CAPTCHA Design and Security Issues

Chapter · January 2019

DOI: 10.1007/978-981-13-1483-4_4

3 authors:

Yang-Wai Chow
University of Wollongong
57 PUBLICATIONS **311** CITATIONS
SEE PROFILE

Willy Susilo
University of Wollongong
674 PUBLICATIONS **11,389** CITATIONS
SEE PROFILE

Pairat Thorncharoensri
University of Wollongong
12 PUBLICATIONS **20** CITATIONS
SEE PROFILE

CAPTCHA Design and Security Issues

Yang-Wai Chow, Willy Susilo and Pairat Thorncharoensri

Institute of Cybersecurity and Cryptology
School of Computing and Information Technology
University of Wollongong, Australia
{caseyc, wsusilo, pairat}@uow.edu.au

Abstract. The concept of reverse Turing tests, or more commonly known as CAPTCHAs, for distinguishing between humans and computers has been around for many years. The widespread use of CAPTCHAs these days has made them an integral part of the Internet for providing online services, which are intended for humans, with some level of protection against automated abuse. Since their inception, much research has focused on investigating various issues surrounding the design and security of CAPTCHAs. A fundamental requirement of CAPTCHAs necessitates that they must be designed to be easy for humans but difficult for computers. However, it is well recognized that the trade-off between usability and security is difficult to balance. In addition, numerous attacks have been developed to defeat CAPTCHAs. In response to this, many different CAPTCHA design variants have been proposed over the years. Despite the fact that CAPTCHAs have been around for more than two decades, the future of CAPTCHAs remains an open question. This chapter presents an overview of research examining a wide range of issues that has been conducted on different types of CAPTCHAs.

Keywords: audio, image, CAPTCHA, machine learning, recognition, security, segmentation, text, usability

1 Introduction

CAPTCHAs, an acronym for *Completely Automated Public Turing test to tell Computers and Humans Apart*, have been around for many years. The term refers to automated tests that can be solved correctly by humans, but current computer programs will have difficulty solving. Over the years, CAPTCHAs have become an integral part of the Internet for deterring the automated abuse of online services that are intended for human users. CAPTCHAs have been used for protecting services against email spam, fraudulent online registrations, Denial of Service (DoS) attacks, online voting fraud, etc. [18].

It is widely recognized that the first use of CAPTCHAs was by the AltaVista search engine in 1997, as a method of deterring the automated submission of URLs to their service. It was reported that the use of this CAPTCHA was successful in reducing the amount of spam by over 95% [6]. This system was

invented by the DEC Systems Research Center, and it was later patented as a computerized method for a server to selectively accept access requests from client computers [16, 48]. Since then, CAPTCHAs have become a ubiquitous part of the Internet, and have even been used in online services provided by major international companies such as Amazon, Baidu, eBay, Facebook, Google, Microsoft, PayPal and Yahoo.

The term CAPTCHA was introduced by von Ahn et al. [77], where they put forward the challenge of using hard Artificial Intelligence (AI) problems for security purposes. While the term CAPTCHA has been adopted as the most widely used term when referring to automated challenge-response tests for ascertaining whether an online service is being used by a human or a computer program (otherwise known as a "bot"), such tests have also been referred to as *Human Interaction Proofs* (HIPs) [6]. In addition, CAPTCHAs are also called *reverse Turing tests*. One of the earliest notions of the reverse Turing test appeared in an unpublished manuscript by Naor [56]. This stems from the idea of the Turing test [75], where the intention of the original Turing test was for a human to attempt to distinguish between another human and a machine based on responses to questions posed to both the human and the machine. A reverse Turing test is one where a computer is used to distinguish between a human and a machine. This has also been referred to as an *automated Turing test*, since a computer is used to automated the process [77].

Despite the fact that many people consider CAPTCHAs to be annoying, CAPTCHAs have seen widespread adoption in many online services. This has resulted in much research on the security of CAPTCHAs, because many techniques for attacking and defeating CAPTCHAs have been developed. This in turn, has given rise to an arms race between the development of new CAPTCHAs and techniques for defeating them. Furthermore, unlike other computer security mechanisms, CAPTCHAs are unique in the sense that a practical CAPTCHA scheme must be design to be secure, i.e. robust against automated attacks, while at the same time it must be usable by humans.

This usability versus security requirement is a difficult act to balance. The design of a robust CAPTCHA must in someway capitalize on the difference in ability between humans and current computer programs [16]. With advances in machine learning and computer vision research, the challenge of whether it is possible to design a CAPTCHA that is easy for humans but difficult for current computer programs is an open challenge. To this end, it is important to examine previous research on CAPTCHAs to be able to identify any potential gaps in ability between humans and computers, as well as to avoid potential pitfalls and design flaws that can easily be exploited in a CAPTCHA.

To date, there are many different types of CAPTCHAs. Visual CAPTCHAs are the most prevalent, and include text-based and image-based CAPTCHAs. Audio CAPTCHAs have been developed in place of visual CAPTCHAs for users who are blind or visually impaired. In addition, other forms of CAPTCHAs, such as 3D-based CAPTCHAs [73], Math CAPTCHAs [40], game-based CAPTCHAs [52] and social authentication [85], have also been proposed. This chapter presents

an overview of research conducted on different types of CAPTCHAs over the years. Research on CAPTCHAs has examined a variety of issues ranging from usability and design to robustness and security.

2 Text-based CAPTCHAs

Of the different types of CAPTCHAs that have been deployed to protect online services, text-based CAPTCHAs are by far the most popular. The challenge presented to a user in this type of CAPTCHA typically involves the user having to identify a sequence of characters in an image, where the text contained within the image is deliberately rendered with some level of distortion and/or noise to deter character recognition programs from correctly solving the challenge. The reason for its widespread usage is due to its intuitiveness, as it is easily understood by users world-wide without much instruction, and most humans have been trained to recognize characters since childhood. In addition, from a security point of view, the brute force search space can be very large, and from a computation perspective, the generation of CAPTCHA challenges can easily be automated without the need for manual effort [18, 84].

2.1 Security and Usability

A fundamental requirement of a practical CAPTCHA necessitates that humans must be able to solve the challenge with a high degree of success, while the likelihood that a computer program can correctly solve it must be very small [60]. As a benchmark, Chellapilla et al. [18] state that the human success rate should approach at least 90%, while the success of computer programs should ideally be less than 1 in 10,000. CAPTCHAs that are difficult or time consuming for legitimate users to solve are often seen as annoying and greatly inconvenience the users.

It has been estimated that with more than 100 million CAPTCHAs being used by humans all over the world everyday, with each case taking a few seconds to solve, this collectively amounts to hundreds of thousands of man-hours per day. Furthermore, each person will have to utilize some mental effort when it comes to solving a CAPTCHA challenge [79]. In addition, demographics play a role in a user's solving abilities, e.g., in general, non-native English speakers perform slower on English-centric CAPTCHAs [13].

The trade-off between security and usability is difficult to balance, as security considerations push designers to increase the difficulty of CAPTCHAs to deter computer programs, while usability requirements restrict them to make a scheme only as difficult as it needs to be. Moreover, while technology and computing techniques are constantly evolving and improving, humans on the other hand, must rely on their inherent abilities and are not likely to get better at solving CAPTCHAs [16]. In a large-scale study on CAPTCHA usability, it was reported that humans often find CAPTCHAs difficult to solve, and that most research mainly focused on making them hard for machines while not paying much attention to usability issues [13].

Design Considerations. The problem faced by CAPTCHA designers is that the security versus usability requirements are often in conflict. A number of design considerations that deal with the trade-off between security and usability are described as follows:

- **Text familiarity:** Text-based CAPTCHAs generated using familiar text (e.g., English words), increases the usability of the resulting CAPTCHA. This is due to the fact that humans find familiar text easier to read as opposed to unfamiliar text [80]. However, language models can be exploited to break CAPTCHAs using a set of words from a dictionary and holistic approaches. Researchers have shown that CAPTCHAs based on language models can successfully be defeated using holistic approaches of recognizing entire words rather than identifying individual characters, which can be difficult if characters are extremely distorted or overlapping [4, 53].

  Instead of using words that can be found in a dictionary, it is possible to use *language-like* strings that are not part of any language. For example, phonetic text or Markov dictionary strings are pronounceable strings that are not actual words of any language. The reason for using pronounceable strings is for the resulting CAPTCHA to have the advantage of text familiarity for humans, while at the same time attempt to circumvent dictionary and holistic attacks. Nevertheless, while humans perform better at correctly identifying pronounceable strings in contrast to purely random character strings, certain characters (e.g., vowels) in pronounceable strings typically appear at higher frequencies when compared with other characters [80].

