SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2015; **8**:2135–2156
Published online 30 October 2014 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.1157

REVIEW ARTICLE

CAPTCHA and its Alternatives: A Review

Mohammad Moradi[1]* and MohammadReza Keyvanpour[2]

[1] Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran
[2] Department of Computer Engineering, Alzahra University, Vanak, Tehran, Iran

ABSTRACT

Nowadays, because of the undeniable impact of the Internet on all aspects of human life, security preserving has received more attention. To reach an acceptable level of security, Completely Automatic Public Turing test to tell Computer and Human Apart or simply CAPTCHA as a security preserving tool has been tailored for situations that need to prevent bots from doing a specific action; for example, signing up and downloading. Simultaneously, it should be also designed in such a way to allow humans to perform the same action. Despite its advantageous applications, there are several important issues such as security, usability, and accessibility that make its use controversial. In this paper, attempts were made to do a comprehensive review on various aspects and state-of-the-art of CAPTCHA in general and its alternatives in particular to help researchers easily focus on specific issues for the sake of proposing new solutions and ideas. Regarding the advancement in CAPTCHA development, new classifications were proposed to categorize different variations of CAPTCHAs and their problems and then compare them. Moreover, different types of CAPTCHAs' alternatives were classified and evaluated by introducing several proposed measures. This evaluation could come in handy for future studies that aim to develop new techniques for overcoming current deficiencies. Copyright © 2014 John Wiley & Sons, Ltd.

KEYWORDS
human interactive proofs; human computer interaction; information security; CAPTCHA; alternatives

***Correspondence**
Mohammad Moradi, Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Nokhbegan Blvd., Qazvin, Iran.
E-mail: Mhd.moradi@qiau.ac.ir

1. INTRODUCTION

Using the Internet for different purposes has turned into one of the daily activities of almost all people all over the world, who research, shop, use applications, and do many other things via the Internet.

As the amount of information and applications grow exponentially, the importance of preserving security attracts more attention, because there are many unfair users like attackers, hackers, and spammers who intend to have unauthorized access to these applications (information), spam, hack accounts, and do other abuses. Thus, the critical role of (information) security [1] in terms of protecting applications and their most important assets —specifically their private information—in such situations has been revealed more than before. In fact, domain of security is very broad and should be able to limit access, protect applications, detect intrusions [2], recover damage, block unauthorized users, and so on. To cope with the increasing number of security threats, several different policies, mechanisms, and tools have been proposed. Completely Automatic Public Turing test to tell Computers and Humans Apart (CAPTCHA) [3] that has been introduced more than a decade ago is currently known as a well-known and applicable security preserving tool. Over the years after its invention, several different types of CAPTCHA have been proposed and employed for various applications. From another point of view, due to the problems and deficiencies of CAPTCHAs, attempts have been made towards presenting alternative approaches.

In this paper, these two interrelated fields were reviewed in detail.

The main contribution of this study was to review, classify, and compare different types of CAPTCHA and its alternatives to draw a portrait of the current situation for future studies. In fact, this paper aimed to be a reference point for the studies that have been conducted on CAPTCHA and its alternative solutions so far in order to direct the upcoming research activities towards inventing more efficient and applicable instances.

The structure of this paper is as follows: in Section 2, the underlying concepts of CAPTCHA, its definitions as well as applications and general issues are discussed.

Section 3 introduces different types of CAPTCHAs. Problems with CAPTCHAs and different categories of alternatives are explained in Sections 4 and 5, respectively. Section 6 evaluates CAPTCHA and its alternatives and makes a comparison. In Section 7, acceptance and popularity of alternative solutions among developers are discussed. Finally, Section 8 predicts the possible future directions in the development of CAPTCHAs and alternative solutions.

2. EXPLANATION OF CAPTCHA

As the introductory section of the paper, in this section, background, applications, and more details about CAPTCHAs will be discussed to present essential information about the topic. These basics are prerequisites for further issues in the rest of the paper.

2.1. Underlying concepts

To cope with the huge amount of possible issues, security policies fall into two main categories of positive and negative security.

The former is an optimistic attitude in which all the actions are allowed for all the users until this access is explicitly denied, while the latter has a pessimistic view and implies that all the actions are restricted for all users, except the explicitly determined ones.

The negative approach towards security seems to be more rational and provides more secure systems.

From another viewpoint, security preserving takes place at two different levels: front-end and back-end. The front-end security mechanisms, as implied by its name, try to preserve security at the outmost layer of applications. Because there is no capability for detecting or recovering attacks in the representation layer, the major goal of front-end security is to limit access of unauthorized users by means of username/password mechanism, client-side validation, and so on. On the other side, in the back-end of applications, there are several different mechanisms to prevent and detect attacks such as firewalls and intrusion detection systems. Moreover, to restrict access levels of privileged users, access control policies are very popular and effective. As another classification, security preserving methods fall into two groups based on their functionality: (1) Detective methods that are responsible for specifying and detecting attacks/intrusion as well as their sources and reasons. Such methods come in handy to recover systems and repair vulnerabilities to avoid further attacks. (2) Preventive methods that are in charge of keeping systems safe by making attacks/intrusion as hard as possible. According to the mentioned separation of security methods, CAPTCHA [3] as a mechanism used for distinguishing between human users and computer programs is a front-end security mechanism and follows the negative policy (because it is assumed by its very nature that all users are computers unless proved otherwise) to prevent web bots from imitating genuine user behaviors (Figure 1).

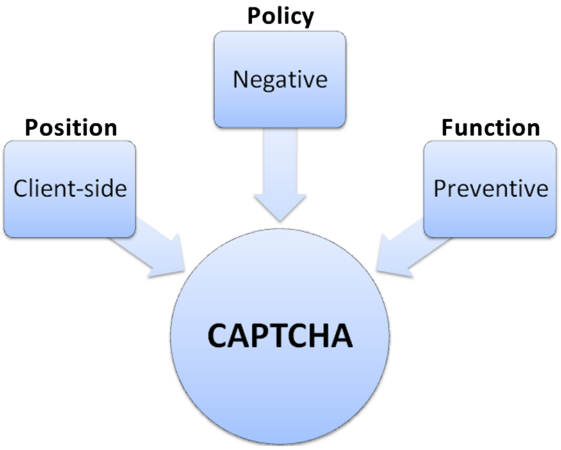


Figure 1. Features of CAPTCHA as a security mechanism.

2.2. Definition

The CAPTCHA, which was formally defined by Von Ahn *et al*. [3] for the first time, is one of the most important types of human interaction proofs (HIPs) systems to distinguish between human users and computer programs (specifically bots) using a type of challenge-response test.

Formally speaking, a CAPTCHA could be considered as a two-rounded authentication protocol as follows [4]:

$$\text{S(ervice)}\rightarrow\text{C(lient)} : \text{ a CAPTCHA } \textit{challenge}$$
$$\text{C}\rightarrow\text{S} : \textit{response}$$

Obviously, such challenges are tests that could be passed by most humans, who have no forms of disabilities, while machines could not easily, quickly, and economically pass through them [5]. Moreover, such tests are often based on hard and open problems in Artificial Intelligence (AI) [6].

2.3. CAPTCHA: nuts and bolts

Usage of CAPTCHAs was introduced in 1997 by AltaVista when developing a simple CAPTCHA that generated images of random text to prevent offensive submission of URLs to their search engines by software robots [7]. Since then, there have been many attempts to generate new and improved CAPTCHAs by other service providers and researchers to protect web applications.

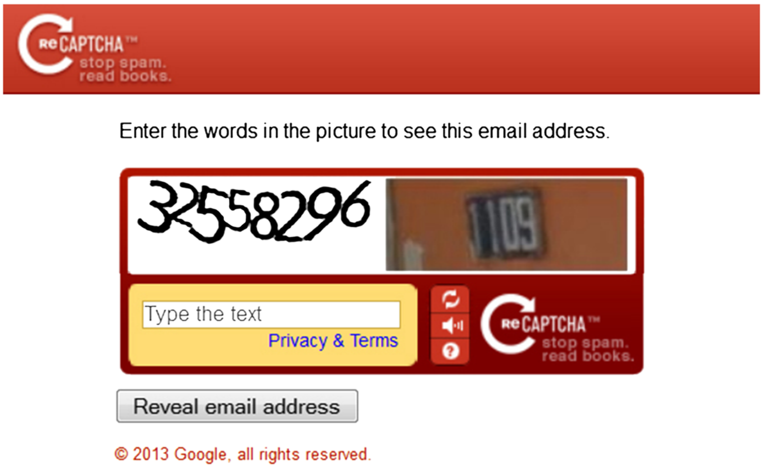
With its increasing popularity, the web has become a fertile field for those who want to abuse or damage others including spammers and hackers. When it comes to the monetary transactions and critical applications that are related to users' credentials, preserving web applications against wicked abusers becomes very important.

Because one of the most common ways of users for interacting with web applications is web forms, CAPTCHAs have employed to protect them. CAPTCHA is especially useful for protecting online forms, although it can be also helpful in keeping spammers from hijacking blogs, forums, or login/sign up procedures and even authenticating users [8].

Generally, CAPTCHAs have several applications for practical security, which include (but are not limited to) the following:

- Preventing comment spam: programs that submit cheap comments, especially with the aim of raising search engine ranks of some websites or advertising, have been around for years. Such comments are called comment spam. Using a CAPTCHA, only humans can leave comments in blogs, and no legitimate comments are ever lost [9].
- Protecting website registration: most of the websites, which have free registration such as email provider services, are the target of bots' attacks that sign up for thousands of email accounts [8]. One of the effective solutions of this problem is to use CAPTCHAs for ensuring that only humans will have free accounts. In general, free services should be protected by a CAPTCHA in order to prevent abuses by automated scripts [9].
- Protecting email addresses: To protect the email addresses posted in clear text from spammers (and screen scraper programs), an effective mechanism is to use CAPTCHAs, specifically reCAPTCHA service, named Mailhide, which helps individuals to protect their email addresses by asking people to solve a reCAPTCHA before viewing the address [10] (Figure 2).
- Online polls: using CAPTCHA is a useful means for holding safe and protected online polls and surveys. The classic example of CAPTCHA's application in this area (Slashdot.com's poll in 1999 [6]) clearly illustrates threats of this kind. Also, there are other examples in which results of (probably unprotected) polls are influenced by massive automated voting [11].
- Preventing dictionary attacks: CAPTCHAs can be also used to prevent dictionary attacks in password systems [12]. The idea is to prevent a computer from being able to iterate via the entire space of passwords by requiring it to solve a CAPTCHA after a certain number of unsuccessful logins [6].
- Securing E-commerce: security is important, but will be crucial when it comes to monetary transactions such as payment processes in E-commerce activities. To provide safety of these processes, users are asked to solve a CAPTCHA prior to clicking the submit button in payment gateway forms and thus protect their credit cards (accounts) against abuses of bots (Figure 3).
- Interactions of social networks: because of the increasing popularity of social networks (or better to say, social networking sites) among different people, they are turning into potential targets of attackers. Thus, as a preventive strategy, using CAPTCHAs may be advantageous to secure some actions that could be automated such as sending private messages [13].

