algo^{logic}

Algologic Technical Report # 02/2017

CAPTCMA – a simple CAPTCHA-like tool, based on mathematics

S. Parthasarathy

drpartha@gmail.com

Our Ref.: CAPTCMA.tex
Ver. : 201702c

Table of contents

Contents

CAPTCMA – a simple CAPTCHA-like tool, based on mathematics

S. Parthasarathy
drpartha@gmail.com

Abstract

CAPTCHA is a popular tool for preventing various kinds of fraud on the w-w-web. It is usually built into interactive forms used in web pages. We describe below, a simple and modified implementation of CAPTCHA. Here, we use mathematics, to raise the security of CAPTCHA by one level higher up.

1 Preamble

A CAPTCHA [1] is a backronym derived from: *Automated Public Turing test to tell Computers and Humans Apart*. CAPTCHA has become a generic name given to tests used for discouraging various forms of fraud on the Internet. These tools try to distinguish between automatically generated inputs and human generated inputs. Because the test is administered by a computer, in contrast to the standard Turing test that is administered by a human, a CAPTCHA is sometimes described as a reverse Turing test.

A CAPTCHA test involves creating an activity which only humans can perform. The most common activity involves recognising a visual cue. This is followed by a challenge involving the visual cue provided. Some CAPTCHAs have been created which involve recognising an audio signal or tune (e.g. a song clipping or a spoken text). The CAPTCHA test succeeds if the user can solve the challenge correctly, based on the visual cue. Thus, CAPTCHA serves as a two-factor authentication scheme. It uses seamlessly, both the strategies:

1. What you have (the visual cue)
2. What you know (the knowhow to interpret and solve the corresponding challenge)

In the first generation of CAPTCHAs, a CAPTCHA image (of a text string) shows a random string which the user has to recognise and retype, to submit a

form. This is a simple problem (seeing) for humans, but a very hard problem for computers which have to use image recognition of a displayed string which is alienated (distorted and polluted). Subsequent improvements had more sophisticated challenges built around a graphical cue. Some examples would be :

- Identify pictures containing a specific object e.g. pictures containing furniture.
- Counting a specific object in a picture e.g. counting the number of children in a group photo
- Naming a city (or country) based on the picture of a landmark located there e.g. naming the city where Tajmahal is located
- Recognising the language from a given written script/text

An interesting extension to the above idea involves challenges based on mathematical texts and a basic knowledge of mathematics.

Using maths based CAPTCHA is not a new idea. One popular method involves generating two random numbers and challenging the user to perform an arithmetic operation (add, substract, multiply, divide) between them. However, this method is limited in scope, since it is limited to the four fundmental arithmetic operators. Moreover, the operands should be limited to small values (usually two-digit numbers), to make it easily solvable by humans. This paper proposes an extension to this idea.

This extension is called CAPTCMA (*Automated Public Turing test to tell Computers and Mathematicians Apart*). We exploit the fact that image recognition is a faculty unique to humans. In addition, only humans can choose the correct action or principle to apply in a given situation. In CAPTCMA, we improve upon this fundamental difference between humans and machines.

CAPTCMA is a good example of the use of mathematical knowledge in an unusual situation. The challenge involves :

1. Recognising and interpreting mathematical text in a distorted/polluted image
2. Applying some mathematical principles, to answer the challenge

The two-factor authentication mentioned earlier is rendered more difficult by CAPTCMA.

- Mechanical image recognition of mathematical texts poses many major challenges:
  - Mathematics follows its own typographic conventions and rules
  - Mathematics uses several abstract symbols
  - Mathematical texts use letters from other languages (esp. Greek)
  - Mathematical texts may contain embedded conventional texts
  - Punctuation marks (e.g. ! * . , / | > <)in mathematical texts have special meaning.
  - The text displayed may be geometrically distorted and interspersed with visual pollutants.
- The interpretation rules (challenge) could belong to any of the various branches of mathematics (e.g. arithmetic, algebra, geometry, trigonometry, calculus, statistics .....). It would be very difficult for a machine to recognise and choose the right branch of mathematics and the right rule inside that branch.

