Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 218 (2023) 2550–2562

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering

# Recent advances of Captcha security analysis: a short literature review

Nghia Trong Dinh[a], Vinh Truong Hoang[b,*]

[a] *Department of Computer Science, Faculty of Electrical Engineering and Computer Science, VSB-Technical University of Ostrava, 17.listopadu 15/2172, 708 33 Ostrava, Czech Republic*

[b] *Faculty of Information Technology, Ho Chi Minh City Open University, 97 Vo Van Tan Street, Ho Chi Minh 722000, Vietnam*

## Abstract

CAPTCHA has long been used to keep bots from misusing web services. Various CAPTCHA schemes have been proposed over the years, principally to increase usability and security against emerging bots and hackers performing malicious operations. However, automated attacks have effectively cracked all common conventional schemes, and the majority of present CAPTCHA methods are also vulnerable to human-assisted relay attacks. Invisible reCAPTCHA and some approaches have not yet been cracked. However, with the introduction of fourth generation bots accurately mimicking human behavior, a secure CAPTCHA would be hardly designed without additional special devices. In this paper, we presented a short literature review of the current CAPTCHA schemes, as well as highlighting new trends and open issues, the challenges, and the opportunities as a solid starting point for designing the future secure and usable CAPTCHA schemes.

*Keywords:* CAPTCHA; AI hard problems; security analysis; machine learning; data engineering

## 1. Introduction

CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) or HIP (Human Interactive Proof) is an automatic security mechanism to distinguish whether the user is a human or a computer program. It creates and scores tests that can be solved by humans but are beyond the capabilities of present computer programs. It has evolved into the most generally utilized standard security measure for preventing automated computer program attacks. With the growth of Web services, denial of service (DoS) attacks by malicious automated programs have become a severe issue, and the Turing test has become a crucial approach for distinguishing people from dangerous automated programs.

* Corresponding author.
*E-mail address:* vinh.th@ou.edu.vn

10.1016/j.procs.2023.01.229

A human judge was authorized to pose a series of questions to two players, one of which was a computer and the other a human, and tell them apart in the original Turing Test. CAPTCHA, like the Turing Test, distinguishes humans from computers, but the judge is now a machine. In general, CAPTCHA is a cryptographic protocol [1] whose underlying hardness assumption is based on an AI problem. CAPTCHA implies a win-win situation: either the captcha is not broken and there is a way to differentiate humans from computers, or the captcha is broken, and a hard AI problem is solved. CAPTCHA is usually a simple visual test or puzzle that a human can complete without much difficulty, but an automated program cannot understand. The test usually consists of letters, numbers or their combination with overlapping and intersection. The CAPTCHA images may be distorted or shown against a complicated background to make them hard to be read by Optical Character Recognition (OCR) software. CAPTCHA has a wide variety of applications on Web and other applications such as: Worms and Spam, Online Polls, Free Email Services, Preventing Dictionary Attacks and also plays a significant role in limiting usage rate. In this study, we gave a brief literature overview of the current CAPTCHA schemes' open concerns, difficulties, and opportunities as a solid starting point for designing the future generation of secure and user-friendly CAPTCHA schemes. The rest of this paper is organized as follows: Section II provides the taxonomy of CAPTCHA attacks. Section III describes CAPTCHA problem analysis. As a result, suggestions and recommendations are provided to build a good CAPTCHA in Section IV. Finally, Section V concludes the paper.

### *1.1. CAPTCHA Evolution*

Text-based CAPTCHAs were the leading technique in the early 2000s. Set of attacks were developed using image processing, pattern recognition, and Machine Learning (ML) algorithms to break popular text-based schemes [3]. Furthermore, anti-recognition and anti-segmentation algorithms were employed in an attempt to improve the security of existing text-based CAPTCHAs. In 2014, Google revealed that developments in AI technology could resolve distorted text variants with 99.8 percent [4], resulting in the decline of text-based CAPTCHA schemes. Since 2004, Computer Vision (CV) problems, including image classification and recognition, were regarded as more difficult AI challenges than text recognition. Following that, many image-based CAPTCHA schemes with drag and drop, image selection, or sliding appeared in order to distinguish humans from computers. However, advanced CV and ML solutions aided in the defeat of the most important image-based CAPTCHA schemes between the years 2013 and 2018. Several image-based CAPTCHA schemes, such as reCAPTCHA V2 scheme, were attacked by ML [5]. Furthermore, approaches such as distortion, background noise mixing, and the use of adversarial instances were proposed as counter-measures against deep learning models. Adversarial examples by Szegedy et al. [6] and others have been suggested to enhance its security against ML-based attacks [7], [8], [9]. However, Na et al. [56] recently suggested a CAPTCHA solver that uses incremental learning on a limited dataset to defeat adversarial CAPTCHAs. To deal with visually impaired users, researchers proposed audio-based CAPTCHAs in addition to text-based and image-based CAPTCHAs. However, language barriers and poor usability limit the effectiveness of these schemes. Furthermore, supervised learning and automated Speech Recognition (ASR) [10] show how these schemes might be exploited. Researchers began developing behavioral-based CAPTCHA schemes in the 2010s to create difficulties based on behavioral features. The first behavioral-based CAPTCHA was launched by Geetest in 2012, while Google released No CAPTCHA reCAPTCHA in 2014 and Invisible CAPTCHA in 2015 and 2017. Bot attacks mimicking the user's behavioral pattern have been demonstrated to be vulnerable to these schemes [14]. Because of the serious privacy concerns, Cloudflare recently decided to discontinue the use of reCAPTCHA [11].

### *1.2. CAPTCHA Codes*

CAPTCHA schemes vary and are constantly improved as a result of advancements in advanced technology, AI, and hacking techniques. CAPTCHA codes are currently classified as cognitive/behavioral-based, video-based, audio-based, image-based, text-based, and others.

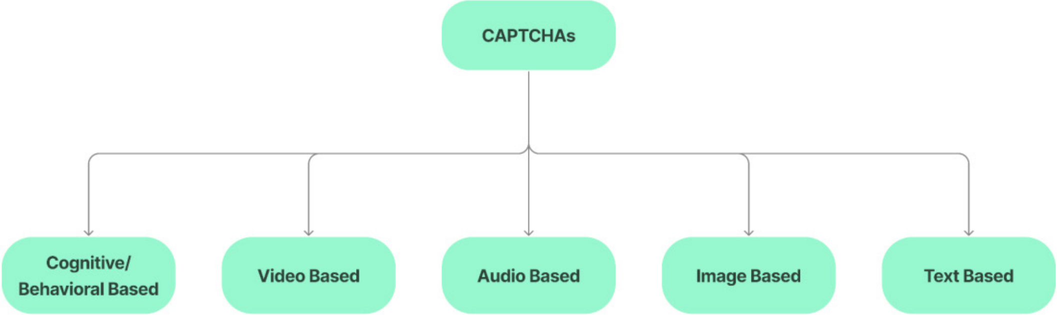

Fig. 1.Main schemes of CAPTCHA

### *1.2.1. Text-based CAPTCHA*

These CAPTCHAs became increasingly applied in the years. In these methods, text is warped and shown to a user as an image and the user must enter this text accurately before passing this test. The AI hardness assumption is that humans can easily read the warped text, but bots using Optical Character Recognition (OCR) techniques find it difficult. The different renderings of the challenge's text can be classified into three subcategories: 2D, 3D, and animation.