On the one hand, CAPTCHAs are very useful and practical; on the other hand, they have several disadvantages and two of the most important ones are as follows: the first drawback is related to the security policy of CAPTCHAs. They assume that all users are computers unless they could pass the test, which can be annoying for genuine users. Another disadvantage is that CAPTCHAs by their nature function much better by blocking spam than by detecting humans (which is their purpose). Such blockage is applicable to the cases when spammers are machines, rather than humans [14]. In such a situation, spammers may defeat CAPTCHAs by some tricks. For example, Slashdot.com [15] described a trick in which spammers run a porn site that is gated by CAPTCHA challenges, which are actually ripped directly from Yahoo's new account creation page. Humans unwittingly solve the challenge on behalf of spammers [14]. In fact, it could be described as a bypass method. Such a trick is also known as man-in-the-middle



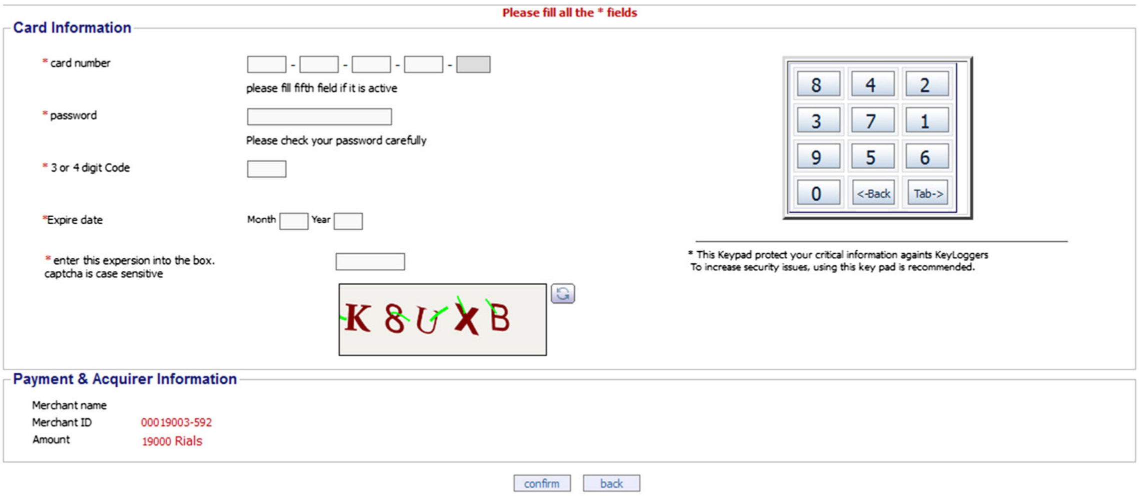
Enter the words in the picture to see this email address.

Type the text

Reveal email address

© 2013 Google, all rights reserved.

Figure 2. An example of reCAPTCHA Mailhide.



Please fill all the * fields

Card Information

* card number □ - □ - □ - □ - □
please fill fifth field if it is active

* password □
Please check your password carefully

* 3 or 4 digit Code □

*Expire date Month □ Year □

* enter this expersion into the box.
captcha is case sensitive □

* This Keypad protect your critical information againts KeyLoggers
To increase security issues, using this key pad is recommended.

Payment & Acquirer Information

Merchant name	
Merchant ID	00019003-592
Amount	19000 Rials

Figure 3. An example of securing payment with CAPTCHA.

(MITM) attack. As a solution, in [16], the authors proposed a cryptographic protocol that stopped this trick. This solution prevented MITM attacks, because the user was not directly connected to the server.

Generally, such tricks as introduced in [17] are known as human-based solving methods that could fall into two major categories: first, similar to the mentioned examples, opportunistic solving that is based on convincing an individual to solve a CAPTCHA as a part of some other unrelated tasks, and second, paid solving that relies on human solvers (workers) who are employed to solve the tests for money [17].

The second drawback is accessibility issues for users with some disabilities, specifically visual impairment. Since most of early CAPTCHAs were text/image-based, such users could not solve them and, as a result, could not use the service. To alleviate such problems, different researchers have proposed other forms of CAPTCHAs such as audio-based ones. These alternatives also have their own problems, namely high costs of implementations.

All in all, CAPTCHAs should be used as a complementary method to preserve security; otherwise, they not only solve any problems but also may become new challenges by themselves.

3. CAPTCHA VARIATIONS

Within the years after proposing the concept, several types of CAPTCHAs have been proposed in the literature, some of which have been practically applied.

From a general perspective and based on the background technologies, CAPTCHAs could be classified into three broad categories of text recognition, image recognition, and speech recognition. Based on their very nature, text- and image recognition types are among visual CAPTCHAs that are used for users without visual impairment and speech (audio)-based ones, as non-visual instances, are (mostly) proposed for the visually impaired users. Under some special conditions (e.g., ambiguity of text or images), audio CAPTCHA may be useful for other users.

Because of the diversity of CAPTCHAs, there could be many possible classifications with respect to their different characteristics. Thus far, to the best of our knowledge, the most sensible and comprehensive classification was proposed in [13]. Although this classification has tried to cover different forms of CAPTCHAs, there are several instances such as interactive and animated CAPTCHAs, which have not been enumerated. According to the present authors, incorporating video CAPTCHA under the category of image CAPTCHAs could be a bit imprecise because of the dynamism of videos by their very nature. Moreover, "cognitive category" seems to be mislabeled, because all CAPTCHAs are in principal cognitive tasks that require users to recognize information.

In fact, this classification is improved with some modifications, as shown in Figure 4.

Because there are implementations composed of different classes (or even multiple tests [18]) in some cases, there is no specific and pre-defined category for reflecting their features. Thus, in the proposed classification, a third category was introduced for such hybrid (or mixed) instances.

In addition to the category of interactive CAPTCHAs, video and animated CAPTCHAs are placed under a new class called "moving object" category. Moreover, to modify the previously mentioned issues, "cognitive" category is renamed as "semantic".

It could be claimed that the presented classification could cover any type of current CAPTCHA implementations.

3.1. Visual CAPTCHAs

The idea behind visual CAPTCHAs is to pose challenges based on user's ability to see and recognize what he or she has seen. These types of tests are generally the most straightforward and easy-to-implement ones despite their

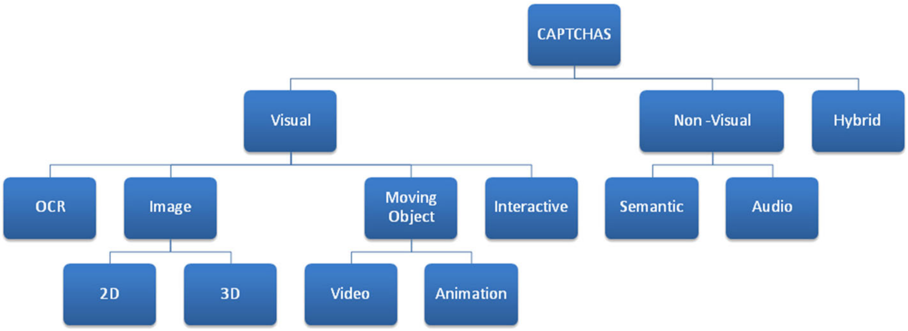


Figure 4. Classification of CAPTCHA Methods.

usability issues for visually impaired users. In fact, such tests rely on weaker (e.g., less precise) vision ability of machines than humans to sift genuine users from bots.

3.1.1. OCR-based CAPTCHAs.

The most widely used and accepted type of CAPTCHA is (optical character recognition) OCR-based or simply text-based ones. In fact, most of the deployed CAPTCHAs are of this category. Such implementation of CAPTCHAs tries to challenge automatic character recognition programs, while human users could easily understand distorted combinations of alphabets and numbers. Rotation, deformation, division, and distortion are some of the techniques used for making text-based CAPTCHAs unrecognizable for machines [19]. There are many instances of text-based CAPTCHAs in the literature including Pessimal Print [20], BaffleText [19], ScatterType [21], and GIMPY[†] and Ez—Gimpy [22]. Moreover, there is another novel technique known as ASCII art CAPTCHA,[‡] which displays a set of fonts created from a combination of characters designed to shape huge versions of keyboard characters (A–Z, 0–9, #, etc.). This set of fonts is called ASCII art characters. Figure 5 shows an example of such CAPTCHAs.

3.1.2. Image-based CAPTCHAs.

As it is difficult for machines to completely identify objects (including their types and features), another type of CAPTCHAs is based on image recognition task. In fact, this type of CAPTCHA takes advantage of computers' weakness in vision to deal with images. Image-based CAPTCHAs, in contrast to text-based ones, usually do not require users to read any text. Instead, they are expected to perform an image recognition task, that is, recognize an object or specific idea from a picture. This is usually combined with the task of grouping objects with similar properties. The key element in image CAPTCHAs—the ability of humans to understand more from a picture than can be extracted automatically—has been termed semantic gap [23]. Among several attempts for designing image recognition CAPTCHAs in recent years, Bongo [3], ESP-Pix [24], Asirra (Figure 6) [25], Imagination [26], and ARTiFACIAL [27] have been the most important ones. As some recent examples, [28–32] could be noted. There are also two important variations of image-based CAPTCHAs: 2D and 3D [33–36]. In fact, all general image-based CAPTCHAs are 2D, and there are no special things about them. However, 3D instances follow different approaches to employ 3D images based on the concept that humans can recognize 3D (character) image better than machines. This way, some use 3D models as challenges, and others create 3D from 2D images. As an implemented solution [37], Rolko's work was based on human imagination and spatial perspective. The basic idea of this method is the rotation of a special 3D model and finding the correct position of rotation.

3.1.3. Moving Object CAPTCHAs.

As a relatively new trend, this category that includes video and animation CAPTCHAs is based on showing a clip to users and asking them to type what they have seen or specify what they have perceived [38]. Although it seems to be more secure, it is more complicated and expensive to implement than other methods. [35,38–40] are some examples of such CAPTCHAs (Figure 7).

3.1.4. Interactive CAPTCHAs.

The major idea behind interactive CAPTCHAs is to involve a higher level of users' interaction in order to provide more secure measures. Thus, this type of tests needs users' mouse click, drag, and so on to be solved. However, most of the so-called interactive CAPTCHAs are not fully interactive, and users are only asked to type the identified/perceived response in the specified place. Except users with some disabilities, solving these CAPTCHAs is usually easy for humans, while bots have some essential difficulties with them. This fact points to the current position of machines' ability to imitate humans' interactive behaviors, especially motion-based acts. References [41–46] are some interactive CAPTCHAs that have been proposed in the literature (Figure 8). Moreover, a new generation of interactive CAPTCHAs introduced in recent years is game-based CAPTCHA [47,48]. The main features of such novel tests are their attractiveness and user-friendliness.

[†] http://www.captcha.net/captchas/gimpy/

[‡] tppCaptcha. Available at http://www.thephppro.com/products/captcha/

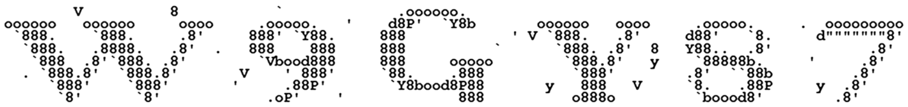
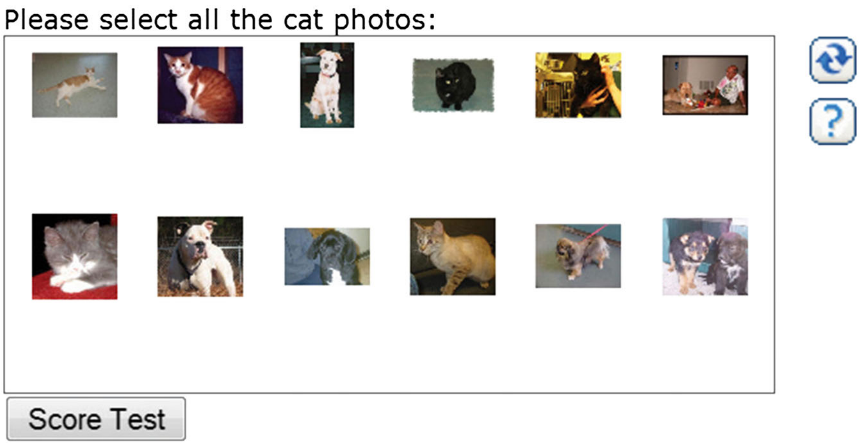
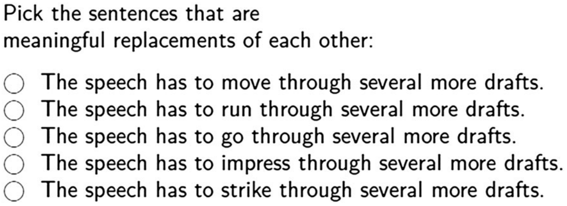


Figure 5. ASCII Art CAPTCHA (tppCaptcha).



