Int. J. Human-Computer Studies 92-93 (2016) 44–54

Contents lists available at ScienceDirect

Int. J. Human-Computer Studies

journal homepage: www.elsevier.com/locate/ijhcs

# Leveraging human computation for pure-text Human Interaction Proofs

Kemal Bicakci, Hakan Ezgi Kiziloz*

*TOBB University of Economics and Technology, Ankara, Turkey*

ARTICLE INFO

*Article history:*
Received 15 August 2015
Received in revised form
9 April 2016
Accepted 18 April 2016
Available online 23 April 2016

*Keywords:*
Human Interaction Proof
Human computation
User study
Security
Usability
Accessibility

ABSTRACT

Even though purely text-based Human Interaction Proofs (HIPs) have desirable usability and accessibility attributes; they could not overcome the security problems yet. Given the fact that fully automated techniques to generate pure-text HIPs securely do not exist, we propose leveraging human computation for this purpose. We design and implement a system called SMARTCHA, which involves a security engine to perform automated proactive checks on the security of human-generated HIPs and a module for combining human computation with automation to increase the number of HIP questions. In our work, we employ HIP operators who generate around 22 000 questions in total for SMARTCHA system. With a user study of 372 participants, we evaluate the usability of SMARTCHA system and observe that users find solving pure-text HIPs of SMARTCHA system significantly more enjoyable than solving reCAPTCHA visual HIPs.

## 1. Introduction

Tests for Human Interaction Proof (HIP), which are supposed to be passed by humans easily, but not by computers, have become the de-facto security countermeasure for many Internet applications. Although many different types of HIPs have been proposed so far, new studies investigating the security and usability trade-off are still worthy. For instance, many HIP tests which involve distorted characters may be broken by automated scripts, and generally, responses to broken HIPs have introduced a more stressful and laborious environment for end users.

Yet another problem is accessibility, i.e., visually impaired users cannot pass these tests. Corporations like Google and Yahoo endeavor to solve this problem by introducing audio HIPs which involves the correct understanding and typing of the letters, digits or words recorded intermittently and/or in a noisy environment. Unfortunately, these audio HIPs are shown to be too difficult to solve. In a large-scale study, average solving accuracy was reported as around 35% for Google's scheme (Bursztein et al., 2010).

One method that has the potential to solve usability and accessibility problems is pure-text HIPs,[1] which do not have any graphical elements and can be presented solely as text. These HIPs can be solved by vision-disabled users by the help of software which reads the intended part of the screen by synthesized voice.[2] However, producing pure-text HIPs that provide an acceptable level of security is an unsolved research problem (Godfrey, 2002). For instance, it is quite possible to solve HIPs automatically with basic parsing techniques if they involve simple arithmetic questions like “what is 3+5?”.

The insecurity of pure-text HIPs can be attributed to insufficient number of base questions from which all test questions are generated and which can be categorized into a few types of questions. For instance, if we closely examine a deployed system (i.e., text-CAPTCHA, Tuley, 2006) which has more than 180 million pure-text HIP tests in its database, we see that all tests fall under a few basic types of questions (detailed in Section 2.1). As a result, a small program searching for known patterns in the tests could easily guess the correct answer with a high success rate.

Our key insight on pure-text HIP tests is as follows. The security provided by pure-text HIPs could be improved if they do not involve a small number of exploitable patterns. In order to have this property, we generate the HIP tests semi-automatically using a large base of diverse questions. The questions are produced

* Corresponding author.
*E-mail addresses:* bicakci@etu.edu.tr (K. Bicakci), hakanezgi@etu.edu.tr (H.E. Kiziloz).

[1] In the literature, HIP tests which involve images showing distorted texts are also sometimes referred as text-based. We use the term “pure-text” to differentiate these two very different approaches.

[2] We are aware how challenging using the screen readers can be; we are only trying to ease the burden on the visually impaired. Interested readers may check Dosono et al. (2015) to understand the limitations of screen readers and Bigham and Cavender (2009) to face encountered problems in this domain, both with empirical results.

http://dx.doi.org/10.1016/j.ijhcs.2016.04.007
1071-5819/© 2016 Elsevier Ltd. All rights reserved.

manually by human operators ensuring that the base questions are not derived from one another. To diversify the human computation for the generation of base questions, we get assistance from a crowdsourcing service such as Amazon's Mechanical Turk (Marketplace for Work, 2005). Although human computation was previously employed for solving HIP tests (Bursztein et al., 2010), our work is the first to use it for generating HIP tests rather than solving them. Due to relaxation of the requirement of being “completely automated”, we avoid using the term CAPTCHA in our work. Instead, we coin a new term and call our approach as SMARTCHA (SeMi Automated Reverse Turing test to tell Computer and Human Apart).