Recently, Super CAPTCHA [12] and 3DCAPTCHA [13] were introduced as text-based CAPTCHA schemes using the same assumptions as Teabag3D. Since 2013, Super CAPTCHA has been available as a WordPress.org plug-in. Suzi et al. [14] recently suggested DotCHA, a 3D text-based CAPTCHA. 3D letters are made of small spheres in each challenge. Each letter is readable at a different twisted rotation angle around a horizontal axis. As a result, 3D text models need to be rotated several times to identify their letters. Another animated CAPTCHA scheme is NuCaptcha [15]. The challenge in NuCaptcha begins with a video of moving white font text, followed by three red characters in a dynamic background. To pass the challenge, the red characters must be typed correctly by the user.

### *1.2.2. Image-based CAPTCHAs*

Due to the recent failure of almost text-based CAPTCHAs, there is growing worry about their protection strength and accessibility. Lately, more designs are focusing on image-based instead of character recognition with the assumption of the general vision challenges being harder than text recognition.

**Interactive-based CAPTCHAs:** These CAPTCHAs are based on the user's interaction, such as swiping gestures or mouse movement, to reveal hidden points in an image. Conti et al. [16] suggested CAPTCHaStar in which the ability of humans to recognize shapes in a cluttered environment is used. The CAPTCHaStar challenge is made up of white pixels called stars that are randomly mixed together. The position of these stars changes depending on where the cursor is. Users must drag the cursor so that the stars form an understandable shape before clicking the left mouse button to pass the CAPTCHA test. Similarly, Okada et al. [17] created Noise CAPTCHA with the same concept. This CAPTCHA is made up of two different sized and noisy images, as well as a hidden object or message in one of the images. Users must drag the small noisy image to identify the hidden object in the large image before clicking the "submit" button to pass the CAPTCHA challenge. Cursor CAPTCHA, proposed by Thomas et al. [18], displays five cursors randomly in a generated image. To pass the challenge, users must overlap the mouse pointer onto a specific cursor.

**Selection-based CAPTCHAs:** These CAPTCHAs require users to choose candidate images from a set of images. Only text or text with a sample image can be used to describe this task. Google released the "No captcha reCAPTCHA" [47] in 2014. Analyzing the browser environment (such as cookies, browser history, etc.), the system determines whether it is encountering a bot or not. The page will display only a checkbox or a selection-based CAPTCHA based on the risk level. The selection-based CAPTCHA challenge renders nine candidate images and a sample image describing the image's required content. In order to pass the challenge, the user must choose images that are similar to the sample. Facebook's image CAPTCHA is similar to reCAPTCHA in its approach. To complete the challenge, users must choose images matching the hint description from a set of twelve images with varying content. Avatar CAPTCHA [20] asks users to select avatar faces from a set of 12 grayscale images that include both human and avatar faces. FR-CAPTCHA [22] and FaceDCAPTCHA [21] are two more face image CAPTCHAs. FR-CAPTCHA requires users to pick up the same person's two face images in a complex background. On the other hand, in FaceDCAPTCHA, users are required to choose between visually warped human face images and non-human face images.

**Click-based CAPTCHAs:** These schemes display text and an image addressing where the user should click in order to pass the challenge. The main limitation of this type is that the challenge needs human intervention in order to generate a new instance. Implicit CAPTCHA [24] is a common example which requires users to click on an identical location of an image. Tang et al. [23] pioneered the use of SACaptcha in which the CAPCHA's some regions linking an explained specific shape must be clicked by users to pass the challenge.

**Draw-based CAPTCHAs:** VAPTCHA (Variation Analysis-Based Public Turing Test to Tell Computers and Humans Apart) [25] consists of an image with a randomly generated trajectory in a challenge. To complete the challenge, users must draw a matching trajectory against this trajectory. Similarly, in MotionCAPTCHA [26] users are also asked to draw a similar shape to the one rendered in the challenge box.

**Slide-based CAPTCHAs:**In these CAPTCHAs, in order to solve a challenge, users must use a slider, such as dragging an image fragment to a correct location, rotating an image orientation or selecting a correct image form. WHAT's Up CAPTCHA [27] displays three rotated images randomly and users must rotate the images to their correct position. Minteye's Slide-To-Fit CAPTCHA [28] displays a swirled image and users must move the provided slider until they see the undistorted image version.

**Drag and drop based CAPTCHAs:**In these CAPTCHAs, users are required to align image pieces to form a complete image by dragging and dropping them. Garb CAPTCHA [29] displays four randomly shuffled pieces of an image. Users are required to reorder these image pieces to get the complete image to pass the CAPTCHA test. Hamid Ali et al. [30] pioneered the use of a puzzle-based CAPTCHA. Four image pieces of an image are required to be dragged and dropped into an empty four cell grid to complete the challenge. Capy CAPTCHA [31] requires users to move a puzzle piece into a missing place in a challenge. This missing place is filled with a random image fraction.

### *1.2.3. Audio-based CAPTCHAs*

For people with visual impairments, a suggested alternative to visual CAPTCHA schemes was audio-based CAPTCHA schemes. They must type what they have heard to pass the test. At Carnegie Mellon University, the researchers introduced audio reCAPTCHA, acquired by Google later. To solve the challenge, users are required to identify eight digits spoken in human noise and only accept one incorrect digit in these digits. The eBay Audio CAPTCHA is made up of six digits in various spoken noisy voices. Microsoft CAPTCHAs are made up of ten digits in different spoken voices mixing noise of some conversations. Yahoo CAPTCHA requires users to enter seven digits after three child-spoken beeps with background noise. The 2013 version of Audio reCAPTCHA requires users to recognize all of the digits divided into three clusters in the challenge. Three or four overlapping digits are found in each cluster. The new version of reCAPTCHA in 2017 included ten spoken digits and background noise.

### *1.2.4. Video-based CAPTCHAs*

In the challenge, a short video is created, reflecting a certain content, users are required to understand and describe it by text. Kluever et al. [32] suggested a CAPTCHA in which with a short video, users are required to watch and then type three words to describe it. Shirali-Shahreza et al. proposed Motion CAPTCHA [34] which requires users to describe the motion of the person in their watching video by choosing one of the sentences.

### *1.2.5. Cognitive-based CAPTCHAs*

CAPTCHA methods based on cognitive abilities that provide increased security have largely replaced traditional Captcha methods. Cognitive abilities are brain-based skills that are the result of a distinct combination of neurobiological and psychological techniques. Knowledge, concentration, memory, judgment and assessment, reasoning and computation, problem solving, and decision making are all aspects of human cognition and behavior. To distinguish between humans and bots, these CAPTCHA methods use biometric (something you are), physical (something you have), and knowledge-based (something you know) factors with or without support of sensors like gyroscope or accelerometer.