Please select all the cat photos:

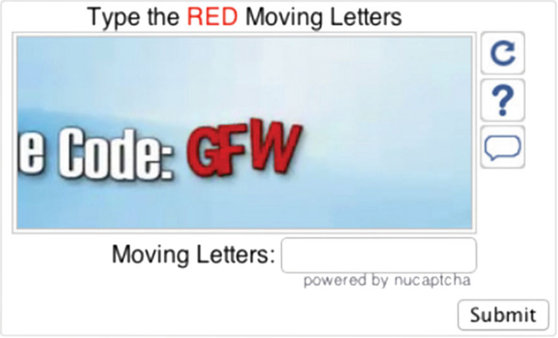
Figure 6. The Asirra CAPTCHA (image taken from [25]).



Pick the sentences that are meaningful replacements of each other:

○ The speech has to move through several more drafts.
○ The speech has to run through several more drafts.
○ The speech has to go through several more drafts.
○ The speech has to impress through several more drafts.
○ The speech has to strike through several more drafts.

Figure 7. An example of NuCAPTCHA (image taken from [40]).



Type the RED Moving Letters

Moving Letters:

Figure 8. A mockup of the 3D interactive CAPTCHA (image taken from [41]).

3.2. Non-visual CAPTCHAs.

In the non-visual class, the subject of challenges is human-only characteristics to recognize sound- and semantic-based tests, instead of necessarily visual ability of users. Although speech-recognition tests may face some usability issues and are subject to be attacked by automatic speech-recognition programs, semantic-based CAPTCHAs are placed in a higher position as they are much harder to be cracked by machines and simultaneously are (relatively) easy for most of users including those with visual/hearing impairment. This is because in such tests, the challenge is a semantic question (not an image or text itself) that could be presented by screen reader. Nonetheless, this type of CAPTCHAs could be problematic for people with cognitive deficiencies.

3.2.1. Audio-based CAPTCHAs.

Another main category of CAPTCHA methods is audio (or sound)-based CAPTCHAs. Similar to image-based methods, they take advantage of computers' weakness in recognizing spoken language in the presence of noise and distortion. This type is usually considered as an alternative to visual CAPTCHAs in the case of visually impaired users. In this method, a word (or a sequence of letters and numbers) is said, and users must type what they hear. The first work in this domain was performed in [49]. As other notable studies, the following references could be mentioned [50–53]. In addition to traditional audio CAPTCHAs, a relatively new approach, for which speech interaction is the best description, is to ask users to repeat a sentence. Then, the response is analyzed to verify whether it is the requested sentence and said by a human, not a speech synthesis system [54]. Lack of the necessary requirements—speaker or microphone based on the situation—is the most important barrier for such methods. Moreover, people with some types of speech-related disability have difficulty with it.

DOI: 10.1002/sec

3.2.2. Semantic CAPTCHAs.

Semantic CAPTCHAs may be the most tolerable ones against attacks as they are (in most cases) far beyond abilities of machines when it comes to answering only-for-human questions. In fact, this category provides users with questions that are easy to understand and answer by humans and relatively (at least at this time) difficult for computers, such as what color is the sky? Nonetheless, such CAPTCHAs may be vulnerable against attacks using a computational knowledge engine, such as Wolfram Alpha[§] or even a search engine. A broad range of questions including logical ones can be used for the implementation of semantic CAPTCHAs. Also, a question with no specific (or more than one specific) answer is possible. Implementation costs of this type of CAPTCHA are very low as they can be presented in the plain text format [55]. There are several studies in the literature in this field such as [56–59] (Figure 9).

3.3. Other approaches

In addition to the mentioned main categories, there are several other methods that implement CAPTCHAs with respect to less-focused aspects such as different spoken languages and personalization. As a matter of fact, the works, which are pointed out in this section, are not out of the scope of the previously mentioned categories. But they present new perspectives on combining classical techniques and concepts with novel ideas to introduce extended instances.

3.3.1. Extended Approaches.

As mentioned earlier, CAPTCHA research community continuously proposes different CAPTCHAs that cannot be completely mentioned or classified here; however, most of the currently proposed CAPTCHAs may fall into the proposed classification as they share common ideas. Anyway, some of the innovative and complementary studies introduced in this section may inspire future research activities.

Because most CAPTCHAs are proposed for those who speak English, non-English users probably have some difficulties with them. Therefore, designing CAPTCHAs in other languages could be an effective idea. As an instance, Shirali-Shahreza and Shirali-Shahreza proposed a CAPTCHA for Persian/Arabic users (Figure 10) [60]. References [61–64] are similar works following such an idea.

A new scheme was proposed in [65], in which a CAPTCHA is displayed over the web page for a fixed time; then, it replaces itself until the CAPTCHA is filled by users. As the refreshing process just works with CAPTCHA and does not affect the web page, it does not bother users. Time limitation of this method makes it harder to be cracked by bots.

In [66], the authors proposed a graphical password scheme that took advantage of CAPTCHAs to implement a graphical password. It is claimed that this method could increase security by adding capability to withstand spyware, shortening size of the password space and resisting brute force attacks.

Ross *et al.* [67] introduced a CAPTCHA based on upright orientation of line drawings rendered from 3D models. The models are selected from a large database, and images are rendered from random viewpoints, affording many different drawings from a single 3D model. Then, it provides users with a set of images, and users are supposed to choose an upright orientation for each image. To pass the test, the claimed users should know the semantic content of each image (Figure 11).

Solve Media[¶] improved—usually causeless—CAPTCHAs in an economical way, in which advertising contents are what users should enter as correct responses to pass the tests. Such an innovative solution could be extended for different purposes as well. For example, such advertisements may be replaced with cultural/educational contents (Figure 12).

One of the latest research directions about this topic is the issue of personalization of CAPTCHAs. Despite its different applications, the key concept of such an operation is to make CAPTCHAs more user-specific with respect to his or her unique characteristics and conditions. One aspect of personalization is individual differences of users' cognitive capabilities that should be investigated and integrated in the user interface design process of CAPTCHAs [68]. Such issues are very important for user preferences and the performance related to the challenges presented by CAPTCHAs. Cognitive style of users is an important factor that could affect usability of CAPTCHAs. As another aspect, personalizing CAPTCHAs could be performed by personalized contents such as geographic information to prevent the third-party human attacks. As an example of such studies, in [69], authors took advantage of the idea of using a geographic scene image (a rotated image-based CAPTCHA combined with Google Map information), which is only privately known to every user. As another recent work, results of [70] indicated that usability of CAPTCHA mechanisms might be supported by personalization techniques based on individual differences in cognitive processing.

3.3.2. Mobile CAPTCHAs.

Over recent years, there are increasing number of people who browse the Internet using mobile devices such as smart phones and tablets. Mobile web users are prompted to solve the same CAPTCHAs designed for desktop computers; however, mobile devices are poorly suited for viewing CAPTCHAs and typing; therefore, there is a critical need for designing mobile-only (device-specific) CAPTCHAs to preserve security of growing and popular mobile applications. There are several studies in the

[§] Wolfram|Alpha: Computational Knowledge Engine. Available at http://www.wolframalpha.com/

[¶] Solve media. Available at http://solvemedia.com/index.html

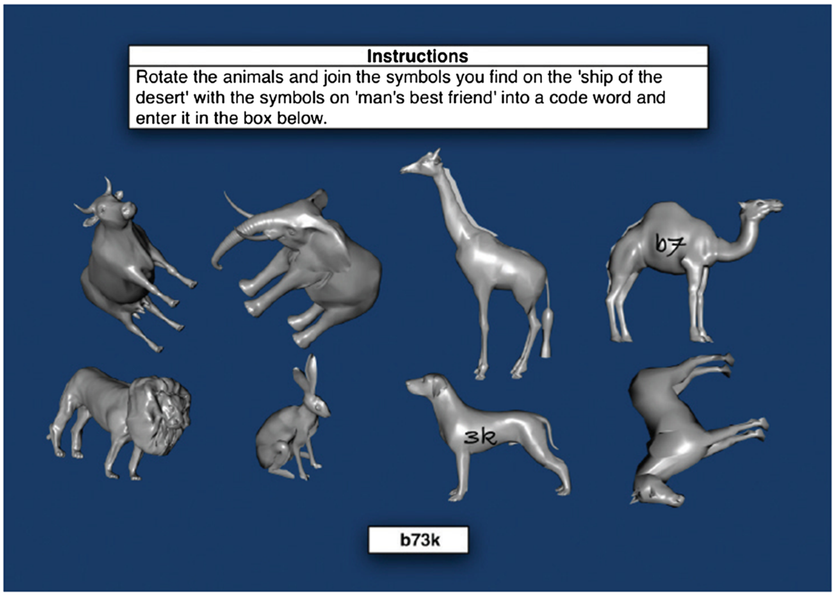

Instructions
Rotate the animals and join the symbols you find on the 'ship of the desert' with the symbols on 'man's best friend' into a code word and enter it in the box below.

Figure 9. An example of cognitive question (CAPTCHA) (image taken from [58]).

محمد
عیسی
مریم
کریم

Figure 10. some examples of the Persian/Arabic CAPTCHA (image taken from [60]).

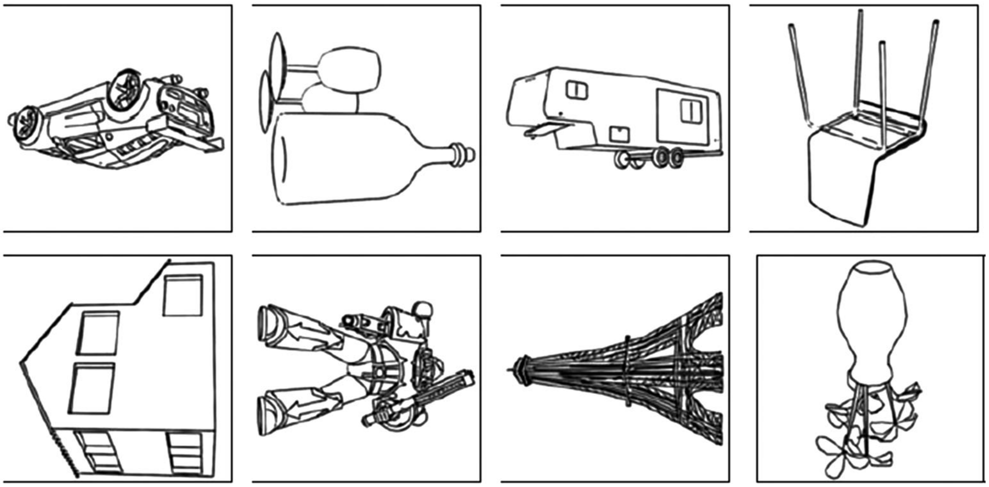
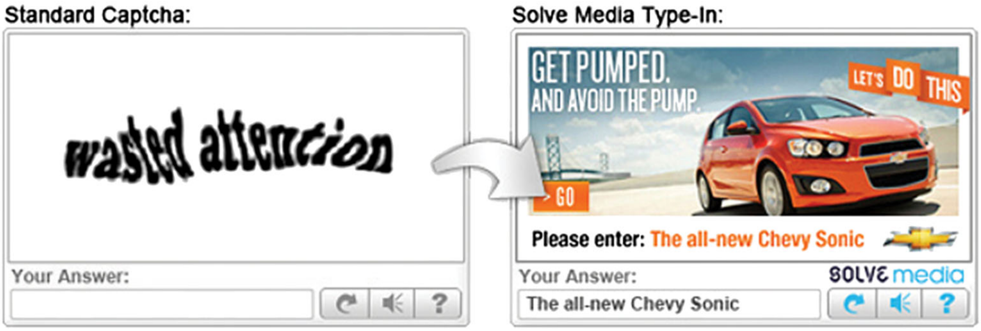


Figure 11. An example of line drawing CAPTCHA (image taken from [67]).

literature about this topic. In [71], a new user-friendly mobile-specific CAPTCHA approach was introduced.

