Gotta CAPTCHA 'Em All: A Survey of 20 Years of the Human-or-computer Dilemma

MERIEM GUERAR and LUCA VERDERAME, DIBRIS, University of Genoa, Italy
MAURO MIGLIARDI, Department of Electronic Engineering, University of Padua, Italy
FRANCESCO PALMIERI, Department of Computer Science, University of Salerno, Italy
ALESSIO MERLO, DIBRIS, University of Genoa, Italy

A recent study has found that malicious bots generated nearly a quarter of overall website traffic in 2019 [102]. These malicious bots perform activities such as price and content scraping, account creation and takeover, credit card fraud, denial of service, and so on. Thus, they represent a serious threat to all businesses in general, but are especially troublesome for e-commerce, travel, and financial services. One of the most common defense mechanisms against bots abusing online services is the introduction of Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA), so it is extremely important to understand which CAPTCHA schemes have been designed and their actual effectiveness against the ever-evolving bots. To this end, this work provides an overview of the current state-of-the-art in the field of CAPTCHA schemes and defines a new classification that includes all the emerging schemes. In addition, for each identified CAPTCHA category, the most successful attack methods are summarized by also describing how CAPTCHA schemes evolved to resist bot attacks, and discussing the limitations of different CAPTCHA schemes from the security, usability, and compatibility point of view. Finally, an assessment of the open issues, challenges, and opportunities for further study is provided, paving the road toward the design of the next-generation secure and user-friendly CAPTCHA schemes.

CCS Concepts: • **Security and privacy → Authentication**; **Graphical/visual passwords;**

Additional Key Words and Phrases: CAPTCHA, bot, CAPTCHA type, security, text CAPTCHA, image CAPTCHA, behavior CAPTCHA, sensor CAPTCHA

ACM Reference format:
Meriem Guerar, Luca Verderame, Mauro Migliardi, Francesco Palmieri, and Alessio Merlo. 2021. Gotta CAPTCHA 'Em All: A Survey of 20 Years of the Human-or-computer Dilemma. *ACM Comput. Surv.* 54, 9, Article 192 (August 2021), 33 pages.
https://doi.org/10.1145/3477142

1 INTRODUCTION

A ***Completely Automated Public Turing tests to tell Computers and Humans Apart* (CAPTCHA)** is, as the name suggests, a challenge-response test used to distinguish between

Authors' addresses: M. Guerar, L. Verderame, and A. Merlo, DIBRIS, University of Genoa, Genoa, Italy; email: {meriem.guerar, luca.verderame, alessio.merlo}@dibris.unige.it, M. Migliardi, Department of Electronic Engineering, University of Padua, Padua, Italy; email: mauro.migliardi@unipd.it; F. Palmieri, Department of Computer Science, University of Salerno, Salerno, Italy; email: fpalmieri@unisa.it.

0360-0300/2021/08-ART192 $15.00
https://doi.org/10.1145/3477142

genuine human users and automated computer programs. CAPTCHAs are commonly used to prevent abuses of online services such as registering thousands of free accounts, obtaining tickets for resale, spreading spam emails, taking over accounts by using brute force [57], or perform credential stuffing attacks [103].

The idea of using a CAPTCHA to check whether the users who are making requests to a web service are humans goes back to 1996 [87]. A year later, AltaVista developed the first practical example of a CAPTCHA scheme, which was based on the inability of **Optical Character Recognition (OCR)** software to recognize a distorted text [76].

In 2000, Von Ahn et al. [126, 127] introduced several practical proposals for designing CAPTCHA schemes based on *hard **Artificial Intelligence (AI)** problems*, i.e., challenges that most humans can solve easily, but computer programs cannot pass.

Most CAPTCHA schemes proposed in the literature follow such an approach and exploit different elements such as character recognition, image understanding, and speech recognition to create challenges that successfully block automated bots. However, the recent advancement of AI in general and **Computer Vision (CV)** in particular has made automated programs significantly better at solving such tests. As a result, almost all of the traditional CAPTCHA schemes have been broken as demonstrated in References [20, 43, 46, 117].

Furthermore, in contrast to Von Ahn et al. expectations, not all the attacks proposed in the literature attempt to solve the underlying AI problem on which these CAPTCHAs are based to break them. Some of them, instead, try to circumvent the AI problem by leveraging the weaknesses in the design of a particular CAPTCHA scheme [40, 62, 64]. These kinds of attacks are known as side-channel attacks.

Over time, designing effective and user-friendly CAPTCHA schemes based on hard AI problems has become very challenging. This has led to the emergence of a new generation of schemes based on behavioral analysis and sensor readings.

In 2014 Google announced that today's Artificial Intelligence technology can solve even the most difficult variant of distorted text at 99.8% accuracy [111] and moved to a CAPTCHA scheme based on behavioral analysis that is considered the dominant CAPTCHA scheme in the market today. In the academic world, many works have shown the vulnerability of the traditional CAPTCHA schemes, nevertheless, many researchers still aim at breaking traditional CAPTCHA schemes and evaluating their security and usability [12, 37, 130, 142], ignoring the emerging CAPTCHA schemes that have not been broken yet. Still, recent works in the literature do not consider these new CAPTCHA schemes neither in their review nor in their security evaluation [26, 134, 140].

Contribution. Different from the existing CAPTCHA surveys (e.g., References [21, 115, 134, 140]), in this work, we present an up-to-date comprehensive CAPTCHA survey that includes both the traditional CAPTCHA schemes and the new generation ones, such as those based on behavior and sensor readings. Then, we propose a novel classification of the existing CAPTCHA literature from 2000 to 2020 based on *10 different groups* (i.e., Text-based, Image-based, Audio-based, Video-based, Game-based, Slider-based, Math-based, Behavior-based, Sensor-based, and CAPTCHA for liveliness detection). To the best of our knowledge, this is the first survey that reviews behavioral-based, sensor-based CAPTCHAs, and CAPTCHA designed for liveliness detection in authentication methods. Furthermore, we survey and analyze all the literature regarding the security evaluation of the existing CAPTCHA schemes and the proposed techniques to break them, showing the weaknesses of the different categories of CAPTCHA schemes. This work also allows us to build a timeline for the security of 77 CAPTCHA schemes illustrating the creation and breaking year along with the breaking percentage. Besides showing the evolution of CAPTCHA over two decades, this timeline provides a clear view of the broken CAPTCHA mechanisms and the ones

that are worth further investigation. In addition, it elucidates the new design trends in CAPTCHA schemes.

Finally, we discuss the evolution of CAPTCHA schemes in terms of new design trends, their security, and their user-friendliness; moreover, we illustrate the open issues, the challenges, and the opportunities for further study, drawing a roadmap for the design of the next generation of secure and user-friendly CAPTCHA schemes.

Structure. The rest of this article is organized as follows: In Section 2, we introduce a comprehensive classification of conventional and recent emerging CAPTCHA schemes. In Section 3, we revise the main attacks against the CAPTCHA schemes described in Section 2. In Section 4, we provide a discussion on the current state-of-the-art of CAPTCHA, highlighting the CAPTCHA evolution and the limitations of each CAPTCHA design from different standpoints. Section 5 discusses open issues, challenges, and opportunities for future work. In Section 6, we draw some conclusions from all the analyses and comparisons performed.

2 CAPTCHA CLASSIFICATION

The traditional classification of CAPTCHA in the literature defines six categories, namely, text-based, image-based, audio-based, video-based, math-based, and game-based CAPTCHA [11, 115]. However, we consider this classification incomplete, because it does not cover the new emerging CAPTCHA schemes. As an example, the most widely adopted CAPTCHA schemes today do not fall into this classification (e.g., reCAPTCHA V2 and Geetest). Nevertheless, even the most recent surveys in the literature adopt this incomplete classification to review and evaluate the security of the existing CAPTCHA schemes [26, 134, 140]. This discrepancy between the relevant literature and the actual state-of-the-art motivated us to propose a more comprehensive classification capable of capturing the new emerging CAPTCHA schemes. We argue that current CAPTCHA schemes can be divided into 10 categories, i.e., *Text-based, Image-based, Audio-based, Video-based, Game-based, Slider-based, Behavior-based, Sensor-based, and CAPTCHAs for liveliness detection in authentication methods.*

It is important to mention that the new CAPTCHA schemes that involve a traditional challenge/response test belong to the old category as well; yet, to highlight the development and the new directions in CAPTCHA design, we will focus on the new added mechanisms.

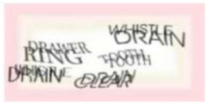
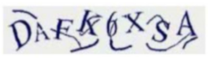
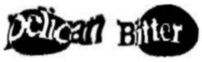
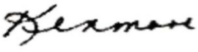
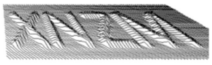
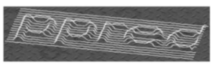
2.1 Text-based CAPTCHAs

Text-based CAPTCHAs are the most popular form of CAPTCHA; in these schemes a text (e.g., a sequence of random characters or words) is distorted and displayed to the user as an image. When words are used, language dependency represents a major limitation of this kind of CAPTCHA scheme. Then, the user is asked to input the text appearing in the image to pass the test. The underlying assumption is that humans can read the distorted text easily, but this is hard for bots using OCR techniques.

Since the interaction required to solve the CAPTCHA (i.e., the input of a text) is the same in almost all text-based CAPTCHAs, we classified the variation of text-based CAPTCHAs according to the different representation of the text of the challenge. Hence, we identified three sub-categories: (1) 2D text-based, (2) 3D text-based, and (3) Animated text-based. Table 1 gathers all the considered text-based CAPTCHA schemes, a relevant graphical sample, and a detailed description of the challenge.

2.1.1 2D Text-based CAPTCHA. The 2D text-based CAPTCHA scheme was initially developed by Andrei Broder and his colleagues at the DEC Systems Research Center in 1997. In the same

Table 1. A Taxonomy of Text-based CAPTCHAs

Type	Scheme	Sample	Year	Challenge Description
2D	GIMPY [129]		2000	Recognize three words out of of seven selected randomly from a dictionary
	EZ-GIMPY [129]		2000	Recognize one English word in a distorted image
	BaffelText [23]		2003	Recognize a pronounceable string of characters with difference masking applied
	Microsoft (MSN) [137]		2002	Recognize eight distorted characters presented with random arcs as clutters
	Google (Gmail) [137]		2006	Recognize characters which are crowded together
	Yahoo [137]		2008	Recognize a string of characters connected by intersecting random lines
	Megaupload		2010	Recognize four overlapped characters with negative intersection areas
	ReCAPTCHA V1 [128]		2008	Recognize distorted text scanned from old books
	Clickable CAPTCHA [25]		2008	Identify English words among non-English words
	Handwritten [106]		2004	Recognize a distorted handwritten text (e.g., city name)
3D	Teabag 3D [95]		2006	Recognize a sequence of characters that appears on a grid in 3D space
	3DCAPTCHA [93]		2006	Recognize a sequence of 3D characters
	Super CAPTCHA [131]		2013	Recognize a sequence of 3D characters
	DotCHA [72]		2019	Drag and rotate the model to identify each letter, then type the answer
Animated	Dracon CAPTCHA [31]		2006	Recognize five characters which fade and blur at various times over the animation frames
	KillBot Professional [90]			Recognize five moving characters among a noisy foreground and/or background
	Atlantis CAPTCHA [90]			Recognize six moving characters among other continuously changing their color
	HelloCAPTCHA [101]		2010	Recognize a sequence of six characters displayed in an animated GIF image
	NuCaptcha [94]		2008	Type the last three red moving characters

year, the AltaVista website used such a method to block bots trying to influence the rank of a set of sites on the AltaVista search engine [7].

In 2000, Von Ahn and Blum, in collaboration with Yahoo, developed **Gimpy CAPTCHA** and **EZ-Gimpy** [129] to prevent spammers from posting malicious advertisements in the chat rooms and to ensure that free accounts were granted only to real individuals. The challenge of the Gimpy CAPTCHA scheme consists of typing correctly at least three out of seven words randomly selected from a dictionary. EZ-Gimpy is a simplified version of Gimpy, showing only a single random word selected from the dictionary. However, the word is rendered to an image using different fonts, background grids, and gradients. Furthermore, the image is altered by using blurring, noise, and distortion effects on letters.

In 2003, Monica Chew and Henry Baird proposed **BaffleText** [23], a text-based CAPTCHA scheme that adopts pseudo-random but pronounceable words along with some masking techniques aiming at preventing the use of OCR software.

In 2010, the popular website for sharing and uploading files **(Megaupload.com)** designed a CAPTCHA scheme based on a new segmentation-resistant mechanism different to that used by **Microsoft**, **Google**, and **Yahoo**. This new mechanism relies on the combination of overlapping characters and the "Gestalt Perception" principle, which is used to hide some contents of the characters where they connect to each other. The Gestalt Perception principle suggests that humans can reconstruct individual characters mentally, while this task is still difficult for computer programs.

The most widely deployed form of text-based CAPTCHA is the first version of **ReCAPTCHA** [128], which had the two-fold aim of protecting websites from bot attacks and digitizing old books. The challenge consists of recognizing two distorted words scanned from old books, one known by the algorithm and one that OCR programs have failed to identify. The challenge is successfully passed if the user correctly recognizes and types the known word. Besides, if the challenge is passed, then the algorithm assumes that the user recognized also the second unknown word.

To improve the usability of text-based CAPTCHAs, Chow et al. [25] introduced the idea of **clickable CAPTCHA**. Their approach consists of combining multiple textual CAPTCHA challenges into a grid of clickable CAPTCHAs (e.g., a 3-by-4 grid). The user has to click on the grid elements that match the challenge requirement. For instance, the challenge can be the identification of English words among non-English words in the grid. Obviously, such a challenge has language dependencies.