- **Color and visual clutter:** The use of color is an important consideration in the design of visual CAPTCHAs. From a usability perspective, the incorporation of color is good for attracting a user's visual attention. At the same time, color can make a CAPTCHA more appealing and less intrusive in its context of use, e.g., if it matches a webpage's color [16]. In addition, appropriate use of color can potentially aid in the recognition and comprehension of text [84].

  On the other hand, the use of color can have negative effects on both the usability and security of a CAPTCHA. This is because the misuse of color can result in the text being difficult to read for humans, e.g., if the text color is similar to the background color. Furthermore, colors may cause problems for people who are color-blind. At the same time, color can result in a computer easily being able to distinguish the text from the background, e.g., if text is displayed in a distinct color. It has been observed that in many CAPTCHAs, the use of color is neither helpful for its usability nor security [84].

  Many CAPTCHAs also adopt the use of visual clutter, e.g., this may be in the form of background textures, noise, or lines to connect characters together, in an attempt to impede automated attacks. This can be detri-

mental to the usability of the CAPTCHA, as humans may have difficulty distinguishing the text from the clutter. As such, the use of color and visual clutter must be carefully considered when designing a CAPTCHA. In general, if color or visual clutter does not contribute to the security of the CAPTCHA, the reason for its use may only be for aesthetic purposes.

– **Distortion:** Affine transformations and distortion of characters are commonly used techniques to impede character recognition using Optical Character Recognition (OCR) software [16]. Transformations alone, which include translation, clockwise/counterclockwise rotation and scaling, are easy for both computers and humans to solve. As such, these are typically combined with some level of distortion. Distortions are elastic deformations that can either be at the level of individual characters (i.e. local warping) or deformations to the overall text (i.e. global warping). In fact, the initial design of the popular "reCAPTCHA" was to deliberately use distorted strings that could not be recognized by OCR programs [79].

  While distortion is a mechanism employed to hinder automated attacks, it has severe implications on the usability of CAPTCHAs, because humans find it difficult to recognize distorted text. For this reason, CAPTCHA systems typically allow the users multiple attempts at solving challenges [84]. An additional consideration when dealing with distorted text from a usability point of view, is that the use of certain character combinations can be confusing. For example, digits and letters like the digit '0' and the letter 'O', the digit '1' and the letter 'l', are common causes of confusion. The same applies to upper and lower case pairs like 'S' and 's', as well as 'Z' and 'z'. Furthermore, certain character combinations like 'VV' can be misinterpreted as 'W', 'rn' as 'm', and so on [84].

2.2 Attacking CAPTCHAs

Since its inception, the security of CAPTCHAs has attracted much attention and has been examined by researchers and practitioners alike. This has resulted in the development of a variety of techniques for attacking and defeating CAPTCHAs. Many researchers have highlighted various flaws in the design of CAPTCHAs that can be exploited, resulting in them being vulnerable to automated attacks. This section presents some of the important work in the area of attacking text-based CAPTCHAs.

In early work on CAPTCHA security, researchers were able to successfully break the "EZ-Gimpy" and "Gimpy" CAPTCHAs. These were CAPTCHAs that were designed by a team at Carnegie Mellon University, and EZ-Gimpy was previously used by Yahoo in their online services to screen out bots [53]. The work by Mori and Malik [53] showed that CAPTCHAs based on language models are susceptible to attacks, because a database of words can be constructed to defeat the CAPTCHA. In their attack, they used a holistic approach of attempting to match the shape contexts of entire words extracted from a CAPTCHA to

a database of known objects. This was possible because the text in EZ-Gimpy and Gimpy were based on a set of English words. They explained that a holistic approach was adopted because in severe clutter, it was difficult to identify individual characters with occluded or ambiguous parts [53]. A pool of candidate words were identified and ranked based the matching of shape contexts, and the one with the best matching score was selected as the solution.

It was reported that this attack was able to break EZ-Gimpy 92% of the time and had a success rate of 33% against the Gimpy CAPTCHA. In addition, using font and lexicon information, the same attack could also be used against "PessimalPrint" [5] and "BaffleText" [20]. These are two other pioneering CAPTCHAs that were designed by the research community. In other early work, it was demonstrated that distortion estimation techniques could be used to break the EZ-Gimpy and Gimpy-r CAPTCHAs with high degrees of success [55]. Since distortion techniques are commonly used to deter character recognition, this research investigated techniques to estimate and remove distortion prior to the character recognition process.

In later work by Chellapilla et al. [17, 19], they were successful in using machine learning to break a variety of CAPTCHAs. This led them to proposing the well known and widely accepted *segmentation-resistant* principle for text-based CAPTCHA design.

Segmentation-Resistant. The segmentation-resistant principle is recognized as one of the foundational guiding principles for the development of text-based CAPTCHAs. In seminal work by a Microsoft research team, Chellapilla et al. [17, 19] demonstrated that machine learning algorithms were capable of successfully solving a variety of CAPTCHA schemes. In their study, they deliberately avoided the use of language models when attempting to break the CAPTCHAs. As such, their work showed that computers were able to outperform humans when it came to the task of recognizing single characters, which had some level of distortion and/or clutter.

This led to the establishment of the segmentation-resistant principle. The task of solving a text-based CAPTCHA consists of two main challenges. The first is *segmentation*; the identification and separation of text contained within a CAPTCHA into its constituting characters in the correct order. The second is *recognition*; this refers to the task of recognizing the individual characters. Since it was shown that computers are better at the recognition task, this implies that if a computer program can adequately segment a CAPTCHA into its constituting characters, the CAPTCHA is essentially broken. Therefore, it is widely accepted that one of the underlying requirements of a secure text-based CAPTCHA scheme is that it must be segmentation-resistant [1, 18].

The team subsequently adopted this principle in the design of a CAPTCHA that was deployed on a number of Microsoft's online services. Unfortunately, it was shown that the CAPTCHA could be segmented using a low-cost attack [83]. This attack showed that even though certain CAPTCHAs may be designed to be resistant against segmentation, segmentation may in fact be possible after

some pre-processing. Nevertheless, despite being able to defeat the CAPTCHA, the authors stated that their work did not negate the segmentation resistance principle [83]. In addition, their work also showed that string length is a factor that must be considered in the design of a text-based CAPTCHA. This is because unlike CAPTCHAs that contain a random number of characters, CAPTCHAs with fix length strings are easier to segment as the total number of characters is known beforehand.

Since the introduction of the segmentation-resistant principle, a variety of CAPTCHAs have been developed with this principle in mind. The mainstream methods of deterring segmentation can broadly be classified into three general categories [14, 61]. It should be noted that the use of these methods have significant implications on the usability of the resulting CAPTCHA. The three general categories are described as follows:

- **Background confusion**: The purpose of this approach is to attempt to prevent segmentation by making it difficult to distinguish between the text and the background. This can be done by using a complex background, by using very similar colors for the text and the background, by adding noise, or any combination of these. In a study by Bursztein et al. [14], they concluded that backgrounds should only be for cosmetic purposes as the use of background confusion as a security mechanism does not add to the robustness of a CAPTCHA. This is because for usability, a human must be able to differentiate between the text and the background.

- **Using lines**: This is an approach where random lines that cross multiple characters are used to deter segmentation. These may consist of small lines that cross some characters, or larger lines that cover the entire CAPTCHA [14]. The aim of this is to attempt to confuse segmentation algorithms by linking characters together.

- **Collapsing**: The idea behind this approach is to increase the difficulty of a segmentation task by joining or overlapping characters. This is typically achieved by removing the space between characters, tilting characters and/or overlapping them, which results in the characters being *crowded* together. This approach is considered to be the most effective mechanism for deterring segmentation [14, 83].

Segmentation Techniques. While a variety of CAPTCHAs have been designed with the segmentation-resistant principle in mind, many of them have been found to be susceptible to segmentation attacks. Over the years, researchers have shown that design flaws can be exploited to segment CAPTCHAs. Among others, the following are several segmentation techniques that have been documented by various researchers [60]:

- **De-noising algorithms**: The addition of noise is a commonly use technique to confuse segmentation algorithms. As such, de-noising techniques are used

to remove random noise from a CAPTCHA before a segmentation operation. Of the various de-noising techniques that have been proposed over the years, the Gibbs algorithm [34], which is also known as the Markov random field technique, has been found to be a very effective de-noising algorithm [14]. The way that this algorithm works is by computing an energy value for every pixel based on its surrounding pixels, and subsequently removing pixels with an energy value that is below a certain threshold. This process is then repeated until there are no more pixels to remove. Other de-noising algorithms include the dilate-erode approach, in which a CAPTCHA image is up-sampled, dilated then eroded. This results in noise, such as lines, being removed while retaining solid thick characters [16].