Following this trend, in [72], a mobile-specific scheme was presented for a CAPTCHA service on Cloud. To be usable for mobile devices, the proposed solution leveraged unique features like touchscreen of mobile gadgets to provide a user-friendly and efficient CAPTCHA. Also, usability of mobile CAPTCHAs with voice and touch input was evaluated in [73]. As an important outcome of this study, it has been shown that users prefer the CAPTCHAs involving touch input on mobile devices. As one of the most recent studies in this context, authors in [74] proposed two new alternative ways for designing CAPTCHAs for mobile devices, in which users said the answer, instead



Your Answer: The all-new Chevy Sonic

Figure 12. An instance of Solve Media invented CAPTCHA.

of typing it with (a) visually provided output stimuli (SeeSay) or (b) auditorily (HearSay). Results of this study unveiled that SeeSay CAPTCHA required less time to be solved and users preferred it over current text-based CAPTCHA solutions.

As one of the enterprise-level instances of mobile-specific CAPTCHAs, *Confident Technologies*[††] have released a solution, in which, instead of asking users to type letters of a word or something like that, they are requested to select several small pictures on the screen as "answers" to a series of brief instructions. Figure 13 shows an example of *Confident's* CAPTCHA.

4. PROBLEMS WITH CAPTCHAS

The CAPTCHAs by themselves have some problems in different phases from preliminary concepts to protection from attacks. These challenges can be classified into three main categories, as shown in Figure 14. This classification divides the problems into three main stages of CAPTCHAs' lifecycle. Implementation phase includes technical issues around the CAPTCHAs that are in the backstage. Security phase points to the threats from outside and considerations from inside (implementation) to protect CAPTCHAs. Third phase pays attention to the problems that affect human interaction with CAPTCHAs. In other words, the mentioned phases specify the issues, threats, and barriers of encountering CAPTCHAs, respectively.

4.1. Implementation

Generally, there are two main directions towards using CAPTCHAs. The first one is to build up CAPTCHA from scratch. Although following this approach could result in more customized and tailored instances for specific requirements and applications, there are several unavoidable issues related to it. Not having enough information about different types of CAPTCHAs, their vulnerabilities and challenges (in implementing a robust test), defects in development and implementation phases, and human-centered problems such as usability issues and total costs are among the most important drawbacks of this approach. Moreover, there are several limitations such as bandwidth issues and copyright considerations—especially for image-and video-based CAPTCHAs, which may affect the development process. The second and logical alternative of this approach is to use a third-party service. Low or (mostly) no implementation costs, active support, high level of confidence (tolerability against attacks), and efficiency are the most important advantages of such services. There are dozens of such services, namely keyCaptcha,[‡‡] sweetCaptcha,[§§] and popular reCAPTCHA.[¶¶] Totally, to compare the two mentioned approaches and also select from among third-party services, four influential measures may be regarded as follows (in the order of importance): efficiency, user-friendliness, variety (of tests, services, and plugins for different systems), and total costs. In fact, there are other factors to mention; however, in terms of security and usability perspectives, the presented ones are at high priority level. For example, reCAPTCHA as a free, widely used, and effective service provides plugins for Wordpress and Mediawiki as well as API for programming languages. Moreover, it presents another service, Mailhide, to protect email addresses.

4.2. Security

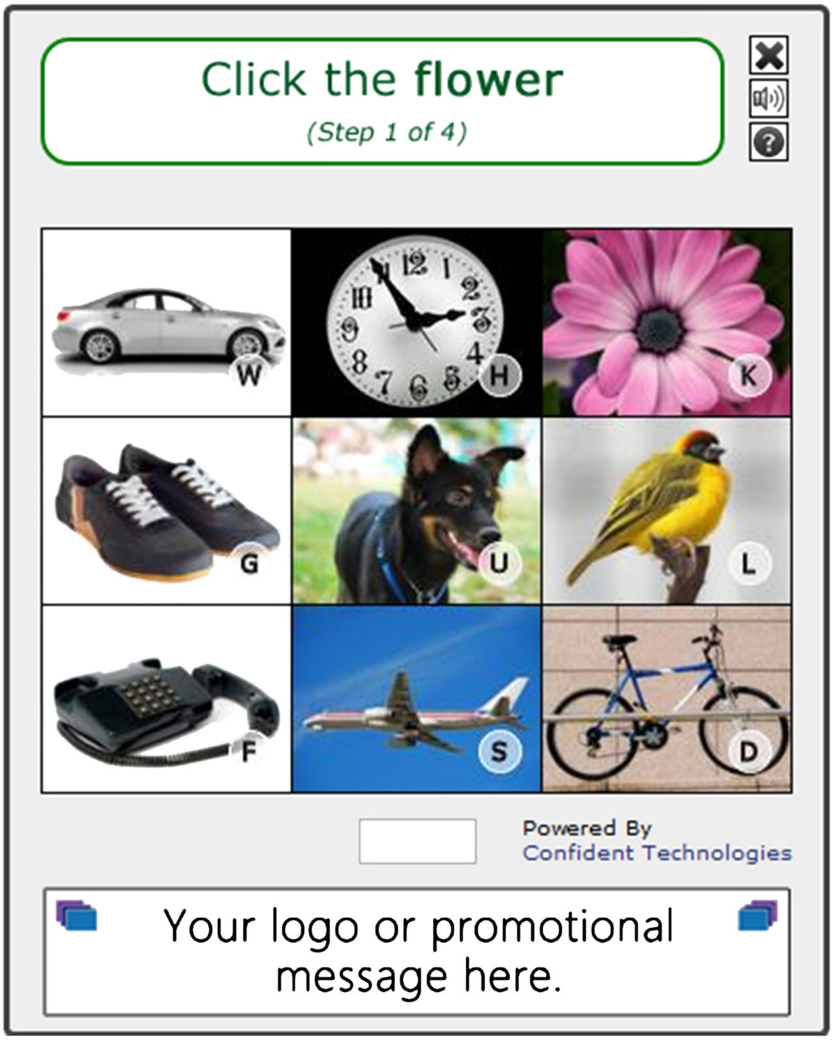
Because CAPTCHAs are mechanisms for protecting resources and applications (specifically, on the web) against intrusive bots, they are targets of numerous attacks in different aspects. Therefore, to be more secure and tolerable, several considerations in design and implementation must be carried out. From the early days of CAPTCHA, there have been many attempts to attack and consequently break them. For this reason, robustness of any CAPTCHA against cracking attempts is one of the major factors for its evaluation. Because of this fact, one of the main contributions of the literature is to invent new methods for breaking the proposed CAPTCHAs. As some examples of these

[††] Confident Technologies. Available at http://confidenttechnologies.com

[‡‡] KeyCAPTCHA. Available at https://www.keycaptcha.com/

[§§] Sweet Captcha. Available at http://sweetcaptcha.com/

[¶¶] reCAPTCHA. Available at http://www.google.com/recaptcha/


Click the flower

(Step 1 of 4)

Powered By
Confident Technologies

Your logo or promotional message here.

Figure 13. Confident technologies' mobile-specific CAPTCHA.

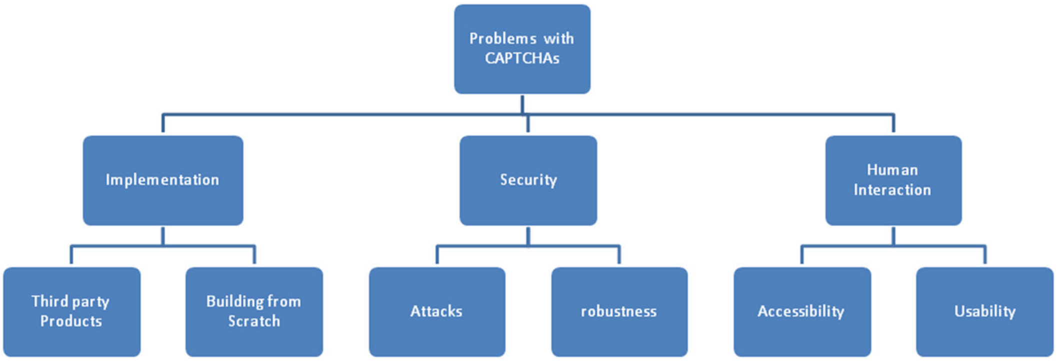


Figure 14. Classification of CAPTCHA problems.

studies, the following instances could be mentioned: Mori and Malik [75] and Thayananthan *et al.* [76] have tried to attack the famous Gimpy method; in [77], Microsoft's CAPTCHA was attacked; attacking image recognition CAPTCHAs has been studied in [78] and [79]; and [80] was a study on breaking audio CAPTCHAs. Moreover, as a recent study, authors of [81], based on their proposed approach for recognizing arbitrary multi-digits, claimed that their model could solve the hardest category of famous reCAPTCHA tests with the accuracy of 99.8%.

In general, most of the attacks could be categorized into the following four classes: Blind Guessing, AI Attacks, Relay Attacks, and Side-Channel Attacks [13]. Although researchers are trying to do their best to design resistant CAPTCHAs against the mentioned attacks, there is another type of attack that could bypass almost any CAPTCHA. This type of attack is based on human intervention, and its most usual form is to place (hard) CAPTCHAs on high volume websites, especially porn websites, and people have to figure out the CAPTCHA for the free content or viewing the content. Also, when it comes to spamming, commercial spammers, instead of relying on bots, prefer to pay to workers in low-paid countries to write spam content. Such a phenomenon was reported in [82]. Although using human users to break CAPTCHAs is relatively expensive and is only rational for commercial purposes, in fact, it could be regarded as a concern for the security of CAPTCHAs. To overcome these third-party attacks, reference [83] was one of the early works that was dedicated for this problem.

4.3. Human interaction

From human users' perspective, the most important aspect of CAPTCHA is its usability. Nowadays, many CAPTCHA implementations are hard to solve by humans as people have to refresh or, in some cases, ignore the service, which is a serious drawback. As an example, there are situations in which, because of unordinary hardness of CAPTCHA, users withdraw from performing actions such as comment posting. Hence, an important aspect to be considered in the process of designing CAPTCHAs is regarding usability metrics.

According to the universal usability [84] concept, information and communication systems should be designed in a way that could be used by a broad range of users including those with some disabilities. Moreover, as CAPTCHAs have some critical applications like the ones in protecting voting systems and monetary transactions, it is more vital for them to be user-friendly and easy to use.

Although there are many studies in the literature that have discussed different aspects of usability of CAPTCHAs [85–89] and [90] performed a comprehensive survey on the usability features of CAPTCHAs, the problems have still remained unsolved.

Because the users' community includes people with different levels of knowledge, age, cognitive abilities, and impairments in addition to various types of equipment, designing a fully accessible CAPTCHA seems to be impossible.

The current solutions cannot cover all situations; for example, cognitive and audio CAPTCHAs have been proposed as alternatives for visual ones in the case of users with visual impairment, while accessibility issues for users with cognitive disabilities or high level of noise in audio CAPTCHAs make them practically unusable. In addition to specific users (those with some types of disabilities), other ones usually have some troubles with CAPTCHAs because of the issues such as inappropriate color, readability, ambiguity, and general presentational problems, which make CAPTCHAs a hateful phenomenon among web users.

Along with the researchers who have developed new methods regarding usability and accessibility features, the answer may be to use some alternatives for CAPTCHAs.

5. ALTERNATIVES TO CAPTCHAS

Although considering accessibility and usability is one of the most important topics in designing effective CAPTCHAs, there are currently some critical issues that make them difficult to use. In fact, as a public thought, CAPTCHAs are not popular for web users. In other words, CAPTCHAs are not the only way for protecting web forms against bots. Therefore, web developing community and researchers have tried to suggest some alternatives for CAPTCHAs to satisfy users. Any proposed alternative should have two major features altogether; it should be user-friendly and preserve security against web bots. Thus, there have been several recommendations that are not based on AI concepts and are simple, yet effective. These tricks have been proposed to cover different applications of CAPTCHAs in a broad range from preventing bots from filling the registration forms to spamming in the form of commenting in the blogs and so forth. In general, CAPTCHA alternatives are based on the concepts of prevention and detection, and all of them could fall into three major categories: (1) interactive, (2) administrative, and (3) cheating bots.