The main research contributions in our work are as follows:

- We design and implement an expandable architecture of a security engine to check security of human-generated pure-text HIPs.
- We investigate viability, effectiveness and performance of employing human computation for generation of pure-text HIPs.
- We introduce “semi-automation” as a novel concept to have theoretically infinite number of tests using human-generated HIPs as base questions.
- With a user study, we evaluate the usability[3] of SMARTCHA and compare it with reCAPTCHA visual HIPs (Official Web Page, 2007).

The rest of the paper is organized as follows. Earlier work on accessible and pure-text HIPs is discussed in Section 2. We introduce our solution called SMARTCHA in Section 3. We perform a user study with 372 participants to test the usability of SMARTCHA system against reCAPTCHA visual HIPs. The methodology and the report of the results are given in Section 4. We finish the paper in Section 5 by presenting concluding remarks and possible future plans for SMARTCHA.

## 2. Earlier work

To our knowledge, the concept of pure-text HIPs was first investigated by Godfrey (2002). In his work, users were shown a paragraph of text in which one of the words was replaced with a bogus word. Users were asked to find out which word was changed. In the same study, Godfrey also presented a successful attack to this method achieving a success rate of 39%. In the attack, a trigram model was used to predict the likelihood of existence of a bogus word.

A second pure-text HIP proposal by Godfrey focused on obfuscating sentences using a trigram model. A valid sentence (i.e., Good sentence) and a randomly generated sentence (i.e., Bad sentence) were retrieved from a corpus. Then, a trigram transformation was applied on both sentences for obfuscation. Users were asked to identify the Good sentence. Godfrey states that humans may be able to do better than guessing.

One other study by Bergmair and Katzenbeisser (2004) suggested using word-sense ambiguity for creating pure-text HIPs. In this proposal, a word which has synonyms, that the synonym has multiple meanings itself, is chosen and new sentences are created using the synonyms. Doing this, some sentences still make sense whereas others become meaningless. Users are asked to choose the correct sentence which still makes sense.

In another study by Ximenes et al. (2006), a pure-text HIP test was designed using Knock-Knock jokes. In a usability study, users were prompted with three text messages of Knock-Knock jokes. Only one of them was a real joke and users were asked to find it. Users have to pass the tests two times in order to prove they are human. Only 30% of the users could pass the test.

Yamamoto et al. (2010) proposed a pure-text HIP asking users to distinguish five sentences obtained from a book or a newspaper from ten made-up sentences which are for instance automatically translated from another language. It was stated that sentences should not be taken directly from Internet due to security issues; yet, the authors did not discuss how to increase the number of sentences automatically. The authors also reported that the usability of the system needs improvement.

Chew and Tygar (2005) introduced a CAPTCHA idea based on collaborative filtering which allows us to ask questions that have no absolute answers. In their study, the system is first trained with answers of real users. Then, new users taking the HIP test are asked questions and their answers are analyzed using the training data in order to see if they match with prior human answers. The authors discussed security requirements for input data and the limitations of the system.

It is not yet possible to generate secure pure-text HIPs in a completely automated fashion (Godfrey, 2002; Bergmair and Katzenbeisser, 2004; Ximenes et al., 2006; Yamamoto et al., 2010; Chew and Tygar, 2005). Furthermore, prior work on accessibility of HIP systems, both pure-text and non-pure-text, generally struggle with this problem. Information about data sets of some of the accessible HIP systems is given in Table 1.

### 2.1. A pure-text service: TextCAPTCHA

Despite earlier negative results on their security properties, we have seen that pure-text HIPs represent a modest success in practice as a simple web service (Tuley, 2006). According to its web page, textCAPTCHA system has more than 180 million HIP tests in its database. The system provides around 271.000 HIP tests daily to WordPress as well as many other websites as a free web service. Even though there is no information on how the test questions are generated, it is certain that automatic techniques are used due to the reported number of questions in the database. An example question in textCAPTCHA database can be given as “Arm, bee or elephant: the body part is?” or “The 2nd colour in green, red and house is?”.

The security of textCAPTCHA service can be broken very easily; a fact also acknowledged on its web site (Tuley, 2006). Since all questions are automatically generated using a few question patterns, identifying these patterns and solving the questions accordingly with a small computer program is straightforward. TextCAPTCHABreaker (Anwar, 2011), a Python application developed as an open source project, reportedly solves HIP tests of textCAPTCHA service correctly with an overall success rate of 99.5% (Anwar, 2011).

## 3. Our solution: SMARTCHA

One of the lessons that can be drawn from the security analysis of textCAPTCHA service given in Section 2.1 is that, if all questions are derived from a few base question types, an automated program could easily give correct answers by exploiting the existing patterns. Hence, to improve security, a reasonable strategy could be the generation of questions not having any common pattern. In our work, we do not attempt to implement the aforementioned strategy with a fully automated method. Instead, we suggest getting the benefit of “human computation” for this purpose.

The concept of human computation was introduced in 1838 (Wayland, 1838). In computer science, it was first used by Alan

[3] For our study, we define the term “usability” as the extent to which a HIP can be solved with correctness, efficiency, and user satisfaction.