- **Histogram-based segmentation**: The idea behind this method is to project a CAPTCHA's pixels to their respective vertical, or in some cases diagonal, pixel positions [2, 82, 83]. By doing so, a histogram can be constructed based on the number of pixels at each position. In general, parts of the histogram that have higher pixel numbers are typically made up of pixels that belong to the characters of a CAPTCHA. On the other hand, parts that contain low pixel numbers usually represent the separation between characters or the end of a character, and are thus potential positions that can be used for segmentation. This method has been shown to be effective against CAPTCHAs in which their characters are only slightly joined, overlapping, or are connected using thin lines [42, 82, 83]. In addition, this method can be used as a step to separate groups of characters, before applying other segmentation techniques.

- **Color Filling Segmentation (CFS)**: This technique is akin to performing a flood fill algorithm and is often used in conjunction with other segmentation techniques. The initial step is to identify a pixel with a color that is associated with text in a CAPTCHA. Once identified, all the neighboring pixels which have the same color and are connected to the initial pixel are traced. This is repeated until all connected pixels are identified. The end result is that the connected pixels will reveal either individual characters or groups of connected characters [14, 32, 31, 83]. In the latter case, this technique will make it easier for other segmentation methods, since sections containing characters have been identified.

- **Opportunistic segmentation**: This approach seeks to exploit regular and predictable features of a CAPTCHA. By relying on educated guesses, potential locations to perform segmentation can be approximated based on prior knowledge of a CAPTCHA. Examples of such prior knowledge include CAPTCHAs that contain a fixed number of characters, where characters often appear at certain locations, and if the characters have roughly the same width. These factors make a CAPTCHA vulnerable to opportunistic segmentation, because it is easy to make educated guesses as to the likely positions for segmentation cuts [14]. Others have also exploited the fact that

in some poorly designed CAPTCHAs, each unique character always contains the same number of pixels. As such, pixel-counting attacks can be used to distinguish between characters [82].

– **Segmentation based on patterns and shapes**: This is a segmentation technique that attempts to identify patterns and shapes that can be used to characterize certain characters. For instance, characters like 'a', 'b', 'd', 'e', 'g', 'o', 'p', and 'q', all contain loops or circular regions, characters like 'i', 'j', and 'l', typically consist of small vertical blocks of pixels, and so on [2, 25, 72]. Once the characteristic patterns and shapes of the characters in a CAPTCHA have been determined, this information can be used to identify particular features in a CAPTCHA that can be exploited to facilitate the segmentation process.

The segmentation techniques described here represent some of the common methods that have been implemented to attack a variety of diverse CAPTCHAs. Many segmentation approaches use a combination and/or variations of the techniques described here. There are also other more specialized attacks and segmentation techniques that have been used to attack specific CAPTCHAs. A systematic study on a number of text-based CAPTCHAs found that solely focusing on being segmentation-resistant alone is not enough to guarantee that a CAPTCHA is secure, because there may be side-channel attacks that can be exploited to defeat a CAPTCHA [9]. In addition, other research has shown that image processing and pattern recognition tools, such as k-means clustering, digital image in-painting, character recognition based on cross-correlation, and so on, can be used to successfully break a variety of CAPTCHAs [47].

While much work on attacking CAPTCHAs has focused devising techniques for breaking individual schemes, many such solutions are only applicable to specific CAPTCHAs or schemes that possess certain exploitable features. With advances in areas such as machine learning, recent research efforts have aimed at developing more universal approaches that can be used to solve text-based CAPTCHAs in general.

Bursztein et al. [10] demonstrated a generic method of solving CAPTCHAs in a single step using machine learning to attack both the segmentation and recognition tasks at the same time. Many previous approaches performed these processes sequentially as separate steps. Their results showed that this approach was successful in defeating various real-world CAPTCHAs, and they concluded that combining the solving of segmentation and recognition problems is likely to be the way forward for automated CAPTCHA attacks [10]. In later work, Gao et al. [33] introduced a simpler and more computationally efficient approach of using Log-Gabor filters as a generic attack on text-based CAPTCHAs. Their experiments demonstrated that this generic method of attack was successful in defeating a wide-range of representative text-based CAPTCHAs.

With the availability of such universal attacks, it has been suggested that traditional text-based CAPTCHAs may be reaching the end of its usefulness in deterring automated programs. Nevertheless, this does not discount the effort

that has gone into the development and understanding of CAPTCHAs over the years, as it has taken a many years of research to achieve such generic attacks. It does however, require researchers and developers to reconsider how reverse Turing tests are to be designed in future, and whether there are any alternatives. This remains an open challenge [10, 33].

2.3 Design Variants

In efforts to overcome the limitations of traditional text-based CAPTCHAs, other design paradigms have been proposed. Examples of such paradigms include CAPTCHAs that are generated using 3D models, as well as CAPTCHA challenges that make use of animation. A number of these approaches are presented here.

3D-based CAPTCHAs. 3D approaches to CAPTCHA design typically involve the rendering of 3D models of text-objects, or of other objects, to an image [41, 49]. Since CAPTCHAs must capitalize on the difference in ability between humans and computers, the assumption in 3D CAPTCHAs is that it is difficult for computer programs to recognize 3D content, whereas this should be easy for humans. The reason for this is because 3D perception is an integral part of the human visual system. However, it has been demonstrated that 3D-based CAPTCHAs are by no means immune to attacks [61, 86].

As an example, a simple 3D CAPTCHA scheme was proposed based on the rendering of 3D text models [43]. However, it can clearly be seen that this approach is not secure against automated attacks, because the front face of characters are rendered under the same lighting conditions and thus appear to have the same shade. In addition, no distortion is applied to the characters. More importantly, this approach fails to take into account the fact that the field of 3D object recognition is a well studied discipline in computer vision.

Nguyen et al. [61] presented techniques that can be used to extract information from 3D text-based CAPTCHAs, and showed that this can be exploited to break a number of such CAPTCHAs. The 3D CAPTCHAs that were investigated in their study were generated by perturbing a regular pattern in a way where a human would recognize 3D text embedded within the pattern. Humans can easily use the resulting CAPTCHA due to the human cognitive ability to perceive 3D from depth cues in 2D images. In contrast, it was assumed that this approach would be difficult for computers. However, the study showed that information about the text could effectively be extracted from the 3D CAPTCHAs, which essentially reduced the CAPTCHA challenge to the task of character recognition.

A number of other 3D CAPTCHA approaches were discussed in Ross et al. [66], in which they pointed out scalability problems with certain 3D CAPTCHAs. This was due to the fact that the generation process involved in some schemes required a substantial amount of manual effort. In their work, they proposed a prototype CAPTCHA scheme that they called "Sketcha". This CAPTCHA is constructed from oriented line drawings of 3D models, where the challenge was

to correctly orient the images. Another idea for 3D CAPTCHAs was proposed using the concept of *emerging images* [50]. The underlying notion for this was to render abstract representations of 3D models in 3D environments, and rely on the human ability to perceive objects as a whole, rather than as individual parts, in order to perceive objects in the scene. In other work, a CAPTCHA that was built on the human ability of perceive 3D text from stereoscopic images was proposed [73].

Animated CAPTCHAs. Unlike traditional CAPTCHAs where the challenge is presented within a single image, animated CAPTCHAs attempt to incorporate a time dimension into the challenge. Several researchers and practitioners have developed animated CAPTCHAs by distributing the information that is required to solve the challenge over a number of animation frames. The assumption in animated CAPTCHAs is that humans can easily perceive information presented over multiple frames, whereas computers will have difficulty processing animated information. This has been dubbed the *zero knowledge per frame* principle, because the required information to solve the CAPTCHA is not completely contained within a single frame [26]. Since the information required to solve an animated CAPTCHA is distributed over a number of frames, this also means that the task of solving animated CAPTCHAs is typically more time consuming compared to single image CAPTCHAs.

Researchers have proposed various ideas for the design of animated CAPTCHAs. For example, a sketch of an animated CAPTCHA was proposed based on moving characters amidst a noisy background [26]. Another idea was to present distorted text on the face of a deforming animated surface [30]. In other work, an animated CAPTCHA based on visual phenomena was proposed. The notion for this approach is that by grouping different entities that move together, objects that are superimposed over a noisy background of the same color will be visible to humans when the objects are moving [57]. Other examples include an animated approach with moving objects where the user's task was to solve the challenge by identifying the correct object and its current location [3], and an animated CAPTCHA based on the concept of motion parallax [23]. In the motion parallax approach, the CAPTCHA was built on the idea that humans are able to perceive depth through motion. Therefore, by placing text at different depths in a 3D scene, humans should be able to perceive foreground characters and separate this from the background when the viewpoint is moving.