In addition to its use for authentication, CAPTCMAs can also be used as a pictorial quiz or as a teaching aid for mathematics.

2 Design principles of CAPTCMA

CAPTCMA is a particular implementation of CAPTCHA. FOSS tools are eminently suitable for creating CAPTCMA.

Each CAPTCMA test is a triplet : <a visual cue >, <a challenge >, <a correct solution to the challenge >. The entire CAPTCMA can therefore be stored as a 2 dimensioned array, each row of which, corresponds to a CAPTCMA test. The array may contain any number of rows (CAPTCMA tests). The tests are chosen randomly from this array. This approach is however considered to be insecure, since the array is visible to everybody.

They can also be stored as a database. Each record of the database will define a CAPTCMA test and will consist of the above three fields. These records can be integrated into HTML forms using PHP and other server-side

scripting tools. Access to the database can be restricted by using password or access keys. This makes the CAPTMA more secure than the array based approach.

However, the most secure option would be to generate the CAPTCMA images and challenges, dynamically. This does not involve storing anything, but is more cumbersome to program. Creating a visual image dynamically, and adding visual distorsions to it is not very easy.

Using mathematics for the visual cues is an extremely rich choice, since it can be visualised by the creator himself. It is easy to create the visual cues for each CAPTCMA test, using tools like GIMP, LaTeX, MathJax etc. The CAPTCMA developer can generate a huge number of visual cues, and the corresponding challenges, using his/her own creativity, imagination, and innovation. The difficulty level of the corresponding challenges can be chosen by the CAPTCMA designer himself/herself. However, it should be remembered that CAPTCMA is not a test of the mathematical skills or knowledge of a person.

A few simple precautions will make it difficult for autmatically/mechanically solving the CAPTCMA test:

- The visual cue should be simple and compact. It should be possible to observe the entire cue in a single eye-shot. Visual distorsions and graphical pollutants should make it difficult for automatic image recognition, but should not render the image illegible to a human.
- The challenge itself should be worded briefly, in unambiguous terms and without using any specialised terminology.
- Care should be taken to see that the wording of the challenge is different for different cues.
- The solution should be unique. It should be possible to type down the response, on a conventional keyboard, or be chosen by a mouse-click.
- Part of the challenge should be displayed in the cue, graphically. In other words the challenge cannot be interpreted without first recognising the cue.
- The solution field itself is never displayed. It is used invisibly for verifying if the response entered by the user is correct or not.

- It would not be a good idea to let the solution be visible anywhere in the visul cue or challenge. For instance, a bad choice of the challenge would be:

  Who is the odd man out in the sequence : 12 6 8 12 7 42 66.

  In the above example, it would be easy to defeat the above challenge by a trial-and-error approach or exhaustive enumeration of all the numbers.

3 CAPTCMA Examples

3.1 Example – 1

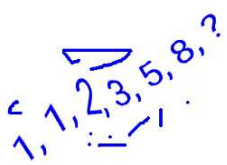


Figure 1: Example 1

The challenge for this example is : *What is the last number in this sequence ?*

In fact, it would be very difficult for a machine to locate and recognise the sequence in this picture. It would be much harder to recognise that it is part of a Fibonacci sequence, since there are several similar integer sequences [2]. By comparison, any human (with some knowledge of mathematics) can spot the answer in no time.

The user is considered to have succeeded when he/she enters the correct answer (13) in the appropriate box in the form where this CAPTCMA challenge is displayed.

3.2 Example – 2

This challenge is based on a trigonometrical identity. A machine may find it difficult to recognise the distorted image of a text like this, and to infer that it is a mathematical expression. It would be much harder (for the machine) to recognise (and solve) the hidden riddle.

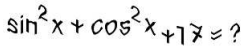


Figure 2: Example 2

By comparison, this same challenge would be a trivial task for any mathematics-aware human being.

The above image could be rendered more difficult to iterpret mechanically. An example is shown below. In this case, we use two graphic methods, super-imposed one-over-the-other, to create a visually polluted the text:

- Distort the basic text
- Add background noise

Care should be taken so that the polluted image does not become illegible to a human.