In contrast to traditional CAPTCHA schemes that use machine-printed text, authors in References [106, 107] proposed **Handwritten CAPTCHAs** that use as challenges synthetic handwritten text images, already known to fool OCR software.

2.1.2 3D Text-based CAPTCHA. 3D text-based CAPTCHA schemes exploit the fact that human beings can easily recognize sequences of 3D characters while bot programs cannot; thus, they represent an advancement in comparison to the 2D text-based CAPTCHA schemes.

One of the first proposals is the **Teabag 3D** designed by the OCR Research Team [95] to identify the weaknesses of 2D text-based CAPTCHA schemes and propose a novel—and more secure—CAPTCHA scheme. Teabag 3D consists of an image with a 3D pattern that contains textual characters (as shown in Table 1). Thanks to the new CAPTCHA scheme, the authors demonstrated that humans could easily recognize the 3D text and, at the same time, automated systems failed in the recognition task.

Similarly, **Super CAPTCHA** [131] and **3DCAPTCHA** [93] are 3D text-based CAPTCHA schemes that were based on those same assumptions and used on several websites. For instance, Super CAPTCHA is also available as a plug-in for WordPress.org since 2013.[1]

[1] https://wordpress.org/plugins/super-capcha/#description.

Imsamai and Phimoltares [68] introduced the **3D CAPTCHA** scheme by rendering a sequence of 3D alphanumeric characters and applying a set of different effects to trick automated recognition systems. Those effects include text rotation, text overlapping, noise addition, scaling, font variation, special characters, and different background textures.

Recently, Suzi et al. [72] introduced a new type of 3D text-based CAPTCHA, called **DotCHA**. The challenge consists of 3D letters composed of several small spheres. Each character is twisted around a horizontal axis so each letter is readable at a different rotation angle. Thus, the user needs to rotate the 3D text model multiple times to identify all the letters. From the usability point of view, DotCHA adds an additional task (i.e., the rotation of the model multiple times) in comparison to the traditional text-based CAPTCHAs that require only the input of the text to solve the challenge.

2.1.3 Animated Text-based CAPTCHA. Animated CAPTCHAs extend text-based schemes by introducing the time dimension. In detail, these CAPTCHA schemes animate the textual content in the challenge in a short clip, thus complicating the extraction task for automated systems.

One of the first proposals of animated CAPTCHA has been introduced by Fischer and Herfet [38] in 2006. Their CAPTCHA scheme is based on the idea of projecting the text onto a deforming animated surface. In 2009, Naumann et al. [88] introduced an animated CAPTCHA based on the perception that the human ocular system tends to group different entities that move together. Hence, the authors developed a new CAPTCHA scheme that shows letters superimposed over a noisy background. The users are able to distinguish the text from the background when the letters are moving.

Similarly, Cui et al. [28] proposed an animated CAPTCHA where the user can get the right characters shown in the animation only when they are moving. They also introduced the "zero-knowledge per frame" principle, which ensures that each frame of the animation does not leak enough information to solve the CAPTCHA challenge.

Besides the CAPTCHA schemes proposed by the scientific community, there are a set of solutions offered either by specific websites or by CAPTCHA service providers.

For instance, the Creo Group [101] introduced in 2010 an animated CAPTCHA, called **HelloCAPTCHA**, freely available through the developers' website. In general, the HelloCAPTCHA challenge consists of a sequence of six characters presented in an animated GIF image. In some challenges, the characters change position and orientation, and in others, they are not all visible at the same time. The idea behind such a scheme is to spread the information over multiple animation frames to prevent a typical OCR attack over a single frame. **NuCaptcha** is another animated CAPTCHA scheme [94]. The challenge consists of a video with scrolling text in white font, followed by three random red characters moving across a dynamic background. The user is required to type the moving red characters to solve the CAPTCHA. **Dracon CAPTCHAs** [31] are animated visual Flash CAPTCHAs. The challenge consists of recognizing five characters displayed at fixed locations and randomly altered by using fade and blur effects. The animation is enriched with noise, e.g., random falling bars in the foreground or small text characters in the background. **KillBot Professional** version [90] is a commercial animated CAPTCHA that claimed among its clients the United States federal government. In detail, the users have to recognize five moving characters displayed in a noisy foreground and background that are composed of lighter colors than the main text characters. **Atlantis CAPTCHA** [90] is an animated CAPTCHA used on the Atlantis website.[2] In such a CAPTCHA, users need to recognize six moving characters among others that are continuously changing their color.

[2] Atlantis-caps.com.

2.2 Image-based CAPTCHAs

An alternative to text-based CAPTCHA schemes are image-based ones. In these schemes, the challenge presented to the user is generally based on understanding a written text describing a task that needs an additional image classification or recognition task to be completed. The textual part has language dependencies. The user interaction or the gesture required to solve the challenge may differ from a scheme to another, therefore, we suggested a classification based on those differences, identifying six different types, as shown in Table 2 and described in the following.

2.2.1 Click-based CAPTCHAs. This type of scheme shows an image and a text that explains where the user needs to click to complete the challenge. A typical example is **Implicit CAPTCHA** [6], where the users are required to click on a specific static place on an image according to the given instruction, e.g., "Click on the climber's glasses" or "Click on the logo on the climber's arm."

The major limitation of such a CAPTCHA scheme is that the challenge cannot be generated automatically, and thus it requires the human intervention to generate a new instance. Recently, a new image-based CAPTCHA, called **SACaptcha**, has been introduced by Tang et al. [121]. Users are required to click on some regions in the image that have a specific shape mentioned in the challenge description to pass the CAPTCHA test.

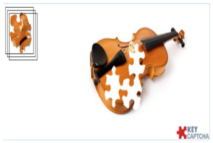
2.2.2 Sliding Image-based CAPTCHAs. In sliding image-based CAPTCHAs, users are required to use the slider to solve an image-based challenge such as adjusting the orientation of an image, selecting the correct form of an image, or moving a fragment of an image to the correct location.

For instance, **WHAT's Up CAPTCHA** [48] presents three randomly rotated images to the users and asks them to use the slider to rotate the images to their upright position. The success rate of a random guess depends on the tolerance of accepted answers. According to the data reported in Reference [48], the success rate of a random guess on one image is 4.48%, but it decreases to 0.009% for three images. Slide-to-fit CAPTCHA [99] by **Minteye** presents a distorted image through a swirl filter with a small slider below the image. Users have to move the slider until the user sees the undistorted version of the image. **Tencent CAPTCHA** asks the users to drag the slider until two puzzle pieces match. One of these puzzle pieces represents the target region in the image, where the users have to place the other piece of the puzzle to have a complete image.

2.2.3 Drag and Drop–based CAPTCHAs. The Drag and drop CAPTCHA scheme requires the users to combine or reorder image pieces by dragging and dropping them to form a complete picture.

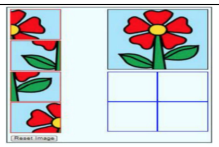
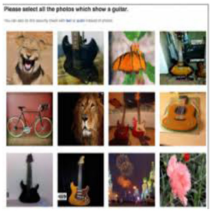
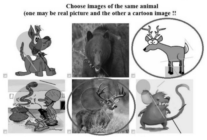
For instance, **Garb CAPTCHA** [132] presents an image divided into four pieces randomly shuffled. To pass the CAPTCHA test, users have to reorder them to reconstruct the original image. Similarly, **Hamid Ali et al.** [60] introduced a puzzle-based CAPTCHA. The challenge consists of dragging and dropping four images or pieces of the same image into an empty grid of four cells. To pass the CAPTCHA test, the position of each image in the grid should be the same as in the reference image. **Gao et al.** [44] proposed an image-based CAPTCHA that uses the jigsaw puzzle. Their CAPTCHA displays an image divided into pieces (i.e., 9, 16, or 25, depending on security level), but only two are not in the original positions. Users have to identify the two pieces and drag one over the other to swap them to solve the puzzle. **Capy CAPTCHA** [17] asks the users to drag one puzzle piece into the correct location within the challenge image. The puzzle void is filled with a fraction from the same or another image rather than a random color. **KeyCAPTCHA** [71] shows an incomplete image along with three puzzle pieces and asks the users to assemble the image as they see it in the reference image displayed in the upper right corner of the frame. The reference image is shown with a small resolution, and it disappears once the cursor is inside the

Table 2. A Taxonomy of Image-based CAPTCHAs

Type	Scheme	Sample	Year	Challenge Description
Click	Implicit CAPTCHA [6]		2005	Click on a specific area of an image (e.g., mountain top)
	SACaptcha [121]		2018	Click on some regions in the image that have a specific shape mentioned in the challenge description
Sliding	WHAT'S UP CAPTCHA [48]		2009	Move the slider to adjust at least three randomly rotated images to their upright orientation
	MintEye CAPTCHA [99]		2012	Move the slider until undistorted version of the image appears
	Tencent (Tencent.com)			Drag the slider until two puzzle pieces match
Drag and Drop	Garb CAPTCHA [132]		2013	Drag and drop the puzzle pieces to their correct position to reconstruct the original image
	Capy CAPTCHA [17]		2012	Drag a puzzle piece to complete a jigsaw
	KeyCAPTCHA [71]		2010	Drag three puzzle pieces to assemble the image
	Gao et al [44]		2010	Identify the two misplaced pieces and swap them

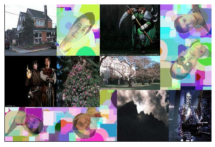
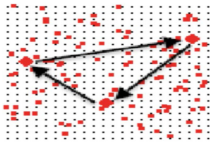
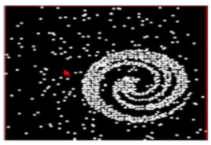
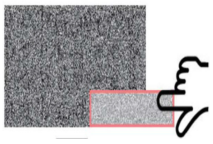
(Continued)

Table 2. Continued

	Hamid Ali et al [60]		2014	Drag and drop four images to an empty grid following the same order in the reference image
	Asirra [35]		2007	Select cats from a set of 12 images of cats and dogs
	No CAPTCHA reCAPTCHA [111]		2014	Select images that have the same content described in the challenge with a sample image
			2014	Select all images with street signs, cars, bridges or some specific object
Selection	Facebook image CAPTCHA			Select the images that correspond to a hint from twelve images with different content
	HumanAuth [89]		2006	Select images with natural contents
	SEMAGE [124]		2011	Select semantically related images from a set of images
	AVATAR [33]		2012	Select avatar faces from a set of 12 images composed of a mix of human and avatar faces

(Continued)

Table 2. Continued

	FR-CAPTCHA [50]		2014	Select real human faces distorted among nonhuman face images
	FaceDCAPTCHA [49]		2014	Select two face images of the same person
	VAPTCHA [139]		2018	Draw an resemblant trajectory to match the reference trajectory
Drawing	Drawing CAPTCHA [113]		2006	Connect a specific dots to each other
	MotionCAPTCHA [1]		2011	Draw a shape displayed in a box
Interactive	CAPTCHAStar		2015	Move the cursor until forming a recognizable shape
	Cursor CAPTCHA [122]		2013	Overlap the cursor on the identical object placed in a random generated image
	Noise CAPTCHA [97]		2012	Move a small noisy image over a large noisy image until a hidden message or object appears

256 frame. To pass the CAPTCHA test, the users have to drag and drop the three puzzle pieces in their
257 correct position.

258 *2.2.4 Selection-based CAPTCHAs.* Selection-based CAPTCHA schemes ask users to select can-
259 didate images from sets of images. The task can be described with text only or with text and a
260 sample image.

A typical CAPTCHA of this kind is **Asirra** [35], which displays 12 images of cats and dogs and asks users to select all cat images among them. Similarly, **HumanAuth CAPTCHA** [89] asks the users to select all images with natural content. It is based on humans' ability to distinguish between images with natural content (e.g., tree, river) and artificial one (e.g., car, watch). In contrast to Asirra and HumanAuth CAPTCHA, **SEMAGE** (SEmantically MAtching imaGEs) CAPTCHA [124] asks users to select semantically related images from a given image set. Thus, the user is required to recognize the content of each image and then understand and identify the semantic relationship between a subset of them.

In 2014, Google introduced the second version of reCAPTCHA based on behavior analysis, called **"No captcha reCAPTCHA"** [47, 111]. In this version the system analyzes the browser environment (e.g., browser history, cookies) and evaluates the risk of being confronted with a bot; if the risk is considered high, then the page displays a selection-based CAPTCHA, otherwise checking a checkbox is enough. The selection-based CAPTCHA challenge consists of a sample image with a keyword describing the content of the image and nine candidate images. The user is required to select images that are similar to the sample to pass the challenge.

Facebook's image CAPTCHA follows the same approach of reCAPTCHA except for the sample image. To pass the challenge, users have to select the images that correspond to the description (i.e., hint) from 12 images with different content. Afterward, Google introduced other variations of image-based reCAPTCHA that ask the user to select images with vehicles, houses, street signs, or other specific objects.

Among others, several selection-based CAPTCHAs rely on face images for their challenges. For instance, **Avatar CAPTCHA** [33] requires users to choose avatar faces from a set of 12 grayscale images composed of a mix of human and avatar faces. Other face-based image CAPTCHAs are **FR-CAPTCHA** [50] and **FaceDCAPTCHA** [49]. FR-CAPTCHA asks users to select two face images of the same person displayed in a complex background. Differently, FaceDCAPTCHA requires users to identify the visually distorted real human faces among nonhuman face images. Unlike Avatar, the human face images used in FR-CAPTCHA and FaceDCAPTCHA are rotated, distorted, or embedded in a complex background.

2.2.5 Drawing-based CAPTCHAs. The CAPTCHA schemes belonging to this category distinguish computers and human beings, thanks to a drawing challenge.