In 2020, Acien et al. [33] suggested BeCAPTCHA-Mouse that distinguishes humans from bots by analyzing mouse trajectories during the challenge. Gametrics [35] differentiates between human and bots by collecting and analyzing the user's mouse movements during the operations of drag and drop to solve a Dynamic Cognitive Game. GEETest and Netease [36] , like Tencent CAPTCHA, require users to complete a sliding image-based CAPTCHA by moving the slider until two puzzle pieces are matched. If users complete the challenge and their sliding behavior is not suspicious, they are considered to have passed the challenge. Siripitakchai et al. [37] proposed EYE-CAPTCHA in which users are required to solve a math-based CAPTCHA by moving their eyes. To complete the challenge, the user must identify the correct answer and use his eyes to move the answer to the center of the screen. In 2014, Google launched "No CAPTCHA reCAPTCHA" (reCAPTCHA V2). All that is required is to check the "I'm not a robot" box. However, user behaviors (such as click, mouse moving and other behaviors) along with other information (browser, cookies, history etc.) are collected and analyzed in the background. If users are suspected of being bots, they need to complete a second image-based reCAPTCHA. In 2017, Invisible reCAPTCHA, an upgraded version of reCAPTCHA V2 was released. The evaluation process is initiated in the background by triggering a JavaScript API call or by users clicking on an existing button. Invisible reCAPTCHA, like the "No CAPTCHA reCAPTCHA" approach, requires a second image-based reCAPTCHA challenge if users are suspected of being bots. Guerar et al. [38], the first person, introduced the physical CAPTCHA for mobile devices, called CAPPCHA (Completely Automated Public Physical test to tell Computers and Humans Apart) in 2015. Users must tilt the device to a specific degree, which is difficult for bots to do. Hupperich et al. [39] introduced Sensor CAPTCHA in 2016, in which users are required to perform a complex gesture (such as fishing, hammering, drinking, etc.) with their mobile devices. The authors of [40] proposed Pedometric CAPTCHA, in which humans are required to walk at least five steps. When the user walks, an acceleration is generated in the mobile device, making it difficult for bots. Mantri et al. [41] suggested a CAPTCHA scheme in which users must meet the requirement of moving the device in accordance with a specific guide showing on the device. Frank et al. [42] instructed users to perform a detectable gesture, recognized by the gyroscope (such as rotating, tilting, or drawing, etc.), on moving the device. Guerar et al. [43] developed Invisible CAPPCHA, which is similar to CAPPCHA in that the challenge is invisible to users. Reading sensors detect user taps as opposed to touch screen events, which bots can easily mimic [68]. Furthermore, this CAPTCHA protects the user's privacy by not sending sensitive data to the server. AccCAPTCHA [44] requires a user to play the rolling ball game. To complete the game, the user must control the ball using the device's motion sensors. SenCAPTCHA was proposed by Feng et al. [45] for locating an animal facial key point. Users are shown a small red ball and an animal image. Then they must control the red ball into the animal's eye center by tilting their devices. GISCHA, a mobile device game-based CAPTCHA, was proposed by Yang et al [46]. To pass the challenge, a user must move the ball to the correct hole. Ababtain et al. [47] suggested the CAPTCHA which requires users to pass a simple game using sensors. They proposed five games, each with several static and one moving objects. Users must move the moving object to hit the correct target static objects in order to pass the challenge. The authors [48] proposed BrightPass, a mobile authentication CAPTCHA to protect PIN/password. Their proposed mechanism uses screen brightness, which automated bots cannot detect, to determine when users should enter a correct digit or a deceptive digit. The authors [49] proposed a PIN-based authentication CAPTCHA used for smartwatches. This mechanism is based on the same concept as CAPPCHA [50]. To enter the password, the bezel must be physically rotated to a specific degree. Similarly, the authors [51] use the digital crown rotation in smartwatches to protect the PIN code.

## 2. CAPTCHA Attack Analysis

| **CAPTCHAs** | **Attack Methods** | **Success Rates** | **Categories** |
|---|---|---|---|
| Gimpy, EZ-Gimpy | Shape context matching [52] | 33%, 92% | Text-based |
| Megaupload CAPTCHA | Segmentation [60] | 78% | Text-based |
| ReCAPTCHA | Neural networks [61] | 99.8% | Text-based |
| Teabag3D, 3DCAPTCHA, Super CAPTCHA | Pixel extraction [13] | 31%, 58%, 27% | Text-based |
| HelloCAPTCHA | PDM (Pixel Delay Map)/CL (Catching Line) [63] | 16% - 100% | Text-based |

| | | | |
|---|---|---|---|
| NuCaptcha | Box shape analysis & SIFT algorithm [64] | 90% | Text-based |
| Asirra | SVM (Support Vector Machine) [65] | 82.7% | Image-based |
| HumanAuth | Side-channel attack [66] | 92% | Image-based |
| Google image-based CAPTCHA | Deep Learning/CNN [67] | 70.78% | Image-based |
| Facebook image-based CAPTCHA | Deep Learning/CNN [67] | 83.5% | Image-based |
| reCAPTCHA V2 | Deep Learning/CNN [36] | 79% - 88% | Image-based |
| Facebook image CAPTCHA | Deep Learning/CNN [36] | 86% | Image-based |
| China Railway CAPTCHA | Deep Learning/CNN [36] | 90% | Image-based |
| Avatar CAPTCHA | CNN [13] | 99% | Image-based |
| FR-CAPTCHA | SVM [68] | 23% | Image-based |
| FaceDCAPTCHA | SVM [68] | 48% | Image-based |
| Minteye CAPTCHA | Sobel operators [69] | 100% | Image-based |
| Tencent CAPTCHA | Deep Learning/CNN [36] | 100% | Image-based |
| Capy CAPTCHA, KeyCAPTCHA, Garb CAPTCHA | JPEG image continuity measurement [70] | 65.1%, 20%, 98.1% | Image-based |
| CAPTCHaStar | Max Concentration [71] | 96% | Image-based |
| Audio reCAPTCHA | SVM [72] | 45% - 58% | Audio-based |
| eBay audio CAPTCHAs | DFT (Discrete Fourier Transform) and supervised learning algorithm [73] | 75% | Audio-based |
| Microsoft and Yahoo audio CAPTCHAs | Non-continuous speech [74] | 49%, 45% | Audio-based |
| Audio reCAPTCHA | HMMs (Hidden Markov Models) [75], Free online speech-to-text services and performing a minimal phonetic mapping [76] | 52%, 85.15% | Audio-based |
| GeeTest, Netease CAPTCHA | Sigmoid function [36] | 96%, 98% | Cognitive-based |
| No CAPTCHA reCAPTCHA | ”divide and conquer” strategy [77] | 96% - 97% | Cognitive-based |