3.3 Example – 3

A commonly used challenge involves two numbers and an arithmetic operation over these two numbers.

Instead of displaying the above challenge, in a text form as :

$$1472 - 907 = ?$$

we will display it graphically after some graphical distorsions, like in this image on the left. Conventional text display can be recognised by OCR tools, whereas the graphical display is much harder to recognise and interpret (for a machine). Notice that the arithmetic operation to be done is spelt out and graphically displayed instead of as a simple textual character (-).

However, challenges which involve the four basic arithmetic operations, are limited in scope. They can be broken by an exhaustive search over all the four operations. We explain below, a strategy to overcome this limitation.

3.4 More options

We can extend the above idea, to make it harder to mechanically interpret CAPTCMA images. We can try any combination of the following strategies.

3.4.1 Extended arithmetic operations

We can try going beyond the four basic arithmetic operators. We can use many (simple) binary arithmetic operations, to discourage automatic/mechanical interpretation of the CAPTCMA tests. The possibilites are

endless.

A few possible options are :

- gcd (a,b)
- lcm (p,q)
- x mod y
- average(x,y)

The operands must be small (one or two digits) so that the user can solve the operation with just a simple eye-shot.

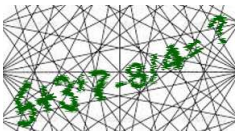
3.4.2 Precedence rules

We can extend this to expressions which involve more complex rules. Here is a simple arithmetic expression which any schoolboy can solve:

$$5 + 3 * 7 - 8/4 = ?$$

provided he knows the rules of arithmetic precedence.

If we present the above expression as an image (with a noisy background), we will have to first extract the above expression from the image, recognise it as an arithmetic expression, parse the expression respecting precedence rules, and arrive at the solution. This would indeed be a major effort to automate.



3.4.3 Cascaded operations

Arithmetic over cascaded operations and nested parantheses is generally tricky. Here is an eaxmple:

$$((-7) * (-7) + ((-7) * (7)) * 4 + 5$$

Warning : All the above strategies may be combined in any sequence, to achieve any level of difficulty for automatic recognition and interpretation. However, since this is not a test of the mathematical skills of the user, it should be used with care. This type of challenge may not be suitable for a layman audience.

4 Wrapup

The above article provides outlines of an unusual usage of mathematics, nicknamed CAPTCMA, to provide a simple but powerful security extensions of the CAPTCHA principle using the following potent facilites:

- Mathematics offers limitless possibilities for creating CAPTCMA challenges.
- Many FOSS tools exist for implementing CAPTCMA.
- These strategies can be combined and used together.

A prototype implementation of CAPTCMA can be found in [3] Several improvements can be made to the prototype implementation. Currently the entire captcha tests are hard coded into a single array in the PHP code. One can transfer this information to an external database, and protect it using access keys or passwords.

If you find this article useful, please send a note to drpartha@gmail.com. Comments, suggestions and remarks are always welcome, as long as they are constructive. This article, as well as other similar articles can be downloaded from [4]

5 About the author

S. Parthasarathy (Dr. Partha) brings with him, a very rich experience (since 1980) in the software industry. He usually teaches discrete mathematics, to students of Computer Science, in India. He is an aggressive supporter of the Free Libre Open Source Software (FOSS) movement, and is a regular contributor to the international effort on Linux/FOSS. He also promotes the use of LaTeX for all academic and technical publications. He was decorated by IEEE (USA) for designing excellent websites for IEEE. His Consulting company provides assistance in web design and maintenance.

You can contact him, by e-mail, on : `drpartha@gmail.com` . For more details about Dr. Partha, please visit : `http://drpartha.org.in/drpartha.htm`

References

[1] Wikipedia, CAPTCHA,
`https://en.wikipedia.org/wiki/CAPTCHA`

[2] OEIS, The On-Line Encyclopedia of Integer Sequences (OEIS)
`https://oeis.org/`

[3] S. Parthasarathy, CAPTMA,
`http://drpartha.org.in/phptest/mycaptcha.php`

[4] S. Parthasarathy,
http://drpartha.org.in/publications/downloadables.htm

* * * *