Shirali-Shahreza introduced the first drawing-based CAPTCHA, named **Drawing CAPTCHA** in 2006 [113]. Users are required to draw lines to connect diamond-shaped dots. These dots are displayed on a screen with noisy background, so users have to identify them first. Another CAPTCHA that falls into this category is **VAPTCHA** (Variation Analysis-based Public Turing Test to Tell Computers and Humans Apart)[139]. The VAPTCHA challenge consists of an image containing a randomly generated reference trajectory. Users are required to draw a resemblant trajectory to match the reference trajectory to complete the verification. If the matching degree is equal to or higher than the minimal match degree defined by the system, then users are classified as humans, otherwise they are assumed to be bots. Similarly, **MotionCAPTCHA** [1] asks users to draw a shape similar to the one displayed in the challenge box.

2.2.6 Interactive-based CAPTCHA. CAPTCHA schemes in this category rely on the user's interaction through mouse movement or swiping gesture to discover a secret position in an image. This position represents the answer to the challenge and it is revealed only after the user's interaction.

For instance, Conti et al. [27] proposed a new CAPTCHA scheme called **CAPTCHaStar**. The proposed CAPTCHA leverages the human ability to recognize shapes in a confusing environment. The underlying assumption is that a machine cannot easily emulate this ability. The CAPTCHaStar challenge consists of white pixels, called stars, randomly mixed during the generation of the

challenge. The position of these stars changes according to the position of the cursor. To pass the CAPTCHA test, users have to move the cursor until the stars aggregate in a recognizable shape, then, click on the left mouse button to send the cursor coordinates to the server. If the cursor is close to the secret position, then users are considered as humans. On mobile devices, CAPTCHaStar requires swiping fingers to move the cursor and tapping the "check" link to submit the final answer.

Similarly, Okada et al. [97] introduced **Noise CAPTCHA**, which is composed of two noisy images with different sizes and a hidden object or message in a specific position in the image. To pass the CAPTCHA test, users have to move the small noisy image over the large image until the hidden object appears, then click on the "submit" button. Similar to CAPTCHaStar, users are considered as humans when they identify the correct (secret) position at which the object or the image becomes visible.

Thomas et al. [122] propose **Cursor CAPTCHA**, which displays five cursor images in a randomly generated image and customizes the cursor image of the mouse pointer. Then, the CAPTCHA asks users to overlap the mouse pointer on an identical cursor image to pass the challenge. At the beginning of the test, users see six cursor images in which two of them are identical, but they are unable to identify the target position until they move the mouse.

2.3 Audio-based CAPTCHAs

Audio-based CAPTCHA schemes were initially proposed as an alternative to visual CAPTCHAs for people who have a visual impairment. To pass the test, they are required to type what they have heard. One of the most popular audio-based CAPTCHA was the **audio reCAPTCHA** proposed by researchers at Carnegie Mellon University and later acquired by Google. To pass the CAPTCHA challenge, users have to recognize eight spoken digits with a background noise composed of human voices speaking backward at varying volumes. Audio reCAPTCHA accepts only one mistake in one of the digits to solve the challenge.

Nevertheless, Sauer et al. [109] showed that this CAPTCHA scheme represents a hard task for blind users. Indeed, their usability study involving six blind participants shows that the participants were able to complete only 46% of the tasks correctly.

Similarly, many popular websites implement audio CAPTCHAs that rely on listening to a random sequence of digits. For instance, **e-Bay Audio CAPTCHA** consists of 6 digits spoken in different voices with regular background noise. **Microsoft CAPTCHAs** are composed of 10 digits spoken in different voices with regular background noise consisting of several simultaneous conversations. **Yahoo CAPTCHA** asks the users to type 7 digits that follow 3 beeps spoken by a child with background noise consisting of other children's voices. The **Audio reCAPTCHA** version, used in 2013, asks the users to identify all digits presented in the challenge composed of three clusters. Each cluster contains 3 or 4 overlapping digits. In 2017, Google released a new version of **reCAPTCHA** with 10 spoken digits and background noise. The available experiences of Audio-based CAPTCHAs are summarized in Table 3.

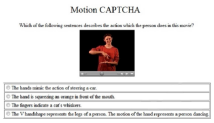
2.4 Video-based CAPTCHAs

CAPTCHA schemes in this category reproduce a short video and then propose a textually described challenge that requires some level of comprehension of the video content.

For instance, Kluever et al. [73] proposed a CAPTCHA that asks the user to watch a video and provide three words that best describe the video. Similarly, Shirali-Shahreza et al. proposed **Motion captcha** [114], which asks the users to watch a video, then they have to select the sentence that describes the motion of the person in the video.

The most common implementations of Video-based CAPTCHAs are reported in Table 3.

Table 3. A Taxonomy of Video- and Audio-based CAPTCHAs

Captcha Type	Captcha Scheme	Sample	Year	Challenge Description
Video-based	Kluever el al [73]		2009	Watch a video and provide three words that best describe the video
	Motion captcha [114]		2008	Select the sentence that describes the motion of the person in the video
Audio-based	Audio ReCAPTCHA (non-continous)		2008	Recognize eight spoken digits with background noise consisting of human voices speaking backwards at varying volumes
	e-Bay audio CAPTCHA			Recognize six digits spoken in different voices with regular background noise
	Microsoft CAPTCHA			Recognize ten digits spoken in different voices with regular background noise consisting of several simultaneous conversations
	Yahoo CAPTCHA			Recognize seven digits that follow three beeps spoken by a child with background noise consisting of other children's voices
	Audio reCAPTCHA (Continuous)		2013	Identify all digits presented in the challenge that consist of three clusters and each cluster contains three or four overlapping digits
	Audio ReCAPTCHA (version 2017)		2017	Recognize ten spoken digits with background noise

2.5 Math-based CAPTCHAs

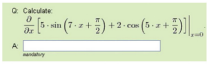
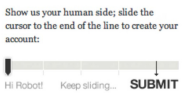
CAPTCHA schemes in this category ask the users to solve a challenge based on a mathematical problem. A typical example of Math-based CAPTCHA is **Arithmetic CAPTCHA** that relies on basic arithmetic operations such as (+,*,-). To solve the challenge, users have to enter the results of a simple math operation such as "2+1= " to prove that they are human. Unlike Arithmetic CAPTCHA, **QRBGS CAPTCHA** [61] usually asks the users to solve a complex equation that involves trigonometric and differential functions. The main problem with such kind of CAPTCHAs is that it assumes that all users have advanced knowledge in mathematics, and it requires a long time to solve the challenge.

2.6 Slider CAPTCHAs

Slider CAPTCHA is another type of CAPTCHA scheme that relies only on the sliding gesture. Unlike sliding image-based CAPTCHAs previously described, image recognition is not part of the challenge. Users have only to move the slider across the screen to prove they are human.

For instance, the CAPTCHA used by **Taobao.com**, which is a Chinese online shopping website owned by Alibaba, asks the users to drag the slider from the start to the end of the sliding bar to

Table 4. A Taxonomy of Math and Slider-based CAPTCHAs

Type	Scheme	Sample	Year	Challenge Description
Math-based	Arithmetic CAPTCHA			Enter the result of the math operation
	QRBGS CAPTCHA [61]		2008	Enter the result of a complex mathematical equation
Slider-based	Taobao			Drag a slider from the start to the end of the sliding bar
	TheyMakeApps CAPTCHA [133]		2010	Move the slider to the end of the line

verify whether they are human or not. Similarly, CAPTCHA used by **TheyMakeApps.com** asks the users to move the slider to the end of the line to submit a form [133]. This type of CAPTCHA has been widely adopted due to its ease of use.

Some well-known examples of Math and Slider-based CAPTCHAs are reported in Table 4.

2.7 Game-based CAPTCHAs

Game-based CAPTCHA schemes have emerged as an alternative that tries to make the task of solving CAPTCHAs a fun activity for the users. These CAPTCHAs are based on the assumption that humans—unlike automated systems—can understand the rules of a game and solve the challenge. Users are required to solve a straightforward game that is often based on image semantics. There are also attempts to make the users enjoy solving math-based CAPTCHAs by offering games such as tic-tac-toe and a dynamic roll-dice game.

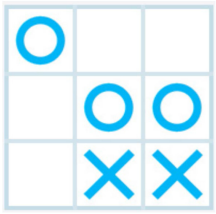
A well-known game-based CAPTCHA is **PlayThru CAPTCHA** [29] designed by a startup called "Are you a human." The challenge requires moving some dynamic objects that have a semantic connection with the static target image. For instance, users might be asked to place food in the refrigerator or feed a baby. Mohamed et al. [80] developed four **Dynamic Cognitive Games (DCG)** similar to PlayThru to investigate both its security and usability. Depending on the game, users are required to drag and drop dynamic objects to match them with others (e.g., match objects with similar shapes) or place them in specific regions (e.g., place the ships on the sea). Their usability study shows that all the four games last less than 10 seconds, and all the participants successfully completed the games within the time out. Regarding the error rate per drag and drop, the authors noticed that the visual matching tasks are less error-prone than the semantic matching tasks.

Another example of game-based CAPTCHA is **SweetCAPTCHA**. Also in this case, the users are required to drag and drop an image with a semantic connection with the target image. For example, users need to drag milk to a cup of coffee, drag the player to the guitar, or drag chopsticks to sushi. Another example is **Tic Tac Toe CAPTCHA**, which proposes to the user an almost complete tic-tac-toe game, where users need a single move to win the game and get three Xs in a row.

Some CAPTCHA designers have tried to have users having fun when they solve CAPTCHAs based on a mathematical problem. A typical example is **Dice CAPTCHA** (i.e., Homo-sapiens Dice version) [30], where users are required to roll some dice and then compute the sum of the digits appearing on them. If the entered sum is correct, then the users are considered humans.

A detailed taxonomy of the most common game-based CAPTCHAs is reported in Table 5.

Table 5. A Taxonomy of Game-based CAPTCHAs

Scheme	Sample	Year	Challenge Description
PlayThru [29]		2012	Play a simple game that consist of moving specific dynamic objects to a specific place according to the image semantics
DCG CAPTCHAs [80]		2014	Depending on the challenge description, drag and drop objects to match them with others or place them in specific regions
SweetCAPTCHA		2011	Drag specific static images to match them with the target image
Dice CAPTCHA [30]		2010	Click on "Roll" to roll the dices, then enter the sum of the numbers appearing on the dices
Tic tac toe CAPTCHA		2011	Complete the game by tapping into the correct position to get a line of 3 Xs (or Os)

2.8 Behavior-based CAPTCHAs

CAPTCHA schemes in this category employ behavioral biometrics such as keystroke dynamics, mouse dynamics, swipe dynamics, and eye movement to distinguish between humans and bots. Most of the proposed schemes involve mouse/swipe dynamics with conventional CAPTCHA schemes (e.g., image-based or game-based).

As an example, Acien et al. [3] proposed in 2020 **BeCAPTCHA-Mouse**, which asks the user to solve an image-based CAPTCHA similar to reCAPTCHA V2. However, such a scheme analyzes the mouse trajectories performed during the task to distinguish between humans and bots. Similarly, **Gametrics** [81] asks the users to solve a Dynamic Cognitive Game CAPTCHA. During the drag and drop operations requested to solve the challenge, the CAPTCHA collects the mouse movement features to distinguish between human and automated systems.

In addition, **GEETest** and **Netease** [141] ask the users to solve a sliding image-based CAPTCHA similar to Tencent CAPTCHA. In detail, the users need to complete an image by dragging the slider to match two puzzle pieces (one reflecting the missing part of the image, the other the correct position in the image). Unlike Tencent CAPTCHA, users are considered humans only when both the puzzle pieces match and the sliding behavior is not considered suspicious.

Furthermore, the same authors of BeCAPTCHA-Mouse proposed a variation for smartphones called **Be-CAPTCHA** [4] that is based on a slider challenge. However, unlike traditional sliding tasks, the algorithm leverages swiping gestures and sensor data to detect human behavior.

Siripitakchai et al. [116] proposed **EYE-CAPTCHA**, which asks the users to solve a math-based CAPTCHA relying on the eye movement. In detail, the challenge prompts a simple math operation in the center on the screen, along with four potential answers at the corners. To solve the challenge, the user has to locate the right answer and move it through his eyes to the center.

Unlike the above-mentioned behavioral CAPTCHAs, the **"No CAPTCHA reCAPTCHA"** (a.k.a., reCAPTCHA V2) deployed by Google in 2014 does not use a traditional CAPTCHA scheme to gather information on the user behavior. On the contrary, it only requires to click on the "I'm not a robot" checkbox. However, in the background, information related to user's behavior (e.g., the mouse movement, where the users click, how long they linger over a checkbox) along with other information such as the installed plugins, the language of the browser, and cookies are collected and analyzed by an engine that evaluates the risk of being confronted with a bot. If the user is classified as human, then no additional tasks are required. Otherwise, the system prompts a traditional image-based reCAPTCHA as a second security layer.

In 2017, Google released another variation of reCAPTCHA V2, called **Invisible reCAPTCHA**. As its name suggests, the challenge is invisible to the user. The verification process is performed in the background, and it is invoked when the user clicks on an existing button on the web page or by a JavaScript API call. Similarly to the "No CAPTCHA reCAPTCHA" approach, Invisible reCAPTCHA requires to solve the traditional image-based reCAPTCHA if and only if the risk analysis engine cannot recognize a human behavior with a given level of confidence.

A detailed taxonomy of the most common behavior-based CAPTCHAs is reported in Table 6.

2.9 Sensor-based CAPTCHAs

The CAPTCHA schemes belonging to this category rely on the data gathered by one or more hardware sensors. These CAPTCHA schemes are typically designed for mobile devices that natively host sensors such as gyroscope or accelerometer. Sensors-based CAPTCHA schemes can be further divided into *physical* and *cognitive*. In the first case, the sensors' data are used to discriminate between a human and a bot. In the latter, the sensors only provide an input channel for the actions of the user.