Despite the assumption that the addition of a time dimension in animated CAPTCHAs makes it difficult for computer programs to solve, techniques that can successful attack animated CAPTCHAs have been developed. Nguyen et al. [58, 59] demonstrated that even though the information required to solve animated CAPTCHAs is distributed over multiple frames, information from the frames can be extracted and collated to solve the CAPTCHA. An example of a method that was developed to defeat animated text-based CAPTCHAs is known as a *Pixel Delay Map*. This was devised based on the notion that characters required to solve an animated CAPTCHA are typically displayed a certain

locations for longer periods of time [58, 59]. The reason for this is to facilitate usability, because even though there may be constant changes in the animation frames, a human has to be given sufficient time to identify the characters. Hence, this can be used to extract characters from multiple animated frames and the extracted characters can subsequently be passed through a character recognition process.

Researchers have also developed techniques for defeating moving image object recognition CAPTCHAs [81]. Their approach was shown to be successful in defeating the "NuCaptcha", which was considered to be the state-of-the-art in animated CAPTCHA design. The challenge in this video based animated CAPTCHA was presented as overlapping characters that were moving in the animated frames. The CAPTCHA was built on the concept that the human mind is able to see the different parts that are moving together, and fill in the blanks to perceive the characters [23, 62]. The attack on moving object CAPTCHAs that was developed by Xu et al. [81], was based on the fact that the characters in most animated CAPTCHAs are rigid objects that do not deform over the multiple animation frames. As such, salient features could be identified and their positions tracked over different frames to extract and segment the characters. A related method was also concurrently developed by Bursztein [9].

Other Variants. A text-based CAPTCHA that was based on the recognition of characters and identifying their locations has previously been proposed [60]. The idea behind this was to display many characters, most of which were not relevant to the challenge, in the CAPTCHA image. These characters were organized in a two dimensional manner and overlapped both vertically and horizontally to deter segmentation. A user is presented with a string of text, and the user's task is to identify the location of each character from the string in the two dimensional arrangement of the text. This method supported a variety of non-keyboard interactions, such as drag-and-drop, mouse movement and clicking, and was thus designed to be suitable for different devices. The purpose was to cater for the modern day ubiquitous use of mobile devices and touch screens. Clickable CAPTCHAs [22] and drag-and-drop CAPTCHAs [15] have also been proposed by other researchers.

Microsoft developed a related two-layer CAPTCHA, in which text was positioned and overlapped in two rows instead of the traditional approach of only having a single line of text. However, this approach was defeated by Gao et al. [31], who demonstrated a method of separating the two rows of text and subsequently extracting the individual characters.

3 Image-based CAPTCHAs

Other than text-based CAPTCHAs, image-based CAPTCHAs are another category of visual CAPTCHAs that have been examined by various researchers and practitioners. In image-based CAPTCHAs, the challenge presented to the user typically involves an image recognition task. Compared with text recognition,

image recognition is seen as a much harder problem for computer programs to solve. This is because there is still a large domain of unsolved image perception and interpretation problems for computers [88]. Moreover, humans find images less bothersome [29].

Thus, this makes image-based challenges an attractive alternative to text-based CAPTCHAs as they can potentially be easy for humans, but difficult for computers. However, unlike text-based CAPTCHAs, which are easy to generate, one of the fundamental requirements of image-based CAPTCHAs is the need to obtain a source of images. Furthermore, there must also be a way to obtain the solution to an image recognition challenge, e.g., using pre-labeled images, to facilitate automated grading.

One of the first research works on image-based CAPTCHAs was presented in Chew and Tygar [21]. In their work, they introduced an image-based CAPTCHA that used labeled photographs that were sourced from Google Image Search [36]. While this was an innovative solution as new images are constantly added to the database, it was highlighted that the problem with this approach was that it is likely that the image labels may not accurately reflect the image content. The reason for this is because the method of labeling photo is based on descriptive text surrounding the image, which may not be accurate [29]. Therefore, this approach relied on manual effort to remove inaccurately labeled images from the CAPTCHA database. It was also pointed out that while the process of constructing an image database could be automated using an image classification program, an attacker could develop a similar classification program to defeat the CAPTCHA [29].

A solution to the problem of obtaining accurately labeled images was described by von Ahn and Dabbish [78]. This work involved getting humans to manually label images by presenting this as an interactive entertainment task in the form of a game called "ESP". The labeled images could be used to build a database for an image-based CAPTCHA, which was the basis of the "PIX" and "ESP-PIX" CAPTCHAs [76]. However, it has been argued that this approach suffered from a number of problems, including the problem of limited scalability and the subjective nature of abstract images made it potentially frustrating for humans [29, 88].

Elson et al. [29] developed an image-based CAPTCHA that they named "Asirra", in which the challenge presented to the user was to correctly distinguish between images of cats and dogs. This is an trivial task for humans, but purported to be difficult for computers. The Asirra CAPTCHA's solution to the problem of obtaining labeled images for its database, was to source images from Petfinder.com [63]. This is a website that promotes the adoption of pets, and many new accurately labeled photographs are added to its database every day. The security of Asirra relied on the large image database size and the difficulty of computers in accurately distinguishing between cats and dogs [29]. Unfortunately, it was shown that machine learning attacks could defeat Asirra. Golle [35] described a classifier that could accurately distinguish between images of

cats and dogs, and demonstrated its success in solving Asirra challenges at a high success rate.

In other work, an image-based CAPTCHA called "ARTiFACIAL" was proposed by Rui and Liu [67]. This CAPTCHA was based on the difference in ability between humans and computers at the task of recognizing human facial features. While humans can easily recognize faces, even in the presence of distortion or occlusion, this is difficult for computer programs especially under varying conditions, such as head orientations, face asymmetry, lighting and shading, and background clutter. To generate a challenge in ARTiFACIAL, a texture of a human face was mapped onto a 3D model of a head. This could then be used to generate images of a face under different conditions, e.g., global head transformation, local facial feature deformations, different illumination conditions and background clutter [67]. As a result, it allowed image challenges and solutions to be generated without manual effort, and was therefore easily scalable. The challenge presented to the user was to correctly identify various facial features. Goswami et al. [38] proposed a related image-based CAPTCHA that was also based on human face recognition. In their approach, a composite image containing distorted real and synthetic human faces amid a complex background was presented to the user. The user's task was to distinguish and select only the real human faces.

"IMAGINATION" is an image-based CAPTCHA generation system that was proposed by Datta et al. [27]. This idea behind this CAPTCHA system was to exploit human imagination through the interpretation of images amidst distortion/clutter. The challenge that was presented by the system consisted of two processes; a click and an annotate process. In the click process, the user was presented with a composite image which consisted of a set of 8 tiled images. The user had to click near the geometric center of the image that he/she wanted to annotate. The selected image underwent a controlled distortion, and the user was presented with a set of word choices. The user then had to select the appropriate word that described the distorted image [27].

Researchers have also proposed a CAPTCHA that was based on notion of using image orientation [37]. In this CAPTCHA, users are presented with images that have undergone some form of random rotation. The user's task is to correctly rotate the images to their upright orientation. This image orientation CAPTCHA attempts to take advantage of the gap between humans and computers in identifying the correct orientation of images. However, the images used in the database must be carefully selected as they must not contain any visual cues that can reveal the upright orientation, and at the same time images that are hard for humans to correctly orientate should also not be used. As such, a set of guidelines in conjunction with a social feedback mechanism was proposed to filter candidate images [37]. A related idea for image orientation CAPTCHA design that was based on the orientation of cropped sub-images was presented in Kim et al. [44].

Other researchers presented a video CAPTCHA, where the user's task was to provide three words describing the content in a video [45]. This video CAPTCHA

scheme was generated based on YouTube videos, which had labels/tags that were supplied by the person who uploaded the video. The user passed the challenge if any of the three words provided by the user matched the labels/tags that were supplied by the video uploader. The researchers developed this CAPTCHA to distinguish between humans and computers based on video understanding. Experiment results on the security and usability of their proposed scheme suggested that it was comparable with other visual CAPTCHAs [45].