5.1. Interactive methods

Interactive methods are those that provide a mechanism to distinguish between human users and bots, just like CAPTCHAs in theory, but in the form of request for doing actions that bots could not usually perform such as motion (mouse)-based operations or presenting human-only assets like social security number. The word interactive means that to pass the test, real human users as active entities should interact with systems or present their own unique assets, while machines (passive entities) usually neither have such an ability (at least at this time) nor include those unique features.

- Unique identity

Since all people have their own unique identifiers and assets for their identification such as national code in Iran and social security number in the USA, these features can be the potential options for distinguishing human users from machines. This method is very common (especially, in the last decade) because of its simplicity and availability; however, it has some critical problems because collecting sensitive data from a large number of people in a host could make it a potential and considerable target for abuse and attack [91]. Therefore, using this approach could be very costly because of the overhead of additional security requirements, except in specific applications.

Another form of unique identity is users' biometric characteristics. A broad range of tests, from fingerprint and retinal scanning to DNA matching, has promised to authoritatively check a person's identity [91] while effectively limiting the ability of web bots to do what they are employed for. As biometrics security is linked to trust, it could be considered the most reliable approach. There are many studies in the literature in this field such as [92], in which a comprehensive review was performed on different aspects, variables, and use cases in biometrics from the perspective of end users. Also, there are studies that have focused on a specific and common form of biometrics, keystroke dynamics [93–95]. However, they have their own drawbacks, the most important of which is the need for expensive hardware infrastructures that are not so prevalent. Because of this problem, using biometrics for preserving security is limited to the highly sensitive applications and specifically military ones in most of the cases.

- SMS/email verification

One of the techniques used by most of the websites for ensuring users' credential and authority is to ask them to verify (mostly) their newly created accounts or (rarely) their transactions via SMS/email sent to them. At first glance, it is feasible and low cost, but it has several barriers due to limited worldwide penetration or coverage for mobile phones or accessibility problems for blind users [91]. In the best condition, this process is time-consuming and non-user-friendly for most of the users.

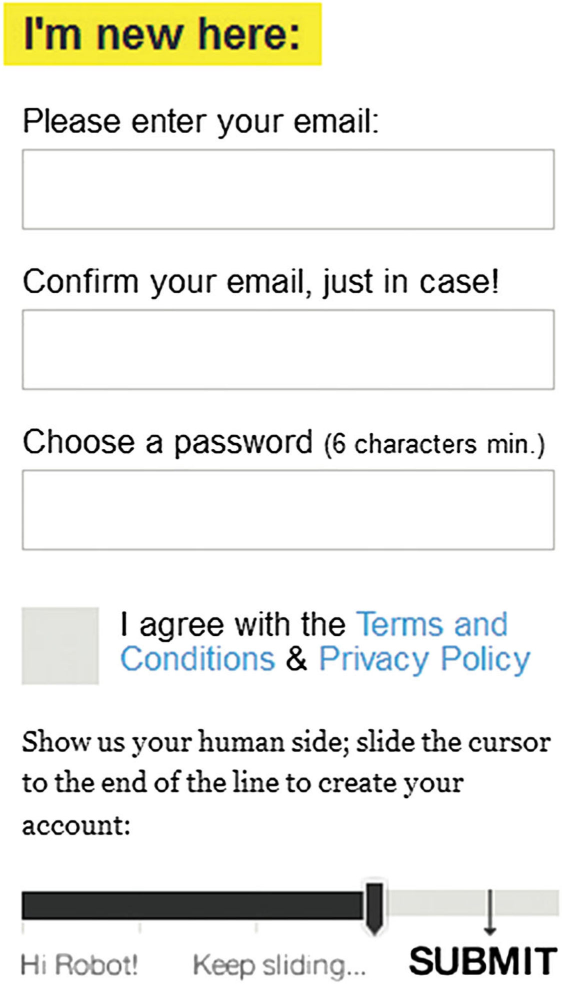
- Motion-based operation

Because bots mostly work with raw codes, they could not usually pass the tests in which some motion-based operations are needed. As some examples of such tests, clicking checkbox, using light pens, and dragging a gauge and slider (Figure 15) can be considered. Also, game-based tests, such as one shown in Figure 16, are other funny alternatives. Although these approaches seem to be effective, they have two major problems. First and foremost, their usability and accessibility for people with some disabilities make them impractical in some situations. Another threat for this idea is automated cursor handling applications [96] that could be used to break such tests.

5.2. Administrative methods

In this category, there are methods in which some high-level supervision tasks take place to detect spam content, prevent bots from attacks, and so on. Using third-party services for content analysis and filtering is one of the most reliable tools that could be used to detect spamming by administrators.

- Spam detection



I'm new here:

Please enter your email:

Confirm your email, just in case!

Choose a password (6 characters min.)

□ I agree with the Terms and Conditions & Privacy Policy

Show us your human side; slide the cursor to the end of the line to create your account:

Figure 15. An example of interactive alternatives to CAPTCHA (http://theymakeapps.com/users/add).

Because one of the most important applications of CAPTCHA is to prevent spam and spammers, a more efficient way to cope with spam-oriented problems is to detect and filter them. Following this trend, there are many studies in the literature that have proposed different techniques for facing the spam phenomenon. One simple way of standing against spam in websites and blog systems is to use third-party plugins and services like Akismet.[‖] Such systems automatically analyze user-submitted data and flag spam. As spam has different types, various methods have been proposed for detecting and filtering them. Authors in [97] performed a study on machine learning techniques for spam filtering in blog comments. Thus, they compared four famous machine learning techniques: Naïve Bayes, K-nearest neighbor, neural networks, and Support Vector Machines. In [98], a novel collaborative blog spam filtering was proposed based on the manual identification of spams and sharing the obtained information through a network

[‖] http://akismet.com/

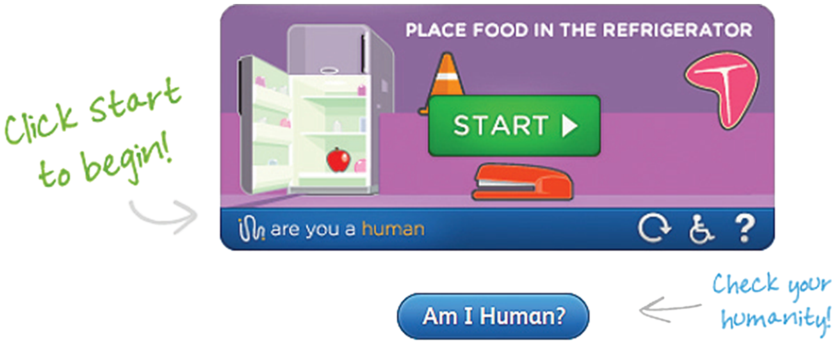


Figure 16. A game-based alternative to CAPTCHA (http://areyouahuman.com/demo/).

of trust. Issue of content-based spam filtering for short text messages, including blog comments, was studied in [99].

In addition to spam filtering and detection methods, it is possible to use heuristic checks, which evaluate behavior of clients. It may be possible to detect the presence of a robotic user based on the volume of data the user requests, series of commonly visited pages, IP addresses, data entry methods, or other signature data that can be collected [91]. Regarding this trend, Prieto *et al.* [100] proposed a new web spam detection system called SAAD based on a set of heuristics and their use in a C4.5 classifier. In [101], the authors tried to automatically detect comment spam through content analysis using some previously non-described features. Also, other spam comment detection methods were studied in [102,103]. Moreover, the possibility of identifying spambots without using CAPTCHA was experimented in [104].

- Logging

For the purposes of auditing and supervising, a not-so-intelligent, yet effective method is to keep a log of everything that happens during a form submission process [105]. The logged data can be invaluable for the later inspection of attacks, hacking attempts, and implementation of solutions. As far as the website with heavy traffic load is concerned, logging could be very costly.

- Server-side validation

To protect applications (especially, on the web), there is an indispensable need for client-side mechanism in terms of submitted data validation, which is not adequate. To provide more security, server-side checking [105] seems to be necessary, specifically for the fields placed in email headers. This careful check should be performed for SQL injection, JavaScript injection, HTML tags, and so on.

- Response time

The idea behind this tricky approach is to calculate the time, during which web forms are filled and submitted [106]. Although it takes a little time for users to complete forms, bots are almost instantaneous. Then, systems could determine bots if the form is filled out in a pre-defined amount of time. Determining the pre-defined time can be carried out by some real-world examinations.

- Centralized sign-on

A centralized sign-on system can alleviate the abusing potential by putting all the impetus on a single system to authenticate users. Systems such as Microsoft Passport offer such centralization [107]. An effective alternative for such systems is to use social login (sign-in) systems using the existing login information from other services, which could be implemented using OpenID.*** This system avoids privacy issues due to not being limited to a single authentication provider. The problem with such federated system could go back to the first place in which users obtain their own ID [107]. For example, to gain a Gmail account, users need to solve a CAPTCHA!

Figure 17 shows an instance of a centralized user base by providing social login functionality.

Another method in this field is to apply certificates for those individuals desiring to verify their identity. These types of systems have been implemented for securing web pages and authenticating emails [91]. In spite of its efficiency, this method is too costly to be implemented in every case.

- Limiting accounts

A way for limiting abuse in the systems such as free email providers is to deliberately throttle new accounts for a specified period [107]; for example, everyone (claimed users) could only send seven emails per day for the first week. However, such strategy could not be the silver bullet, because it could be bypassed, for example, by recruiting tens of spammers. In fact, such a limitation may influence genuine users as well. Like other approaches, this one could be handy in specific applications.

*** OpenID Foundation website. Available at http://openid.net/

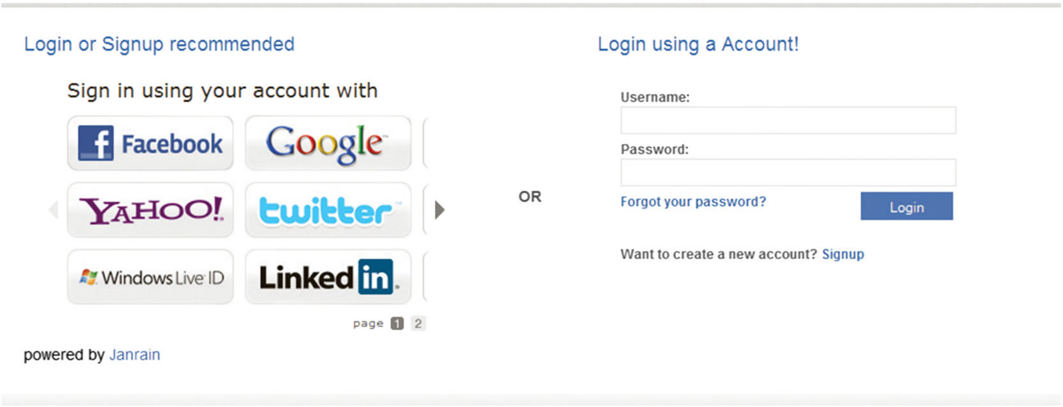


Figure 17. Example of social login functionality (http://www.mahalo.com/login).

- Detecting presence of JavaScript

This option is open for pages running the JavaScript code. When JavaScript is executed, a human is definitely using the page [105]. Thus, developers can create a way to check this activity. The main problem of this option happens when a user deactivates JavaScript.

- Human verification

In the case of specific and critical applications, perhaps human intervention is required for the verification of accounts or user-submitted data. Doing so provides a higher level of security and assurance. However, this approach may not be applied in real-world applications, in which there are a huge number of users; direct human verification could be placed after an automatic checking that excludes potential spam content/activities.

5.3. Cheating bots

To prevent spam bots from doing malicious actions, another working strategy is to cheat and recognize them by monitoring the consequences. In fact, cheating can be defined as reverse techniques that identify bots by attracting them to do actions that human users could not do. Also, trapping bots can be regarded as another solution. As an important forthcoming threat for such approach, it could be remarked that if these strategies would be employed on a large scale, then bots will be probably reprogrammed in order to overcome them.