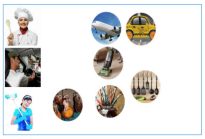
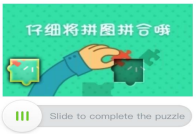
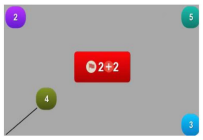
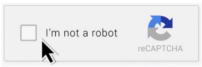
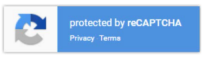
A detailed taxonomy of the available sensor-based CAPTCHA experiences is reported in Table 7.

2.9.1 Physical CAPTCHAs. The first physical CAPTCHA for mobile devices has been introduced by Guerar et al. [54] in 2015. The proposed CAPTCHA scheme, called **CAPPCHA** (Completely Automated Public Physical test to tell Computers and Humans Apart), requires the users to tilt the device to a specific degree to prove they are humans. The challenge exploits the impossibility for a software bot to perform a physical task such as moving the device. Furthermore, thanks to the use of dedicated hardware sensors, the CAPTCHA scheme does not require randomizing the challenge or executing sophisticated gestures. Therefore, the authors suggested a simple gesture such as tilting the device to a specific degree, which can be detected easily through motion sensors such as the accelerometer and gyroscope.

Similarly, in 2016, Hupperich et al. [66] proposed **Sensor CAPTCHA**, which asks the users to move the device to prove they are humans. Unlike CAPPCHA, Sensor CAPTCHA asks the users to perform a complex gesture such as hammering, fishing, drinking, or turning the body while holding the mobile device.

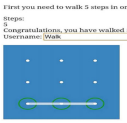
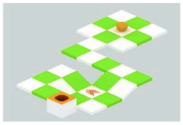
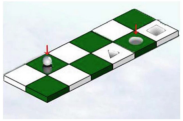
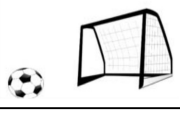
In Reference [74], the authors suggested **Pedometric CAPTCHA**, which requires walking at least five steps to be considered human. The idea behind this is to create an acceleration in the mobile device while the user is walking that cannot be generated by a bot. **Mantri et al.**[78] proposed a CAPTCHA scheme that asks the users to move the device according to a specific pattern displayed on the screen. For instance, the user is required to write an "S" letter while holding the device and then press the "submit" button. Similarly, **Frank et al.** [39] asks the users to move the device to perform a gesture that can be detected by the gyroscope, such as tilting the device, rotating the device or drawing a three-dimensional shape or letter while holding the device.

Table 6. A Taxonomy of Behavior-based CAPTCHAs

Scheme	Sample	Year	Challenge Description
BeCAPTCHA-Mouse [81]		2020	Solve a selection image-based CAPTCHA
Gametrics [81]		2016	Drag-drop a subset of the moving objects to their orresponding targets which are static
GEETest (geetest.com)		2012	Drag the slider until two puzzle pieces match
Netease [141]			Drag the slider until two puzzle pieces match
Be-CAPTCHA [4]		2020	Drag a slider from the start to the end of the sliding bar
Eye-CAPTCHA [116]		2017	User locates the answer of a simple math operation displayed in the screen and move it using his eyes to the center
No CAPTCHA reCAPTCHA [47]		2014	Click on I'm not a robot Checkbox
Invisible reCAPTCHA [47]		2017	No visible challenge, it is invoked via a Javascript API or when the user clicks on an existing button on the website

In Reference [53], Guerar et al. proposed **Invisible CAPPCHA** based on the same idea of CAPPCHA, although—as the name suggests—the challenge is invisible to the users. The authors noticed that most of the online services that require protection against automation abuses in mobile devices require the interaction with the touchscreen (e.g., fill a form, write a comment, tap on a button, perform the login). Such physical interactions cause micro-movements of the device that can be tracked by motion sensors such as the accelerometer. Based on their observation, they leveraged the implicit user's taps to make the challenge transparent to the users and thus more user-friendly. Unlike the Invisible reCAPTCHA designed by Google, Invisible CAPPCHA is based on humans' ability to perform a physical task and not on the way they perform the task. Also, the tap gesture is detected through sensor readings rather than touchscreen events that can be easily simulated by the bots [102]. Furthermore, no sensitive data are provided to the server side as the

Table 7. A Taxonomy of Sensor-based CAPTCHAs

Type	Scheme	Sample	Year	Challenge Description
Physical	CAPPCHA [54]		2015	Tilt the device to a specific degree
	Pedometric CAPTCHA [74]		2017	Walk at least 5 steps
	Mantri et al. [78]		2017	Move the device according to a specified pattern displayed on the screen
	Sensor CAPTCHA [66]		2016	Perform gestures such as hammering, fishing, turning the body while holding the mobile device
	Invisible CAPPCHA [53]		2018	No task is required
	Frank et al. [39]		2018	Move the device to perform an action (e.g., tilt the device, draw a shapes, letters or patterns)
Cognitive	AccCAPTCHA [75]		2013	Play a simple rolling ball game or other well-known games (e.g., enigma, racing game)
	GISCHA [138]		2013	Play a simple game that consist of moving a ball to the destination hole with a specific shape
	SenCAPTCHA [36]		2020	Identify the animal eye position, then tilt the device to move the ball to this position
	Ababtain et al. [2]		2019	Play a simple game that consist of one moving object (i.e., ball) and one or multiple target objects (e.g., Goal). Users move the device to match the ball with the target object

interpretation of the sensor data is completely performed inside trusted hardware in the client side and thus it preserves the user's privacy.

2.9.2 Cognitive Sensor-based CAPTCHAs. Similar to the traditional CAPTCHAs, Cognitive sensor-based CAPTCHAs ask the users to solve a cognitive challenge (e.g., recognizing an image, or solving a game, selecting images based on expert medical knowledge [96]), yet they use sensors as their input to solve the challenge rather than the conventional taping or swiping gestures. To this aim, we classified these CAPTCHAs as sensor-based CAPTCHA rather than including them in one of the categories mentioned above to highlight the current research trends.

A typical example of this category is **AccCAPTCHA** [75], where the challenge requires to play a simple game such as the rolling ball game. Thanks to the device's motion sensors, the user can move the ball to complete the game.

Yang et al. [138] proposed **GISCHA**, a game-based image semantic CAPTCHA for mobile devices. The challenge consists of a rolling ball and destination holes with different shapes. The direction of the rolling ball can be controlled by turning the mobile device to different angles. The users are considered as human if they successfully move the ball to the destination hole shaped like a circle. Similarly, the CAPTCHA designed by Ababtain et al. [2] asks the users to solve a simple game to prove that they are humans, also in this case, using the sensors as their input. They suggested five games where all of them use one dynamic object and one or multiple static objects. To pass the test, the users have to move the dynamic object so it touches specific static objects that are considered as targets.

Recently, Feng et al. [75] proposed **SenCAPTCHA**, which is based on the difficulty of finding an animal facial key point. Such a CAPTCHA scheme proposes an image of an animal along with a small red ball. The users are required to tilt their devices to move the red ball into the center of that animal's eye. The idea behind using the sensor readings is to avoid the traditional input modalities (i.e., typing, selecting images) that can be inconvenient on devices with small screen sizes.

2.10 CAPTCHAs for Liveliness Detection in Authentication Methods

Today, one of the biggest problems that threatens every website with a login is the use of malicious bots for credential stuffing and credential cracking. This is due to the availability of billions of breached credentials. Imperva [67] reported that a recent credential stuffing attack lasted 60 hours and included 44 million login attempts. In the literature, there are many proposals that attempt to embed a form of CAPTCHA in the authentication methods to stop these attacks.

In 2010, Stefan Popoveniuc [100] proposed an authentication method called SpeakUP for remote unsupervised voting. They added text-based CAPTCHA to voice biometrics. To log in, the voters are required to read out loud a 2D text CAPTCHA displayed on the screen that is associated with the candidate for whom they wish to vote. The voters are identified by the biometric characteristics of their voices. For further security, the author suggested to capture a video of the voter while solving the CAPTCHA.

Recently, Uzun et al. [123] proposed a real-time CAPTCHA system called rtCaptcha for defending against automated attacks on facial authentication systems. Similar to SpeakUp CAPTCHA, once the authentication session starts, users are required to take a video while pronouncing out loud the 2D text CAPTCHA presented as a challenge to prove they are humans. The session will time out if no response is received after a predefined period.

In Reference [55], BrightPass, an authentication method for mobile social media networks has been proposed. It adds liveliness detection mechanism to PIN/password to prevent the automated process of iterating through the entire password space and from testing all the stolen passwords.

The underlying mechanism leverages screen brightness, which cannot be captured by malicious programs, to tell users when to input a correct PIN digit and when to input a misleading lie digit.

In References [56, 59], a novel PIN-based authentication method for smartwatches that embeds a form of physical CAPTCHA has been presented. It uses the same principle behind CAPPCHA [52]. Users have to physically rotate the bezel to a specific degree to input the PIN digits. Using a trusted hardware (i.e., the bezel) this mechanism prevents any automated program from performing a brute force or credential stuffing attacks. This mechanism can be also used separately from PIN-based authentication. Similarly, authors in Reference [58] leverage the rotation of the smartwatch digital crown to prevent automated attacks against the PIN code.

3 SECURITY OF CAPTCHA SCHEMES

The different proposals of CAPTCHA schemes aim to discern between human and computing systems, thanks to a challenge. Instead, from an attacker perspective, the goal is to break the CAPTCHA scheme, i.e., to solve the proposed challenge with an automated system and still be recognized as a human. The general process of breaking traditional CAPTCHAs can be divided into the following phases/stages: pre-processing, segmentation, and recognition. Pre-processing techniques (e.g., image binarization, image thinning, and noise removal) are usually used to remove background patterns, separate the foreground from the background, and eliminate noise before the segmentation and recognition phases [104]. In some cases, extraction techniques are used before pre-processing [92], such as **Pixel Delay Map (PDM), Catching Line (CL)**, and **Frame Selection (FS)**. Segmentation techniques are used to split the CAPTCHA image into segments that contain individual objects to facilitate recognition. Well-known techniques that have been used in breaking CAPTCHAs are vertical histogram, color-filling, snake segmentation [104], and JSEG. Many efforts have been put into breaking the different CAPTCHAs by the scientific community in past years. To do so, attackers can rely on a set of attacking methodologies that can be grouped in:

- **Object recognition attacks.** This type of attack includes object recognition attacks, pixel-count, dictionary, and database attacks [104]. The common techniques used for object recognition are pattern matching (e.g., shape context matching [83], correlation algorithm [84]), OCR recognition, **Scale-Invariant Feature Transform (SIFT)** and, recently, deep learning. In particular, the most used deep learning models for CAPTCHA recognition are CNN, RNN, and LSTM-RNN [42, 46, 105].
- **Random Guess Attacks.** In this type of attack, attackers try to break the CAPTCHA scheme by guessing the correct answer. Therefore, CAPTCHAs with a small number of different challenges are vulnerable to this attack.
- **Human Solver Relay Attacks.** The bot forwards the CAPTCHA challenges to remote human workers to solve the CAPTCHAs in exchange for a small income. The human workers solve the challenges and send the correct responses to the bot that can solve the CAPTCHA accordingly.

In the following, we outline the existing techniques for attacking the different CAPTCHA schemes presented in Section 2. Furthermore, we plot them in a timeline graph (Figure 1) to report if the scheme has been broken (represented in the graph with a red bar), the number of years that occurred to find a successful attack, and the best breaking percentage achieved. As shown in Figure 1, most CAPTCHA schemes have been successfully broken with a high success rate in few years.

3.1 Attacks against Text-based CAPTCHA

A lot of works suggested methods to break the different type of text-based CAPTCHAs. In 2003, Mori and Malik [83] proposed a method based on shape context matching to break both Gimpy and

Fig. 1. Timeline of the security breaking of the CAPTCHA schemes.

EZ-Gimpy CAPTCHAs with 33% and 92% accuracy, respectively. In Reference [84], EZ-Gimpy was also broken with a success rate of 99% using a correlation algorithm and a direct distortion estimation algorithm. In 2005, Chellapilla et al. [18, 19] were able to break various text-based CAPTCHAs by using machine learning and suggested a secure CAPTCHA scheme based on hard-segmentation problems. In 2008, Yan and El Ahmad showed that some segmentation-resistant CAPTCHAs could be broken, including the ones used by Microsoft, Google, and Yahoo [136, 137]. Later, other researchers attempted also to break these CAPTCHA schemes and they achieved higher success rates [118, 143]. El Ahmad and Yan [34] were able to break Megaupload CAPTCHA with a success

rate of 78%. In 2014, researchers from Google [46] broke the hardest category of ReCAPTCHA using neural networks with an accuracy of 99.8%.