Table 1. Comparison of some CAPTCHA attacks

### *2.1. Attack against Text-based CAPTCHA*

Text-based CAPTCHAs were the first CAPTCHA scheme and still remain the most popular. Mori and Malik [52] introduced an attack method of shape matching in 2003 to pass Gimpy and EZ-Gimpy CAPTCHAs with accuracy of 33 percent and 92 percent respectively. The proposed method [53] used a correlation algorithm and a direct distortion estimation algorithm to successfully break EZ-Gimpy with a success rate of 99 percent. Chellapilla et al. [54], [55] created a highly secure CAPTCHA of anti-segmentation in 2005 after passing various text-based CAPTCHAs with machine learning. In 2008, several anti-segmentation CAPTCHAs, used by Google, Microsoft, and Yahoo, were demonstrated to be able to be cracked by El Ahmad and Yan [56], [57]. Later, other researchers attempted to pass these CAPTCHAs with higher success rates [58], [59]. El Ahmad and Yan [60] also broke Megaupload CAPTCHA with 78 percent of success. Google researchers [61] used neural networks to break the hardest category of ReCAPTCHA in 2014, with an accuracy of 99.8 percent. The authors [13] suggested 3D CAPTCHA attack methods without OCR softwares. In several 3D-based CAPTCHAs, such as 3DCAPTCHA, Teabag 3D, and Super CAPTCHA, they extracted pixels from the characters for automated challenge recognition. Using such a technique, the authors were able to break 3DCAPTCHA, Teabag 3D, and Super CAPTCHA with success rates of 58 percent, 31 percent and 27 percent respectively. Furthermore, the same authors [62] were able to pass Teabag 3D by using the 3D textual objects' side surface information. In the animated-based CAPTCHAs, Nguyen et al. [63] demonstrated how to easily extract information across multiple animated frames by using CL (Catching Line) or PDM (Pixel Delay Map). These methods successfully defeated animated CAPTCHAs such as KillBot Professional, iCAPTCHA, Dracon CAPTCHA, and Atlantis.Due to their vulnerability to segmentation attacks, the same methods were used in [63] to defeat HelloCAPTCHA variants with a success rate ranging from 16 percent to 100 percent. NuCaptcha is a segmentation-resistant animated CAPTCHA that works by overlapping and cramming together to counter PDM or CL attack methods. Elie Bursztein [64] separated objects in each frame with a success rate of 90% using an interest points (SIFT algorithm) density evaluation and bounding box shape analysis.

### *2.2. Attack against Image-based CAPTCHA*

Golle [65] was successful in breaking the Asirra scheme. To accomplish this, SVM (Support Vector Machine) was used to classify cats and dogs with a success rate of 82.7 percent. Hernandez-Castro et al. in [66] suggested a side-channel attack breaking HumanAuth with an accuracy rate of 92 percent. Facebook image-based CAPTCHA and Google image-based CAPTCHA were bypassed by Sivakorn et al. [67] with success rates of 83.5 percent and 70.78 percent respectively. The authors [36] achieved success rates of 79 and 88 percent with the new and old variations of reCAPTCHA V2. They also defeated China Railway CAPTCHA and Facebook image CAPTCHA with success rates of 90 percent and 86 percent respectively. Besides, these authors broke different image-based CAPTCHA schemes, including the Tencent CAPTCHA with a success rate of 100 percent. Convolutional Neural Networks (CNN) [13] was applied to successfully break Avatar CAPTCHA, with a success rate of 99 percent. Both FaceDCAPTCHA and FR-CAPTCHA were defeated by Gao et al. [68] with success rates of 48 percent and 23 percent respectively. Minteye CAPTCHA was defeated in [69] by utilizing the length of the image's edges and Sobel operators. The attack method chooses the image with the smallest sum of edges based on the fact that a swirled image takes the longer edges. Hernandez-Castro et al. [70] suggested a low-cost attack using JPEG to measure image continuity. Using this side-channel attack, they successfully broke Capy CAPTCHA, Garb CAPTCHA and KeyCAPTCHA with success rates of 65.1 percent, 98.1 percent and 20 percent respectively. Gougeon and Lacharme [71] were recently able to defeat CAPTCHAaStar with a success rate of 96 percent. They also demonstrated that the parameter tuning does not prevent this CAPTCHA from their attack on pixel concentration (stars) during image formation.

### *2.3. Attack against Audio-based CAPTCHA*

Tam et al. [72] experimented with an SVM-based approach to defeat audio reCAPTCHA with a success rate of 45 percent for the exact matching solution and a success rate of 58 percent for a "one mistake" passing condition. Burzstein and Bethard [73] demonstrated a success rate of 75% in bypassing eBay's audio CAPTCHAs. Their method analyzes the wave file using a Discrete Fourier Transform (DFT) and then clusters the energy spikes. Then, to recognize speech patterns, a supervised learning algorithm is employed to train audio data. The authors [74] introduced a CAPTCHA breaker with non-continuous speech that broke Yahoo and Microsoft audio CAPTCHAs with success rates of 45 percent and 49 percent respectively. The classification stage in this solver was supervised, whereas the automated segmentation stage was unsupervised. Amazon Mechanical Turk was used to label them, and the scraped CAPTCHAs were classified using the Regularized Least-Squares Classification (RLSC) algorithm. Due to the presence of semantic vocal noise, their system could only solve reCAPTCHA with a success rate of 1.5 percent. Sano et al. [75] suggested a CAPTCHA breaker for continuous speech to defeat anti-segmentation CAPTCHAs that overlap target voices. For speech recognition, Hidden Markov Models (HMMs) were employed and tested on the 2013 version of audio reCAPTCHA with a success rate of 52 percent. Bock et al. [76] presented unCaptcha that can bypass the 2017 version of audio reCAPTCHA with a success rate of 85.15 percent by utilizing free online services of speech-to-text and performing a minimal phonetic mapping for accuracy improvement.

### *2.4. Attack against Cognitive-based CAPTCHA*

Using four simulation functions (Softmax, Sigmoid, Tanh and ReLu) to mimic human behaviors, Zhao et al. [36] successfully bypassed sliding-based CAPTCHA such as GeeTest and Netease CAPTCHA with success rates of 96 and 98 percent respectively. By creating a tracking cookie for automated bots, Sivakorn et al. [67] were able to fool Google's risk analysis system. As a result, after nine days of automated bots browsing various Google services, the solver can check the box of "I'm not a robot". Besides, the authors suggested a simple attack with a success rate of 70.78 percent for defeating the second reCAPTCHA V2 challenge. To break No CAPTCHA reCAPTCHA, the authors [77] applied the "divide and conquer" strategy. They were successful 97.4 percent of the time on a 100 x 100 grid and 96.7 percent of the time on a 1000 x 1000 screen resolution.

### *2.5. Attack against Other CAPTCHA*

Kluever et al. [32] developed a tag frequency-based approach to attack their proposed video-based CAPTCHA with a success rate of 13 percent. Hernandez-Castro et al. [78] were successful in breaking QRBGS CAPTCHA by the side-channel attack with a success rate of 44.54 percent. Mohamed et al. [79] demonstrated that dictionary-based attacks are able to defeat DCG CAPTCHAs.

### *2.6. Other Attacks*

Side-channel attacks are processes that attempt to solve an issue that is considerably easier than the original. The intended solution is built around a difficult challenge (AI-hard problem), whereas the actual solution is built around any design or implementation issues to avoid the more difficult approach. These attacks rely on randomness deviations, missing uniform randomness, to identify a link between the challenges and their responses. In this case, the challenge provides (unintentionally, "leaked" or "side-channel") knowledge on the answer. ASIRRA’s side-channel attacks are briefly described in this section [80]. ASIRRA is made up of over 25.000 photos, half of which are classed as cats or dogs. These photographs were processed by a classifier that, without utilizing any image recognition techniques, was able to discriminate between cat and dog pictures with about accuracy of 60 percent. HumanAuth's authors opted to mix a PNG image with a random JPG image picked from the library to prevent easy image library indexing. Choosing a new watermark that has a greater impact on the original image, would come at the expense of human usability.