A systematic study on a number of image-based CAPTCHAs was conducted by Zhu et al. [88]. This study evaluated and presented attacks against several such CAPTCHAs and described the reasons why they could be broken. Their work analyzed a number of image-based CAPTCHAs including Asirra, the video CAPTCHA, the image orientation CAPTCHA and others. These were evaluated in terms of whether manual effort was involved in CAPTCHA generation, ease of automated grading, whether it was easy for humans and computers to solve, its scalability, etc. In addition, they described methods for attacking the ARTiFACIAL and IMAGINATION CAPTCHAs. This led the researchers to proposing a number of guidelines for the design of robust image-based CAPTCHAs. These guidelines state that image recognition CAPTCHAs must rely on *semantic information*, multiple types of objects and prevent machine learning attacks by avoiding the possibility of exploiting a priori knowledge. Based on the lessons learned and their set of guidelines, they subsequently developed an image-based CAPTCHA called "Cortcha", which was designed to be scalable, did not require manual labeling and could use an unlimited number of object types [88]. In Cortcha, the user was presented with a selection of candidate objects and an inpainted image. The user's task was to choose a candidate object, drag it around and drop it at a position in the inpainted image where it would look natural and semantically meaningful.

While semantic information has been described as a requirement for robust image-based CAPTCHAs, Sivakorn et al. [69] developed an attack for breaking semantic image CAPTCHAs using deep learning. Their attack extracts semantic information from images using image annotation services and libraries, to solve a CAPTCHA challenge by identifying image content and selecting candidate images that depict similar objects. This approach was shown to be highly successful in automatically solving Google's image-based version of reCAPTCHA. Furthermore, it also achieved very high accuracy in attacking an image CAPTCHA used by Facebook. As a result of their study, they proposed several safeguards for reducing the accuracy of their automated attack [69]. Among others, these recommendations include introducing content homogeneity, employing more complicated semantic relations, and using adversarial images to reduce classification accuracy.

4 Audio CAPTCHAs

While text-based and image-based CAPTCHAs have received much attention and have been commonly deployed on many online services, these are visual

CAPTCHAs that cannot be used by people who are blind or visually impaired. As such, audio CAPTCHAs were introduced to provide an accessible alternative to visual CAPTCHAs. This category of CAPTCHA usually involves the user performing some form of speech recognition task in the midst of distorted speech [46]. However, despite the fact that audio CAPTCHAs are commonly used alongside visual CAPTCHAs, the security of this type of CAPTCHA is often overlooked [12].

Studies have shown that audio CAPTCHA are more difficult to use and more time consuming, when compared with their visual counterparts for both blind and sighted users [7, 13]. Part of the reason for this is because visual CAPTCHAs are perceived as a whole even when focus is on the answer box, whereas audio playback is linear and reviewing an audio CAPTCHA requires users to replay it from the beginning before focusing on the answer box. An optimized interface for addressing the usability of audio CAPTCHAs was presented in Bigham and Cavender [7].

In the work conducted by Soupionis and Gritzalis [71], they presented an overview of several existing audio CAPTCHA implementations at the time. These audio CAPTCHAs were evaluated in terms of their different characteristics, which included attributes such as duration, vocabulary, production procedure and noise. Their work was aimed at developing an audio CAPTCHA that was suitable for use in Voice over Internet Protocol (VoIP) applications. To this end, they suggested a set of required VoIP CAPTCHA attributes and presented an implementation of the proposed audio CAPTCHA. However, the security of the proposed CAPTCHA was not throughly examined.

To deter automated speech recognition attacks by computers, distortion and background noise is typically added to an audio CAPTCHA. Background noise like running water has different characteristics from the human voice and can easily be distinguished, whereas noise in the form of background human voices from multiple speakers is more difficult to solve. Similar to methods for breaking text-based CAPTCHAs, Tam et al. [74] showed how machine learning techniques can also be used to solve audio CAPTCHAs. In their approach, a heuristic was developed based on the use of energy peaks to segment audio for a training set. They showed that this approach was able to successfully defeat three different audio CAPTCHAs.

In another study on the security of audio CAPTCHAs, Bursztein and Bethard [12] demonstrated that while off-the-shelf speech recognition programs may perform poorly when it comes to attacking audio CAPTCHAs, it is possible to develop an automated program for solving them. They developed a tool called "Decaptcha" that used supervised learning to recognize speech patterns. Decaptcha looks at voice energy spikes by isolating such spikes in a wave file using a discreet Fourier transform [12]. It was shown that this approach of analyzing energy spikes was able to break an audio CAPTCHA at a high success rate. The work on Decaptcha was extended in Bursztein at el. [11] and used to successfully defeat a number of audio CAPTCHAs, including those used by Digg, eBay, Microsoft, reCAPTCHA and Yahoo. These CAPTCHAs were described

as non-continuous audio CAPTCHAs, as they consisted of a sequence of spoken letters and/or digits that were distorted using various types of noise, such as white noise, Gaussian noise, echos, semantic noises and so on [11]. It was reported that Decaptcha had lower performance when it came to semantic noises that were used in reCAPTCHA, and at the same time this type of noise was the least harmful to human understanding. As such, it was recommended that future audio CAPTCHAs should investigate the use of semantic noise.

Different methods of attacking audio versions of reCAPTCHA were presented in a number of other studies [8, 68, 70]. Sano et al. [68] described the fact that previous attacks on audio CAPTCHAs were aimed at solving non-continuous audio CAPTCHAs. These methods of attack consisted of two phases, namely, a segmentation stage and a classification stage [11, 74]. However, the security mechanism in continuous audio CAPTCHAs relies on the difficulty in performing accurate segmentation. They described a method of using Hidden Markov models to construct a speech recognition solver to attack a version of reCAPTCHA that used continuous audio. To improve the security of audio CAPTCHAs, they suggested that increasing the number of voices and adopting semantic noise can potentially achieve this. Bock et al. [8] on the other hand presented a low-resource method of attack that used free, non-specialized and publicly available speech recognition services to successfully defeat an audio version of reCAPTCHA. In addition, Solanki et al. [70] also demonstrated low-cost attacks against seven major audio CAPTCHAs using off-the-shelf speech recognition services.

5 Other Designs and Issues

Although much of the research on CAPTCHAs has focused on their security in deterring automated attacks by computers, attackers can bypass the security of CAPTCHAs by using *relay attacks*. This may be in the form of CAPTCHA smuggling, in which an attacker redirects the task of solving a CAPTCHA to an unsuspecting victim, or by creating a paid CAPTCHA solving service where human labor is employed to solve CAPTCHA challenges. Egele et al. [28] described CAPTCHA smuggling and discussed the feasibility of creating CAPTCHA farms for implementing such man-in-the-middle attacks. In other research, Motoyama et al. [54] examined the economics of behind paid CAPTCHA solving services. Either way, relay attacks bypass the protection offered by CAPTCHAs and side steps the need for developing automated programs for solving CAPTCHAs by using humans to solve the challenges.

The notion of Game CAPTCHAs is seen as an approach for potentially making the task of solving CAPTCHAs more fun for users. While there are different types of game CAPTCHAs, a number of studies have looked at a category of such CAPTCHAs known as Dynamic Cognitive Game (DCG) CAPTCHAs [51, 52]. DCG CAPTCHAs present a challenge to the user in the form of a simple game that requires the user to perform cognitive tasks, which are typically accomplished by interacting with a series of dynamic images. For example, it might take the form of moving objects within an images, where the user's task is to

drag-and-drop the objects to match specific targets. Despite the fact that such CAPTCHAs will generally take a longer time to solve, users may potentially find the game-like activities enjoyable.

For accessibility, visual CAPTCHAs are often used alongside audio CAPTCHAs. It has been suggested that audio CAPTCHAs might be weaker than visual CAPTCHAs [11]. This is possibly due to the human visual system occupying a much larger section of the human brain when compared to the human audio processing system. Furthermore, with advances in modern signal processing and machine learning, the difference in audio capabilities between humans and computers is possibly less significant than when compared to the difference in visual processing capabilities [11]. This implies that an attacker can exploit this by attacking the security of the weaker alternative. In addition, attackers can also exploit other CAPTCHA design and implementation flaws. Examples of these were explored in side-channel attacks on a Math CAPTCHA, in which solutions were not uniformly distributed, as well as a gender classification CAPTCHA [39, 40].

In related work, CAPTCHAs have been proposed for other purposes including as graphical passwords [87] and for bot prevention in multiplayer online games [24]. Researchers have also proposed a CAPTCHA inspired photograph-based *social authentication* approach to verifying a user's identity. Yardi et al. [85] presented system called "Lineup" where the task presented to users was to identify people in photos whom they know. This system relied on the social network graph in Facebook to build its photo authentication framework. However, Polakis et al. [65] demonstrated a technique of defeating such social authentication using face recognition software. They later proposed a more robust social authentication approach, where challenges were generated by selecting photos that failed face recognition software and transforming those photos to deter image matching techniques [64].