In Reference [93], the authors discovered a set of attacks against 3D CAPTCHAs, even without the usage of OCR programs. In detail, they were able to successfully extract a set of pixels from the characters of several 3D CAPTCHA schemes (i.e., Teabag 3D, 3dcaptcha, and Super CAPTCHA) that can be used for automated recognition of the challenge. Thanks to such a technique, the authors were able to achieve success rates of 31%, 58%, and 27% in breaking Teabag 3D, 3dcaptcha, and Super CAPTCHA, respectively. Furthermore, the same authors in Reference [91] were able to break Teabag 3D with a higher success rate (i.e., 76%) by exploiting the side surface information contained in the 3D text objects. Nguyen et al. [90] showed that the information across multiple animation frames in animated CAPTCHA schemes could be easily extracted using simple techniques such as the **PDM (Pixel Delay Map)** or **CL (Catching Line)** methods. They used these methods to defeat several animated CAPTCHAs with a high success rate, including iCAPTCHA, Atlantis, KillBot Professional, and Dracon CAPTCHA. In Reference [92], the same methods have been used to defeat different types of HelloCAPTCHA schemes with a success rate between 16%–100%, due to their weakness against segmentation attacks. Unlike HelloCAPTCHA, NuCaptcha is an animated CAPTCHA designed to be segmentation resistant. Since the characters are overlapped and crowded together, the PDM or CL methods used to defeat HelloCAPTCHA are not effective to separate the characters. However, NuCaptcha has been broken using more sophisticated attacks [13, 135]. Elie Bursztein [13] achieved a success rate of 90% by using bounding box shape analysis and an interest points (SIFT algorithm) density evaluation to isolate objects in each frame. Then he tracked these objects across multiple frames and kept only the 50 frames that contain the CAPTCHA answer.

3.2 Attacks against Image-based CAPTCHA

Many attacks have been suggested in the literature to bypass the different type of image-based CAPTCHAs. Golle [45] was able to break the Asirra scheme with a success rate of 10.3%. To do so, he used different features to train an **SVM (Support Vector Machine)** classifier to identify cats and dogs with 82.7% accuracy (i.e., accuracy for a single image). Hernandez-Castro et al. in Reference [62] proposed a side-channel attack that bypassed the HumanAuth challenge with a 92% success rate. Sivakorn et al. [117] have successfully attacked both Google and Facebook image-based CAPTCHA with success rates of 70.78% and 83.5%, respectively. In Reference [141], the authors broke the new and the old variation of reCAPTCHA V2 with 79% and 88% success rates, respectively. Furthermore, they broke the Facebook image CAPTCHA and the China Railway CAPTCHA with success rates of 86% and 90%, respectively. Cheung [22] successfully broke Avatar CAPTCHA using **Convolutional Neural Networks (CNN)**, with a high success rate of 99%. Gao et al. [41] broke both FR-CAPTCHA and FaceDCAPTCHA with success rates of 23% and 48%, respectively.

The Minteye CAPTCHA scheme was broken in Reference [69] by exploiting the concept of Sobel operators and the length of the edges of the image. The idea behind this attack is based on the observation that the more an image is swirled, the longer the edges in the image become. So, the breaking methods consists in summing the length of the edges in the image and then selecting the image with the lowest sum of edges as the correct answer.

In Reference [141], the authors broke different schemes of image-based CAPTCHAs, including the Tencent CAPTCHA. In detail, their proposal achieved 100% success rate even during the motion of the sliding puzzle to the target region. Hernandez-Castro et al. [64] proposed a very low-cost attack that does not attempt to solve image recognition or shape recognition problems but instead uses JPEG to measure the continuity of the image. Through this side-channel attack, they were able

to bypass the most popular sliding-based CAPTCHAs. In detail, they break Capy CAPTCHA with a 65.1% success rate, and by applying minor modifications, they were able to break KeyCAPTCHA and Garb CAPTCHA as well with success rates of 20% and 98.1%, respectively. Conti et al. [27] pointed out that Jigsaw CAPTCHA proposed by Gao et al. [44] is vulnerable to relay attack and random guess attack with a success rate of 6.66%. Lin et al. [77] broke Drawing CAPTCHA with an accuracy of 75%. They proposed an effective erosion-based breaking algorithm based on their observation of the difference between the size of the diamond-shaped dots and the dots used in the background as noise.

Although CAPTCHaStar authors tested its resiliency against several types of automated attacks such as traditional attacks, automated attacks using ad hoc heuristics, and attacks based on machine learning, recently, Gougeon and Lacharme [51] were able to break this CAPTCHA with a 96% success rate. In addition, they pointed out that the modification of the parameters does not prevent CAPTCHaStar against their proposed attack, which is based on the concentration of pixels (i.e., stars) during the formation of the image. In Reference [27] the authors pointed out that the resiliency of Cursor CAPTCHA to machine learning-based attacks and stream relay attack is low.

3.3 Attacks against Audio-based CAPTCHA

Tam et al. [120] were the first to evaluate the robustness of audio CAPTCHAs against automated attacks. They were able to break audio reCAPTCHA using an SVM-based approach. They achieved a success rate of 45% when they matched the solution exactly and 58% when they leveraged a "one mistake" passing condition. Burzstein and Bethard [15] introduced Decaptcha, a system that was able to bypass the eBay's audio CAPTCHAs with a 75% success rate. Their system applies a **Discrete Fourier Transform (DFT)** to the wave file and then isolates the energy spikes. Afterward, it uses a supervised learning algorithm to recognize speech patterns. In Reference [14] the authors proposed a CAPTCHA solver based on the non-continuous speech, which defeated the Microsoft and the Yahoo audio CAPTCHAs with a success rate of 49% and 45%, respectively. The segmentation phase was unsupervised, while the classification phase was supervised. They used the **Regularized Least-Squares Classification (RLSC)** algorithm for classification and Amazon Mechanical Turk to label scraped CAPTCHAs. However, their system was able to solve reCAPTCHA with only 1.5% success rate, due to the presence of semantic vocal noise. Sano et al. [108] developed a CAPTCHA solver for continuous CAPTCHAs that use overlapping target voices as defensive techniques to make automated segmentation difficult. Their system applied **Hidden Markov Models (HMMs)** for speech recognition. It was tested on the version of audio reCAPTCHA used in 2013, and the results show that it was able to break this version of continuous reCAPTCHA with a success rate of 52%. Bock et al. [9] introduced unCaptcha, an automated system that can bypass audio reCAPTCHA released in 2017 with an 85.15% success rate. They attained these results by leveraging free online speech-to-text services and performing a minimal phonetic mapping to enhance accuracy.

3.4 Attacks against Behavior-based CAPTCHA

Although Sliding-based behavioral CAPTCHA schemes attempted to increase the security of sliding CAPTCHAs by detecting malicious behaviors, recently, Zhao et al. [141] were able to bypass such a detection by leveraging four simulation functions (i.e., Sigmoid, Softmax, ReLu, and Tanh) to mimic human behaviors. Their proposed attack against the GeeTest and Netease CAPTCHA schemes achieves the best success rate of 96% and 98% respectively, by using the Sigmoid function. Furthermore, Sivakorn et al. [117] found that Google's tracking cookies can be used to influence the risk analysis and, thus, bypass the reCAPTCHA V2 restrictions. In detail, the authors designed a tracking cookie for bots that was able, after nine days of automated browsing across different

Google services, to fool the Google risk analysis system into thinking that the traffic is made by human beings and, consequently, to check the "I'm not a robot" box. Furthermore, the authors proposed a low-cost attack that breaks the second layer of reCAPTCHA V2 with a success rate of 70.78%. In Reference [5], the authors used a "divide and conquer" strategy to defeat the No CAPTCHA reCAPTCHA scheme for any grid resolution. They achieved a success rate of 97.4% on a 100×100 grid and 96.7% on a 1000×1000 screen resolution.

3.5 Attacks against the Other Type of CAPTCHA

Kluever et al. [73] performed a tag frequency-based attack to evaluate the security of their proposed video-based CAPTCHA and achieved a success rate of 13%. Hernandez-Castro et al. [61] were able to break QRBGS CAPTCHA using a side-channel attack with a success rate of 44.54%. In Reference [80], Mohamed et al. reported that DCG CAPTCHAs, including PlayThru, are vulnerable to dictionary-based automated attacks. In Reference [32], a developer proposed a solver that automatically bypasses SweetCAPTCHA. In Reference [125], different variations of slider CAPTCHAs, including the Taobao scheme, have been bypassed by using a simple JavaScript code and puppeteer.

4 EVOLUTION OF CAPTCHA SCHEMES

The evolution of CAPTCHA schemes follows the advancements of technology to break them. In the early 2000s, text-based CAPTCHAs were the dominant solutions to discern between human and automated users. To this aim, security experts developed a set of attacks to break the most popular text-based schemes by leveraging image processing, pattern recognition, and machine learning algorithms [16]. Furthermore, the scientific community attempted to enhance the security of existing text-based CAPTCHAs by applying anti-segmentation and anti-recognition techniques. However, these countermeasures made text-based CAPTCHAs challenging even for human users, resulting in a higher error rate and limited usability that reduces text-based schemes' popularity. Finally, in 2014 a research conducted by Google demonstrated that the advancements in the AI technology could solve the most complicated variants of distorted text at 99.8% accuracy [46], leading to the decline of the text-based CAPTCHA schemes.

The security weaknesses of text-based CAPTCHAs and its usability issues, especially with the advent of mobile devices, led many researchers to look for alternatives. Since 2004, many of them have focused on exploiting **Computer Vision (CV)** problems such as image classification and object recognition that were considered harder AI problems than character recognition at that time. Chew and Tygar [24] were among the first researchers using labeled images to design image-based CAPTCHAs. After that, many images-based CAPTCHAs schemes have been proposed to create challenges that require selection, drag and drop, or sliding of images to discern between human and automated usages. However, the advancement in CV and machine learning and the advent of **Machine Learning as a service (MLaaS)** solutions boosted the breaking of the major image-based CAPTCHA schemes between 2013 and 2018.

For instance, the authors of Reference [141] exploited ML to perform attacks against several image-based CAPTCHAs, including the image-based reCAPTCHA V2 scheme.

Furthermore, the authors proposed several countermeasures, including the use of distortion techniques on characters on the background image or in the hint, the addition of noise on background images, and the use of adversarial examples to hinder deep learning models. In this regard, the concept of adversarial examples was first introduced by Szegedy et al. [119], and, since then, many researchers proposed CAPTCHA schemes based on adversarial examples to improve its security against ML-based bot attacks [65, 98, 112]. However, Na et al. [85] recently proposed an efficient CAPTCHA solver that breaks adversarial CAPTCHAs using incremental learning with only a small dataset. The authors demonstrated that existing defense methods (e.g., References

[98, 112]) that use adversarial examples in CAPTCHA schemes are not effective against their proposed adaptive CAPTCHA solver.

In conjunction with the advent of text-based and image-based CAPTCHAS, the security experts proposed Audio-based CAPTCHAs to cope with visually impaired users. However, those schemes are limited by language barriers and low usability, as discussed in Reference [8]. Furthermore, they are also weak against supervised learning and **automatic Speech Recognition (ASR)** attacks [70].

Starting from the 2010s, the research community introduced behavioral-based CAPTCHA schemes to build challenges based on behavioral biometrics measurements. The first deployed behavioral-based CAPTCHA was introduced in 2012 by the Geetest company, while in 2014, Google released No CAPTCHA reCAPTCHA and later on Invisible CAPTCHA (2017).

Still, most of the commercial and academic proposals are based on mouse dynamics, which have been shown to be vulnerable to bots attacks that attempt to mimic the user's behavioral pattern [102, 141]. As shown in the timeline of Figure 1, the most widespread behavioral CAPTCHAs (i.e., No CAPTCHA reCAPTCHA, GEETest, and Netease) have been broken with a high success rate [5, 141] in past years.

In addition, behavioral-based CAPTCHA schemes raise serious privacy concerns as described in References [10, 66, 110]. For instance, Reference [10] demonstrated how demographic attributes such as gender, age group, and education level could be extracted while solving a simple game CAPTCHA (e.g., Gametrics) by capturing user's innate cognitive abilities and behavioral patterns. Due to such concerns, Cloudflare recently decided to move away from reCAPTCHA [79].

Finally, the latest research directions exploit the data gathered from sensors to build challenges that are difficult to be emulated by automated bots. At the time of writing, no study has been done to review or analyze the security strength of sensor-based CAPTCHAs, and none of the proposed solutions has been successfully bypassed.

5 OPEN ISSUES, CHALLENGES, AND OPPORTUNITIES

In this section, we identify the open issues in designing robust and usable CAPTCHA schemes, as well as the main challenges that a CAPTCHA designer might have to deal with, and opportunities for further study.

5.1 Resilience to Both Automated and Human Solver Relay Attacks

A CAPTCHA scheme can be considered highly secure when both the automated attack success rate is less than 0.01% [86, 137] and it is resilient to human solver relay attacks. Unfortunately, in the literature, most studies dedicated to the design of CAPTCHA schemes focus only on automated attacks, while only few of them take into account the resilience to human solver relay attacks.

The security level of traditional CAPTCHA schemes depends on the hardness of some AI problem. However, the progress of AI techniques and computing power has led to the breaking of these CAPTCHA schemes with high success rates [9, 46, 117, 141]. Therefore, to design the next generation CAPTCHA schemes, it is important to move away from schemes based on hard AI problems toward other approaches less vulnerable to learning-based attacks [63]. Recently, big companies such as Google, Alibaba, and Tencent have migrated towards behavior-based CAPTCHA schemes, while there is an initiative aiming at deploying a sensor-based CAPTCHA scheme that uses the same key concept of Invisible CAPPCHA [53] by a company called Brave [10].

As presented in detail in Section 3, all the popular conventional CAPTCHA schemes have been broken with high success rate by automated attacks, and most of them are also vulnerable to human solver relay attacks (the most notable exceptions being CAPTCHaStar, PlayThru, and Dynamic Cognitive Game CAPTCHA). Similarly, popular behavior-based CAPTCHA schemes have

also been broken with high success rate by automated attacks, and all of them are vulnerable to human solver attacks. Invisible reCAPTCHA and other academic proposals have not been broken yet, however with the advent of the fourth-generation bots that rotate through thousands of different IP addresses and mimic accurately the human behavior, it would be difficult to design a secure CAPTCHA based solely on the user behavior data that can be gathered in a normal (i.e., with no additional sensors or special hardware) environment. None of the sensor-based CAPTCHA has been broken yet by automated attacks; however, similar to the other types of CAPTCHA schemes, most of them are vulnerable to human solver relay attacks. The exception to this vulnerability is represented by the ones that have been specifically designed to resist this kind of attack (e.g., Invisible CAPPCHA). Another weakness of sensor-based CAPTCHA schemes is the limited number of challenges. This is due to the fact that designing a large number of usable gestures, for instance, to ensure high security against automated attacks, is very challenging. However, this weakness may be solved relying on trusted hardware.

