Exploring the Usability of CAPTCHAS on Smartphones: Comparisons and Recommendations

Gerardo Reynaga
School of Computer Science
Carleton University
Ottawa, Canada
Email: gerardor@scs.carleton.ca

Sonia Chiasson
School of Computer Science
Carleton University
Ottawa, Canada
Email: chiasson@scs.carleton.ca

Paul C. van Oorschot
School of Computer Science
Carleton University
Ottawa, Canada
Email: paulv@scs.carleton.ca

Abstract—**Completely Automated Public Turing tests to tell Computers and Humans Apart (captchas) are challenge-response tests used as a security mechanism on the web to distinguish human users from automated programs. While captchas are often necessary to stop abuse of resources, most existing schemes are intended for traditional desktop computing environments rather than for mobile device usage. In this paper we present a comparative user study of nine captcha schemes on smartphones to assess whether alternative input mechanisms help improve the usability of captchas in smartphones, and to evaluate the usability of modified schemes intended to be more suitable for smartphones. The results show that although participants find virtual keyboards on smartphones prone to errors they prefer them as input mechanism over other alternatives. We also found that the content of the challenge is highly relevant in users' perceptions when it comes to captchas on smartphones. Based on our experiences, we offer a set of ten specific recommendations for the implementation of captchas on smartphones.**

I. Introduction

Mobile usage has grown substantially in the last few years and the trend continues [1], [2]. Over 58% of the US population now owns a smartphone [2] and saturation is even higher in other places such as the Hong Kong, the UK, and Australia [3]. In fact, 84% of mobile users in the US had used their devices for shopping within the first quarter of 2013 [4]. Facilitating mobile web interactions is clearly an important focus area, as is enforcing web security.

In particular, we are looking at one aspect of secure web interactions: the usability of captchas on smartphones. *Captchas* are challenge-response tests used as a security mechanism on the web to distinguish human users from automated programs in an effort to reduce abuse of web resources. Captchas are not only deployed in web sites; applications such as Snapchat are including captchas as part of their interface [5]. While extensive research is available on captchas for traditional desktop computing [6], [7], [8], [9], work on captchas for mobile devices is incipient and limited despite the popularity of mobile web browsing. Since most developers and web admins deploy existing captchas, our previous work on the usability of captchas on smartphones included a heuristic evaluation and preliminary study of four desktop captcha schemes displayed on smartphones and uncovered significant usability issues [10].

Even though virtual mobile keyboards have improved and several types exist, our empirical data shows that current keyboards continue to be a cause of errors and frustrations for users. Swapping between numeric and alphabetic keyboards is a source of mistakes while solving challenges, for example. In this paper, we investigate our own customizations to existing captcha schemes specifically intended for smartphone usage and compare the usability of these to the original designs. In total, we compare the usability of nine captcha schemes on smartphones. The evaluated captcha schemes include prototype implementations of schemes which offer an API, commercial captchas, and academic captchas. The security of some of these schemes has been explored in previous research [11], [12], [13]. Our goal is to explore whether adapting input mechanisms to solve captcha challenges helps usability on smartphones, and to evaluate the usability of the modified schemes intended to be more suitable for smartphones. Five existing schemes (mobile and non-mobile), are evaluated on smartphones; and the four new prototypes are similarly evaluated in a user study where users experience the captchas on smartphones. Besides results specific to these schemes, we provide ten generic recommendations for the implementation of captchas on smartphones. Given the exploratory nature of the study, and its design (each participant tested only 4 out of 9 schemes), we simply highlight the positives and negatives of the different schemes rather than provide an statistical analysis to highlight which scheme performs best.

The remainder of the paper is organized as follows: Section II introduces and describes previous and related work. In Section III we describe the evaluated schemes. In Sections IV and V, we outline and describe our methodology, procedure, and data collection for the user study, and present the results. We discuss lessons learnt as well as recommendations in Section VI. Lastly, we offer some discussion and concluding remarks in Sections VII and VIII.

II. Related Work

Captchas can be categorised according to the type of cognitive challenge presented. Character-recognition (CR)

USEC '15, 8 February 2015, San Diego, CA, USA
Copyright 2015 Internet Society, ISBN 1-891562-40-1
http://dx.doi.org/10.14722/usec.2015.23006

captchas involve static images of distorted characters; Audio captchas (AUD) use words or spoken characters as the challenge; Image-recognition captchas (IR) involve classification or recognition of images or objects other than characters; Cognitive-based captchas (COG) include puzzles, questions, and other challenges related to the semantics of images or language constructs. For both CR and IR, we further subdivide then into dynamic subclasses. That is, the CR-dynamic class encompasses dynamic movement of text as the challenge and the IR-dynamic class uses moving objects as the challenge. These two can be grouped as a cross-class category: moving-image object recognition captchas (MIOR) [13].

The usability of captchas for traditional desktop computing has been previously explored. Early work studying the friction between distortions and the ability of humans to solve challenges is presented by Chellapilla *et al.* [14]. Yan and El Ahmad [7] analyse three captcha classes: CR, IR, and Audio. They look at captchas in terms of distortion, content and presentation. Recently, Bursztein *et al.* analyse how the content of the challenge affects the accuracy and perception of Google's text-based captchas [8]. In previous work, Bursztein *et al.* assessed the accuracy of humans solving text-based and audio captchas [15].

While the majority of usability analysis has been done for captchas on desktops, attention to evaluations of captchas for mobile device usage has been limited. Wismer *et al.* [16] explore voice and touch input of existing captchas on mobile devices. Their evaluation focuses on voice and touch input using Apple's iPad and they found significant problems. Reynaga and Chiasson [10] present an exploratory analysis based on a user study and a heuristic evaluation of captchas on smartphones. Although not designed for mobile devices, Gossweiler *et al.* [17] present an IR captcha scheme that could be adapted for mobile usage. Their scheme consists of rotating an image to its upright / natural position with a slider. They suggest that the mobile version would allow direct image rotation with finger gestures. Specific to mobile devices, Chow *et al.* [18] introduce the idea of presenting several CR captchas in a grid of clickable captchas. The answer is input by using the phone's (NOKIA 5200) keyboard and selecting the grid elements which satisfy the challenge. Shirali-Shahreza *et al.* [19] explore speech as input mechanism for CR challenges and AUD challenges. Lin *et al.* [20] propose two mobile specific captcha schemes. The first is an IR scheme called "captcha zoo". It requires users to choose certain 3D target animals from a set of containing two types of animals. This approach is similar to Asirra [21]. The second scheme is a four-character CR challenge with a custom eight-icon keyboard displaying characters from the challenge plus four additional characters. On the commercial side, NuCaptcha [22] offers a mobile version of their MIOR desktop captcha.

