bot detection CAPTCHA, already exist and require only a single click to respond to challenges, improving accessibility. However, according to DataDome Blog [26], a bot protection service for websites, solutions like reCAPTCHA v3 are deployed ten times less frequently than reCAPTCHA v2. Furthermore, Hossen and Hei in [27] designed an audio adversarial CAPTCHA specifically to counter ASR systems while maintaining good usability for humans. Their research demonstrated a maximum attack success rate of 17.6% for a tested speech-to-text service, highlighting its potential resilience against automated transcription attacks.

5. Ethics and Reproducibility

This study examines the misuse of ASR and speech-to-text technologies to bypass audio CAPTCHAs. While highlighting these vulnerabilities is crucial for improving system robustness, it is equally important to mitigate the risk of enabling malicious activities.

To prevent misuse of our findings, we intentionally omitted specific implementation details—such as code and configurations for automated CAPTCHA resolution—ensuring our insights serve academic and security purposes only.

All experiments were conducted in a controlled environment (Section 3.1) to prevent any unintended effects on production systems. Although the CAPTCHA page is hosted locally, requests and validation attempts are still sent to Google servers, imposing a negligible load.

All related audio files and results will be publicly shared after publication, and the complete code will be available to vetted researchers contacting us via the provided email addresses.

6. Conclusion

In this paper, we demonstrate that audio CAPTCHAs can be bypassed using ASR transcription models, enabling automated systems to solve challenges intended to distinguish humans from bots. Our findings reveal that even individuals with limited technical expertise can leverage readily available tools to compromise audio CAPTCHA systems with a high degree of accuracy.

To evaluate the effectiveness of these bypass techniques, we conducted extensive experiments using transcription models such as Whisper and DeepSpeech, alongside cloud-based services including Google Speech-to-Text, Deepgram, and Azure AI Speech. The results demonstrate a significant capacity to accurately solve audio CAPTCHA challenges, highlighting a serious threat to the robustness of this security mechanism. Although audio CAPTCHAs currently provide valuable accessibility for visually impaired users, they are increasingly vulnerable in the era of modern AI advancements.

These results underscore the need to transition from traditional voice CAPTCHAs to more advanced systems that minimize or eliminate user interaction. Modern alternatives—such as reCAPTCHA v3, invisible CAPTCHAs, and behavioral analysis-based solutions—offer robust protection against automated attacks without compromising user experience. The main challenge now is to incentivize administrators to replace vulnerable voice CAPTCHAs with these automated, next-generation solutions.

References

[1] L. von Ahn *et al.*, "Telling Humans and Computers Apart Automatically," *Commun. ACM*, vol. 47, no. 2, Feb. 2004.

[2] Google Security, "reCaptcha v2," https://developers.google.com/recaptcha/intro.

[3] BuiltWith, "CAPTCHA Usage Distribution in the Top 1 Million Sites," https://trends.builtwith.com/widgets/captcha.

[4] W. Xu, J. Chen, Y. Li, Q. Zhang, and Z. Wang, "Benchmarking cloud-based speech-to-text services in noisy environments," *arXiv*, 2021. [Online]. Available: https://arxiv.org/abs/2105.03409

[5] J. Min and L. Wang, "Integration of large language models into speech recognition systems," in *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2307.06530

[6] L. von Ahn, M. Blum, N. J. Hopper, and J. Langford, "Captcha: Using hard ai problems for security," *Advances in Cryptology — EUROCRYPT 2003*, pp. 294–311, 2003.

[7] R. Mistry *et al.*, "DeCaptcha: Cracking CAPTCHA using Deep Learning Techniques," in *IEEE ISCON*, October 2021, pp. 1–6.

[8] Y. Gao *et al.*, "Research on the Security of Visual Reasoning CAPTCHA," in *30th USENIX Security*, Aug. 2021, pp. 3291–3308.

[9] E. Bursztein, S. Martin, and J. C. Mitchell, "Text-based captcha strengths and weaknesses," *Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS)*, pp. 125–138, 2011.

[10] H. Gao, J. Yan, F. Liu, Z. Tu, and C. Su, "A robust captcha design based on multi-label image classification," *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 240–248, 2013.

[11] A. Searles *et al.*, "An Empirical Study & Evaluation of Modern CAPTCHAs," in *32nd USENIX Security*, Aug. 2023.

[12] A. Plesner *et al.*, "Breaking reCAPTCHAv2," in *COMPSAC*. IEEE, July 2024, pp. 1047–1056.

[13] K. Bock, A. Klein, and F. Breitinger, "Breaking audio captchas: An overview and a novel attack approach," in *Proceedings of the 12th International Conference on Availability, Reliability and Security (ARES)*. ACM, 2017, pp. 1–10.

[14] A. Nguyen, V. Nguyen, S. Tran, and B. Le, "Breaking recaptcha: Attacking the google's audio captcha," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 342–349, 2014.

[15] Y. Qin, Y. Wang, Y. Li, and J. Zhang, "Adversarial examples for audio captcha security enhancement," *arXiv preprint arXiv:2203.02735*, 2022.

[16] I. Hossen and X. Hei, "aaeCAPTCHA: The Design and Implementation of Audio Adversarial CAPTCHA," in *2022 IEEE Euro S&P*, June 2022, pp. 430–447.

[17] H. Abdullah *et al.*, "Attacks as Defenses: Designing Robust Audio CAPTCHAs Using Attacks on Automatic Speech Recognition Systems," in *NDSS*, 2023.

[18] R. Jiang *et al.*, "Diff-CAPTCHA: An Image-based CAPTCHA with Security Enhanced by Denoising Diffusion Model," *CoRR*, vol. abs/2308.08367, 2023.

[19] S. Solanki, G. Krishnan, V. Sampath, and J. Polakis, "In (cyber)space bots can hear you speak: Breaking audio captchas using OTS speech recognition," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017*, B. Thuraisingham, B. Biggio, D. M. Freeman, B. Miller, and A. Sinha, Eds. ACM, 2017, pp. 69–80. [Online]. Available: https://doi.org/10.1145/3128572.3140443

[20] Google Cloud, "Speech-to-Text API," https://cloud.google.com/speech-to-text/docs.

[21] Microsoft, "Azure AI Speech," https://azure.microsoft.com/fr-fr/products/ai-services/ai-speech/.

[22] Deepgram, "Deepgram Speech to Text API," https://deepgram.com/product/speech-to-text.

[23] A. Radford *et al.*, "Robust Speech Recognition via Large-Scale Weak Supervision," in *Proc. ICML*, 2023.

[24] A. Y. Hannun *et al.*, "Deep Speech: Scaling up end-to-end speech recognition," *CoRR*, vol. abs/1412.5567, 2014.

[25] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," *arXiv preprint arXiv:1912.06670*, 2019.

[26] DataDome Blog, "ReCAPTCHA v2 vs. v3: Efficient bot protection? [2024 Update]," https://datadome.co/guides/captcha/recaptchav2-recaptchav3-efficient-bot-protection/.

[27] I. Hossen and X. Hei, "aaecaptcha: The design and implementation of audio adversarial captcha," in *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, 2022, pp. 430–447.