A Proposal of Human Interactive Proof in the Text Domain*

Pablo Ximenes[1], André dos Santos[2], Marcial Fernandez[1], Joaquim Celestino Jr.[1]

[1]Communications Networks and Information Security Laboratory (LARCES) –
Universidade Estadual do Ceará (UECE) – Fortaleza, CE – BRAZIL

[2]College of Computing – Georgia Institute of Technology – Atlanta, GA – USA.

{pablo,marcial,celestino}@larces.uece.br, andre@cc.gatech.edu

Abstract. *This work describes a proposal for a novel type of Human Interactive Proof based on natural language Artificial Intelligence problems. It is a test to tell humans and computers apart that requires only plain text to be assembled. Although this is a work in progress, this paper shows the real feasibility of constructing such a Human Interactive Proof.*

1. Introduction

Human interactive proofs (HIP) are increasingly present in nowadays computer systems. The idea behind the concept is to make a certain system able to identify whether it is dealing with a human being or an automatic computational procedure, or robot. HIP's are mainly based on the famous "imitation game" proposed by Alan Turing [8]. The imitation game is a simple game in which a human interrogator would have to test two unknown entities aiming to discover their nature concerning humanity. The only available initial data for the test is the fact that one of the entities is a computer program (or robot) and the other is a real person. The interrogator would have to find out which one is each through a series of questions. After a while, if the human interrogator can not tell the difference between one entity answering the questions and a human being, this entity would have passed the test. The "Turing Test" is a test based on the same ideas of those from the "Imitation Game". Passing a Turing Test proves one of two statements: 1) the entity is in fact a real person; 2) computers are able to have sufficient level of intelligence so they can emulate humans. Failing a Turing Test would mean that the entity is in fact a computer system.

Although Artificial Intelligence has evolved a great deal throughout the years, it is common knowledge that computer systems are not yet able to emulate all aspects of human cognition in a way that would be feasible for them to emulate humans. Thus, one can conclude based on this assumption that, conceptually, a "Turing Test" is in fact a technique for telling humans and computers apart.

HIP's are widely used in the Information Security Field of Computer Science as a countermeasure against certain types of malicious activities like computer attacks that depend on repetitive automatic behavior. If every one of the repetitive actions from the attacker would require an HIP, it would be assured that each action is performed by a human being. Thus, the attack itself would suffer from all the limitations related to a

**This research was sponsored by UOL (www.uol.com.br), through its UOL Bolsa Pesquisa program, process number 200503311809*

human being, such as slow requests (slow typing), physiological needs (a human need to sleep), and difficulty in parallelizing an attack (a multitude of people attacking together). These limitations would, in a final analysis, make many attacks impractical. Some examples of attacks that can be stopped by means of HIP techniques are mass electronic mailing (SPAM) [1], dictionary attacks [5], online poll frauds [1], and automatic registering in internet (especially web based) services [1].

Although HIP is a powerful technique for information security, every HIP currently implemented depends on some advanced capability of the computing device presenting the proof. Examples of such requirements are powerful and high definition color graphics and audio. These capabilities often are not present on a great number of computing devices such as cellular phones, smart card readers, and data collectors. This lack of capability from certain devices limits the applicability of HIP's, preventing the use of this powerful information security tool for applications that require the use of such devices. This work proposes a novel type of HIP that is assembled by the exclusive use of text with no ties to advanced capability requirements of graphics and/or audio. Therefore, the proposed solution will enable computing devices with limited capabilities to use HIP's.

2. CAPTCHA

The Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) [1] is the most used form of HIP. A CAPTCHA is a type of Turing Test, but with characteristics that make it public and completely automated. These characteristics define some basic aspects that must be followed in order for an HIP to be considered a CAPTCHA. Firstly, the test is supposed to be automated with little or no human intervention. This way, it must be conducted solely by a computer program. This differs from the original "Turing Test" which is conducted by a human. Thus, a CAPTCHA is commonly known as a Reverse Turing Test (RTT). Secondly, the test must remain public so its efficacy is not bound to the secrecy of its database or the structure of the test itself. Should an attacker download the entire test's database and acquire all knowledge regarding its structure, he should not be able to develop a computer program capable of passing the test. Thus, the security of a CAPTCHA should remain dependent only on a hard Artificial Intelligence (AI) problem. Hard AI problems are those agreed upon as currently impossible to solve by the AI community [1].