III. Captcha Schemes

We selected 5 existing schemes, mobile-friendly and desktop, representing each of the main captcha categories: CR, IR, and MIOR; including two mobile-friendly schemes. The target schemes are summarized below and depicted in Figure 1.

1) *reCaptcha* [23] is widely deployed on the Internet; this CR challenge consists of recognizing and typing two words or numbers.
2) *NuCaptcha* [22] is a commercial MIOR scheme consisting of either reading alphanumeric characters that overlap as they swing independently left to right (statically pinned at the centre of each letter), or reading a uniquely colored code word in a phrase that loops endlessly in the captcha window.
3) *Asirra* [21] is a research IR captcha that was available[1] for deployment; the challenge consisted of selecting images of cats from a grid of 12 images containing dogs and cats. The image database comes from a pet adoption service called Petfinder [24]. These images are pre-labeled by the person uploading each pet's image.
4) *Picatcha* [25] is a commercial mobile-friendly IR captcha. This captcha shows eight images from which the user has to select a specific subset. For example, a user might be asked to select icons of horses, monitors, or hands. The number of correct target images varies from challenge to challenge.
5) *Emerging* [13] is an academic proposal which addresses security flaws found in NuCaptcha. The challenge consists of recognizing three alphanumeric characters. The three characters move on a wave across a canvas in an endlessly loop. Both the challenge characters and the background consist of moving black and white pixels. The movement renders visible the characters, but observing any one frame does not reveal the characters from among the noise.

A. Mobile prototype schemes

We also designed mobile adaptations of some of the above five schemes with the intention of simplifying the user interaction on mobile devices. Examples of these are also available in Figure 1, annotated with "(prototype)".

For these four schemes, we eliminated the use of the standard keyboard as an input mechanism, incorporated touch gestures where possible, and tried to minimize occlusion of the challenge. Our prototype schemes are described as follows.

1) Gesture reCaptcha: This scheme is a CR captcha which employs reCaptcha's API as the source of its challenges. The challenge is displayed without modification, but the input mechanism is altered. A drawing canvas is included beneath the challenge. Users input each character individually by drawing each character on the canvas. The drawing is then recognized and the interpreted character is added to the input text field.

2) Gesture Emerging: Similarly to Gesture reCaptcha, we adapted Emerging captcha to use a canvas for drawing each of the three characters by using gestures. The character is recognized using handwriting recognition and is automatically entered in the text input field.

3) Asirra Slide: Using Asirra's publicly-available API, we modified the user interface of the challenge. Rather than showing a grid of images, we updated Asirra to display each of the images individually in a carousel. Participants could slide the images back and forth, and select the cat images with a long press. While the underlying task remained the same, the

[1] The service was closed permanently in October 2014.

modified user interface allowed for larger images that fit on the mobile screen without the need for zooming and provided a larger area for clicking on items.

4) reCaptcha Buttons: We created another variant of reCaptcha using the reCaptcha's API. The challenge was displayed normally, but the input mechanism was modified. This prototype aimed to enlarge the keyboard buttons by grouping the [a-z0-9] characters in six buttons. Five buttons grouped alphabetic characters and one grouped numeric values. When pressed, each button showed a pop-up menu containing the characters for that group and participants selected characters from the pop-up menu.

IV. User Study

We conducted a usability evaluation comparing these captcha schemes. The goals of the study were to assess the usability of the captcha schemes on smartphones and identify the users' preferences and opinions of the various schemes.

We chose a controlled lab study because besides collecting quantitative performance data, it gave us the opportunity to collect participants' impromptu reactions and comments, and allowed us to interview participants about their experience. This type of information is invaluable in learning *why* certain prototypes are unacceptable or difficult for users and learning which prototypes are deemed most acceptable.

Sessions lasted approximately 45 minutes. Participants were offered a $15 honorarium for their time. This research has been approved by our institution's Research Ethics Board. Participants used their own device for the study. This enabled us to cover a range of scenarios and to ensure that any problems uncovered were not due to unfamiliarity with the device itself. We had a smartphone available, but no participant chose to use it. In the real world, users have a plethora of browsers and smartphone models. Several implementation issues were uncovered by allowing participants to use their own phones.

A. Participants

The 28 participants (16 females, 12 males, mean age 35.17, SD 10) were graduate students (5), undergraduate students (6), professionals (9), research assistants (4) and faculty members (4). None had participated in prior captcha studies. The average self-reported expertise using smartphones was 6.76 out of 10, SD 2.6. The average length of phone ownership was 32 months, SD 32.32. All except three reported having encountered captchas before the study. Nineteen participants browsed the Internet on their smartphones *daily*, two browsed *once a week*, two browsed *several times a week*, two *less than once a week*, and two declined to answer.

B. Study Design

Participants were divided into two groups, each group evaluated four captcha schemes. A within-subjects experimental design was used for each group.

A *challenge* refers to a single captcha puzzle to be solved by the user. Each participant was asked to solve ten challenges per scheme; some participants kept solving challenges after being told they could stop. This process was not automated because we used the demo sites provided by the scheme owners whenever possible and not all sites provided APIs to embed the system. Participants completed an average of 35.64 challenges. In total, 998 challenges were attempted.

The order of presentation for the schemes was counterbalanced according to a 4×4 Latin Square for each group to eliminate biases from learning effects. For each scheme, challenges were randomly selected.

We collected performance data and subjective data. Performance was measured by noting the overall time, number of successes, refresh/skips (as explained later), and errors while answering the challenges. The participants also responded to a demographics questionnaire and a satisfaction survey. The questionnaires were implemented using Limesurvey.[2]

C. Interface Implementation

A simple web-based user interface was designed where users could enter their user name and select a scheme to evaluate. Participants were directed by the experimenter as to which scheme to select next according to their prescribed presentation order.

Where possible, we used the live demo sites offered by the original developers. Visiting the original demo sites allowed testing of the latest version of the schemes and meant that the systems functioned exactly as intended by their developers. When this was not possible, alternative methods were used as described below. Customizations and implementations were made using PHP, HTML5, CSS3, and JavaScript.