6 Conclusion

Over the years, much research has gone into developing a greater understanding of CAPTCHAs. Numerous CAPTCHA schemes, which include a variety of visual and audio CAPTCHAs, have been proposed to date, and various studies have examined these diverse schemes in terms of their human usability, robustness against automated attacks, semantic information, and so on. With advances in technology and the various techniques that have been developed for attacking CAPTCHAs, researchers nowadays have greater insight into the field of CAPTCHA design and security, and must consider different strategies for developing CAPTCHAs. As such, even though it has been many years since the notion of CAPTCHAs was first introduced, the question as to whether it is possible to design a CAPTCHA that is easy for humans but difficult for computers, still remains an open challenge.

References

1. A. S. E. Ahmad, J. Yan, and L. Marshall. The robustness of a new CAPTCHA. In M. Costa and E. Kirda, editors, *Proceedings of the Third European Workshop on System Security, EUROSEC 2010, Paris, France, April 13, 2010*, pages 36–41. ACM, 2010.
2. A. S. E. Ahmad, J. Yan, and M. Tayara. The robustness of Google CAPTCHAs. *University of Newcastle, UK, Technical Report*, 1278:1–15, 2011.
3. E. Athanasopoulos and S. Antonatos. Enhanced CAPTCHAs: Using animation to tell humans and computers apart. In H. Leitold and E. P. Markatos, editors, *Communications and Multimedia Security, 10th IFIP TC-6 TC-11 International Conference, CMS 2006, Heraklion, Crete, Greece, October 19-21, 2006, Proceedings*, volume 4237 of *Lecture Notes in Computer Science*, pages 97–108. Springer, 2006.
4. P. Baecher, N. Büscher, M. Fischlin, and B. Milde. Breaking recaptcha: A holistic approach via shape recognition. In J. Camenisch, S. Fischer-Hübner, Y. Murayama, A. Portmann, and C. Rieder, editors, *Future Challenges in Security and Privacy for Academia and Industry - 26th IFIP TC 11 International Information Security Conference, SEC 2011, Lucerne, Switzerland, June 7-9, 2011. Proceedings*, volume 354 of *IFIP Advances in Information and Communication Technology*, pages 56–67. Springer, 2011.
5. H. S. Baird, A. L. Coates, and R. J. Fateman. PessimalPrint: a reverse Turing test. *International Journal on Document Analysis and Recognition*, 5(2-3):158–163, 2003.
6. H. S. Baird and K. Popat. Human interactive proofs and document image analysis. In D. P. Lopresti, J. Hu, and R. S. Kashi, editors, *Document Analysis Systems V, 5th International Workshop, DAS 2002, Princeton, NJ, USA, August 19-21, 2002, Proceedings*, volume 2423 of *Lecture Notes in Computer Science*, pages 507–518. Springer, 2002.
7. J. P. Bigham and A. Cavender. Evaluating existing audio CAPTCHAs and an interface optimized for non-visual use. In D. R. O. Jr., R. B. Arthur, K. Hinckley, M. R. Morris, S. E. Hudson, and S. Greenberg, editors, *Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI 2009, Boston, MA, USA, April 4-9, 2009*, pages 1829–1838. ACM, 2009.
8. K. Bock, D. Patel, G. Hughey, and D. Levin. unCaptcha: A low-resource defeat of reCaptcha's audio challenge. In W. Enck and C. Mulliner, editors, *11th USENIX Workshop on Offensive Technologies, WOOT 2017, Vancouver, BC, Canada, August 14-15, 2017*. USENIX Association, 2017.
9. E. Bursztein. *How we broke the nucaptcha video scheme and what we propose to fix it.* https://www.elie.net/blog/security/how-we-broke-the-nucaptcha-video-scheme-and-what-we-propose-to-fix-it.
10. E. Bursztein, J. Aigrain, A. Moscicki, and J. C. Mitchell. The end is nigh: Generic solving of text-based captchas. In S. Bratus and F. F. X. Lindner, editors, *8th USENIX Workshop on Offensive Technologies, WOOT '14, San Diego, CA, USA, August 19, 2014*. USENIX Association, 2014.
11. E. Bursztein, R. Beauxis, H. S. Paskov, D. Perito, C. Fabry, and J. C. Mitchell. The failure of noise-based non-continuous audio captchas. In *32nd IEEE Symposium on Security and Privacy, S&P 2011, 22-25 May 2011, Berkeley, California, USA*, pages 19–31. IEEE Computer Society, 2011.

12. E. Bursztein and S. Bethard. Decaptcha: Breaking 75% of eBay audio CAPTCHAs. In *Proceedings of the 3rd USENIX Conference on Offensive Technologies*, WOOT'09, pages 8–8, Berkeley, CA, USA, 2009. USENIX Association.
13. E. Bursztein, S. Bethard, C. Fabry, J. C. Mitchell, and D. Jurafsky. How good are humans at solving captchas? A large scale evaluation. In *31st IEEE Symposium on Security and Privacy, S&P 2010, 16-19 May 2010, Berleley/Oakland, California, USA*, pages 399–413. IEEE Computer Society, 2010.
14. E. Bursztein, M. Martin, and J. C. Mitchell. Text-based CAPTCHA strengths and weaknesses. In Y. Chen, G. Danezis, and V. Shmatikov, editors, *Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS 2011, Chicago, Illinois, USA, October 17-21, 2011*, pages 125–138. ACM, 2011.
15. S. K. Chaudhari, A. R. Deshpande, S. B. Bendale, and R. V. Kotian. 3D drag-n-drop CAPTCHA enhanced security through CAPTCHA. In *Proceedings of the International Conference & Workshop on Emerging Trends in Technology*, ICWET '11, pages 598–601, New York, NY, USA, 2011. ACM.
16. K. Chellapilla, K. Larson, P. Y. Simard, and M. Czerwinski. Building segmentation based human-friendly human interaction proofs (HIPs). In H. S. Baird and D. P. Lopresti, editors, *Human Interactive Proofs, Second International Workshop, HIP 2005, Bethlehem, PA, USA, May 19-20, 2005, Proceedings*, volume 3517 of *Lecture Notes in Computer Science*, pages 1–26. Springer, 2005.
17. K. Chellapilla, K. Larson, P. Y. Simard, and M. Czerwinski. Computers beat humans at single character recognition in reading based human interaction proofs (HIPs). In *CEAS 2005 - Second Conference on Email and Anti-Spam, July 21-22, 2005, Stanford University, California, USA*, 2005.
18. K. Chellapilla, K. Larson, P. Y. Simard, and M. Czerwinski. Designing human friendly human interaction proofs (HIPs). In G. C. van der Veer and C. Gale, editors, *Proceedings of the 2005 Conference on Human Factors in Computing Systems, CHI 2005, Portland, Oregon, USA, April 2-7, 2005*, pages 711–720. ACM, 2005.
19. K. Chellapilla and P. Y. Simard. Using machine learning to break visual human interaction proofs (HIPs). In *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada]*, pages 265–272, 2004.
20. M. Chew and H. S. Baird. BaffleText: a human interactive proof. In T. Kanungo, E. H. B. Smith, J. Hu, and P. B. Kantor, editors, *Document Recognition and Retrieval X, Santa Clara, California, USA, January 22-23, 2003, Proceedings*, volume 5010 of *SPIE Proceedings*, pages 305–316. SPIE, 2003.
21. M. Chew and J. D. Tygar. Image recognition CAPTCHAs. In K. Zhang and Y. Zheng, editors, *Information Security, 7th International Conference, ISC 2004, Palo Alto, CA, USA, September 27-29, 2004, Proceedings*, volume 3225 of *Lecture Notes in Computer Science*, pages 268–279. Springer, 2004.
22. R. Chow, P. Golle, M. Jakobsson, L. Wang, and X. Wang. Making CAPTCHAs clickable. In M. Spasojevic and M. D. Corner, editors, *Proceedings of the 9th Workshop on Mobile Computing Systems and Applications, HotMobile 2008, Napa Valley, California, USA, February 25-26, 2008*, pages 91–94. ACM, 2008.
23. Y. Chow and W. Susilo. AniCAP: An animated 3D CAPTCHA scheme based on motion parallax. In D. Lin, G. Tsudik, and X. Wang, editors, *Cryptology and Network Security - 10th International Conference, CANS 2011, Sanya, China, December 10-12, 2011. Proceedings*, volume 7092 of *Lecture Notes in Computer Science*, pages 255–271. Springer, 2011.