As Hard AI problems used by CAPTCHA systems must also be easy for humans to solve, they are generally related to aspects of human cognition. Examples of such problems for computers are optical character recognition (OCR), audio recognition, natural language processing, and image recognition. The same problems, when related to human cognitive abilities are, respectively, reading text in images, listening to text in audio samples, understanding the meaning of a text excerpt, and understanding and/or identifying an image sample. It is evident that a human being would have no problems solving those problems, as for a computer program this would not be a trivial task.

The most common implementation of CAPTCHA is the reading CAPTCHA, with examples in [1]. This type of CAPTCHA presents an image containing a text sequence. The user is supposed to read the text within the image and type it accordingly. A variation of this CAPTCHA is the listening CAPTCHA [2], where the user must type what can be listened on an audio sample.

3. A Proposal of CAPTCHA in the Text Domain

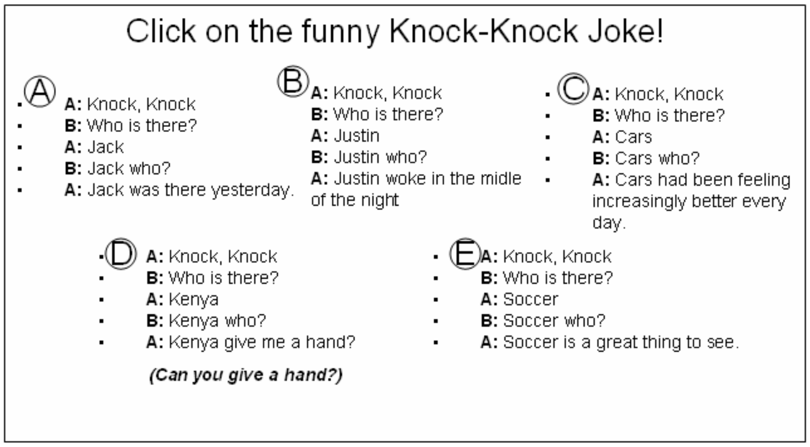
The construction of an HIP (or CAPTCHA) in the text-domain is often cited as an important open problem [1,6,4,9]. To our knowledge, the only formal attempts to construct a CAPTCHA in the text domain are those proposed by [4] and [9]. These proposals are mainly related to AI problems in the computational linguistics field. Despite the fact [4] and [9] show very interesting investigative paths, they lack to comply with one of the main principles of the CAPTCHA paradigm which is the public characteristic of the system (known by cryptography community as the Kerckhoff's principle) as they depend on some level of secrecy for the test to work.

To overcome that, the test generation process must use a one-way AI encoding procedure. One-way AI encoding should be understood as the process of encoding a piece of information as a Hard AI problem, so that this information is intelligible only by humans. For example, the one-way AI encoding in the reading CAPTCHA means creating a picture that shows the letters that humans can read but OCR techniques are inefficient in extracting the text. For a text-only based CAPTCHA this would mean the automatic generation of text excerpts that would transmit a piece of information that only humans could understand.