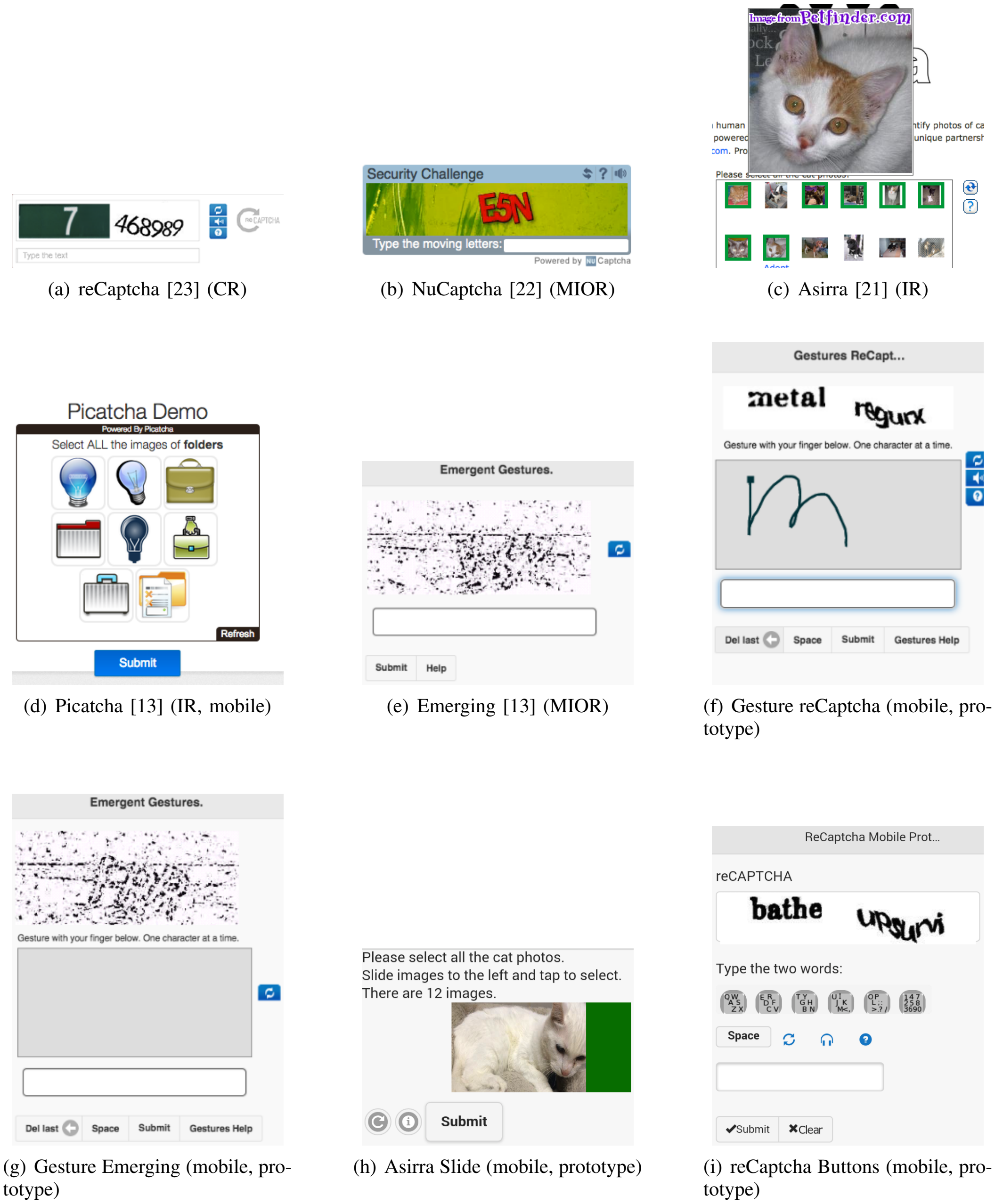
The existing captchas were presented as follows. For NuCaptcha and Picatcha, participants were redirected to the respective demo sites. We created a plain webpage that only included the embedded challenge for reCaptcha and Asirra and we used the available APIs to generate and evaluate the challenges. Emerging captcha is a prototype system. We created a plain webpage embedding the challenge and had a pool of 20 challenges available from which to randomly select.

The Gesture reCaptcha prototype customized the user interface for the reCaptcha API by adding a gesture canvas and translate the user's input to a text string submitted as input to the API. The canvas was 300px×140px. Due to limitations of the available gesture recognizers and results of pilot testing, we used a Wizard of Oz (WoZ) approach to translate user gestures to text input. The researcher would unobtrusively observe the user's drawn characters in real-time, type the character on a remote device, and send it to the user's device. Delay was minimal. Participants thought that the researcher was taking notes relating to observations and were unaware of this aspect of the prototype.

The Gesture Emerging prototype used a similar gesture canvas and WoZ approach to capture user's drawn characters and translating it to text input.

The Asirra Slides prototype provided a custom user interface using the carousel slide component. It used the Asirra API in the back-end to serve and evaluate challenges. The slide images were 150px×150px in size.

[2] LimeSurvey: http://www.limesurvey.org/



(a) reCaptcha [23] (CR)

(b) NuCaptcha [22] (MIOR)

(c) Asirra [21] (IR)

(d) Picatcha [13] (IR, mobile)

(e) Emerging [13] (MIOR)

(f) Gesture reCaptcha (mobile, prototype)

(g) Gesture Emerging (mobile, prototype)

(h) Asirra Slide (mobile, prototype)

(i) reCaptcha Buttons (mobile, prototype)

Fig. 1. Screenshots of the evaluated captcha schemes.

Table I summarizes the evaluated schemes. The "Hosting" column refers to the site which participants visited to solve challenges. *API* refers to a webpage on our lab server using the scheme's public API. Schemes in which participants were redirected to another server are noted as *Demo*, while those denoted *Coded* were custom prototypes served from our lab server. The *Mobile* column identifies whether the scheme is specifically designed for smartphone usage. The *Data Collection* column summarizes how we gathered performance data for the scheme. We were able to *Instrument* the webpages so that most of the data collection was accomplished programatically. However, this was not possible for the two schemes denoted with *Video*; for these, we video recorded the interaction and manually extracted the necessary data after the session.

Scheme	No. Participants	Hosting	Mobile	Data Collection
reCaptcha	18	API	✗	Instrumented
NuCaptcha	10	Demo	✓	Video
Asirra	15	API	✗	Instrumented
Picatcha	10	Demo	✓	Video
Emerging	10	Coded	✓	Instrumented
Gesture reCaptcha	12	API	✓	Instrumented
Gesture Emerging	10	Coded	✓	Instrumented
Asirra Slides	15	API	✓	Instrumented
reCaptcha Buttons	5	API	✓	Instrumented

TABLE I. OVERVIEW OF EXPERIMENTAL SETUP.