In 2009, Philippe Golle [81] introduced the effective attacks on ASIRRA based on analyzing the CAPTCHA’s features, such as font, shape, texture and color. By employing image processing, this approach divides the photographs into a cell grid of texture and color (grayscale), which is then fed into support-vector machine (SVM) classifiers with the success classification of 83 percent.

If a CAPTCHA is based on a public knowledge database (i.e., labeled photos), there are numerous potential attacks against that database:

- Database indexing attacks: the database can be downloaded (at least partially) to obtain the information needed to solve the CAPTCHA.
- Database poisoning attacks: With an open and unprotected CAPTCHA database, our information can be uploaded to help us solve the CAPTCHA with this information.

Moreover, CAPTCHAs are intended to be completed by humans, but there exist markets for labor services solving CAPTCHAs [118] (usually in cheap labor regions) and relay attacks, which transmit CAPTCHA challenges to humans who benefit from solving them [82].

## 3. CAPTCHA Problem Analysis

### *3.1. Attack Threats*

With the evolution of automated attacks, the differences in solving CAPTCHAs between humans and automated bots may become irrelevant:

- Should a human who is browsing another website or is presented with another program's GUI be ineligible to solve our CAPTCHAs?
- Is a computer program that has been human assisted still an automatic attack?

Because it is difficult to distinguish between humans and bots, CAPTCHA schemes require additional mechanisms to improve their security:

- Measure a "human" quality, ability, or behavior to distinguish between humans and computers.
- Differentiate between humans and human-assisted algorithms to prevent magnifying or human-assisted attacks.
- Prevent relay attacks by differentiating between humans who see the CAPTCHA on the original CAPTCHA site and those who see it on another site/interface [119].
- Prevent human farm attacks by employing methods to thwart or make more difficult the use of farms of solvers in solving the CAPTCHA.

### *3.2. AI Hardness Not Transmitted*

The majority of CAPTCHAs have been vulnerable as a result of one of the following issues:

1. They are based on a much more specific and weaker underlying problem than the original one intended.
2. Flaws from design or implementation make them much easier to be bypassed by employing procedures analyzing their challenges. As a result, these procedures are known as side-channel attacks because they attempt to solve a much easier problem than the one intended by the CAPTCHA designers [78] [80].
3. The difficulty of an AI-unsolved problem is hard to convey to a CAPTCHA design. We don't know how to categorize or deeply understand an AI-hardness, so a CAPTCHA challenge of this AI-hardness may be not difficult enough for automated bots.

### *3.3. Design Flaws*

One common mistake is to select a non-uniformly distributed subset of possible answers. QRBGS (MathCAPTCHA) is one such example, with its designers employing one-digit figures in their arithmetic operations. As a result, the answers are likely to be small integers. Megaupload CAPTCHA is another example, which avoids using the values O, I, J, and 0. Worse, it always employs the three-letter-then-a-digit scheme, which makes it more user-friendly while also making it significantly less powerful. Teabag's challenges [13] use only three-character lengths and avoid characters that are hard to distinguish in 3D projections. Characters 'S', 'Z', '3', 'P', 'b', 'w', 'M', 't', and 'd' appeared more than 3 percent in a sample of 100 challenges, while a major set of other 34 characters, including '1' and '0', did not appear (possibly to avoid coincidence with 'I' and 'O').

Any relative idea in CAPTCHA design that is not based on randomness can allow challenge analysis, leading to side-channel attacks or challenge categorization analysis. Because the distribution of letter sizes in Teabag is not uniform, the frontal borders of the characters can be chosen based on their area size. There is also pixel correlation, which allows for back border detection. Simple algorithms, such as pixel continuity, can detect growing background areas. In some challenges, the non-character image portion can be removed completely or nearly completely [13]. Another example is the Megaupload CAPTCHA, which always prints the letters and digits in the same font style, Antique Olive (as identified by Identifont). Characters are rotated at specific angles, clockwise or counter-clockwise, with the first letter clockwise and the second counter-clockwise. It also prevents the overlap of more than two characters.

The challenge may provide (unintentionally, "leaked" or "side-channel") information based on the answer content. Side-channel attacks can be used to bypass the challenges by leveraging the leaked information. Besides, it is not always necessary to make it easy for a CAPTCHA to determine whether or not their answers are correct. Avoid knowing whether an answer to a challenge is correct or incorrect, or any other way of knowing if it is close to being correct, if at all possible. We can communicate this information to the user via an intermediary communication mechanism (such as email accounts, which must also be controlled to limit emailing times) or we can transfer it to the user such that it is hard to be distinguished by automated bots.

Another typical mistake, making CAPTCHA dependent on the challenger is a bad idea, and it's even worse if this dependence can be known or guessed. ASIRRA, for example, displays pets in Petfinder that are near the challenger's position in order to increase the chances of adoption for the pets displayed in the CAPTCHA (using IP geolocation). This flaw is critical because it facilitates many types of attacks, including database poisoning and database indexing.

### *3.4. Implementation Flaws*

Some CAPTCHA systems can be completely bypassed by leveraging the session ID of a previously used CAPTCHA. That is due to poor implementation, but it was not unusual a few years ago. Some developers still encode the answer to the challenge in the URL or a form field. Using this mistake, many challenges can be requested with the same answer. As a result, a mean attack [83] can be launched by calculating the median values of those challenges.

Another mistake implementation is sending the client a hash of the answer, such as MD5 hash, as a key. If the number of answers is limited or not distributed uniformly, the hashes of these answers can be easily learned enough to solve the challenges. Besides, using small fixed pools of challenges is one of common implementation flaws. HumanAuth, for example, uses fewer than a hundred images, even masking them with logos, that are easily characterized or indexed [80]. Furthermore, HumanAuth only generates challenge answers with values 0 or a small integer. This allows another type of attack: if the answer 0 fails, we will answer with a series of integers beginning with the smallest absolute values. Another common mistake is that QRBGS challenges, as an example, are not created on demand, but rather are repeated. Furthermore, some systems employ an extremely risky communication method with the CAPTCHA server, which is easily exploitable [84].

### *3.5. Preserving the* user'*s privacy*

In contrast to traditional CAPTCHA schemes, new sensor and behavioral based CAPTCHA schemes have been shown to raise privacy concerns such as user behavioral data, cookies and sensor data sent to remote servers. Some researchers proposed sending only the test results to the server, rather than the sensor data, as a solution. However, trusted hardware is required to prevent client-side hacking. As a result, the privacy of users should be strongly considered during the design phase of new CAPTCHA schemes.

### *3.6. Compatibility with all devices*

A robust and usable CAPTCHA is obviously expected to be compatible with a wide range of devices. The most promising CAPTCHA schemes, on the other hand, rely heavily on a single device. For example, CAPTCHA schemes based on touch-and-tap dynamics or mouse dynamics require device specialization. Sensor-based CAPTCHA schemes, which require sensors found only in smartwatches, tablets, or smartphones, are difficult to implement on the majority of users' devices.