One very promising investigative approach for the construction of a one-way AI encoding procedure for text-only CAPTCHA's, thus respecting Kerckhoff's principle is "punning". "Punning", or wordplay, is the phenomena where two words that have similar sound present different meanings according to the context. This is subject of several studies by the computational humor community [7, 9]. Those studies have stated that a wordplay in the context of a joke is far more simple for computers to generate than to understand. We believe this gap is sufficient enough to generate wordplay based text excerpts that computers will not "get", but will be easily recognizable by humans. Figure 1 shows a possible example of a CAPTCHA based on Knock-Knock (KK) jokes, a special type of joke found on the English language. A regular KK joke is a dialog between two people that uses wordplay in the punchline and can be summarized using the following structure:
Line$_1$: "Knock, Knock"
Line$_2$: "Who is there?"
Line$_3$: any phrase
Line$_4$: Line$_3$ followed by "who?"
Line$_5$: One or several sentences containing one of the following:
Type$_1$: Line$_3$
Type$_2$: a wordplay on Line$_3$
Type$_3$: a meaningful response to Line$_3$.

While studying the computational aspects regarding KK Jokes, Taylor [9] demonstrated that is "easy" for computers to generate meaningful KK Jokes, whereas their recognition through computational means is far from trivial, thus enforcing our theory.

Our proposal is to use the linguistic aspects behind the phenomena of "punning" in the construction of a strong HIP that respects all principles of the CAPTCHA paradigm including Kerckhoff's principle. We believe that those aspects include different types of linguistic constructs other than humor. We are currently studying the cognitive and computational aspects behind "punning" in the hope to model and formalize a one-way AI encoding procedure for this domain.



Click on the funny Knock-Knock Joke!

A
- **A:** Knock, Knock
- **B:** Who is there?
- **A:** Jack
- **B:** Jack who?
- **A:** Jack was there yesterday.

B
A: Knock, Knock
B: Who is there?
A: Justin
B: Justin who?
A: Justin woke in the midle of the night

C
- **A:** Knock, Knock
- **B:** Who is there?
- **A:** Cars
- **B:** Cars who?
- **A:** Cars had been feeling increasingly better every day.

D
- **A:** Knock, Knock
- **B:** Who is there?
- **A:** Kenya
- **B:** Kenya who?
- **A:** Kenya give me a hand?

(Can you give a hand?)

E
- **A:** Knock, Knock
- **B:** Who is there?
- **A:** Soccer
- **B:** Soccer who?
- **A:** Soccer is a great thing to see.

Figure 1. A possible example of the proposed CAPTCHA system

4. Conclusion

Constructing a CAPTCHA in the text domain is not a trivial task. This paper presented a promising and solid investigative approach towards creating a text only HIP that truly respects all principles regarding the CAPTCHA paradigm. This approach is related to "punning" as a means of distinguishing men and machines. It was specially demonstrated the feasibility of such an HIP with the use of wordplay based jokes (such as Knock-Knock jokes). Although computational humor shows as an interesting investigative path, the cognitive aspects behind "punning" can be found in other types of natural language and linguistics problems, suggesting a variety of possible investigations.

References

[1] Luis von Ahn et al., CAPTCHA: using hard ai problems for security. In Advances in Cryptology, Eurocrypt 2003, vl. 2656 of Springer LNCS, pp. 294–311, May 2003.

[2] Tsz-Yan Chan. Using a text-to-speech synthesizer to generate a reverse turing test. In Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence, p. 226. IEEE Computer Society, 2003.

[4] Philip Brighten Godfrey. Text-based CAPTCHA algorithms. In First Workshop on Human Interactive Proofs, 2002. Unpublished Manuscript.

[5] Beunny Pinkas and Tomas Sander. Securing passwords against dictionary attacks. In Proceedings of the 9th ACM Conference on Computer and Communications Security, pp. 161-170. ACM Press, 2002.

[6] Bartosz Przydatek. On the (im)possibility of a text-only captcha. In First Workshop on Human Interactive Proofs, 2002. Unpublished Abstract.

[7] Graeme Ritchie "Prospects for Computational Humour," *Proceedings of 7th IEEE International Workshop on Robot and Human Communication*, Takamatsu, Japan, pp. 283-291, 1998

[8] Alan M. Turing. Computing machinery and intelligence. Mind, 49:433–460, 1950.

[9] Julia Taylor, Master Thesis, University of Cincinnati, 2004.