D. Procedure

Each participant completed a one-on-one session with the experimenter; sessions which included NuCaptcha and

Picatcha schemes were video and audio recorded. The study protocol consisted of the following steps: 1) *Briefing session*. The experimenter explained the goals of the study, detailing the study steps, and asking them to read and sign the consent form. 2) *Demographics questionnaire*. Participants answered an online demographic questionnaire at the end of the first evaluated scheme. 3) *Captcha testing*. Participants visited a host page with links to the captcha schemes from the smartphone. Participants tested four schemes each. 4) *Satisfaction questionnaire*. After completing the challenges for a scheme, participants completed an online satisfaction questionnaire collecting their opinion and satisfaction of the scheme.

E. Data Collection

Three methods were used to collect data: logs, questionnaires and observations. Unless otherwise indicated in Table I each system was instrumented to log users' interactions with the system. We recorded the overall time (receiving, answering submitting, and getting the reply) for each challenge. We also tracked the frequency counts for success, refresh/skip, help button clicks, and errors while answering the challenges. For NuCaptcha and Picatcha, this information was manually extracted from the videos.

V. Results

We now report the result of the user study. Our intention is to explore what worked well and identify areas where usability was problematic. In particular, we did not include inferential statistics. We felt that these would be misleading given that some of the schemes used WoZ.

A. Data Analysis

To be identified as a *Success*, the user's response had to be entirely correct. An *Error* occurred when the user's response did not match the challenge's solution and was indicated as incorrect by the captcha site. A *Skipped* outcome occurred when the participant pressed the "Get Another Challenge" button and was presented with a different challenge. *Overall Time:* The overall time was measured as the time between when the challenge was displayed to when the response was received. This included the time to input the answer, as well as the time it took the form to receive the reply. The total success, error, skip rates and time were calculated based on the average of each participant's averages. Times for skipped challenges were not included since we observed users making the "skip" decision very quickly and this may unfairly skew the results towards shorter mean times. However, we include challenges that resulted in errors because in these cases participants actively tried to solve the challenge.

The Gesture Emerging and Gesture reCaptcha schemes used the Wizard of Oz technique. To allow the experimenter to input the drawn character, an overhead of 3.5 seconds per character was added. This overhead was included in the Overall Time calculations and related statistical analysis.

Our satisfaction questionnaire used 10-point Likert-scales to evaluate agreement or disagreement with the questions (1 - Strongly Disagree, 10 - Strongly Agree).

Scheme	N	Success	Error	Skips	Mean Time (SD) in seconds
reCaptcha	190	91%	9%	0%	25.2 (17.50)
NuCaptcha	155	98%	2%	0%	8.5 (2.92)
Asirra	97	80%	19%	1%	29.2 (9.83)
Picatcha	120	80%	17%	3%	12.3 (4.97)
Emerging	116	98%	2%	0%	22.4 (6.46)
Gesture reCaptcha	102	87%	3%	10%	55.3 (12.49)
Gesture Emerging	115	88%	12%	0%	44.5 (12.65)
Asirra Slides	103	75%	25%	0%	30.6 (12.98)

TABLE II. Summary of performance results. reCaptcha Buttons is not reported since it was eliminated earlier on.

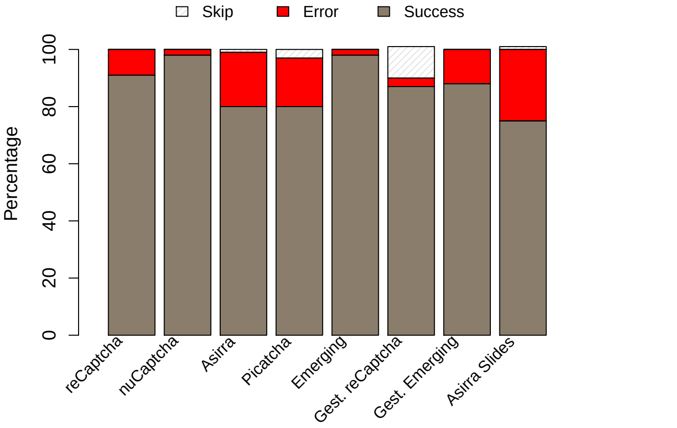


Fig. 2. Percentages of Success, Error and Skips per scheme. reCaptcha Buttons is not reported since it was eliminated earlier on.

Pilot testing: The reCaptcha Buttons prototype offered a modified user interface for reCaptcha meant to enlarge buttons while leaving screen space for the challenge. However, early pilot testing showed that this user interface had significant usability issues. We decided that this was not a viable alternative and we discontinued testing after five users. Therefore, our results focus on the remaining eight schemes. In the remaining trials this scheme was replaced by Gesture reCaptcha.

B. Performance

Table II summarizes the performance outcomes for the evaluated schemes. Success, error and skips are presented as percentages. The time and standard deviation are expressed in seconds.[3] Figure 2 visually summarizes the percentages of success, error and skips for the evaluated the schemes.

Success: As shown in the table, users were most successful at solving the text-based Captchas. At 98%, NuCaptcha and Emerging resulted in the most successful outcomes, but all text-based schemes had success rates of 90% or above. Looking at reCaptcha and Emerging, the variants with virtual keyboards had slightly higher success rates than their gesture-based counterparts. Note that in this paper we are comparing usability aspects, but not security - and therefore, schemes which users both are more accurate on (more correct answers), and prefer better, are not necessarily the best captchas (for example, they might also be much more easily defeated by automated programs than others).

[3] Our first Gesture reCaptcha participant used an open source handwriting recognizer and its accuracy was unacceptable. As a result, the solving times for this participant were removed for these calculations.

Errors: Although success rates were relatively high, we examine the sources of errors to gain insight into where problems occur. Asirra and Asirra Slides resulted in the highest number of errors. For Asirra, the small image size, the quality of images and the need for panning were the likely sources of errors. For Asirra Slides, many participants initially had correct responses but accidentally deselected images by tapping and sliding before submitting. Picatcha had images which often confused participants and they would fail to recognize some of the target images. With Gesture Emerging, one participant submitted wrong answers because his browser (IEMobile/11.0) would not support the gesture canvas. Gesture reCaptcha, NuCaptcha, and Emerging had extremely low error rates.