## **4. How to Design a Good CAPTCHA**

### *4.1. Good Properties*

Any new CAPTCHA design should be put into production in a test site, without other protections (to focus on the CAPTCHA's hardness), for a long enough period of time to allow research. These new CAPTCHAs should include the following features to improve security against automated bots:

- In all parameters, there should be randomness and a uniform distribution. For example, for a text CAPTCHA: uniform number of areas, lines, pixels with random properties (color, group, group size, etc.), variable number of characters, various typefaces, image size, and etc.
- There should be no simpler CAPTCHA challenges: subtypes or alternatives should have the same level of difficulty (such as visual and audio CAPTCHAs).
- The challenge should be as close to the original AI problem as possible.
- The design should include features that detect automatic bypass or prevent relay attacks.
- Challenges should be distributed uniformly and independent of users and answers. Furthermore, the answers should be distributed randomly and uniformly. There should be no statistical relationship between the challenges and the answers.
- Make it difficult for automated bots to determine whether or not their answers are correct by using adversarial samples, response mechanisms, or communication methods with CAPTCHA servers.

### *4.2. Security Assurance*

- Answer repetition: if an attacker is able to collect a finite quantity of challenges with the same answers, it must be confirmed that this attacker will not be able to create a better answer than a random answer. It means that there is no better attack than trial and error.

- Challenge repetition: If our CAPTCHA has only a finite set of different challenges and we don't know how to solve them, there should be no better strategy than trial and error, with a low success rate.
- Non categorization: If our CAPTCHA is made up of different types of challenges, there should be no way to tell them apart automatically or to classify the difficulty of various challenges.

### *4.3. Security Test*

For this test, we propose to create a large enough set of elements (T = test, A = answer) of tests. We look for non-uniformities in this distribution using general randomness and statistical analysis tools [116]:

- Inconsistencies in the distribution of A (potential blind attack).
- Inconsistencies in the distribution of T (type-of-challenge categorization and challenge analysis).
- Correlations among T and A (potential side-channel attack).

These tests can be performed for some simple properties of T, such as color histograms, area sizes, histograms, distances between similar areas, maximum and minimum for a block of bytes, bit correlation with given vectors, and etc. This can be used to estimate the security parameters of any CAPTCHA proposal, avoiding pitfalls such as irrelevant parameter values that cause leakage of information [80] [78].

## Conclusion

CAPTCHA is a competition between humans and computers. Computers attempt to mimic everything humans can do. On the contrary, Humans rely on AI’s hardness and cognition capability to challenge computers. Obviously, with the rapid and continuous development of technology, computers outfitted with the most robust and cutting-edge software and hardware are capable of solving AI’s most difficult problems at any time.

In this work, we have provided a short literature review of current CAPTCHA schemes, as well as highlighted new trends and open issues, challenges, and opportunities for further research of the next generation of secure and user-friendly CAPTCHA schemes. We expect that this work will serve as a good starting point for new CAPTCHA designers in order to avoid some common design and implementation flaws, as well as for the development of new security assessment and assurance level evaluation methodologies.

## References

[1] L von Ahn, M Blum, J Langford, CAPTCHA: Using Hard AI Problems for Security (2003).

[2] M Swain, Knowledge-based System. in: W Dubitzky et al. (eds) Encyclopedia of Systems Biology (2013).

[3] E Bursztein, M Martin, J Mitchell. Text-Based CAPTCHA Strengths and Weaknesses. In Proceedings of the 18th ACM Conference on Computer and Communications Security, Chicago, Illinois, USA, (2011).

[4] Ian J. Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay D. Shet. Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks. CoRR abs/1312.6082 (2014).

[5] Binbin Zhao, Haiqin Weng, Shouling Ji, Jianhai Chen, Ting Wang, Qinming He, and Reheem Beyah. Towards Evaluating the Security of Real-World Deployed Image CAPTCHAs. In Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security (Toronto, Canada) (AISec ’18). Association for Computing Machinery, New York, NY, USA, 85–96, (2018).

[6] Ch Szegedy, W Zaremba, I Sutskever, J Bruna, D Erhan, IJ Goodfellow, R Fergus. Intriguing properties of neural networks. In the 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, (2014).

[7] D. Hitaj, B. Hitaj, S. Jajodia, and L. V. Mancini. Capture the Bot: Using Adversarial Examples to Improve CAPTCHA Robustness to Bot Attacks. IEEE Intelligent Systems, 1-1, (2020).

[8] M. Osadchy, J. Hernandez-Castro, S. Gibson, O. Dunkelman, and D. Pérez-Cabo. No Bot Expects the Deep-CAPTCHA! Introducing Immutable Adversarial Examples, With Applications to CAPTCHA Generation. IEEE Transactions on Information Forensics and Security 12, 2640–2653, (2017).

[9] Chenghui Shi, Xiaogang Xu, Shouling Ji, Kai Bu, Jianhai Chen, Raheem Beyah, and Ting Wang. AdversarialCAPTCHAs. arXiv:1901.01107 [cs.CR] (2019).

[10] Mohit Jain, Rohun Tripathi, Ishita Bhansali, and Pratyush Kumar. Automatic Generation and Evaluation of Usable and Secure Audio ReCAPTCHA. In The 21st International ACM SIGACCESS Conference on Computers and Accessibility (Pittsburgh, PA, USA) (ASSETS ’19). Association for Computing Machinery, New York, NY, USA, 355–366, (2019).

[11] Sergi Isasi Matthew Prince. Moving from reCAPTCHA to hCaptcha. https://blog.cloudflare.com/moving-from-recaptcha-to-hcaptcha (2020).

[12] Michael L Wells. Exciting Features in Super CAPTCHA, (2003).

[13] Vu Duc Nguyen, Yang-Wai Chow, and Willy Susilo. On the security of text-based 3D CAPTCHAs, (2014).

[14] Suzi Kim and Sunghee Choi. DotCHA: A 3D Text-Based Scatter-Type CAPTCHA. In Web Engineering, Maxim Bakaev, Flavius Frasincar, and In-Young Ko (Eds.). Springer International Publishing, Cham, 238–252, (2019).

[15] NuCaptcha Inc, NuCaptcha, https://www.nucaptcha.com (2018).

[16] Mauro Conti, Claudio Guarisco, and Riccardo Spolaor. CAPTCHaStar! A Novel CAPTCHA Based on Interactive Shape Discovery. In Applied Cryptography and Network Security, Mark Manulis, Ahmad-Reza Sadeghi, and Steve Schneider (Eds.). Springer International Publishing, Cham, 611–628, (2016).

[17] M. Okada and S. Matsuyama. New CAPTCHA for smartphones and tablet PCs. In 2012 IEEE Consumer Communications and Networking Conference (CCNC), 34–35, (2012).

[18] V. A. Thomas and K. Kaur. 2013. Cursor CAPTCHA — Implementing CAPTCHA using mouse cursor. In Tenth International Conference on Wireless and Optical Communications Networks (WOCN), 1–5, (2013).

[19] Vinay Shet. Are you a robot? Introducing "No CAPTCHA reCAPTCHA", (2014).

[20] D. D'Souza, P. C. Polina, and R. V. Yampolskiy. Avatar CAPTCHA: Telling computers and humans apart via face classification. In IEEE International Conference on Electro/Information Technology, 1–6, (2012).