On the basis of the above observations, we identified the following open problems that require further study to design robust and usable CAPTCHA schemes: It is necessary to investigate (1) the resilience of currently unbroken behavior-based CAPTCHAs against fourth-generation bots; (2) the security strength of sensor-based CAPTCHA schemes against replay attacks, sensor manipulation [82], and human solver relay attacks; (3) the security of CAPTCHA schemes that make validation process at the client-side either with or without secure hardware, as they may be hacked.

5.2 Friction-heavy vs. Frictionless Challenges

CAPTCHA schemes are well known as a source of annoyance to users. This is due to the fact that most of the time designers trying to make the scheme more secure also make the challenge harder for humans. It is important to reduce the friction in general and the cognitive overload associated to the challenges. Creating user-friendly CAPTCHAs, yet, is not always an easy task, and in many cases there is a tradeoff between security and usability. Some CAPTCHA schemes achieve complete transparency to users (i.e., invisible reCAPTCHA, invisible CAPPCHA) removing all cognitive challenges. However, it is worth noting that not all the CAPTCHA schemes in the same category (i.e., behavioral-based and sensor-based) are automatically endowed with the same level of usability. In fact, while some of them require a simple task such as clicking on a check box or tilting the device, others require less user-friendly tasks such as solving a complex cognitive task, performing a physical task such as walking a few steps, or performing complex gestures.

To the best of our knowledge, there is no study fully dedicated to the analysis of the usability of behavior-based and sensor-based CAPTCHA schemes. Therefore, we argue that such a study would allow assessing the level of usability of all the CAPTCHA schemes proposed in the behavioral-based and sensor-based categories.

5.3 Preserving the User's Privacy

Unlike traditional CAPTCHA schemes, it has been shown that the new behavior-based and sensor-based CAPTCHA schemes may raise a privacy issue when information such as user's behavioral data, sensor data, and cookies that can be used for tracking are sent to a remote server. As a solution, some researchers suggested to send solely the results of the test to the server, instead of the sensor data. However, trusted hardware is then required to prevent hacking at the client side. Further study is needed to identify methodologies capable of preventing client-side hacking without requiring trusted hardware. Besides, the user's privacy should be taken into strong consideration in general from the very start of the design phase of new CAPTCHA schemes.

5.4 Compatibility with All Devices

A robust and usable CAPTCHA scheme that is compatible with different form factors is obviously highly desirable, however, the most promising CAPTCHA schemes category in terms of security and usability present a significant dependency on a specific form factor. For instance, behavioral-based CAPTCHA schemes strongly rely on mouse dynamics or on touch-and-tap dynamics, hence they require form-factor specialization. Sensor-based CAPTCHA schemes require sensors that are available only in tablets, smartphones, and smartwatches (e.g., References [56, 58]), hence they are currently unavailable on a large portion of users' devices, and further studies to find potential surrogates of sensors data, possibly relying on trusted hardware on desktops and laptops, are needed.

6 CONCLUSION

CAPTCHA has been widely used as a security mechanism to prevent bots from abusing online services. Over the years, different types of CAPTCHA schemes have been proposed, mainly to improve the usability and the security against new threats presented by evolving bots. The studies in the literature usually focus on the conventional CAPTCHA schemes, i.e., text-, image-, and audio-based schemes, and do not take into account either new types of schemes or novel threats such as human solver relay attacks, sensor manipulation [82], and the risk of privacy breaches. In this article, we have provided a comprehensive review of the related research involving two decades by also highlighting the new trends and open issues. We have first presented a comprehensive classification of the current CAPTCHA schemes that includes both traditional and new ones. Then, to evaluate their drawbacks from the security point of view, we have provided a detailed summary on the attack methods that have been used to break CAPTCHA schemes in each category. Finally, we have discussed the current state-of-the-art in the field of CAPTCHA schemes design, highlighting the open issues, the challenges, and the opportunities for further research that constitute the road toward the design of the next generation of secure and user-friendly CAPTCHA schemes.

REFERENCES

[1] josscrowcroft.com. 2011. MotionCAPTCHA v0.2, Stop Spam, Draw Shapes. Retrieved from http://www.josscrowcroft.com/demos/motioncaptcha/.

[2] E. Ababtain and D. Engels. 2019. Gestures based CAPTCHAs the use of sensor readings to solve CAPTCHA challenge on Smartphones. In *International Conference on Computational Science and Computational Intelligence (CSCI'19)*. 113–119. DOI: https://doi.org/10.1109/CSCI49370.2019.00026

[3] Alejandro Acien, Aythami Morales, Julian Fiérrez, and Rubén Vera-Rodriguez. 2020. BeCAPTCHA-Mouse: Synthetic mouse trajectories and improved bot detection. *ArXiv* abs/2005.00890 (2020).

[4] Alejandro Acien, Aythami Morales, Julian Fierrez, Ruben Vera-Rodriguez, and Ivan Bartolome. 2020. Be-CAPTCHA: Detecting human behavior in smartphone interaction using multiple inbuilt sensors. In *AAAI Workshop on Artificial for Cyber Security (AICS'20)*.

[5] Ismail Akrout, Amal Feriani, and Mohamed Akrout. 2019. Hacking Google reCAPTCHA v3 using reinforcement learning. *ArXiv* abs/1903.01003 (2019).

[6] Henry S. Baird and Jon L. Bentley. 2005. 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. DOI: https://doi.org/10.1117/12.590944

[7] M. Tariq Banday and Nisar A. Shah. 2011. A study of CAPTCHAs for securing web services. *arXiv preprint arXiv:1112.5605* (2011).

[8] Jeffrey P. Bigham and Anna C. Cavender. 2009. Evaluating existing audio CAPTCHAs and an interface optimized for non-visual use. In *SIGCHI Conference on Human Factors in Computing Systems (CHI'09)*. Association for Computing Machinery, New York, NY, 1829–1838. DOI: https://doi.org/10.1145/1518701.1518983

[9] Kevin Bock, Daven Patel, George Hughey, and Dave Levin. 2017. UnCaptcha: A low-resource defeat of Recaptcha's audio challenge. In *11th USENIX Conference on Offensive Technologies (WOOT'17)*. USENIX Association, 7.

[10] Brave. 2019. zkSENSE: A privacy-preserving mechanism for bot detection in mobile devices. Retrieved from https://brave.com/zksense-a-privacy-preserving-mechanism-for-bot-detection-in-mobile-devices/.

[11] Darko Brodić and Alessia Amelio. 2020. *Types of CAPTCHA*. Springer International Publishing, Cham, 29–32. DOI: https://doi.org/10.1007/978-3-030-29345-1_6

[12] Darko Brodic and Alessia Amelio. 2019. Exploring the usability of the text-based CAPTCHA on tablet computers. *Connect. Sci.* 31, 4 (2019), 430–444. DOI: https://doi.org/10.1080/09540091.2019.1609417.

[13] Elie Bursztein. 2012. How we broke the NuCaptcha video scheme and what we propose to fix it. Retrieved from https://elie.net/blog/security/how-we-broke-the-nucaptcha-video-scheme-and-what-we-propose-to-fix-it/.

[14] E. Bursztein, R. Beauxis, H. Paskov, D. Perito, C. Fabry, and J. Mitchell. 2011. The failure of noise-based non-continuous audio Captchas. In *IEEE Symposium on Security and Privacy*. 19–31. DOI: https://doi.org/10.1109/SP.2011.14

[15] Elie Bursztein and Steven Bethard. 2009. Decaptcha: Breaking 75% of eBay audio CAPTCHAs. In *3rd USENIX Conference on Offensive Technologies*. USENIX Association.

[16] Elie Bursztein, Matthieu Martin, and John Mitchell. 2011. Text-based CAPTCHA strengths and weaknesses. In *18th ACM Conference on Computer and Communications Security (CCS'11)*. Association for Computing Machinery, New York, NY, 125–138. DOI: https://doi.org/10.1145/2046707.2046724

[17] Capy Inc. 2018. Capy Puzzle CAPTCHA. Retrieved from https://www.capy.me/products/puzzle_captcha/.

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

[19] Kumar Chellapilla, Kevin Larson, Patrice Y. Simard, and Mary Czerwinski. 2005. Building segmentation based human-friendly human interaction proofs (HIPs). In *Human Interactive Proofs*, Henry S. Baird and Daniel P. Lopresti (Eds.). Springer Berlin, 1–26. DOI: https://doi.org/10.1007/11427896_1

[20] Kumar Chellapilla and Patrice Y. Simard. 2004. Using machine learning to break visual human interaction proofs (HIPs). In *17th International Conference on Neural Information Processing Systems (NIPS'04)*. The MIT Press, Cambridge, MA, 265–272. DOI: https://doi.org/10.5555/2976040.2976074

[21] J. Chen, Xiangyang Luo, Yanqing Guo, Y. Zhang, and Daofu Gong. 2017. A survey on breaking technique of text-based CAPTCHA. *Secur. Commun. Netw.* 2017 (2017), 6898617:1–6898617:15.

[22] B. Cheung. 2012. Convolutional neural networks applied to human face classification. In *11th International Conference on Machine Learning and Applications*. 580–583. DOI: https://doi.org/10.1109/ICMLA.2012.177

[23] Monica Chew and Henry S. Baird. 2003. BaffleText: a human interactive proof. In *Document Recognition and Retrieval X*, Tapas Kanungo, Elisa H. Barney Smith, Jianying Hu, and Paul B. Kantor (Eds.), Vol. 5010. International Society for Optics and Photonics, SPIE, 305–316. DOI: https://doi.org/10.1117/12.479682

[24] Monica Chew and J. D. Tygar. 2004. Image recognition CAPTCHAs. In *Information Security*, Kan Zhang and Yuliang Zheng (Eds.). Springer Berlin, 268–279. DOI: https://doi.org/10.1007/978-3-540-30144-8_23

[25] Richard Chow, Philippe Golle, Markus Jakobsson, Lusha Wang, and XiaoFeng Wang. 2008. Making CAPTCHAs clickable. In *9th Workshop on Mobile Computing Systems and Applications (HotMobile'08)*. Association for Computing Machinery, New York, NY, 91–94. DOI: https://doi.org/10.1145/1411759.1411783

[26] Yang-Wai Chow, Willy Susilo, and Pairat Thorncharoensri. 2019. *CAPTCHA Design and Security Issues*. Springer Singapore, 69–92. DOI: https://doi.org/10.1007/978-981-13-1483-4_4

[27] Mauro Conti, Claudio Guarisco, and Riccardo Spolaor. 2016. 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. DOI: https://doi.org/10.1007/978-3-319-39555-5_33

[28] J. Cui, J. Mei, X. Wang, D. Zhang, and W. Zhang. 2009. A CAPTCHA implementation based on 3D animation. In *International Conference on Multimedia Information Networking and Security*. 179–182. DOI: https://doi.org/10.1109/MINES.2009.298

[29] Corey Cummings. 2012. PlayThru: A Gaming Alternative to CAPTCHA Bot Checks. Retrieved from https://techli.com/playthru-captcha-alternative/30109/.

[30] dice-captcha.com. 2010. Dice CAPTCHA. Retrieved from http://dice-captcha.com/demo-dice-captcha.php.

[31] Dracon Projects. 2006. Dracon Visual Flash CAPTCHA. Retrieved from https://www.dracon.biz/captcha.php.

[32] drdre1. 2016. Sweet CAPTCHA solver. Retrieved from https://github.com/drdre1/Adultddl-Sweet-Captcha-Solver.

[33] D. D'Souza, P. C. Polina, and R. V. Yampolskiy. 2012. Avatar CAPTCHA: Telling computers and humans apart via face classification. In *IEEE International Conference on Electro/Information Technology*. 1–6. DOI: https://doi.org/10.1109/EIT.2012.6220734

[34] Ahmad Salah El Ahmad, Jeff Yan, and Lindsay Marshall. 2010. The robustness of a new CAPTCHA. In *3rd European Workshop on System Security (EUROSEC'10)*. Association for Computing Machinery, New York, NY, 36–41. DOI: https://doi.org/10.1145/1752046.1752052

[35] Jeremy Elson, John R Douceur, Jon Howell, and Jared Saul. 2007. Asirra: A CAPTCHA that exploits interest-aligned manual image categorization. In *14th ACM Conference on Computer and Communications Security (CCS'07)*. Association for Computing Machinery, New York, NY, 366–374. DOI: https://doi.org/10.1145/1315245.1315291

[36] Yunhe Feng, Qing Cao, Hairong Qi, and Scott Ruoti. 2020. SenCAPTCHA: A mobile-first CAPTCHA using orientation sensors. In *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies Conference*. 1–26. DOI: https://doi.org/10.1145/3397312

[37] Diogo Daniel Ferreira, Luís Leira, Petya Mihaylova, and Petia Georgieva. 2019. Breaking text-based CAPTCHA with sparse convolutional neural networks. In *Pattern Recognition and Image Analysis*, Aythami Morales, Julian Fierrez, José Salvador Sánchez, and Bernardete Ribeiro (Eds.). Springer International Publishing, Cham, 404–415. DOI: https://doi.org/10.1007/978-3-030-31321-0_35

[38] I. Fischer and T. Herfet. 2006. Visual CAPTCHAs for document authentication. In *IEEE Workshop on Multimedia Signal Processing*. 471–474. DOI: https://doi.org/10.1109/MMSP.2006.285353

[39] Brandon Z. Frank and Joseph A. Latone. 2018. Verifying a user utilizing gyroscopic movement. Retrieved from http://www.freepatentsonline.com/9942768.html. Patent 9942768.