24. Y. Chow, W. Susilo, and H. Zhou. CAPTCHA challenges for massively multiplayer online games: Mini-game CAPTCHAs. In A. Sourin and O. Sourina, editors, *2010 International Conference on CyberWorlds, Singapore, October 20-22, 2010*, pages 254–261. IEEE Computer Society, 2010.
25. C. Cruz-Perez, O. Starostenko, F. Uceda-Ponga, V. A. Aquino, and L. Reyes-Cabrera. Breaking reCAPTCHAs with unpredictable collapse: Heuristic character segmentation and recognition. In J. A. Carrasco-Ochoa, J. F. M. Trinidad, J. A. Olvera-López, and K. L. Boyer, editors, *Pattern Recognition - 4th Mexican Conference, MCPR 2012, Huatulco, Mexico, June 27-30, 2012. Proceedings*, volume 7329 of *Lecture Notes in Computer Science*, pages 155–165. Springer, 2012.
26. J. S. Cui, J. T. Mei, W. Z. Zhang, X. Wang, and D. Zhang. A CAPTCHA implementation based on moving objects recognition problem. In *2010 International Conference on E-Business and E-Government*, pages 1277–1280, May 2010.
27. R. Datta, J. Li, and J. Z. Wang. IMAGINATION: a robust image-based CAPTCHA generation system. In H. Zhang, T. Chua, R. Steinmetz, M. S. Kankanhalli, and L. Wilcox, editors, *Proceedings of the 13th ACM International Conference on Multimedia, Singapore, November 6-11, 2005*, pages 331–334. ACM, 2005.
28. M. Egele, L. Bilge, E. Kirda, and C. Kruegel. CAPTCHA smuggling: hijacking web browsing sessions to create CAPTCHA farms. In S. Y. Shin, S. Ossowski, M. Schumacher, M. J. Palakal, and C. Hung, editors, *Proceedings of the 2010 ACM Symposium on Applied Computing (SAC), Sierre, Switzerland, March 22-26, 2010*, pages 1865–1870. ACM, 2010.
29. J. Elson, J. R. Douceur, J. Howell, and J. Saul. Asirra: a CAPTCHA that exploits interest-aligned manual image categorization. In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28-31, 2007*, pages 366–374. ACM, 2007.
30. I. Fischer and T. Herfet. Visual CAPTCHAs for document authentication. In *8th IEEE International Workshop on Multimedia Signal Processing (MMSP 2006)*, pages 471–474, 2006.
31. H. Gao, M. Tang, Y. Liu, P. Zhang, and X. Liu. Research on the security of Microsoft's two-layer captcha. *IEEE Trans. Information Forensics and Security*, 12(7):1671–1685, 2017.
32. H. Gao, W. Wang, J. Qi, X. Wang, X. Liu, and J. Yan. The robustness of hollow captchas. In A. Sadeghi, V. D. Gligor, and M. Yung, editors, *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*, pages 1075–1086. ACM, 2013.
33. H. Gao, J. Yan, F. Cao, Z. Zhang, L. Lei, M. Tang, P. Zhang, X. Zhou, X. Wang, and J. Li. A simple generic attack on text captchas. In *23nd Annual Network and Distributed System Security Symposium, NDSS 2016, San Diego, California, USA, February 21-24, 2016*. The Internet Society, 2016.
34. S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):721–741, 1984.
35. P. Golle. Machine learning attacks against the Asirra CAPTCHA. In P. Ning, P. F. Syverson, and S. Jha, editors, *Proceedings of the 2008 ACM Conference on Computer and Communications Security, CCS 2008, Alexandria, Virginia, USA, October 27-31, 2008*, pages 535–542. ACM, 2008.
36. Google Inc. *Google Image Search*. https://images.google.com/.

37. R. Gossweiler, M. Kamvar, and S. Baluja. What's up CAPTCHA?: a CAPTCHA based on image orientation. In J. Quemada, G. León, Y. S. Maarek, and W. Nejdl, editors, *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, pages 841–850. ACM, 2009.
38. G. Goswami, B. M. Powell, M. Vatsa, R. Singh, and A. Noore. FaceDCAPTCHA: Face detection based color image CAPTCHA. *Future Generation Computer Systems*, 31:59–68, 2014.
39. C. J. Hernández-Castro, M. D. R.-Moreno, D. F. Barrero, and S. Gibson. Using machine learning to identify common flaws in CAPTCHA design: FunCAPTCHA case analysis. *Computers & Security*, 70:744–756, 2017.
40. C. J. Hernández-Castro and A. Ribagorda. Pitfalls in CAPTCHA design and implementation: The Math CAPTCHA, a case study. *Computers & Security*, 29(1):141–157, 2010.
41. M. E. Hoque, D. J. Russomanno, and M. Yeasin. 2D captchas from 3D models. In *Proceedings of the IEEE SoutheastCon 2006*, pages 165–170, March 2006.
42. S. Huang, Y. Lee, G. Bell, and Z. Ou. An efficient segmentation algorithm for CAPTCHAs with line cluttering and character warping. *Multimedia Tools Appl.*, 48(2):267–289, 2010.
43. M. Imsamai and S. Phimoltares. 3D CAPTCHA: A next generation of the CAPTCHA. In *Proceedings of the International Conference on Information Science and Applications (ICISA 2010), Seoul, South Korea, 21-23 April, 2010*, pages 1–8. IEEE Computer Society, 2010.
44. J. Kim, W. Chung, and H. Cho. A new image-based CAPTCHA using the orientation of the polygonally cropped sub-images. *The Visual Computer*, 26(6-8):1135–1143, 2010.
45. K. A. Kluever and R. Zanibbi. Balancing usability and security in a video CAPTCHA. In L. F. Cranor, editor, *Proceedings of the 5th Symposium on Usable Privacy and Security, SOUPS 2009, Mountain View, California, USA, July 15-17, 2009*, ACM International Conference Proceeding Series. ACM, 2009.
46. G. Kochanski, D. P. Lopresti, and C. Shih. A reverse turing test using speech. In J. H. L. Hansen and B. L. Pellom, editors, *7th International Conference on Spoken Language Processing, ICSLP2002 - INTERSPEECH 2002, Denver, Colorado, USA, September 16-20, 2002*. ISCA, 2002.
47. S. Li, S. A. H. Shah, M. A. U. Khan, S. A. Khayam, A. Sadeghi, and R. Schmitz. Breaking e-banking captchas. In C. Gates, M. Franz, and J. P. McDermott, editors, *Twenty-Sixth Annual Computer Security Applications Conference, ACSAC 2010, Austin, Texas, USA, 6-10 December 2010*, pages 171–180. ACM, 2010.
48. M. Lillibridge, M. Abadi, K. Bharat, and A. Broder. Method for selectively restricting access to computer systems, Feb. 27 2001. US Patent 6,195,698.
49. C. R. Macias and E. Izquierdo. Visual word-based captcha using 3d characters. In *3rd International Conference on Imaging for Crime Detection and Prevention (ICDP 2009)*, pages 1–5, Dec 2009.
50. N. J. Mitra, H. Chu, T. Lee, L. Wolf, H. Yeshurun, and D. Cohen-Or. Emerging images. *ACM Trans. Graph.*, 28(5):163:1–163:8, 2009.
51. M. Mohamed, S. Gao, N. Sachdeva, N. Saxena, C. Zhang, P. Kumaraguru, and P. C. van Oorschot. On the security and usability of dynamic cognitive game CAPTCHAs. *Journal of Computer Security*, 25(3):205–230, 2017.
52. M. Mohamed, N. Sachdeva, M. Georgescu, S. Gao, N. Saxena, C. Zhang, P. Kumaraguru, P. C. van Oorschot, and W. Chen. A three-way investigation of a game-captcha: automated attacks, relay attacks and usability. In S. Moriai, T. Jaeger,