- The Honeypot method

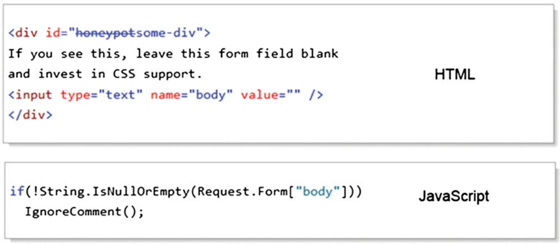
In 2007, Phil Haack proposed a smart method [108] for detecting bots using a honeypot, which is also known as hidden input field trap. This method is based on an additional field in the web forms that is hidden to users. Because the spam bots process raw HTML rather than rendering the source code, therefore, they could not detect the trick. If the data are inserted in this hidden field, it could be found that it has not been carried out by a human user [106]. Moreover, this approach can be made more complicated and extended using JavaScript and CSS styling. Figure 18 illustrates sample codes for the implementation of this trick.

- Switching form fields

Another option for cheating bots is to switch the order of form fields or make users to, for example, enter the required information in the reverse order of fields, for example, entering name in a phone field [109]. Using JavaScript and CSS, this approach could be implemented more efficiently. This method can be regarded in another way by creating form field names in a dynamic way.

- Confirmation page trick

A good trick for trapping bots and preventing them can be when users enter information into one page and activate the submit button to proceed to the next page. In such cases, the next page may contain the information previously entered by the user and also a confirmation button [110], which prevents bots from successfully entering information into the system, because they would not have been programmed with the expectation of such an extra step [111]. Of course, to respect genuine users, it would be clearly explained to them.



```
<div id="~~honeypot~~some-div">
If you see this, leave this form field blank
and invest in CSS support.
<input type="text" name="body" value="" />
</div>
```

```
if(!String.IsNullOrEmpty(Request.Form["body"]))
  IgnoreComment();
```

Figure 18. A simple implementation of Honeypot trick (codes taken from [108]).

6. EVALUATION

Selecting the best alternatives for CAPTCHAs is a crucial function, because disproportion between requirements and solutions not only solves the problems but also addresses new ones.

Therefore, to run an evaluation on these alternatives as well as CAPTCHAs themselves, several measures have been introduced by the authors as follows. There are several different potential criteria to be selected for evaluation; however, the following measures have been proposed with regard to their priorities and importance. In fact, these measures may provide a sensible framework for evaluating the current CAPTCHAs and their alternatives as they cover the most important aspects that might be encountered in selecting/implementing a solution. In fact, the most important considerations in selecting an efficient solution can be based on the target community and the resources that should be protected.

- *Cost*: implementation and maintenance (modifying, upgrading) of deployed solution should be economical or at least in a reasonable cost. For example, biometrics is not a rational solution for an entertainment website. From a high-level viewpoint, this factor is one of the most influential ones for management decision making. In fact, one of the most important reasons of using CAPTCHAs is their (usually) low cost of implementation, because many free solutions are available and the commercial ones are also not so expensive. Thus, to be efficient and able to engage users (security administrators) in employing an alternative solution (instead of CAPTCHA), total costs should be as minimum as possible. From this viewpoint, the alternatives belonging to the cheating bots' category are more proportionate and applicable. However, in applying this measure to CAPTCHAs, considering the need for storing and collecting images including copyright charges for image-based ones and the need for producing and analyzing the sounds, which probably require costly equipment (for speech-recognition instances), text-based CAPTCHAs are more economical. In general, using a third-party solution such as reCAPTCHA is definitely more economical and has less (or even without) supervisory overheads.
- *Efficiency*: efficiency refers to the functionality of solutions with regard to the overhead due to management operations. In fact, a balance should be made between them. Depending on the cases, factors may have different weights for evaluation. For example, in a military case, to provide more functionality, high management overhead is acceptable. In other words, efficiency could be defined as the assurance impact of a method to work as expected and designed for. Determining how much a technique is efficient is in fact a hard task, because it is non-functional and there are several factors that may influence it. Nonetheless, in the context of CAPTCHA and its alternatives, if efficiency is thought of as the capability of a given method to distinguish machines from human users, semantic CAPTCHAs (despite their own intrinsic problems) seem to be more powerful and effective. Also, administrative alternatives are more efficient, because they pose more precise criteria as their challenges.
- *Effect on usability*: As in CAPTCHAs, usability issues are the most important barriers for designing and implementing alternative solutions. To this end, the goal should be to provide users with simple and user-friendly interfaces with respect to the universal usability concepts. Complexity should go to the back-end as possible. Frankly speaking, proposing a completely usable CAPTCHA/alternative for all types of users (probably) is an impossible mission. The best approach for reaching the highest level of usability conformance, as mentioned earlier, is to personalize solutions based on cognitive/motional ability of users. In other words, user-specific solutions should be implemented rather than case-specific ones. All in all, semantic CAPTCHAs and alternative techniques that try to cheat the bots are more usable solutions, because they have minimum interaction with genuine users. However, semantic CAPTCHAs are subject to critical problems, especially for individuals facing cognitive disorders.
- *Robustness*: from security perspective, solutions must be as robust as possible in order not to be breakable by attackers. Thus, the first and foremost step is to survey recent advancement in cracking similar applications to protect the solution against them. As a matter of fact, back-end-related solutions (administrative) are more secure and more expensive. Furthermore, efficiency and robustness are two sides of the same coin, because an efficient solution that could not tolerate attacks is in fact inefficient. This issue is not necessarily true in a reverse direction; however, it can be said that robustness is one of the key factors that shape effectiveness. Realistically, no mechanisms could be completely secure against attacks, and the related literature continuously proposes cracking techniques for every method. Accordingly, alternative solutions are generally more tolerable against attacks, among which administrative methods are at a higher level of robustness. In the context of CAPTCHA, semantically designed CAPTCHAs seem to be more sustainable, because currently, machines are relatively weak in recognizing/answering semantic questions.

Based on the aforementioned criteria, in the following tables (Tables I and II), different categories of CAPTCHAs and their alternatives are evaluated (by the present authors).

In the context of CAPTCHAs, selecting the best choice is a difficult task because of the non-functionality of security mechanisms and unequal characteristics of users. In fact, there is not one best choice to cover all the occasions. Nevertheless, semantic CAPTCHAs, due to their low implementation costs and difficulty of machines in

Table I. Evaluation of CAPTCHAs.

Method	Cost	Efficiency	Effect on usability	Robustness	Drawbacks	Assessment
OCR	Low	Medium	High	Medium	Well-studied topic (existence of attack and cracking methods)	Most straightforward, easy to implement and common type of solutions, not very reliable for sensitive situations
Image	Medium	Medium	High	Medium	Copyright issues, well-studied topic (existence of attack and cracking methods)	Popular method with many third-party implemented instances, easy to implement
Moving Objects	High	High	High	High	Usability issues, Bandwidth consuming problems, high costs of implementation	Costly method yet effective one, not so usable and relatively tolerable against attacks
Interactive	High	High	Highest	High	Cause severe effects on usability	Tolerable against attack, not so usable because of the need for users' interaction
Semantic	Lowest	Highest	Low	Highest	Cause problems for users with cognitive deficiency	Low cost, (mostly) usable and robust
Audio	High	High	Medium	High	Need to additional requirements (speaker, headphone, etc.)	Relatively effective and costly with some usability issues, good as secondary solution

Table II. Evaluation of CAPTCHAs' alternatives.

Method	Cost	Efficiency	Effect on usability	Robustness	Drawbacks	Assessment
Interactive	High	Medium	Highest	Medium	Efficiency depends on user interaction	Well-suited for specific and sensitive applications
Administrative	High	Highest	Low	Highest	Overheads of supervision tasks, time-consuming	Because it is more user-friendly and transparent, it is the best class
Cheating bots	Lowest	Medium (to high)	Lowest	Medium	Non-standard usage	Non-standard and tricky approach that could affect structure of forms and pages

recognition, are the best ones. However, from the user-centered design perspective, interactive image-based (game-based) CAPTCHAs seem to be more interesting and user-friendly.

Also, according to the evaluation in Table II, when requirements are high security and efficiency for sensitive resources, administrative methods are the first choice. However, to use such methods, there are two major obstacles: higher cost of implementation and supervision tasks that make it practical only for large and susceptible systems.

The most efficient solution for the small- and medium-scaled websites and applications is to use a mixed approach wherever required. In most cases, such a strategy works well.

To bold the differences between CAPTCHAs and alternative solutions as an evaluation means to specify the best options, in the following table (Table III), a high-level comparison of such solutions is presented. Based on these general (proposed) measures, the claim on the lack of the best choice for every situation has been proved. Indeed, the most influential factor for selecting a solution is features and conditions of target users. Moreover, because most of the time users are not of a specific group with the same cognitive/motional styles and abilities, the best approach is to employ a case-specific (or even a mixed) approach based on the sensitivity of resources to be protected.

7. DISCUSSION

Although it is revealed that in some cases, alternative methods are far better (efficient, secure, and user-friendly) than CAPTCHAs, there is a fact that in comparing these two groups of options, CAPTCHAs are more popular and applicable among users (security administrators, websites owners, etc.). However, for some specific applications such as spam detection, alternatives are so popular (e.g., using Akismet Plugin). In this section, some important reasons are mentioned for this inclination.

Table III. High-level comparison of CAPTCHAs and alternative solutions.

Approach	Total costs	Efficiency	Effect on usability	Robustness	Drawbacks	Assessment
CAPTCHA	Medium	Medium	High	Medium (depends on the implementation)	Usability issues, non-user- friendliness, well-studied attacks	Popular among administrators (first-choice), easy to implement, well-studied topic with many implemented variations
Alternatives	Low to high (depends on the case)	High	Medium	Medium (depends on the implementation)	Unorganized solutions, management overheads, being not so popular	Seems to be more effective, some of its variations are costly, good for sensitive applications, less-focused topic

- Most administrators do not code their CAPTCHAs from scratch; for different reasons including robustness and popularity, low implementation costs, and less management overhead, they use the third-party solutions. Obviously in such cases, implementing an alternative solution is not rational, except in critical and sensitive situations.
- Because most of the alternative solutions are often newcomers and there is not sufficient information about them, a notable number of administrators are not aware of them at all. Moreover, unlike CAPTCHAs, there are not many famous third-party solutions that present such alternatives to users. Simply, nowadays, alternative solutions are in their infancy both from diversity and popularity perspectives.
- Unfortunately, most of the alternative solutions are not well-structured and packaged to be easily practical and implementable for most users. Lack of such features makes them informal and non-user-friendly (users here refer to managers and admins).
- Unawareness about the severe effects of CAPTCHAs on conversion rate, usability, accessibility, and users' satisfaction is the most important factor for website owners to continue using them. It has been seen that in some cases, managers that are aware of their consequences (and simultaneously unaware about the alternatives) have removed CAPTCHAs from their websites. They are more concerned about their users than attackers.

8. FUTURE DIRECTIONS

Regarding the evolution of security mechanisms over years, due to the critical issues of CAPTCHAs, especially in the usability context, it is definitely predictable that in the near future, alternative solutions will be major players of the field of distinguishing human and machine tests. Nonetheless, until those days, to respond to the community's requests, there are some ongoing trends as follows:

With respect to the aesthetical aspects of UI and flourishing user experience design, future CAPTCHAs should be more user-friendly and well-designed to alleviate their intrinsic inconvenience. In addition, making CAPTCHAs more interesting (e.g., by presenting game-based examples) could work to this end.

Moreover, because mobile devices are becoming so popular, to leverage their capabilities, an ongoing trend is to develop gesture-based CAPTCHAs. Furthermore, developing universally usable CAPTCHAs as the most difficult task in this context may be the concern of researchers.

One of the possible future directions for the alternatives could be packaging them in the form of plugins for some of the proposed solutions in order to make them organized and structured.

Taking benefits of new technologies such as HTML5's Canvas is also another approach for inventing alternatives for CAPTCHAs. Moreover, to cheat bots, it is possible to use script, which unlike CAPTCHAs, works in the background to redirect or puzzle machines.

9. CONCLUSIONS

The CAPTCHA, as a mechanism for preventing web bots and scripts from imitating functions of human users, has several applications including securing online voting, E-commerce, and sign up/login. There are different types of CAPTCHAs such as text-based and image-based to be implemented in different situations. In addition to intrinsic challenges on implementing and securing CAPTCHAs, usability and accessibility issues are the major barriers in their deployment. To overcome these serious problems that could affect the efficiency of CAPTCHAs, several alternatives have been proposed. Based on the underlying concepts, the alternatives fall into three categories as follows: (1) interactive, (2) administrative, and (3) cheating bots.