Skips: Very few skips were observed across all schemes. Five schemes had no skips at all, and the remainder had skip rates under 11%; see Table II. Although it happened rarely, participants were quick to decide if they were skipping a challenge or attempting to solve it. Once committed to solving a challenge, participants followed through and we saw no skipping partway through completion.

Overall Time: Times are also summarized in Table II. NuCaptcha demonstrated the shortest overall solving time. Although Emerging and Gesture Emerging challenges were also three characters in length, these challenges could not be read and solved until the animation started. Unfortunately, loading times for these prototypes were slow[4] and this affected the overall solving times. This could likely be addressed in a production implementation. Similarly, the gesture schemes using WoZ had significantly longer solving times due to manual translation of gesture into text. Although we provide a summary of the times for completeness, we do not consider our timing information to represent a realistic comparison because of these limitations.

C. Participant Opinion

Participants provided feedback through Likert-scale responses, free-form questionnaire responses, and verbal comments during the sessions.

Likert-scale questions: Figure 3 reports the mean Likert-scale responses assessing each scheme. The Likert-scale ranged from 1 = Strongly Disagree (dark red) to 10 = Strongly Agree (dark green). The questions marked with (*) were inverted to avoid bias; as a result the scores for these statements were reversed before calculating the means. The inverted statements are cross-checks on other questions (*i.e.,* q.5 cross-checks q.1, and q.8 cross-checks q.6). A higher score always refers to a positive opinion for the scheme.

The statements are as follows:

1) *Accurate solving:* It was easy to accurately solve the challenges
2) *Understandability:* The challenges were easy to understand
3) *Memorability:* If I didn't use this mechanism for a few weeks, I would still remember how to answer the challenges

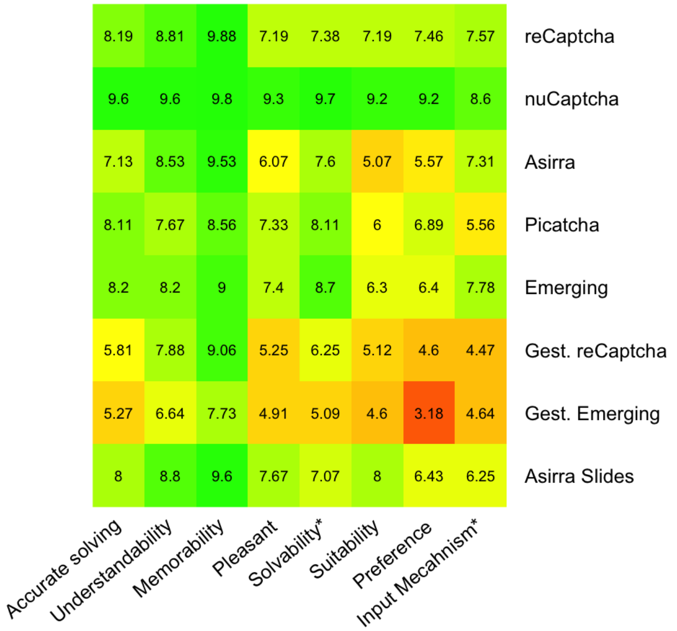


Fig. 3. Mean scores for the Likert-scale questionnaire responses. 1 = most negative, 10 = most positive. * denotes inverted questions

4) *Pleasant:* This captcha mechanism was pleasant to use
5) *Solvability:* I found it hard to solve challenges presented on this captcha scheme*
6) *Suitability:* I found this mechanism well suited for the smartphone
7) *Preference:* On a smartphone, I would prefer using this captcha mechanism compared to other captchas
8) *Input mechanism:* The input mechanism is more prone to mistakes than traditional input mechanisms* (*e.g.,* virtual keyboard)

As shown in the graph, both gesture-based schemes were scored poorly. Gesture Emerging ranked the lowest for preference. From our observations, participants were especially unhappy with challenge loading times and canvas performance. The latter was due to the WoZ experimental design, and the former due to file size. These two factors had a major impact on the participant's perception of the schemes. At the opposite end, participants particularly enjoyed NuCaptcha and rated it highly on all aspects. Both Asirra Slides and Emergent scored positively on all questions, indicating that they were also fairly well-received.

Comments: The perception questionnaire included free-form space for comments. Sample comments from participants are included in Table III. We organize comments according to the main themes uncovered.

Many participant comments related to the scheme's input mechanisms. Many wanted the input mechanism to be tailored specifically for the captcha task. For example, some participants wanted a numeric virtual keyboard to be automatically activated if the challenge was number-based. One participant commented "The keyboard is too big on this" after the keyboard pushed the challenge partially out of the viewport. Others commented on the response time for the gesture-based

[4] In some cases more than 10 seconds while the rest of the schemes the loading time was under 3 seconds.

Scheme	Comments
reCaptcha	"Important on a smartphone to only ask for either digits or text."
NuCaptcha	"The colour helped in terms of making it easier to read."
Asirra	"I prefer seeing all the pictures of cats and dogs at the same time."
Picatcha	"Images could be confusing for some, culturally specific."
Emerging	"The video was slow to load sometimes."
Gesture reCaptcha	"Sometimes difficult to get the intended results" [inconsistency and poor performance of character recognizer]
Gesture Emerging	"It was mainly hard because of the time between accepting the gesture, and the canvas not responding."
Asirra Slides	"The implementation was pretty buggy, but I think the idea was alright."

TABLE III. SAMPLE COMMENTS FROM PARTICIPANTS.

schemes. There was concern over the recognizer and the time it takes to process characters: "it was slower than typing". This is clearly related to the WoZ technique, but participants were unaware of this fact.

Participants identified that preferred schemes and input mechanisms would vary for different user demographics. For example, "If the recognizer would work I think younger people may prefer this method. I think they are not used to keyboards." Although not age-related, we have some evidence of such polarized preferences within our study. Our participants were almost equally divided in their preference regarding the grid layout and the sliding carousel for Asirra. Several participants remarked that some of the schemes may have cultural and language dependencies which would limit broad adoption and usage.

The content of the challenges was highly important in user's perception. For example, participants expressed preference of selecting cats vs dogs. Unsurprisingly, participants also expressed a clear preference for simpler, shorter challenges with little or no distortion. They voiced displeasure with highly distorted and time-consuming challenges. Participants also preferred words over random letters or a mix of alphanumeric characters, and noted that image-quality significantly affected the difficulty of Asirra challenges. The simplest, shortest challenges would clearly be the most usable in the sense of quickest and easiest, but herein we are studying only usability, and security is an important consideration as well (simpler might well in some cases be less secure; certainly for simpler versions or simpler instances from a single scheme, e.g., recognizing 1 letter vs. recognizing 5 letters Regarding bandwidth usage, users did not want to waste their data plan on captchas and quickly got impatient with slowly loading challenges and response times.