and K. Sakurai, editors, *9th ACM Symposium on Information, Computer and Communications Security, ASIA CCS '14, Kyoto, Japan - June 03 - 06, 2014*, pages 195–206. ACM, 2014.
53. G. Mori and J. Malik. Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), 16-22 June 2003, Madison, WI, USA*, pages 134–144. IEEE Computer Society, 2003.
54. M. Motoyama, K. Levchenko, C. Kanich, D. McCoy, G. M. Voelker, and S. Savage. Re: CAPTCHAs-understanding CAPTCHA-solving services in an economic context. In *19th USENIX Security Symposium, Washington, DC, USA, August 11-13, 2010, Proceedings*, pages 435–462. USENIX Association, 2010.
55. G. Moy, N. Jones, C. Harkless, and R. Potter. Distortion estimation techniques in solving visual captchas. In *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), with CD-ROM, 27 June - 2 July 2004, Washington, DC, USA*, pages 23–28. IEEE Computer Society, 2004.
56. M. Naor. *Verification of a Human in the Loop or Identification via the Turing Test*, 1996. http://www.wisdom.weizmann.ac.il/∼naor/PAPERS/human.pdf.
57. A. B. Naumann, T. Franke, and C. Bauckhage. Investigating CAPTCHAs based on visual phenomena. In T. Gross, J. Gulliksen, P. Kotzé, L. Oestreicher, P. A. Palanque, R. O. Prates, and M. Winckler, editors, *Human-Computer Interaction - INTERACT 2009, 12th IFIP TC 13 International Conference, Uppsala, Sweden, August 24-28, 2009, Proceedings, Part II*, volume 5727 of *Lecture Notes in Computer Science*, pages 745–748. Springer, 2009.
58. V. D. Nguyen, Y. Chow, and W. Susilo. Attacking animated CAPTCHAs via character extraction. In J. Pieprzyk, A. Sadeghi, and M. Manulis, editors, *Cryptology and Network Security, 11th International Conference, CANS 2012, Darmstadt, Germany, December 12-14, 2012. Proceedings*, volume 7712, pages 98–113. Springer, 2012.
59. V. D. Nguyen, Y. Chow, and W. Susilo. Breaking an animated CAPTCHA scheme. In F. Bao, P. Samarati, and J. Zhou, editors, *Applied Cryptography and Network Security - 10th International Conference, ACNS 2012, Singapore, June 26-29, 2012. Proceedings*, volume 7341 of *Lecture Notes in Computer Science*, pages 12–29. Springer, 2012.
60. V. D. Nguyen, Y. Chow, and W. Susilo. A CAPTCHA scheme based on the identification of character locations. In X. Huang and J. Zhou, editors, *Information Security Practice and Experience - 10th International Conference, ISPEC 2014, Fuzhou, China, May 5-8, 2014. Proceedings*, volume 8434 of *Lecture Notes in Computer Science*, pages 60–74. Springer, 2014.
61. V. D. Nguyen, Y. Chow, and W. Susilo. On the security of text-based 3D CAPTCHAs. *Computers & Security*, 45:84–99, 2014.
62. NuCaptcha Inc. *NuCaptcha*. http://www.nucaptcha.com/.
63. Petfinder. *Petfinder*. https://www.petfinder.com/.
64. I. Polakis, P. Ilia, F. Maggi, M. Lancini, G. Kontaxis, S. Zanero, S. Ioannidis, and A. D. Keromytis. Faces in the distorting mirror: Revisiting photo-based social authentication. In G. Ahn, M. Yung, and N. Li, editors, *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014*, pages 501–512. ACM, 2014.
65. I. Polakis, M. Lancini, G. Kontaxis, F. Maggi, S. Ioannidis, A. D. Keromytis, and S. Zanero. All your face are belong to us: breaking Facebook's social authentication. In R. H. Zakon, editor, *28th Annual Computer Security Applications Conference, ACSAC 2012, Orlando, FL, USA, 3-7 December 2012*, pages 399–408. ACM, 2012.

66. S. A. Ross, J. A. Halderman, and A. Finkelstein. Sketcha: a captcha based on line drawings of 3D models. In M. Rappa, P. Jones, J. Freire, and S. Chakrabarti, editors, *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 821–830. ACM, 2010.
67. Y. Rui and Z. Liu. ARTiFACIAL: Automated reverse Turing test using FACIAL features. *Multimedia Syst.*, 9(6):493–502, 2004.
68. S. Sano, T. Otsuka, K. Itoyama, and H. G. Okuno. HMM-based attacks on Google's ReCAPTCHA with continuous visual and audio symbols. *JIP*, 23(6):814–826, 2015.
69. S. Sivakorn, I. Polakis, and A. D. Keromytis. I am robot: (deep) learning to break semantic image CAPTCHAs. In *IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, March 21-24, 2016*, pages 388–403. IEEE, 2016.
70. S. Solanki, G. Krishnan, V. Sampath, and J. Polakis. In (cyber)space bots can hear you speak: Breaking audio CAPTCHAs using OTS speech recognition. In B. M. Thuraisingham, B. Biggio, D. M. Freeman, B. Miller, and A. Sinha, editors, *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017*, pages 69–80. ACM, 2017.
71. Y. Soupionis and D. Gritzalis. Audio CAPTCHA: Existing solutions assessment and a new implementation for VoIP telephony. *Computers & Security*, 29(5):603–618, July 2010.
72. O. Starostenko, C. Cruz-Perez, F. Uceda-Ponga, and V. A. Aquino. Breaking text-based captchas with variable word and character orientation. *Pattern Recognition*, 48(4):1101–1112, 2015.
73. W. Susilo, Y. Chow, and H. Zhou. STE3D-CAP: stereoscopic 3D CAPTCHA. In S. Heng, R. N. Wright, and B. Goi, editors, *Cryptology and Network Security - 9th International Conference, CANS 2010, Kuala Lumpur, Malaysia, December 12-14, 2010. Proceedings*, volume 6467 of *Lecture Notes in Computer Science*, pages 221–240. Springer, 2010.
74. J. Tam, J. Simsa, S. Hyde, and L. von Ahn. Breaking audio CAPTCHAs. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 1625–1632. Curran Associates, Inc., 2008.
75. A. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.
76. C. M. University. *The official CAPTCHA site.* https://www.captcha.net/.
77. L. von Ahn, M. Blum, N. J. Hopper, and J. Langford. CAPTCHA: using hard AI problems for security. In E. Biham, editor, *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, volume 2656 of *Lecture Notes in Computer Science*, pages 294–311. Springer, 2003.
78. L. von Ahn and L. Dabbish. Labeling images with a computer game. In E. Dykstra-Erickson and M. Tscheligi, editors, *Proceedings of the 2004 Conference on Human Factors in Computing Systems, CHI 2004, Vienna, Austria, April 24 - 29, 2004*, pages 319–326. ACM, 2004.
79. L. von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum. re-CAPTCHA: Human-based character recognition via web security measures. *Science*, 321(5895):1465–1468, 2008.

80. S.-Y. Wang, H. S. Baird, and J. L. Bentley. Captcha challenge tradeoffs: Familiarity of strings versus degradation of images. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 164–167, 2006.
81. Y. Xu, G. Reynaga, S. Chiasson, J. Frahm, F. Monrose, and P. C. van Oorschot. Security analysis and related usability of motion-based captchas: Decoding codewords in motion. *IEEE Trans. Dependable Sec. Comput.*, 11(5):480–493, 2014.
82. J. Yan and A. S. E. Ahmad. Breaking visual captchas with naive pattern recognition algorithms. In *23rd Annual Computer Security Applications Conference (ACSAC 2007), December 10-14, 2007, Miami Beach, Florida, USA*, pages 279–291. IEEE Computer Society, 2007.
83. J. Yan and A. S. E. Ahmad. A low-cost attack on a Microsoft captcha. In P. Ning, P. F. Syverson, and S. Jha, editors, *Proceedings of the 2008 ACM Conference on Computer and Communications Security, CCS 2008, Alexandria, Virginia, USA, October 27-31, 2008*, pages 543–554. ACM, 2008.
84. J. Yan and A. S. E. Ahmad. Usability of captchas or usability issues in CAPTCHA design. In L. F. Cranor, editor, *Proceedings of the 4th Symposium on Usable Privacy and Security, SOUPS 2008, Pittsburgh, Pennsylvania, USA, July 23-25, 2008*, ACM International Conference Proceeding Series, pages 44–52. ACM, 2008.
85. S. Yardi, N. Feamster, and A. Bruckman. Photo-based authentication using social networks. In C. Faloutsos, T. Karagiannis, and P. Rodriguez, editors, *Proceedings of the first Workshop on Online Social Networks, WOSN 2008, Seattle, WA, USA, August 17-22, 2008*, pages 55–60. ACM, 2008.
86. Q. Ye, Y. Chen, and B. Zhu. The robustness of a new 3D CAPTCHA. In J. Ramel, M. Liwicki, J. Ogier, K. Kise, and R. Smith, editors, *11th IAPR International Workshop on Document Analysis Systems, DAS 2014, Tours, France, April 7-10, 2014*, pages 319–323. IEEE Computer Society, 2014.
87. B. B. Zhu, J. Yan, G. Bao, M. Yang, and N. Xu. Captcha as graphical passwords - A new security primitive based on hard AI problems. *IEEE Trans. Information Forensics and Security*, 9(6):891–904, 2014.
88. B. B. Zhu, J. Yan, Q. Li, C. Yang, J. Liu, N. Xu, M. Yi, and K. Cai. Attacks and design of image recognition CAPTCHAs. In E. Al-Shaer, A. D. Keromytis, and V. Shmatikov, editors, *Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, Chicago, Illinois, USA, October 4-8, 2010*, pages 187–200. ACM, 2010.