SEMAGE: A New Image-based Two-Factor CAPTCHA

Shardul Vikram
Texas A&M University
College Station, Texas
shardul.vikram@tamu.edu

Yinan Fan
Texas A&M University
College Station, Texas
yinanf@neo.tamu.edu

Guofei Gu
Texas A&M University
College Station, Texas
guofei@cse.tamu.edu

ABSTRACT

We present SEMAGE (**SE**mantically **MA**tching ima**GE**s), a new image-based CAPTCHA that capitalizes on the human ability to define and comprehend image content and to establish *semantic relationships* between them. A SEMAGE challenge asks a user to select *semantically related* images from a given image set. SEMAGE has a two-factor design where in order to pass a challenge the user is required to figure out the content of each image and then understand and identify semantic relationship between a subset of them. Most of the current state-of-the-art image-based systems like Assira [20] only require the user to solve the first level, i.e., image recognition. Utilizing the semantic correlation between images to create more secure and user-friendly challenges makes SEMAGE novel. SEMAGE does not suffer from limitations of traditional image-based approaches such as lacking customization and adaptability. SEMAGE unlike the current text-based systems is also very user-friendly with a high fun factor. These features make it very attractive to web service providers. In addition, SEMAGE is language independent and highly flexible for customizations (both in terms of security and usability levels). SEMAGE is also mobile devices friendly as it does not require the user to type anything. We conduct a first-of-its-kind large-scale user study involving 174 users to gauge and compare accuracy and usability of SEMAGE with existing state-of-the-art CAPTCHA systems like reCAPTCHA (text-based) [6] and Asirra (image-based) [20]. The user study further reinstates our points and shows that users achieve high accuracy using our system and consider our system to be fun and easy.

Categories and Subject Descriptors

K.6.5 [**[Computing Milieux**]: Management of Computing and Information Systems - Security and Protection

General Terms

Security

Keywords

CAPTCHA, Semantic-based Interactional Proofs,Two-factor CAPTCHA

1. INTRODUCTION

New web applications and services emerge everyday in all areas of life. More people are getting used to having online services, such as email services, forums, and specialized interest groups. For the service providers, one important aspect to consider is to make sure that the services and resources are allocated to the targeted customers. Malicious usage of services, such as using a 'bot' to register legal accounts [9], can take up valuable resources and distribute malicious information thereafter. Thus it is important for the service provider to be able to distinguish a bot from human users, and CAPTCHA systems are widely used for this purpose.

CAPTCHA stands for "Completely Automated Public Tests to tell Computers and Humans Apart" [29, 28, 27, 15, 9]. The idea is to introduce a difficult AI problem so that either the purpose of distinguishing bots and legitimate users is served, or that an AI breakthrough is achieved [29, 28]. The robustness of CAPTCHA systems relies not on the secrecy of the database, but on the intrinsic difficulty of the problem. The difficulty of solving a CAPTCHA problem for a bot and for a human often increases in similar curves. As CAPTCHA systems are rarely stand-alone and are often integrated as an auxiliary part for applications such as online registration, it is unrealistic to ask for the user's concentration for longer than a few seconds. Hence a complicated challenge requiring the humans to devote more time would make it unrealistic to be deployed on real world systems.

Identifying distorted letters, answering questions based on images are a few techniques that are in use to defeat bots, with the former being the most widespread. However with the increasing advances in the field of computer vision, bots have been known to break text CAPTCHAs using techniques such as OCR (Optical Character Recognition) and segmentation [30, 26, 16, 2, 19]. Increasing the complexity of the text-based systems by introducing more noise and distortion to make the challenge difficult for bots also makes them less user friendly and less usable to normal users.

Image-based systems were then proposed to increase the usability of CAPTCHA systems [20, 3, 17, 23, 18, 7, 25, 32]. However, many current state-of-the-art image-based systems such as Asirra [20] suffer from the lack of flexibility and adaptability. Assira challenges focus on image recognition only, requiring the user to identify all cats among a series of images of cats and dogs. Specialized attacks using machine learning techniques have achieved a high rate of success against systems like Asirra, as shown by Golle [22]. Moreover the inherent choice presented to the bot is always binary (an image is either a cat or a dog), making it more susceptible to template fitting attacks, which will be further discussed in Section 4.2. We propose SEMAGE, a novel image-based CAPTCHA system, which has a two-factor model requiring the user to recognize the image and identify images that share a semantic relationship.