Although the current alternatives are effective and useful, there are some problems with (some of) them due to higher costs of implementation. Moreover, usability and accessibility problems with these alternatives have remained unsolved in some cases.

All in all, to gain more security, the silver bullet is to establish a mixed protection mechanism including CAPTCHA and its alternatives wherever required.

Regarding the mentioned issues, in this paper, different types of CAPTCHAs and their alternative solutions, their latest advancement, and issues were reviewed; then, they were evaluated by introducing several criteria. The main intention of this paper was to provide a reference point of the current studies and trends for future works.

ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for their helpful comments and suggestions.

REFERENCES

1. Von Solms B. Information security—a multidimensional discipline. *Computers & Security* 2001; **20** (6):504–508.
2. Verwoerd T, Hunt R. Intrusion detection techniques and approaches. *Computer Communications* 2002; **25**(15):1356–1365.
3. Von Ahn L, Blum M, Langford J. Telling humans and computers apart automatically. *Communications of the ACM* 2004; **47**(2):56–60.
4. Yan J. Bot, cyborg and automated Turing test. *Security Protocols,* Lecture Notes in Computer Science, Vol. **5087**, Springer: Berlin Heidelberg, 2009; 190–197.
5. Chellapilla K, Larson K, Simard P, Czerwinski M. Designing human friendly human interaction proofs (HIPs). *Proceedings of SIGCHI Conference on Human factors in computing systems* (CHI 05), 2005; 711–720.
6. Von Ahn L, Blum M, Hopper NJ, Langford J. CAPTCHA: using hard AI problems for security. *Advances in Cryptology—EUROCRYPT 2003*, Vol. **2656**, Lecture Notes in Computer Science, Springer: Berlin Heidelberg, 2003; 294–311.
7. Pope C, Kaur K. Is it human or computer? Defending E-Commerce with CAPTCHA. *IT Professional* 2005; **7**(2):43–49.
8. Jeng AB, Tseng CC, Tseng DF, Wang JC. A study of CAPTCHA and its application to user authentication. *Proceedings of the Second International Conference on Computational Collective Intelligence: Technologies and Applications*, ICCCI'10, Vol. **2**, 2010; 433–440.
9. What is a CAPTCHA? Available at: http://www.google.com/recaptcha/captcha [accessed June 2013]
10. reCAPTCHA Mailhide. Available at: http://www.google.com/recaptcha/mailhide [accessed June 2013].
11. Basso A, Miraglia M. Avoiding massive automated voting in Internet polls. *Electronic Notes in Theoretical Computer Science (ENTCS)* 2008; **197**(2):149–157.
12. Pinkas B, Sander T. Securing passwords against dictionary attacks. *Proceedings of the 9th ACM conference on Computer and communications security*, Washington, DC, USA, 2002; 18–22. DOI: 10.1145/586110.586133
13. Hidalgo JMG, Alvarez G. Chapter 3—CAPTCHAs: an artificial intelligence application to web security. In *Advances in Computers*, Vol. **83**, Zelkowitz MV (ed). Elsevier, 2011; 109–181.
14. Atwood J. CAPTCHA effectiveness. Available at: http://www.codinghorror.com/blog/2006/10/captcha-effectiveness.html [accessed June 2013]
15. Porn rewards users to get past anti-spam Captchas. Available at: http://yro.slashdot.org/story/04/01/28/1344207/porn-rewards-users-to-get-past-anti-spam-captchas [accessed March 2013]
16. Petrillo UF, Mastroianni G, Visconti I. The design and implementation of a secure CAPTCHA against man-in-the-middle attacks. *Security and Communication Networks* 2013. doi:10.1002/sec.825.
17. Motoyama M, Levchenko K, Kanich C, McCoy D, Voelker GM, Savage S. Re: CAPTCHAs—understanding CAPTCHA—solving services in an economic context. *Proceedings of USENIX Security Symposium*, 2010.
18. Longe OB, Robert ABC, Onwudebelu U. Checking Internet masquerading using multiple CAPTCHA challenge-response systems. *Proceedings of 2nd International Conference on Adaptive Science & Technology*, ICAST, IEEE, 2009; 244–249.
19. Chew M, Baird HS. Baffletext: a human interactive proof. *Proceedings of SPIE-IS &T Electronic Imaging, Document Recognition and Retrieval X* 2003; **5010**:305–316. doi:10.1117/12.479682.
20. Baird HS, Coates AL, Fateman RJ. Pessimalprint: a reverse Turing test. *International Journal on Document Analysis and Recognition (IJDAR)* 2003; **5**(2–3):158–163.
21. Baird HS, Moll MA, Wang SY. Scattertype: a legible but hard-to-segment CAPTCHA. *Proceedings of 8th International Conference on Document Analysis and Recognition*, ICDAR' 05, Vol. **2**, 2005; 935–939.
22. Mori G, Malik J. Recognizing objects in adversarial clutter: breaking a visual CAPTCHA. *Proceedings of the IEEE computer society conference on Computer vision and pattern recognition*, CVPR'03, 2003; 134–141.
23. Smeulders A, Worring M, Santini S, Gupta A, Jain R. Content-based image retrieval at the end of the early years. *Transactions on Pattern Analysis and Machine Intelligence, IEEE* 2000; **22**(12):1349.
24. Von Ahn L, Blum M, Langford J. Telling humans and computers apart automatically—how lazy

cryptographers do AI. *Communications of the ACM - Information Cities* 2004; **47**(2):56–60.

25. Elson J, Douceur JR, Howell J, Saul J. Asirra: a CAPTCHA that exploits interest-aligned manual image categorization. *Proceedings of 14th ACM conference on Computer and communications security*, ACM, New York, CCS '07, 2007; 366–374. DOI: 10.1145/1315245.1315291
26. Datta R, Li J, Wang JZ. Imagination: a robust image-based CAPTCHA generation system. *Proceedings of the 13th annual ACM international conference on Multimedia*, MULTIMEDIA '05, 2005; 331–334.
27. Rui Y, Liu Z. ARTiFACIAL: automated reverse Turing test using facial features. *Proceedings of the 11th ACM International Conference on Multimedia*, ACM, New York, MULTIMEDIA '03, 2003; 295–298. DOI: 10.1145/957013.957075
28. Raj SBE, Jayanthi VS, Muthulakshmi V. A novel architecture for the generation of picture based CAPTCHA. *Advanced Computing, Networking and Security*, Lecture Notes in Computer Science, Vol. **7135**, Springer: Berlin Heidelberg, 2012; 568–574.
29. Kim J, Kim S, Yang J, Ryu JH, Wohn K. FaceCAPTCHA: a CAPTCHA that identifies the gender of face images unrecognized by existing gender classifiers. *Multimedia Tools and Applications* 2013. doi:10.1007/s11042-013-1422-z.
30. Hsieh CC, Wu ZY. Anti-SIFT images based CAPTCHA using versatile characters. *Proceedings of International Conference on Information Science and Applications* (ICISA), IEEE, 2013; 1–4.
31. Aadhirai R, Kumar PJ, Vishnupriya S. Image CAPTCHA: based on human understanding of real world distances. *Proceedings of 4th International Conference on Intelligent Human Computer Interaction* (IHCI), IEEE, 2012; 1–6.
32. Obimbo C, Halligan A, De Freitas P. CaptchAll: an improvement on the modern text-based CAPTCHA. *Procedia Computer Science* 2013; **20**:496–501.
33. Hoque ME, Russomanno DJ, Yeasin M. 2D Captchas from 3D models. *Proceedings of the IEEE. SoutheastCon Conference*, Memphis, IEEE, 2006; 165–170. DOI: 10.1109/second.2006.1629343
34. Imsamai M, Phimoltares S. 3D CAPTCHA: a next generation of the CAPTCHA. *Proceedings of the International Conference on Information Science and Applications*, ICISA' 2010, IEEE, 2010; 1–8.
35. Ince IF, Salman YB, Yildirim ME, Yang TC. Execution time prediction for 3D interactive CAPTCHA by keystroke level model. *Proceedings of the 4th International Conference on Computer Sciences and Convergence Information Technology*, ICCIT '09, IEEE, 2009; 1057–1061.
36. Chow YW, Susilo W. Enhanced STE3D-CAP: a novel 3d CAPTCHA family. *Information Security Practice and Experience*, Lecture Notes in Computer Science, Vol. **7232**, Springer: Berlin Heidelberg, 2012; 170–181.
37. 3D CAPTCHA. Available at: http://www.3dcaptcha.net/index.php?lang=en [accessed June 2013]
38. Cui JS, Mei JT, Wang X, Zhang D, Zhang WZ. A CAPTCHA implementation based on 3D animation. *Proceedings of the International Conference on Multimedia Information Networking and Security*, MINES '09, Vol. **2**, 2009; 179–182.
39. Kluever KA, Zanibbi R. Balancing usability and security in a video Captcha. *Proceedings of the 5th Symposium on Usable Privacy and Security*, SOUPS '09, 2009; 1–11.
40. NuCaptcha, Inc. NuCaptcha: most secure and usable Captcha. Available at: http://www.nucaptcha.com/ [accessed December 2012]
41. Winter-Hjelm C, Kleming MH, Bakken RH. An interactive 3D CAPTCHA with semantic information. *Proceedings of the Norwegian AI Society*, NAIS' 2009, ACM, 2009; 157–160.
42. Shirali-Shahreza S, Ganjali Y, Balakrishnan R. Verifying human users in speech-based interactions. *Proceedings of 12th Annual Conference of the International Speech Communication Association*, ISCA 2011, Florence, Italy, 2011.
43. Chew M, Tygar JD. Image recognition CAPTCHAs. *Proceedings of the 7th Information Security Conference*, ISC' 04, Lecture Notes in Computer Science, Springer: Berlin/Heidelberg, 2004; 268–279.
44. Shirali-Shahreza M, Shirali-Shahreza S. Drawing captcha. *Proceedings of the 28th International Conference on Information Technology Interfaces*, IEEE, 2006; 475–480.
45. Desai A, Patadia P. Drag and drop: a better approach to CAPTCHA. *Proceedings of the India Conference*, INDICON, Annual IEEE, 2009; 1–4. DOI: 10.1109/INDCON.2009.5409359
46. Chow R, Golle P, Jakobsson M, Wang L, Wang X. Making CAPTCHAs clickable. *Proceedings of the 9th workshop on Mobile computing systems and applications*, HotMobile '08, Napa Valley, California, ACM, 2008; 91–94.
47. Yang TI, Koong CS, Tseng CC. Game-based image semantic CAPTCHA on handset devices. *Multimedia Tools and Applications* 2013. doi:10.1007/s11042-013-1666-7.
48. Kani J, Nishigaki M. Gamified CAPTCHA. *Human Aspects of Information Security, Privacy, and Trust*, Lecture Notes in Computer Science, Vol. **8030**, Springer: Berlin Heidelberg, 2013; 39–48. DOI: 10.1007/978-3-642-39345-7_5