D. Experimenter Observations

The experimenter noted verbal comments as participants solved challenges and noted any behavioural observations. We categorised the comments and observations according to the following themes.

Phone handling: Some participants varied the position of the phone depending on the type of challenge. Some placed the smartphone on the table when typing was involved but held the phone in their hands when using gestures, selecting or sliding images. Secondly, although most of the participants kept the phone in portrait mode, some participants rotated their phones from portrait to landscape while attempting to avoid challenge obfuscation. Obfuscation was most problematic with Asirra. We observed as well that pinching and zooming sometimes caused other phone features to come up. For example, while zooming and panning, one participant wanted to drag the page and instead the OS drawer (top bar) was activated, increasing the solving time for that challenge. Several participants had protective cases or screen overlays on their smartphones; this did not seem to interfere with the solving of the challenges.

Software observations: Many errors (challenge responses marked as incorrect) were due to implementation issues and browser incompatibilities. For example, we noted that two phones models by Lenovo and Samsung, using the Mobile Safari/534.30 browser resulted in a second line drawn outside the canvas in response to gestures on the drawing canvas; the line was parallel to the drawn gesture. This was obviously a source of confusion for participants in the gesture-based schemes. We did not observe the same problem on any other browsers. Similarly to desktop solving, many participants were not aware that CR answers are generally case insensitive. Some participants were typing each character in capital letters, activating the shift key each time. The help buttons were never used by participants throughout the entire study.

Challenges and schemes: We noticed several scheme-specific issues and discuss our main observations.

The most obvious difficulties were observed with Asirra. Asirra used a pre-tagged public image database for its challenges. The low quality of some images negatively impacted participants' performance and experience; we expect that this would also occur on a desktop. Some participants accidentally clicked up to three times on the "Adopt Me" link. In response, the browser left the current page and opened Petfinder's website. Upon returning to the scheme's challenge page, a new challenge would be presented. Furthermore, a couple of participants were confused about which image to select; they attempted to select the enlarged image rather than the image from the grid.

Asirra Slides resulted in slightly more positive comments than Asirra. Comments included "This is fun" or "Sliding is easier for this type of input, but the keyboard suits me fine". However, participants were ultimately divided on which variant was best. Four participants expressed a preference for seeing all the images at once in a grid and three participants said they liked sliding through the images. The remainder did not voice an opinion.

While using reCaptcha, one participant commented "it's frustrating [referring to the challenge], can I hear it?". Then the participant tried an audio challenge but rather than function as expected, the browser wanted to download an MP3 audio file. This highlights the shortcomings and challenges audio captchas still face on smartphones.

Distraction from main task: While participants generally accepted the need for captchas, they wanted it to be as quick and unobtrusive as possible. One participant noted for Picatcha "I didn't like it, it's like a game, you don't find a purpose, it's a distractor you want to do your main task, not play. You want to send your form". Participants were unhappy with schemes that may annoy or distract the user from their main task. This observation is especially noteworthy for designers of game-based captchas or those choosing what scheme to deploy on websites.

Other distractions were also noted. Asirra and Picatcha could take participants to external sites. Asirra had the “Adopt Me” link under each of the images and Picatcha’s success notification image would redirect participants to another site if it was clicked.

Gesture input: On a few smartphones, the gesture canvas pushed part of the challenge off of the visible portion of the interface. However, this was more problematic for Gesture reCaptcha than Gesture Emerging. Gesture Emerging has a three-letter challenge that allows the participant to observe the challenge and answer it without continuously having to see it.

The gesture interface was new to participants and they needed some time to learn how to use it effectively. Most tried to write the complete answer on the canvas at once even though we told participants to draw the characters individually. Participants voiced their uncertainty: “With the gestures I have to learn how to use it”. It is likely that the recognizer would need to either offer personalized gesture training or the user would need to learn to “properly” gesture each character. We received comments including “easy challenges, but the gestures were tricky”, “easy to gesture” and “I would prefer the gestures if properly recognized”. In spite of their uncertainty, participants did not venture to click on the help button where a table on how to draw characters was available.

E. Summary of Results

We recorded the following performance measures: success rates, errors and overall solving times. Participants were able to solve the majority of challenges in all schemes. As described above, NuCaptcha showed both the most successful outcomes for the user study and was favoured by participants in the questionnaire responses. The Emerging scheme also showed good overall performance. Its loading times were an issue for user satisfaction and performance. Both were text-based schemes.

We tested several different modalities for input mechanisms: virtual keyboard, drawn gestures, carousel slides, or tap-based image selection. Surprisingly given the difficulties with smartphone keyboards, we found that keyboard input was most accurate. NuCaptcha and Emerging both had 98% success rate. It appears that familiarity with the keyboard was enough to overcome known difficulties with typing on small touch screens.

Participants found the gesture input interesting, but performance issues relating to the WoZ implementation led to more negative perceptions than initially expected. In their current state, participants felt that gestures added complexity compared to typing. Users may make errors while typing, but they knew exactly how to resolve the problem when it occurred.

Participant preferred simple and quick schemes such as NuCaptcha and Emerging. For schemes where concerns were noted, participants’ survey responses and experimenter observations revealed valuable feedback regarding implementation and deployment.

VI. Recommendations

In this section we provide a series of improvements and suggestions to adapt and deploy captcha schemes on mobile websites or responsive websites (responsive websites are sites that are designed to offer an optimal viewing experience regardless of the device used) concerned with bots. Our previous work [10] also included some general recommendations. While a few of these overlap, the current study provides more detailed insights across a broader range of schemes. For example, it offers details relevant to distinct input mechanisms. Therefore, the recommendations in this paper are more specific than our previous set. While our recommendations focus on usability aspects, it is important that security considerations [26], [27], [28] are also taken into account before deploying a captcha scheme. Examples of features such as simple distortions, colours, and lines have been shown to offer little additional security in the face of a determined attacker [27], [29], [12]. These features are often used in simplistic schemes failing to increase the security of the scheme.