[40] Christoph Fritsch, Michael Netter, Andreas Reisser, and Günther Pernul. 2010. Attacking image recognition Captchas. In *Trust, Privacy and Security in Digital Business*, Sokratis Katsikas, Javier Lopez, and Miguel Soriano (Eds.). Springer Berlin, 13–25. DOI: https://doi.org/10.1007/978-3-642-35130-3_23

[41] 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. DOI: https://doi.org/10.1109/CIT/IUCC/DASC/PICOM.2015.332

[42] Haichang Gao, Wei Wang, Jiao Qi, Xuqin Wang, Xiyang Liu, and Jeff Yan. 2013. The robustness of hollow CAPTCHAs. In *ACM SIGSAC Conference on Computer & Communications Security (CCS'13)*. Association for Computing Machinery, New York, NY, 1075–1086. DOI: https://doi.org/10.1145/2508859.2516732

[43] H. Gao, J. Yan, Fang Cao, Zhengya Zhang, Lei Lei, Mengyun Tang, P. Zhang, X. Zhou, Xuqin Wang, and J. Li. 2016. A simple generic attack on text captchas. In *Network and Distributed System Security Symposium*.

[44] H. Gao, D. Yao, H. Liu, X. Liu, and L. Wang. 2010. A novel image based CAPTCHA using jigsaw puzzle. In *13th IEEE International Conference on Computational Science and Engineering*. 351–356. DOI: https://doi.org/10.1109/CSE.2010.53

[45] Philippe Golle. 2008. Machine learning attacks against the Asirra CAPTCHA. In *15th ACM Conference on Computer and Communications Security (CCS'08)*. Association for Computing Machinery, New York, NY, 535–542. DOI: https://doi.org/10.1145/1455770.1455838

[46] Ian J. Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay D. Shet. 2014. Multi-digit number recognition from street view imagery using deep convolutional neural networks. *CoRR* abs/1312.6082 (2014).

AQ2 retrieved on 08/25/2021 [47] Google. [n.d.]. Choosing the type of reCAPTCHA. Retrieved from https://developers.google.com/recaptcha/docs/versions.

[48] Rich Gossweiler, Maryam Kamvar, and Shumeet Baluja. 2009. What's up CAPTCHA? A CAPTCHA based on image orientation. In *18th International Conference on World Wide Web (WWW'09)*. Association for Computing Machinery, New York, NY, 841–850. DOI: https://doi.org/10.1145/1526709.1526822

[49] Gaurav Goswami, Brian Powell, Mayank Vatsa, Richa Singh, and Afzel Noore. 2014. FaceDCAPTCHA: Face detection based color image CAPTCHA. *Fut. Gen. Comput. Syst.ems* 31 (2014), 59–68. DOI: https://doi.org/10.1016/j.future.2012.08.013

[50] Gaurav Goswami, Brian M. Powell, Mayank Vatsa, Richa Singh, and Afzel Noore. 2014. FR-CAPTCHA: CAPTCHA based on recognizing human faces. *PLoS ONE* 9 (2014). DOI: https://doi.org/10.1371/journal.pone.0091708

[51] Thomas Gougeon and Patrick Lacharme. 2018. How to break CAPTSHaStar. In *International Conference on Information Systems Security and Privacy*. DOI: https://doi.org/10.5220/0006577600410051

[52] Meriem Guerar, Alessio Merlo, and Mauro Migliardi. 2018. Completely automated public physical test to tell computers and humans apart: A usability study on mobile devices. *Fut. Gen. Comput. Syst.* 82 (2018), 617–630. DOI: https://doi.org/10.1016/j.future.2017.03.012

[53] Meriem Guerar, Alessio Merlo, Mauro Migliardi, and Francesco Palmieri. 2018. Invisible CAPPCHA: A usable mechanism to distinguish between malware and humans on the mobile IoT. *Comput. Secur.* 78 (2018), 255–266. DOI: https://doi.org/10.1016/j.cose.2018.06.007

[54] M. Guerar, M. Migliardi, A. Merlo, M. Benmohammed, and B. Messabih. 2015. 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'15)*. 203–210. DOI: https://doi.org/10.1109/HPCSim.2015.7237041

[55] M. Guerar, M. Migliardi, A. Merlo, M. Benmohammed, F. Palmieri, and A. Castiglione. 2018. Using screen brightness to improve security in mobile social network access. *IEEE Trans. Depend. Sec. Comput.* 15, 4 (2018), 621–632. DOI: https://doi.org/10.1109/TDSC.2016.2601603

[56] Meriem Guerar, Mauro Migliardi, Francesco Palmieri, Luca Verderame, and Alessio Merlo. 2020. Securing PIN-based authentication in smartwatches with just two gestures. *Concurr. Comput.: Pract. Exper.* 32, 18 (2020), e5549.

[57] Meriem Guerar, Benmohammed Mohamed, and Vincent Alimi. 2016. Color wheel pin: Usable and resilient ATM authentication. *J. High Speed Netw.* 22 (06 2016), 231–240. DOI: https://doi.org/10.3233/JHS-160545

[58] Meriem Guerar, Luca Verderame, Alessio Merlo, Francesco Palmieri, Mauro Migliardi, and Luca Vallerini. 2020. CirclePIN: A novel authentication mechanism for smartwatches to prevent unauthorized access to IoT devices. *ACM Trans. Cyber-Phys. Syst.* 4, 3 (Mar. 2020). DOI: https://doi.org/10.1145/3365995

[59] M. Guerar, L. Verderame, M. Migliardi, and A. Merlo. 2019. 2GesturePIN: Securing PIN-Based authentication on smartwatches. In *IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'19)*. 327–333. DOI: https://doi.org/10.1109/WETICE.2019.00074

[60] F. A. B. Hamid Ali and F. B. Karim. 2014. Development of CAPTCHA system based on puzzle. In *International Conference on Computer, Communications, and Control Technology (I4CT'14)*. 426–428. DOI: https://doi.org/10.1109/I4CT.2014.6914219

[61] Carlos Javier Hernandez-Castro and Arturo Ribagorda. 2010. Pitfalls in CAPTCHA design and implementation: The Math CAPTCHA, a case study. *Comput. Secur.* 29, 1 (2010), 141–157. DOI: https://doi.org/10.1016/j.cose.2009.06.006

[62] C. J. Hernandez-Castro, A. Ribagorda, and Y. Saez. 2010. Side-channel attack on the HumanAuth CAPTCHA. In *International Conference on Security and Cryptography (SECRYPT'10)*. 1–7.

[63] Carlos Javier Hernández-Castro, Shujun Li, and María D. R-Moreno. 2020. All about uncertainties and traps: Statistical oracle-based attacks on a new CAPTCHA protection against oracle attacks. *Comput. Secur.* 92 (2020), 101758. DOI: https://doi.org/10.1016/j.cose.2020.101758

[64] C. J. Hernández-Castro, M. D. R-Moreno, and D. F. Barrero. 2015. Using JPEG to measure image continuity and break capy and other puzzle CAPTCHAs. *IEEE Internet Comput.* 19, 6 (2015), 46–53. DOI: https://doi.org/10.1109/MIC.2015.127

[65] D. Hitaj, B. Hitaj, S. Jajodia, and L. V. Mancini. 2020. Capture the bot: Using adversarial examples to improve CAPTCHA robustness to bot attacks. *IEEE Intell. Syst.* (2020), 1–1. DOI: https://doi.org/10.1109/MIS.2020.3036156

[66] Thomas Hupperich, Katharina Krombholz, and Thorsten Holz. 2016. 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. DOI: https://doi.org/10.1007/978-3-319-45572-3_3

[67] Imperva. 2020. 2020 Bad Bot Report. Retrieved from https://www.imperva.com/resources/reports/Imperva_BadBot_Report_V2.0.pdf.

[68] M. Imsamai and S. Phimoltares. 2010. 3D CAPTCHA: A next generation of the CAPTCHA. In *International Conference on Information Science and Applications*. 1–8. DOI: https://doi.org/10.1109/ICISA.2010.5480258

AQ3 J. W. Andrews

[69] Jack. 2013. Breaking the MintEye image CAPTCHA in 23 lines of Python. Retrieved from http://www.jwandrews.co.uk/2013/01/breaking-the-minteye-image-captcha-in-23-lines-of-python/.

Q3

[70] Mohit Jain, Rohun Tripathi, Ishita Bhansali, and Pratyush Kumar. 2019. Automatic generation and evaluation of usable and secure audio ReCAPTCHA. In *21st International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS'19)*. Association for Computing Machinery, New York, NY, 355–366. DOI: https://doi.org/10.1145/3308561.3353777

[71] KeyCAPTCHA Inc. 2010. KeyCAPTCHA. Retrieved from https://www.keycaptcha.com/.

[72] Suzi Kim and Sunghee Choi. 2019. 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. DOI: https://doi.org/10.1007/978-3-030-19274-7_18

[73] Kurt Alfred Kluever and Richard Zanibbi. 2009. Balancing usability and security in a video CAPTCHA. In *5th Symposium on Usable Privacy and Security (SOUPS'09)*. Association for Computing Machinery, New York, NY. DOI: https://doi.org/10.1145/1572532.1572551

[74] S. Kulkarni and H. S. Fadewar. 2017. Pedometric CAPTCHA for mobile Internet users. In *2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT'17)*. 600–604. DOI: https://doi.org/10.1109/RTEICT.2017.8256667

[75] Ching-Jung Liao, Chang-Ju Yang, Jin-Tan Yang, Hsiang-Yang Hsu, and Jhih-Wei Liu. 2013. A game and accelerometer-based CAPTCHA scheme for mobile learning system. In *Proceedings of EdMedia + Innovate Learning 2013*, Jan Herrington, Alec Couros, and Valerie Irvine (Eds.). Association for the Advancement of Computing in Education (AACE), Victoria, Canada, 1385–1390. Retrieved from https://www.learntechlib.org/p/112139.

[76] Mark D. Lillibridge, Martin Abadi, Krishna Bharat, and Andrei Broder. 2001. Method for selectively restricting access to computer systems. Retrieved from http://www.freepatentsonline.com/6195698.html. Patent 6195698.

[77] Rosa Lin, Shih-Yu Huang, Graeme B. Bell, and Yeuan-Kuen Lee. 2011. A new CAPTCHA interface design for mobile devices. In *12th Australasian User Interface Conference (AUIC'11)*. Australian Computer Society, Inc., AUS, 3–8.

[78] Viraj C. Mantri and Prateek Mehrotra. 2018. User authentication based on physical movement information. Retrieved from http://www.freepatentsonline.com/9864854.html. Patent 9864854.

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

[80] Manar Mohamed, Niharika Sachdeva, Michael Georgescu, Song Gao, Nitesh Saxena, Chengcui Zhang, Ponnurangam Kumaraguru, Paul C. van Oorschot, and Wei-Bang Chen. 2014. A three-way investigation of a game-CAPTCHA: Automated attacks, relay attacks and usability. In *9th ACM Symposium on Information, Computer and Communications Security (ASIA CCS'14)*. Association for Computing Machinery, New York, NY, 195–206. DOI: https://doi.org/10.1145/2590296.2590298

[81] Manar Mohamed and Nitesh Saxena. 2016. Gametrics: Towards attack-resilient behavioral authentication with simple cognitive games. *32nd Annual Conference on Computer Security Applications*. DOI: https://doi.org/10.1145/2991079.2991096

[82] M. Mohamed, B. Shrestha, and N. Saxena. 2017. SMASheD: Sniffing and manipulating Android sensor data for offensive purposes. *IEEE Trans. Inf. Forens. Secur.* 12, 4 (2017), 901–913. DOI: https://doi.org/10.1109/TIFS.2016.2620278

[83] G. Mori and J. Malik. 2003. Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. DOI: https://doi.org/10.1109/CVPR.2003.1211347

[84] G. Moy, N. Jones, C. Harkless, and R. Potter. 2004. Distortion estimation techniques in solving visual CAPTCHAs. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. DOI: https://doi.org/10.1109/CVPR.2004.1315140.

[85] Dongbin Na, Namgyu Park, Sangwoo Ji, and Jong Kim. 2020. CAPTCHAs are still in danger: An efficient scheme to bypass adversarial CAPTCHAs. In *Information Security Applications*, Ilsun You (Ed.). Springer International Publishing, Cham, 31–44.

[86] Rabih Al Nachar, Elie Inaty, Patrick J. Bonnin, and Yasser Alayli. 2015. Breaking down Captcha using edge corners and fuzzy logic segmentation/recognition technique. *Secur. Commun. Netw.* 8, 18 (2015), 3995–4012. DOI: https://doi.org/10.1002/sec.1316

[87] Moni Naor. 1996. Verification of a human in the loop or identification via the Turing test. Retrieved from http://www.wisdom.weizmann.ac.il/~{}naor/PAPERS/humanabs.html.

[88] Anja B. Naumann, Thomas Franke, and Christian Bauckhage. 2009. Investigating CAPTCHAs based on visual phenomena. In *Human-Computer Interaction–INTERACT 2009*, Tom Gross, Jan Gulliksen, Paula Kotzé, Lars Oestreicher, Philippe Palanque, Raquel Oliveira Prates, and Marco Winckler (Eds.). Springer Berlin, 745–748. DOI: https://doi.org/10.1007/978-3-642-03658-3_79

[89] Neo. 2006. Blog post, [HumanAuth] Verification code for natural patterns. Retrieved from http://www.neo.com.tw/archives/965.