[21] Gaurav Goswami, Brian Powell, Mayank Vatsa, Richa Singh, and Afzel Noore. FaceDCAPTCHA: Face detection-based color image CAPTCHA. Future Generation Computer Systems 31, 59–68, (2014).

[22] Gaurav Goswami, Brian M. Powell, Mayank Vatsa, Richa Singh, and Afzel Noore. FR-CAPTCHA: CAPTCHA Based on Recognizing Human Faces. PLoS ONE 9 (2014).

[23] M. Tang, H. Gao, Y. Zhang, Y. Liu, P. Zhang, and P. Wang. Research on Deep Learning Techniques in Breaking Text-Based Captchas and Designing Image-Based Captcha. IEEE Transactions on Information Forensics and Security 13, 2522–2537, (2018).

[24] Henry S. Baird and Jon L. Bentley. Implicit CAPTCHAs. In Document Recognition and Retrieval XII, Elisa H. Barney Smith and Kazem Taghva (Eds.), Vol. 5676. International Society for Optics and Photonics, SPIE, 191 – 196, (2005).

[25] Yuan, Jingxia Chongqing. Variation Analysis-Based Public Turing Test to Tell Computers and Humans Apart (2018).

[26] MotionCAPTCHA v0.2, Stop Spam, Draw Shapes, (2011).

[27] Rich Gossweiler, Maryam Kamvar, and Shumeet Baluja. What's up CAPTCHA? A CAPTCHA Based on Image Orientation. In Proceedings of the 18th International Conference on World Wide Web (Madrid, Spain) (WWW '09). Association for Computing Machinery, New York, NY, USA, 841–850, (2009).

[28] Blog post, Minteye offers no-type CAPTCHA as a security twist, (2012).

[29] Garb CAPTCHA, (2013).

[30] F. A. B. Hamid Ali and F. B. Karim. Development of the CAPTCHA system based on puzzles. In International Conference on Computer, Communications, and Control Technology (I4CT), 426–428, (2014).

[31] Capy Inc, Capy Puzzle CAPTCHA, (2018).

[32] Kurt Alfred Kluever and Richard Zanibbi. Balancing Usability and Security in a Video CAPTCHA. In Proceedings of the 5th Symposium on Usable Privacy and Security (Mountain View, California, USA) (SOUPS '09). Association for Computing Machinery, New York, NY, USA, Article 14, 11 pages, (2009).

[33] Alejandro Acien, Aythami Morales, Julian Fiérrez, and Rubén Vera-Rodriguez. BeCAPTCHA-Mouse: Synthetic Mouse Trajectories and Improved Bot Detection, (2020).

[34] M. Shirali-Shahreza and S. Shirali-Shahreza. Motion CAPTCHA. In Conference on Human System Interactions. 1042–1044, (2008).

[35] Manar Mohamed and Nitesh Saxena. Gametrics: towards attack-resilient behavioral authentication with simple cognitive games. Proceedings of the 32nd Annual Conference on Computer Security Applications (2016).

[36] Binbin Zhao, Haiqin Weng, Shouling Ji, Jianhai Chen, Ting Wang, Qinming He, and Reheem Beyah. Towards Evaluating the Security of Real-World Deployed Image CAPTCHAs. In Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security (Toronto, Canada) (AISec '18). Association for Computing Machinery, New York, NY, USA, 85–96, (2018).

[37] A. Siripitakchai, S. Phimoltares, and A. Mahaweerawat. EYE-CAPTCHA: An enhanced CAPTCHA using eye movement. In the 3rd IEEE International Conference on Computer and Communications (ICCC), 2120–2126, (2017).

[38] M. Guerar, M. Migliardi, A. Merlo, M. Benmohammed, and B. Messabih. A Completely Automatic Public Physical test to tell Computers and Humans Apart: A way to enhance authentication schemes in mobile devices. In International Conference on High Performance Computing Simulation (HPCS), 203–210, (2015).

[39] Thomas Hupperich, Katharina Krombholz, and Thorsten Holz. Sensor Captchas: On the Usability of Instrumenting Hardware Sensors to Prove Liveliness. In Trust and Trustworthy Computing, Michael Franz and Panos Papadimitratos (Eds.). Springer International Publishing, Cham, 40–59, (2016).

[40] S. Kulkarni and H. S. Fadewar. Pedometric CAPTCHA for mobile Internet users. In the 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT), 600–604, (2017).

[41] Viraj C. Mantri and Prateek Mehrotra. User authentication based on physical movement information, (2018).

[42] Brandon Z. Frank and Joseph A. Latone. Verifying a user utilizing gyroscopic movement, (2018).

[43] Meriem Guerar, Alessio Merlo, Mauro Migliardi, and Francesco Palmieri. Invisible CAPPCHA: A usable mechanism to distinguish between malware and humans on the mobile IoT. Computers & Security 78, 255–266, (2018).

[44] Ching-Jung Liao, Chang-Ju Yang, Jin-Tan Yang, Hsiang-Yang Hsu, and Jhih-Wei Liu. A Game and Accelerometer-based CAPTCHA Scheme for Mobile Learning System. In Proceedings of EdMedia & Innovate Learning, Jan Herrington, Alec Couros, and Valerie Irvine (Eds.). Association for the Advancement of Computing in Education (AACE), Victoria, Canada, 1385–1390, (2013).

[45] Yunhe Feng, Qing Cao, Hairong Qi, and Scott Ruoti. SenCAPTCHA: A Mobile-First CAPTCHA Using Orientation Sensors. In Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, Vol. 4, 1–26, (2020).

[46] Tzu-I Yang, Chorng-Shiuh Koong, and Chien-Chao Tseng. Game-based image semantic CAPTCHA on handset devices. Multimedia Tools and Applications 74, 5141–5156, (2013).

[47] E. Ababtain and D. Engels. Gestures Based CAPTCHAs the Use of Sensor Readings to Solve CAPTCHA Challenge on Smartphones. In International Conference on Computational Science and Computational Intelligence (CSCI), 113–119, (2019).

[48] M. Guerar, M. Migliardi, A. Merlo, M. Benmohammed, F. Palmieri, and A. Castiglione. Using Screen Brightness to Improve Security in Mobile Social Network Access. IEEE Transactions on Dependable and Secure Computing 15, 4, 621–632, (2018).

[49] M. Guerar, L. Verderame, M. Migliardi, and A. Merlo. 2GesturePIN: Securing PIN-Based Authentication on Smartwatches. In IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 327–333, (2019).

[50] Meriem Guerar, Alessio Merlo, and Mauro Migliardi. Completely Automated Public Physical test to tell Computers and Humans Apart: A usability study on mobile devices. Future Generation Computer Systems 82, 617 – 630, (2018).

[51] Meriem Guerar, Luca Verderame, Alessio Merlo, Francesco Palmieri, Mauro Migliardi, and Luca Vallerini. CirclePIN: A Novel Authentication Mechanism for Smartwatches to Prevent Unauthorized Access to IoT Devices. ACM Trans. Cyber-Phys. Syst. 4, 3, Article 34, 19 pages, (March 2020).

[52] G. Mori and J. Malik. Recognizing objects in adversarial clutter: breaking a visual CAPTCHA. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Proceedings., Vol. 1. I–I, (2003).

[53] G. Moy, N. Jones, C. Harkless, and R. Potter. Distortion estimation techniques in solving visual CAPTCHAs. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., Vol. 2. II–II, (2004).