Recommendations (design of challenges)

1) Design with one-task only focus. Avoid optional features on the captchas (e.g., Asirra’s Adopt Me link). While these may appear desirable for other reasons, they hinder usability on small screens because solving captchas is already more difficult and time-consuming than on a desktop.
2) Use input mechanisms that are cross-platform compatible and that do not interfere with normal operation of the browser. In particular, touch events are often handled differently depending on the browser. For example, the default behaviour of double tapping in some browsers is to trigger a zoom-in, and thus this gesture should not be mapped to another action within the captcha.
3) Make sure errors (skip and help buttons or dialogues) do not completely distract the user from the main task or force the user to restart the main task (*e.g.,* filling a web form).

Recommendations (screen layout)

4) Consider isolating the captcha from the rest of the web form. For example, create a wizard in which solving the challenge is one of the steps to complete the web form; or create a pop-up layer with the captcha in it. This isolation maximizes screen real estate for the captcha and makes it easier to ensure that incorrect challenges do not unnecessarily disturb the main task (*e.g.,* completing a web form).
5) Strive for a minimalist interface. Avoid cluttering the captcha challenge with non-essential buttons and instructions, in particular because the input interface may require a significant portion of the screen.
6) Keep orientation and size of the captcha consistent with the rest of the web form. Users should not need to change the phone orientation or need to zoom/pan in order to solve challenges.

Recommendations (environment of use)

7) Perform adequate cross-platform testing to ensure that all web elements work across all browsers. For example, a canvas web element to capture gestures may behave differently or not work in some browsers, and some commercial audio captchas may be non-functional.
8) Avoid features which may fail in commonly expected environmental conditions (such as noisy environments - bad for audio; bright or dim conditions - bad for low contrast challenges; social interactions - bad for a cognitive demanding challenge; etc.).
9) Minimize bandwidth usage as well as image, animation, game and audio challenge file sizes. Seek design options with acceptably small bandwidth requirements.
10) Captcha designs should require only default browser features. Avoid designs which require non-standard software, fixed-size screens, extra plug-ins or tools on the client mobile device.

VII. Discussion

As part of our investigation of captchas on smartphones, we designed four modified captcha schemes intended to be more usable for this environment. Unfortunately, our studies found that these were not as successful as we had hoped. However, the positive results observed from some of the existing schemes were promising. Both NuCaptcha and Emerging performed well overall in terms of usability. Given that NuCaptcha has been broken [13], perhaps Emergent captcha would serve as a reasonable alternative if implemented in a manner that takes into account our proposed recommendations.

Bursztein *et al.* [8] noted a disconnect between users' preferences and their ability to accurately solve challenges. Our work extends this finding by demonstrating that it is applicable across a wide range of captcha schemes. Although our participants could solve challenges successfully, they did not necessarily like the schemes. For example, Gesture Emerging had similarly high correctness outcomes to NuCaptcha, but Gesture Emerging scored considerably lower on the perception questionnaires. This reinforces the need for considering both correctness and perception when evaluating schemes. The security is also essential to any captcha scheme, but not studied within the context of this paper.

Regardless of the variant tested, participants showed some reluctance to learn new input mechanisms. Most of the participants preferred the now familiar virtual keyboard despite its small size and known inconveniences. Participants felt that they at least knew how to cope with the keyboard when something went wrong; they had not devised such coping mechanisms for the new input mechanisms. We are reluctant to completely disregard gestures as an input mechanism, considering a number of recent applications (Evernote and Google apps, for example) now include gesture recognition for character input because it is more convenient than typing in several scenarios. We believe that it is likely that over time, gesture input may become as familiar as the virtual keyboard and may mature into a viable alternative for captchas.

Given the exploratory nature of the study, we think that it is best to simply highlight the positives and negatives of the different schemes rather than statistically analyse the outcomes of the schemes.

Although gesture-recognition technology is reasonably advanced, web browser implementations are still lacking. Gesture recognition in current mobile browsers have several constraints: browser support for W3C standards varies, there are limitations in HTML specifications, and the availability and implementation of efficient and robust multi-stroke recognizers is lacking. Until some of these constraints are addressed, it will be difficult to use gestures for web-based captchas.

Limitations and Future Work

Although we feel that our study was valuable and offered important insight into the usability of captchas on smartphones, we recognize that our approach had several limitations.

First, our study had a limited sample size and had users try multiple schemes in one session. This had the advantages of experiments being able to closely observe and discuss usability with each participant and allowing users to compare schemes. However, this decreased ecological validity. A more realistic scenario would have had participants complete one captcha as part of web form submission in a natural environment; but we would have sacrificed the amount of data collected and the opportunity to observe and talk to participants.

Using WoZ impacted the performance and perception of the gesture schemes. It would be desirable to do a full implementation with a robust gesture recognizer to more fully assess the usability of gestures as an input mechanism. We pilot tested reCaptcha with the $P Point-Cloud Recognizer [30] but found its accuracy unacceptable. As a result, we discontinued it use and moved to a WoZ user study design.

The two Emerging variants used Gifs to display their challenges. These files were large in size, which in some cases caused slow loading of the challenges. Given that Emerging performed well despite these limitations, we believe that it would be worthwhile to invest in a more robust implementation for further testing.

Several users wanted to write all characters together when responding using gesture input. Clearly this is desirable from a usability perspective, but it would increase the recognizer's complexity; the recognizer would have to deal with segmentation and recognition, not only recognition. Depending on the length of the challenge, inputting all characters at once may unreasonably restrict the size of each gesture to allow for enough room to complete all characters.

We remind the reader that in this paper we are comparing the usability aspects of the evaluated captcha schemes rather than their security. Nonetheless, the security is central to any scheme.

As future work, we plan to explore further alternative input mechanisms for captchas such as smartphone sensors and multi-finger gestures.

VIII. Conclusion

The aim of this work was to explore whether alternative input mechanisms help improve the usability of captchas on smartphones, and to evaluate the usability of the modified schemes intended to be more suitable for smartphones. In total, we compared the usability of nine captcha schemes on smartphones, including four proposed alternatives. The results show that although participants find virtual keyboards prone to errors, they prefer them for solving character recognition challenges over the other input alternatives studied herein. We believe that including the results of the prototypes is valuable to the community even if their usability was lower than we had hoped. We also found that gesture input in web forms requires robust and reliable implementation of recognizers for users to accept it as a viable option for solving captcha challenges. Perhaps unsurprisingly, participants' preferences are dominated by simple, short challenges with little or no distortion. Character recognition challenges were preferred over image recognition challenges. Another key finding was the disconnect between users' preferences and their ability to correctly solve challenges. Thus a sole measure of captcha solving success and failures does not accurately indicate usability. Our studies suggest that although alternative or novel input mechanisms are approached with caution by users, there is room for more research and improvement. This work enabled us to provide a set of recommendations and suggestions to adapt captchas schemes for websites catering to mobile users. We hope that our study provides useful insight to produce effective, usable mobile captchas.