ACSAC '11 Dec. 5-9, 2011, Orlando, Florida USA

The introduction of semantic correlation makes SEMAGE more robust from similar machine learning attacks. Other image-based systems like ESP-PIX [3] and SQ-PIX [7] are language dependent and have usability concerns. We survey more CAPTCHA systems and their limitations in Section 2.

In this paper, we propose SEMAGE (**Se**mantically **Ma**tching Ima**ge**s), a two-factor CAPTCHA system. In SEMAGE, we present the user with a set of candidate images, out of which a subset of them would be semantically related. The challenge for the user is to identify the semantically related images based on the context defined by the system. Note that the images in the correct set need not be images of the same object, a set of semantically related images may be images of entities with different physical attributes but sharing the same meaning in the defined context. Consider for example the user being asked to identify similar images with the context being similar images should have the same origin, the candidate set could contain images such as a wooden log, a wooden chair, a matchstick, an electronic item, an animal, and a human, with the chair, matchstick and log being the similar set.

The challenge in solving a SEMAGE CAPTCHA system is twofold: (1) a user has to figure out the content of the individual images, i.e., image recognition, (2) and understand the semantic relationships between them and correctly identify the matching images. This challenge solving ability comes naturally to humans as humans automatically employ their cognitive ability and common sense without even realizing the inherent difficulty of the task. The same challenge for a bot would require both understanding images and identifying relationships between them, constituting a difficult AI problem. Our two-factor design aims at increasing the difficulty level for a bot and improving usability for humans, without sacrificing the robustness of the system.

What makes SEMAGE novel is the idea of presenting the user with a two-factor challenge of "identifying images with similar semantics under the given context". The idea of choosing images exhibiting semantic similarity has a much broader scope than simple selection of images of animals of the same species (cats in the case of Assira). This feature differentiates SEMAGE from other state-of-the-art image-based CAPTCHAs that only require the user to solve the first level, which is image recognition. Computers are hard to comprehend and identify the semantic content of an image, making SEMAGE very robust to bots. We present and discuss what semantic similarity entails in Section 3.

We also implement one very simplified sample instance of SEMAGE using real and cartoon images of animals. The relationship query asks the user to pick up images (real and cartoon) of the same species. This particular implementation has two immediate benefits: (1) Adds fun factor for the user without adding burden on the recognition part since a human can easily make a connection between a real image of an animal and a cartoon image; (2) Scales up the difficulty level for bots as the cartoon images need not even resemble the real physical attributes of the animal. Moreover, SEMAGE provides an easy-to-operate interface to indicate correct answers making it an ideal choice for touch-based systems and smart-phones where typing is more difficult. A sample simplified SEMAGE challenge is shown in Figure 1 which illustrates the idea. A human can easily identify the images marked in a circle as similar but a bot would not be able to relate the real and cartoon images due to difference in shape and texture. Note that this is just one way of creating a SEMAGE challenge. Any other *semantic relationship* can be used as the identifying factor apart from our particular simplified implementation.

The main contributions of this paper are as follows:

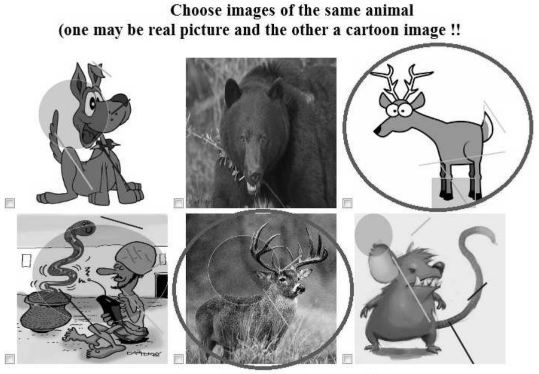

Choose images of the same animal
(one may be real picture and the other a cartoon image !!)