[90] Vu Duc Nguyen, Yang-Wai Chow, and Willy Susilo. 2012. Attacking animated CAPTCHAs via character extraction. In *Cryptology and Network Security*, Josef Pieprzyk, Ahmad-Reza Sadeghi, and Mark Manulis (Eds.). Springer Berlin, 98–113. DOI: https://doi.org/10.1007/978-3-642-35404-5_9

[91] Vu Duc Nguyen, Yang-Wai Chow, and Willy Susilo. 2012. Breaking a 3D-based CAPTCHA scheme. In *Conference on Information Security and Cryptology*, Howon Kim (Ed.). Springer Berlin, 391–405. DOI: https://doi.org/10.1007/978-3-642-31912-9_26

[92] Vu Duc Nguyen, Yang-Wai Chow, and Willy Susilo. 2012. Breaking an Animated CAPTCHA Scheme. In *Applied Cryptography and Network Security*, Feng Bao, Pierangela Samarati, and Jianying Zhou (Eds.). Springer Berlin, 12–29. DOI: https://doi.org/10.1007/978-3-642-31284-7_2

[93] Vu Duc Nguyen, Yang-Wai Chow, and Willy Susilo. 2014. On the security of text-based 3D CAPTCHAs. *Comput. Secur.* 45 (2014), 84–99. DOI: https://doi.org/10.1016/j.cose.2014.05.004

[94] NuCaptcha Inc. 2018. NuCaptcha. Retrieved from https://www.nucaptcha.com.

[95] OCR Research Team. 2006. Teabag 3D evolution. Retrieved from https://ocr-research.org.ua/teabag.html.

[96] Marek R. Ogiela, Natalia Krzyworzeka, and Lidia Ogiela. 2018. Application of knowledge-based cognitive CAPTCHA in Cloud of Things security. *Concurr. Comput.: Pract. Exper.* 30, 21 (2018), e4769. DOI: https://doi.org/10.1002/cpe.4769

[97] M. Okada and S. Matsuyama. 2012. New CAPTCHA for smartphones and tablet PC. In *IEEE Consumer Communications and Networking Conference (CCNC'12)*. 34–35. DOI: https://doi.org/10.1109/CCNC.2012.6181038

[98] M. Osadchy, J. Hernandez-Castro, S. Gibson, O. Dunkelman, and D. Pérez-Cabo. 2017. No bot expects the DeepCAPTCHA! introducing immutable adversarial examples, with applications to CAPTCHA generation. *IEEE Trans. Inf. Forens. Secur.* 12, 11 (2017), 2640–2653. DOI: https://doi.org/10.1109/TIFS.2017.2718479

[99] Nancy Owano. 2012. Phys.org Blog post, Minteye offers no-type CAPTCHA as a security twist. Retrieved from https://phys.org/news/2012-12-minteye-no-type-captcha.html.

[100] Stefan Popoveniuc. 2010. SpeakUp: remote unsupervised voting. In *Industrial Track Applied Cryptography and Network Security*.

[101] Program Product. 2010. *HelloCAPTCHA*. Retrieved from http://www.hellocaptcha.com/.

[102] Radware. 2020. The Big Bad Bot Problem 2020.

[103] Steven Rees-Pullman. 2020. Is credential stuffing the new phishing? *Comput. Fraud Secur.* 2020, 7 (2020), 16–19. DOI: https://doi.org/10.1016/S1361-3723(20)30076-2

[104] Narges Roshanbin and James Miller. 2013. A survey and analysis of current CAPTCHA approaches. *J. Web Eng.* 12, 1–2 (Feb. 2013), 1–40. DOI: https://doi.org/10.5555/2481562.2481563

[105] C. Rui, Y. Jing, H. Rong-gui, and H. Shu-guang. 2013. A novel LSTM-RNN decoding algorithm in CAPTCHA recognition. In *3rd International Conference on Instrumentation, Measurement, Computer, Communication and Control*. 766–771. DOI: https://doi.org/10.1109/IMCCC.2013.171

[106] A. Rusu and V. Govindaraju. 2004. Handwritten CAPTCHA: Using the difference in the abilities of humans and machines in reading handwritten words. In *9th International Workshop on Frontiers in Handwriting Recognition*. 226–231. DOI: https://doi.org/10.1109/IWFHR.2004.54

[107] Amalia Rusu and Venu Govindaraju. 2005. Visual CAPTCHA with handwritten image analysis. In *Human Interactive Proofs*, Henry S. Baird and Daniel P. Lopresti (Eds.). Springer Berlin, 42–52. DOI: https://doi.org/10.1109/10.1007/11427896_3

[108] Shotaro Sano, Takuma Otsuka, and Hiroshi G. Okuno. 2013. 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, 36–52. DOI: https://doi.org/10.1007/978-3-642-41383-4_3

[109] Graig Sauer, Jonathan Holman, Jonathan Lazar, Harry Hochheiser, and Jinjuan Feng. 2010. Accessible privacy and security: A universally usable human-interaction proof tool. *Univers. Access Inf. Soc.* 9, 3 (Aug. 2010), 239–248. DOI: https://doi.org/10.1007/s10209-009-0171-2

[110] Katharine Schwab. 2019. Google's new reCAPTCHA has a dark side. Retrieved from https://www.fastcompany.com/90369697/googles-new-recaptcha-has-a-dark-side.

[111] Vinay Shet. 2014. Are you a robot? Introducing "No CAPTCHA reCAPTCHA." Retrieved from https://security.googleblog.com/2014/12/are-you-robot-introducing-no-captcha.html.

[112] Chenghui Shi, Xiaogang Xu, Shouling Ji, Kai Bu, Jianhai Chen, Raheem Beyah, and Ting Wang. 2019. Adversarial CAPTCHAs. *arXiv preprint arXiv:1901.01107* (2019).

[113] M. Shirali-Shahreza and S. Shirali-Shahreza. 2006. Drawing CAPTCHA. In *28th International Conference on Information Technology Interfaces*. 475–480. DOI: https://doi.org/10.1109/ITI.2006.1708527

[114] M. Shirali-Shahreza and S. Shirali-Shahreza. 2008. Motion CAPTCHA. In *Conference on Human System Interactions*. 1042–1044. DOI: https://doi.org/10.1109/HSI.2008.4581589

[115] Ved Prakash Singh and Preet Pal. 2014. Survey of different types of CAPTCHA. *Int. J. Comput. Sci. Inf. Technol.* 5, 2 (2014), 2242–2245.

[116] A. Siripitakchai, S. Phimoltares, and A. Mahaweerawat. 2017. EYE-CAPTCHA: An enhanced CAPTCHA using eye movement. In *3rd IEEE International Conference on Computer and Communications (ICCC'17)*. 2120–2126. DOI: https://doi.org/10.1109/CompComm.2017.8322911

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

[118] Oleg Starostenko, Claudia Cruz-Perez, Fernando Uceda-Ponga, and Vicente Alarcon-Aquino. 2015. Breaking text-based CAPTCHAs with variable word and character orientation. *Pattern Recog.* 48, 4 (2015), 1101–1112. DOI: https://doi.org/10.1016/j.patcog.2014.09.006

[119] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1312.6199.

[120] Jennifer Tam, Sean Hyde, Jiri Simsa, and Luis Von Ahn. 2008. Breaking audio CAPTCHAs. In *21st International Conference on Neural Information Processing Systems (NIPS'08)*. Curran Associates Inc., Red Hook, NY, 1625–1632.

[121] M. Tang, H. Gao, Y. Zhang, Y. Liu, P. Zhang, and P. Wang. 2018. Research on deep learning techniques in breaking text-based Captchas and designing image-based Captcha. *IEEE Trans. Inf. Forens. Secur.* 13, 10 (2018), 2522–2537. DOI: https://doi.org/10.1109/TIFS.2018.2821096

[122] V. A. Thomas and K. Kaur. 2013. Cursor CAPTCHA—Implementing CAPTCHA using mouse cursor. In *10th International Conference on Wireless and Optical Communications Networks (WOCN'13)*. 1–5. DOI: https://doi.org/10.1109/WOCN.2013.6616188

[123] Erkam Uzun, Simon Pak Ho Chung, Irfan Essa, and Wenke Lee. 2018. rtCaptcha: A real-time CAPTCHA based liveness detection system. In *Network and Distributed System Security Symposium*. DOI: https://doi.org/10.14722/ndss.2018.23253

[124] Shardul Vikram, Yinan Fan, and Guofei Gu. 2011. SEMAGE: A new image-based two-factor CAPTCHA. In *27th Annual Computer Security Applications Conference (ACSAC'11)*. Association for Computing Machinery, New York, NY, 237–246. DOI: https://doi.org/10.1145/2076732.2076766

[125] Filip Vitas. 2019. How to bypass "slider CAPTCHA" with JS and Puppeteer. Retrieved from https://medium.com/@filipvitas/how-to-bypass-slider-captcha-with-js-and-puppeteer-cd5e28105e3c.

[126] Luis von Ahn, Manuel Blum, Nicholas Hopper, and John Langford. 2000. CAPTCHA: Telling Humans and Computers Apart Automatically. Retrieved from http://www.captcha.net/.

[127] Luis Von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. 2003. CAPTCHA: Using hard AI problems for security. In *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 294–311. DOI: https://doi.org/10.1007/3-540-39200-9_18

[128] Luis von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. 2008. reCAPTCHA: Human-based character recognition via web security measures. *Science* 321, 5895 (2008), 1465–1468. DOI: https://doi.org/10.1126/science.1160379.

[129] Luis von Ahn, Manuel Blum, Nick Hopper, John Langford, and Udi Manber. 2000. GIMPY. Retrieved from http://www.captcha.net/captchas/gimpy/.

[130] P. Wang, H. Gao, Z. Shi, Z. Yuan, and J. Hu. 2020. Simple and easy: Transfer learning-based attacks to text CAPTCHA. *IEEE Access* 8 (2020), 59044–59058. DOI: https://doi.org/10.1109/ACCESS.2020.2982945

[131] Michael L. Wells. 2003. Exciting Features in Super CAPTCHA. Retrieved from https://goldsborowebdevelopment.com/2013/06/exciting-features-in-super-captcha/.

[132] Wordpress.org. 2013. Garb CAPTCHA. Retrieved from https://wordpress.org/plugins/captcha-garb/.

[133] Luke Wroblewski. 2010. A Sliding Alternative to CAPTCHA? Retrieved from https://www.lukew.com/ff/entry.asp?1138.

[134] Xin Xu, Lei Liu, and Bo Li. 2020. A survey of CAPTCHA technologies to distinguish between human and computer. *Neurocomputing* (2020). DOI: https://doi.org/10.1016/j.neucom.2019.08.109

[135] Y. Xu, G. Reynaga, S. Chiasson, J. Frahm, F. Monrose, and P. C. van Oorschot. 2014. Security analysis and related usability of motion-based CAPTCHAs: Decoding codewords in motion. *IEEE Trans. Depend. Sec. Comput.* 11, 5 (2014), 480–493. DOI: https://doi.org/10.1109/TDSC.2013.52

[136] Jeff Yan and Ahmad Salah El Ahmad. 2008. *Is Cheap Labour Behind the scene? - Low-cost Automated Attacks on Yahoo CAPTCHAs*. Technical Report. School of Computing Science, Newcastle University, England.

[137] Jeff Yan and Ahmad Salah El Ahmad. 2008. A low-cost attack on a Microsoft Captcha. In *15th ACM Conference on Computer and Communications Security (CCS'08)*. Association for Computing Machinery, New York, NY, 543–554. DOI: https://doi.org/10.1145/1455770.1455839

[138] Tzu-I Yang, Chorng-Shiuh Koong, and Chien-Chao Tseng. 2013. Game-based image semantic CAPTCHA on handset devices. *Multimedia Tools Applic.* 74 (2013), 5141–5156. DOI: https://doi.org/10.1007/s11042-013-1666-7

[139] C. N. Yuan, and Jingxia Chongqing. 2018. Variation Analysis-Based Public Turing Test to Tell Computers and Humans Apart. Retrieved from http://www.freepatentsonline.com/y2018/0253542.html.

[140] Y. Zhang, H. Gao, G. Pei, S. Luo, G. Chang, and N. Cheng. 2019. A survey of research on CAPTCHA designing and breaking techniques. In *18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE'19)*. 75–84. DOI: https://doi.org/10.1109/TrustCom/BigDataSE.2019.00020

[141] Binbin Zhao, Haiqin Weng, Shouling Ji, Jianhai Chen, Ting Wang, Qinming He, and Reheem Beyah. 2018. Towards evaluating the security of real-world deployed image CAPTCHAs. In *11th ACM Workshop on Artificial Intelligence and Security (AISec'18)*. Association for Computing Machinery, New York, NY, 85–96. DOI: https://doi.org/10.1145/3270101.3270104

AQ5 Remove 143

[142] Y. Zi, H. Gao, Z. Cheng, and Y. Liu. 2020. An end-to-end attack on text CAPTCHAs. *IEEE Trans. Inf. Forens. Secur.* 15 (2020), 753–766. DOI: https://doi.org/10.1109/TIFS.2019.2928622

[143] ~~Y. Zi, H. Gao, Z. Cheng, and Y. Liu. 2020. An end-to-end attack on text CAPTCHAs. *IEEE Trans. Inf. Forens. Secur.* 15 (2020), 753–766. DOI: https://doi.org/10.1109/TIFS.2019.2928622~~ Q5

Received March 2021; revised May 2021; accepted July 2021

AUTHOR QUERIES

Q1: AU: Please provide complete mailing addresses for all authors.
Q2: AU: Please supply year of publication for Reference 47.
Q3: AU: Please supply author last name for Reference 69.
Q4: AU: Please supply publication or retrieval info for Reference 102.
Q5: AU: References 142 and 143 are identical.