Acknowledgment

Sonia Chiasson holds a Canada Research Chair in Human Oriented Computer Security and Paul van Oorschot holds a Canada Research Chair in Authentication and Computer Security. Both authors acknowledge the Natural Sciences and Engineering Research Council of Canada (NSERC) for funding theirs Chairs and Discovery Grants. The authors also acknowledge funding from NSERC SurfNet.

References

[1] "Mobile majority: U.S. Smartphone Ownership Tops 60%," 2013, http://www.nielsen.com/us/en/newswire/2013/mobile-majority–u-s–smartphone-ownership-tops-60-.html.

[2] J. Brenner, "Pew Internet: Mobile," http://www.pewinternet.org/fact-sheets/mobile-technology-fact-sheet/, 2014.

[3] "The asian mobile consumer decoded," http://www.nielsen.com/us/en/newswire/2013/the-asian-mobile-consumer-decoded0.html, 2013.

[4] "Who is the mobile shopper?" http://www.nielsen.com/us/en/newswire/2013/who-is-the-mobile-shopper-.html, 2013.

[5] Snapchat, https://www.snapchat.com/, 2014, snapchat.

[6] L. von Ahn, M. Blum, N. Hopper, and J. Langford, "CAPTCHA: Using hard AI problems for security," in *EUROCRYPT*, 2003, vol. 2656.

[7] J. Yan and A. S. El Ahmad, "Usability of CAPTCHAs or usability issues in CAPTCHA design," ser. SOUPS '08. New York, NY, USA: ACM, 2008, pp. 44–52.

[8] E. Bursztein, A. Moscicki, C. Fabry, S. Bethard, J. C. Mitchell, and D. Jurafsky, "Easy does it: More usable captchas," ser. CHI '14. ACM, 2014, pp. 2637–2646.

[9] A. El Ahmad, J. Yan, and W. Ng, "Captcha design: colour, usability and security," *Internet Computing, IEEE*, vol. 16, no. 2, pp. 44–51, 2012.

[10] G. Reynaga and S. Chiasson, "The usability of CAPTCHAs on smartphones," in *International Conf. on Security and Cryptography, (SECRYPT 2013)*. SCITEPRESS, August 2013, pp. 427–434.

[11] P. Golle, "Machine learning attacks against the Asirra CAPTCHA," ser. CCS '08. ACM, 2008, pp. 535–542.

[12] E. Bursztein, J. Aigrain, A. Moscicki, and J. C. Mitchell, "The end is nigh: Generic solving of text-based captchas," in *8th USENIX Workshop on Offensive Technologies (WOOT 14)*. USENIX Association, 2014.

[13] Y. Xu, G. Reynaga, S. Chiasson, J.-M. Frahm, F. Monrose, and P. C. van Oorschot, "Security analysis and related usability of motion-based captchas: Decoding codewords in motion," *IEEE TDSC*, vol. 11, no. 5, pp. 480–493, Sept 2014.

[14] K. Chellapilla, K. Larson, P. Simard, and M. Czerwinski, "Building segmentation based human-friendly human interaction proofs (HIPs)," in *Human Interactive Proofs*, ser. Lecture Notes in Computer Science, H. Baird and D. Lopresti, Eds. Springer Berlin / Heidelberg, 2005, vol. 3517, pp. 173–185.

[15] E. Bursztein, S. Bethard, C. Fabry, J. C. Mitchell, and D. Jurafsky, "How good are humans at solving CAPTCHAs? A large scale evaluation." in *IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2010, pp. 399–413.

[16] A. J. Wismer, K. C. Madathil, R. Koikkara, K. A. Juang, and J. S. Greenstein, "Evaluating the usability of captchas on a mobile device with voice and touch input," in *Proc. of the Human Factors and Ergonomics Society Annual Meeting*, vol. 56. SAGE Publications, 2012, pp. 1228–1232.

[17] R. Gossweiler, M. Kamvar, and S. Baluja, "What's up CAPTCHA?: a CAPTCHA based on image orientation," ser. WWW '09. New York, NY, USA: ACM, 2009, pp. 841–850.

[18] R. Chow, P. Golle, M. Jakobsson, L. Wang, and X. Wang, "Making captchas clickable," ser. HotMobile '08. ACM, 2008, pp. 91–94.

[19] S. Shirali-Shahreza, G. Penn, R. Balakrishnan, and Y. Ganjali, "Seesay and hearsay captcha for mobile interaction," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '13. New York, NY, USA: ACM, 2013, pp. 2147–2156.

[20] R. Lin, S.-Y. Huang, G. B. Bell, and Y.-K. Lee, "A new captcha interface design for mobile devices," in *ACSW 2011: Australasian User Interface Conf.*, 2011.

[21] Microsoft. (2012) Asirra (Animal Species Image Recognition for Restricting Access). http://research.microsoft.com/en-us/um/redmond/projects/asirra/. Last Accessed: Jan 2013.

[22] NuCaptcha. (2014) White paper: Nucaptcha and traditional captcha. http://www.nucaptcha.com/resources/whitepapers.

[23] Google, "reCaptcha: Stop Spam, Read Books." http://www.google.com/recaptcha, 2013.

[24] PetFinder, "Petfinder, Pet Adoption," https://www.petfinder.com/, 2014.

[25] Picatcha. (2014) PICATCHA: Image Captcha. http://www.picatcha.com.

[26] E. Bursztein and S. Bethard, "Decaptcha: breaking 75% of eBay audio CAPTCHAs," in *Proc. of the 3rd USENIX Conf. on Offensive Technologies*, ser. WOOT'09, 2009.

[27] J. Yan and A. El Ahmad, "Breaking visual CAPTCHAs with naive pattern recognition algorithms," ser. ACSAC, Dec. 2007, pp. 279–291.

[28] B. B. Zhu, J. Yan, Q. Li, C. Yang, J. Liu, N. Xu, M. Yi, and K. Cai, "Attacks and design of image recognition CAPTCHAs," ser. CCS '10. ACM, 2010, pp. 187–200.

[29] J. Yan and A. S. El Ahmad, "A low-cost attack on a Microsoft Captcha," ser. ACM CCS '08, pp. 543–554.

[30] V. Radu-Daniel, A. Lisa, and W. Jacob O., "$P Point-Cloud Recognizer," http://depts.washington.edu/aimgroup/proj/dollar/pdollar.html, 2014.