[54] Kumar Chellapilla, Kevin Larson, Patrice Simard, and Mary Czerwinski. Computers beat humans at single character recognition in reading based human interaction proofs (HIPs). In the 2nd Conference on Email and Anti-Spam, (2005).

[55] Kumar Chellapilla, Kevin Larson, Patrice Y. Simard, and Mary Czerwinski. Building Segmentation Based Human-Friendly Human Interaction Proofs (HIPs). In Human Interactive Proofs, Henry S. Baird and Daniel P. Lopresti (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 1–26, (2005).

[56] Jeff Yan and Ahmad Salah, El Ahmad. A Low-Cost Attack on a Microsoft Captcha. In Proceedings of the 15th ACM Conference on Computer and Communications Security (Alexandria, Virginia, USA) (CCS '08). Association for Computing Machinery, New York, NY, USA, 543–554, (2008).

[57] Jeff Yan and Ahmad Salah, El Ahmad. Is cheap labour behind the scene? - Low-cost automated attacks on Yahoo CAPTCHAs. Technical Report. School of Computing Science, Newcastle University, England, (2008).

[58] Oleg Starostenko, Claudia Cruz-Perez, Fernando Uceda-Ponga, and Vicente Alarcon-Aquino. Breaking text-based CAPTCHAs with variable word and character orientation. Pattern Recognition 48, 1101–1112, (2015).

[59] Y. Zi, H. Gao, Z. Cheng, and Y. Liu. 2020. An End-to-End Attack on Text CAPTCHAs. IEEE Transactions on Information Forensics and Security 15, 753–766, (2020).

[60] Ahmad Salah El Ahmad, Jeff Yan, and Lindsay Marshall. The Robustness of a New CAPTCHA. In Proceedings of the Third European Workshop on System Security (Paris, France) (EUROSEC '10). Association for Computing Machinery, New York, NY, USA, 36–41, (2010).

[61] Ian J. Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay D. Shet. Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks. CoRR abs/1312.6082 (2014).

[62] Vu Duc Nguyen, Yang-Wai Chow, and Willy Susilo. Breaking a 3D-Based CAPTCHA Scheme. In Information Security and Cryptology - ICISC 2011, Howon Kim (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 391–405, (2012).

[63] Vu Duc Nguyen, Yang-Wai Chow, and Willy Susilo. Attacking Animated CAPTCHAs via Character Extraction. In Cryptology and Network Security, Josef Pieprzyk, Ahmad-Reza Sadeghi, and Mark Manulis (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 98–113, (2012).

[64] Elie Bursztein. How we broke the nucaptcha video scheme and what we propose to fix it, (2012).

[65] Philippe Golle. Machine Learning Attacks against the Asirra CAPTCHA. In Proceedings of the 15th ACM Conference on Computer and Communications Security (Alexandria, Virginia, USA) (CCS '08). Association for Computing Machinery, New York, NY, USA, 535–542, (2008).

[66] C. J. Hernandez-Castro, A. Ribagorda, and Y. Saez. Side-channel attack on the HumanAuth CAPTCHA. In 2010 International Conference on Security and Cryptography (SECRYPT), 1–7, (2010).

[67] Suphannee Sivakorn, Jason Polakis, and Angelos D. Keromytis. I'm not a human: Breaking the Google reCAPTCHA. In BlackHat, (2016).

[68] H. Gao, L. Lei, X. Zhou, J. Li, and X. Liu. 2015. The Robustness of Face-Based CAPTCHAs. In IEEE International Conference on Computer and Information Technology, Ubiquitous Computing and Communications, Dependable Autonomic and Secure Computing, Pervasive Intelligence and Computing, 2248–2255, (2015).

[69] Jack. Breaking the MintEye image CAPTCHA in 23 lines of Python, (2013).

[70] C. J. Hernández-Castro, M. D. R-Moreno, and D. F. Barrero. Using JPEG to Measure Image Continuity and Break Capy and Other Puzzle CAPTCHAs. IEEE Internet Computing 19, 46–53, (2015).

[71] Thomas Gougeon and Patrick Lacharme. How to Break CaptchaStar. In ICISSP, (2018).

[72] Jennifer Tam, Sean Hyde, Jiri Simsa, and Luis Von Ahn. Breaking Audio CAPTCHAs. In Proceedings of the 21st International Conference on Neural Information Processing Systems (Vancouver, British Columbia, Canada) (NIPS'08). Curran Associates Inc., Red Hook, NY, USA, 1625–1632, (2008).

[73] Elie Bursztein and Steven Bethard. Decaptcha: breaking 75% of eBay audio CAPTCHAs. In Proceedings of the 3rd USENIX conference on Offensive technologies, Vol. 1. USENIX Association, 8, (2009).

[74] E. Bursztein, R. Beauxis, H. Paskov, D. Perito, C. Fabry, and J. Mitchell. The Failure of Noise-Based Non-continuous Audio Captchas. In 2011 IEEE Symposium on Security and Privacy, 19–31, (2011).

[75] Shotaro Sano, Takuma Otsuka, and Hiroshi G. Okuno. Solving Google's Continuous Audio CAPTCHA with HMM-Based Automatic Speech Recognition. In Advances in Information and Computer Security, Kazuo Sakiyama and Masayuki Terada (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 36–52, (2013).

[76] Kevin Bock, Daven Patel, George Hughey, and Dave Levin. UnCaptcha: A Low-Resource Defeat of Recaptcha's Audio Challenge. In Proceedings of the 11th USENIX Conference on Offensive Technologies (Vancouver, BC, Canada) (WOOT'17). USENIX Association, USA, 7, (2017).

[77] Ismail Akrout, Amal Feriani, and Mohamed Akrout. Hacking Google reCAPTCHA v3 using Reinforcement Learning. ArXiv abs/1903.01003 (2019).

[78] Carlos Javier Hernandez-Castro and Arturo Ribagorda. Pitfalls in CAPTCHA design and implementation: The Math CAPTCHA, a case study. Computers & Security 29, 141 – 157, (2010).

[79] Manar Mohamed, Niharika Sachdeva, Michael Georgescu, Song Gao, Nitesh Saxena, Chengcui Zhang, Ponnurangam Kumaraguru, Paul C. van Oorschot, and Wei-Bang Chen. A Three-Way Investigation of a Game-CAPTCHA: Automated Attacks, Relay Attacks and Usability. In Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security (Kyoto, Japan) (ASIA CCS '14). Association for Computing Machinery, New York, NY, USA, 195–206, (2014).

[80] Hernandez-Castro, C.J., Ribagorda, A., Saez, Y.: Side-channel attack on labeling CAPTCHAs, (2009).

[81] Golle, P.: Machine Learning Attacks Against the Asirra CAPTCHA. In: ACM CCS, (2008).

[82] Ran Halprin. Dependent captchas: Preventing the relay attack. Weizmann Institute of Science, (2009).

[83] Wieser W. Captcha recognition via averaging. DOI= http://www. triplespark.net/misc/captcha, (2007).

[84] Caine, A., Hengartner, U.: The AI Hardness of CAPTCHAs does not imply Robust Network Security. In: IFIP, Trust Management, vol. 238, pp. 367–382, (2007).