49. Chan N. Abstract of sound oriented CAPTCHA. *Proceedings of the Workshop on Human Interactive Proofs*, Palo Alto, CA, 2002; 35.
50. Chan TY. Using a text-to-speech synthesizer to generate a reverse Turing test. *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence*, ICTAI' 03, IEEE, 2003; 226–232.
51. Kochanski G, Lopresti D, Shih C. A reverse Turing test using speech. *Proceedings of the 7th International Conference on Spoken Language Processing*, 2002; 1357–1360.
52. Holman J, Lazar J, Feng JH, D'Arcy J. Developing usable CAPTCHAs for blind users. *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, Assets '07, ACM, 2007; 245–246.
53. Schlaikjer A. A dual-use speech CAPTCHA: aiding visually impaired web users while providing transcriptions of audio streams. *Technical Report CMU-LTI-07-014*, Carnegie Mellon University, 2007.
54. Shirali-Shahreza S, Ganjali Y, Balakrishnan R. Verifying human users in speech-based interactions. *Proceedings of 12th Annual Conference of the International Speech Communication Association* (ISCA 2011), Florence, Italy, 2011.
55. Pablo X, Santos A, Marcial F, Joaquim C. A captcha in the text domain. In *Proceedings of OTM Workshops*, Robert M, Zahir T, Pilar H (eds). Lecture Notes in Computer Science, Springer: Berlin/Heidelberg, 2006; **4277**(1): 605–615.
56. Lupkowski P, Urbanski M. SemCAPTCHA—user-friendly alternative for OCR-based CAPTCHA systems. *Proceedings of the International Multiconference on Computer Science and Information Technology*, IMCSIT' 2008, IEEE, 2008; 325–329.
57. Chew M, Tygar J. Collaborative filtering CAPTCHAs. *Proceedings of the 2nd International Workshop on Human Interactive Proofs*, HIP' 2005, Lecture Notes in Computer Science, Vol. **3517**, 2005; 66–81.
58. Richard B, Stefan K. Towards human interactive proofs in the text-domain. *Proceedings of the 7th Information Security Conference*, Springer Verlag: Berlin/Heidelberg, 2004; 257–267.
59. Yamamoto T, Tygar JD, Nishigaki M. CAPTCHA using strangeness in machine translation. *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications*, AINA' 2010, IEEE, 2010; 430–437.
60. Shirali-Shahreza MH, Shirali-Shahreza M. Persian/Arabic CAPTCHA. *International Journal on Computer Science and Information Systems* 2006; **1**(2):63–75.
61. Shirali-Shahreza MH, Shirali-Shahreza M. Persian/Arabic Baffletext CAPTCHA. *Journal of Universal Computer Science* 2006; **12**(12):1783–1796.
62. Shirali-Shahreza MH, Shirali-Shahreza M. Multilingual CAPTCHA. *Proceedings of the 5th IEEE International Conference on Computational Cybernetics*, IEEE, 2007; 135–139.
63. Shirali-Shahreza MH, Shirali-Shahreza M. Nastaliq CAPTCHA. *Iranian Journal of Electrical and Computer Engineering* 2007; **5**(2):109–114(in Persian).
64. Xu S, Lau F, Cheung WK, Pan Y. Automatic generation of artistic Chinese calligraphy. *Intelligent Systems, IEEE* 2005; **20**(3):32–39.
65. Yadava P, Sahu C, Shukla S. Time-variant Captcha: generating strong Captcha Security by reducing time to automated computer programs. *Journal of Emerging Trends in Computing and Information Sciences* 2011; **2**(12):701–704.
66. Wang L, Chang X, Ren Z, Gao H, Liu X, Aickelin U. Against spyware using CAPTCHA in graphical password scheme. *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications*, AINA' 10, IEEE, 2010; 760–767.
67. Ross SA, Halderman JA, Finkelstein A. Sketcha: a Captcha based on line drawings of 3D models. *Proceedings of the 19th international conference on World wide web*, Raleigh, North Carolina, USA, WWW '10, 2010; 821–830.
68. Belk M, Fidas C, Germanakos P, Samaras G. Do cognitive styles of users affect preference and performance related to CAPTCHA challenges? *Extended Abstracts of the ACM SIGCHI Conference on Human Factors in Computing Systems* (CHI 2012), Austin, Texas, USA, ACM, 2012; 1487–1492.
69. Wei TE, Jeng AB, Lee HM. GeoCAPTCHA—a novel personalized CAPTCHA using geographic concept to defend against 3 rd Party Human Attack. *Proceedings of 31st International Performance Computing and Communications Conference* (IPCCC), IEEE, 2012; 392–399.
70. Belk M, Germanakos P, Fidas C, Holzinger A, Samaras G. Towards the personalization of CAPTCHA mechanisms based on individual differences in cognitive processing. *Human Factors in Computing and Informatics, Lecture Notes in Computer Science* 2013; **7946**:409–426.
71. Lin R, Huang SY, Bell GB, Lee YK. A new CAPTCHA interface design for mobile devices. In *Proceedings of the Twelfth Australasian User Interface Conference* (AUIC '11), Vol. **117**, Lutteroth C, Shen H (eds). Australian Computer Society, Inc.: Darlinghurst, Australia, 2011; 3–8.
72. Saxena A, Chauhan NS, Sravan KR, Vangal AS, Rodrguez DP. A new scheme for mobile based CAPTCHA service on Cloud. *Proceedings of*

International Conference on Cloud Computing in Emerging Markets (CCEM), IEEE, 2012; 1–6.

73. Wismer AJ, Madathil KC, Koikkara R, Juang KA, Greenstein JS. Evaluating the usability of CAPTCHAs on a mobile device with voice and touch input. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, SAGE Publications, 2012; **56**(1): 1228–1232.
74. Shirali-Shahreza S, Penn G, Balakrishnan R, Ganjali Y. SeeSay and HearSay CAPTCHA for mobile interaction. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, ACM, New York, NY, USA, 2013; 2147–2156. DOI: 10.1145/2470654.2481295
75. Mori G, Malik J. Recognizing objects in adversarial clutter: breaking a visual CAPTCHA. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR' 03, IEEE, Vol. **1**, 2003; 134–141.
76. Thayananthan A, Stenger B, Torr PH, Cipolla R. Shape context and chamfer matching in cluttered scenes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR' 03, IEEE, Vol. **1**, 2003; 127–133.
77. Yan J, El Ahmad AS. A low-cost attack on a Microsoft Captcha. *Proceedings of the 15th ACM conference on Computer and communications security*, Alexandria, Virginia, USA, CCS'08, ACM, 2008; 543–554.
78. Zhu BB, Yan J, Li Q, *et al*. Attacks and design of image recognition CAPTCHAs. *Proceedings of the 17th ACM conference on Computer and communications security*, Chicago, Illinois, USA, CCS'10, ACM, 2010; 187–200.
79. Fritsch C, Netter M, Reisser A, Pernul G. Attacking image recognition Captchas—a naive but effective approach. *Proceedings of the 7th international conference on Trust, privacy and security in digital business*, TrustBus'10, 2010; 13–25.
80. Tam J, Simsa J, Hyde S, Ahn LV. Breaking audio CAPTCHAs. *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems*, MIT Press, Vancouver, British Columbia, Canada, 2009; 1625–1632.
81. Goodfellow IJ, Bulatov Y, Ibarz J, Arnoud S, Shet V. Multi-digit number recognition from street view imagery using deep convolutional neural networks. arXiv preprint, 2013; arXiv preprint arXiv:1312.6082.
82. Russian serfs paid $3 a day to break CAPTCHAs. Available at: http://www.theregister.co.uk/Print/2008/03/14/captcha_serfs/ [accessed June 2013]
83. Truong HD, Turner CF, Zou CC. iCAPTCHA: the next generation of CAPTCHA designed to defend against 3rd party human attacks. *Proceedings of IEEE International Conference on Communications*, ICC' 11, Kyoto, Japan, IEEE, 2011; 1–6.
84. Shneiderman B. Universal usability. *Communications of the ACM* 2000; **43**(5):84–91.
85. Sauer G, Holman J, Lazar J, Hochheiser H, Feng J. Accessible privacy and security: a universally usable human-interaction proof tool. *Universal Access in the Information Society* 2010; **9**(3):239–248.
86. Yan J, El Ahmad AS. Usability of CAPTCHAs or usability issues in CAPTCHA design. *Proceedings of the 4th symposium on Usable privacy and security*, Pittsburgh, Pennsylvania, SOUPS '08, ACM, 2008; 44–52.
87. Hochheiser H, Feng J, Lazar J. Challenges in universally usable privacy and security. *Symposium on Usable Privacy and Security*, SOUPS, Pittsburgh, PA, USA, 2008.
88. Sutherland C. Usability and security of text-based CAPTCHAs. *UMM CSci Senior Seminar Conference*, 2012.
89. Kluever KA, Zanibbi R. Video CAPTCHAs: usability vs. security. *Proceedings of IEEE Western New York Image Processing Workshop*, WNYIP '08, IEEE, 2008.
90. Pakdel R, Ithnin N, Hashemi M. CAPTCHA: a survey of usability features. *Research Journal of Information Technology* 2011; **3**:215–228.
91. Inaccessibility of CAPTCHA. *W3C Working Group Note*, 2005. Available at: http://www.w3.org/TR/turingtest/ [accessed November 2012]
92. Bernecker O. Biometrics: security: an end user perspective. *Information Security Technical Report* 2006; **11**(3):111–118.
93. Guven A, Sogukpinar I. Understanding users' keystroke patterns for computer access security. *Computers & Security* 2003; **22**(8):695–706.
94. Karnan M, Akila M, Krishnaraj N. Biometric personal authentication using keystroke dynamics: a review. *Applied Soft Computing* 2011; **11**(2):1565–1573.
95. Limpanuparb T. The enhancement of password security system using keystroke verification. *NECTEC Technical Journal* 2004; **4**:531–537.
96. SeleniumHQ Browser Automation. Available at: http://docs.seleniumhq.org/ [accessed January 2013]
97. Romero C, Valdez MG, Alanis A. A comparative study of machine learning techniques in blog comments spam filtering. *Proceedings of The IEEE International Joint Conference on Neural Networks* (IJCNN), 2010; 1–7.
98. Han S, Ahn YY, Moon S, Jeong H. Collaborative blog spam filtering using adaptive percolation search. *Proceedings of Workshop on the Weblogging Ecosystem* (WWW'06), 2006.

99. Cormack GV, Gómez Hidalgo JM, Sánz EP. Spam filtering for short messages. *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, ACM, 2007; 313–320.
100. Prieto VM, Álvarez M, López-García R, Cacheda F. Analysis and detection of web spam by means of web content. *Multidisciplinary Information Retrieval*, Lecture Notes in Computer Science, Vol. **7356**, Springer: Berlin Heidelberg, 2012; 43–57.
101. Huang C, Jiang Q, Zhang Y. Detecting comment spam through content analysis. *Web-Age Information Management*, Lecture Notes in Computer Science, Vol. **6185**, Springer: Berlin Heidelberg, 2010; 222–233.
102. Kantchelian A, Ma J, Huang L, Afroz S, Joseph A, Tygar JD. Robust detection of comment spam using entropy rate. *Proceedings of the 5th ACM Workshop on Security and Artificial Intelligence*, ACM, 2012; 59–70.
103. Huang C, Jiang Q, Zhang Y. Detecting comment spam through content analysis. *Proceedings of Web-Age Information Management*, Lecture Notes in Computer Science Volume **6185**, Springer: Berlin Heidelberg, 2010; 222–233.
104. Potdar V, Ridzuan F, Hayati P, *et al*. Spam 2.0: the problem ahead. *Proceedings of Computational Science and Its Applications–ICCSA*, Lecture Notes in Computer Science, Vol. **6017**, Springer: Berlin Heidelberg, 2010; 400–411.
105. Buckler C. 10 things to check before using a CAPTCHA. Available at: http://www.sitepoint.com/captcha-alternatives/ [accessed December 2012]
106. Bushell D. In search of the perfect CAPTCHA. Available at: http://coding.smashingmagazine.com/2011/03/04/in-search-of-the-perfect-captcha/ [accessed December 2012]
107. Edwards J. Beyond CAPTCHA: no bots allowed! Available at: http://www.sitepoint.com/captcha-problems-alternatives/ [accessed December 2012]
108. Haack P. Honeypot Captcha. Available at: http://haacked.com/archive/2007/09/10/honeypot-captcha.aspx [accessed November 2012]
109. Alternatives to CAPTCHAs. Available at: http://www.simplecaptchas.com/alternatives.php [accessed April 2012]
110. Pogue D. Use it better: 8 alternatives to the Hated Captcha. Available at: http://www.scientificamerican.com/article.cfm?id=pogue-8-alternatives-to-hated-captcha [accessed November 2012]
111. Keeping out the bad bots. Available at: http://www.timezoneoneblog.com/2013/03/04/keeping-out-the-bad-bots/ [accessed December 2012]

DOI: 10.1002/sec