Figure 1: Sample SEMAGE challenge; the encircled images are similar.

- We propose SEMAGE, a new image-based two-factor CAPTCHA that has several unique features. The design of a SEMAGE allows easy tuning of the security level and usability level depending on the nature and popularity of the website. The images of the SEMAGE challenges can vary to suit the needs of different websites. In fact in most cases given a labeled database it is very easy and intuitive to come up with a definition "semantic relationship" and SEMAGE implementation. We also provide an in-depth security analysis and show how SEMAGE is more robust to many attacks than existing systems.
- We further conduct a large-scale user study with 174 participants using a simple sample SEMAGE implementation. We compare our system with state-of-the-art text-based CAPTCHA system reCAPTCHA [6] and image-based system Asirra [20] on the metrics of usability and fun factor. As discussed in details in Section 5, results show that our system is easy to use and participants reported a high level of 'fun' factor.

2. BACKGROUND

CAPTCHA systems, text-based in particular, have been in widespread use as the first line of defense against bots on the web. Recently, with the improvements in computer vision technology, text-based systems have become susceptible to bot attacks with a high success rate [30, 26, 16, 2, 19, 13]. Hence a lot of work has proposed alternate CAPTCHA systems such as image-based [20, 3, 17, 23, 18, 7, 25, 32] and audio-based systems [14, 10, 1, 21].

2.1 Text-based Systems

Generally, text-based CAPTCHA systems ask the user to discern letters or numbers. GIMPY is one classic example [4]. Attacks on text-based systems mostly employ OCR (optical character recognition) algorithms. These algorithms first segment the images into small blocks each containing only one letter, and use pattern recognition algorithms to match the letters in each block to standard letter template features [30, 26, 16]. The later task is considered a well solved AI problem. In counter-attack to these algorithms, text-based CAPTCHA systems employ the following techniques to enhance robustness [15, 19]:

- Adding noises in the form of scattered lines and dots to the background to counter-attack segmentation algorithms.
- Characters are connected or overlapped so that attacking algorithms cannot correctly segment image into correct blocks.
- Characters are twisted to increase difficulty in character recog-

nition.

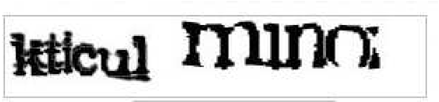


Figure 2: A text-based CAPTCHA example

However, all the above techniques increase the difficulty level for humans too. Connecting characters together makes the task harder for humans. For example, when the character ‘r’ and ‘n’ are connected, it looks like the character ‘m’. Twisted characters not only gnaw on user’s nerves, but also are sometimes impossible to identify correctly. Figure 2 shows one such difficult-to-solve text-based challenge.

Text-base system faces one inevitable situation: humans find the CAPTCHA challenge unpleasant as CAPTCHA gets more complicated. This is probably why popular websites such as MSN hotmail opted for simple and clean CAPTCHA , which could be attacked with a success rate over 80% [30]. Some systems use distinctive color for each character and add colored background using non-text colors, both of these additions can be easily removed by an automated program, which add no more difficulty for the bot [31].

Popular systems such as ‘reCAPTCHA’ [6] use dictionary words that are labeled as unrecognizable by real automatic OCR programs running on real tasks of digitizing books, and evaluate correctness by other user’s input. However, reCAPTCHA also suffers from decreased usability and user satisfaction due to the high distortion and noise in the challenge.

2.2 Audio-based Systems

Audio-based CAPTCHA systems [1, 14, 10, 21] remedy the fact that visual CAPTCHA systems are not accessible to visually-impaired people. In a typical audio CAPTCHA system, letters or digits are presented in randomly spaced intervals, in the form of audio pronunciation. To make the test more robust against bots, background noises are added to the audio files. These systems are highly dependent on the audio hardware and the user only has a certain small amount of time to identify each character. In some sense, audio CAPTCHA systems can be considered as the acoustic version of text-based systems. Although the visual cues are replaced with acoustic cues and the algorithms vary, the underlying idea of attacking is the same - features are extracted and classified to recognize the letters [12]. The difficulty curve for bot and humans are similar. Thus audio CAPTCHA systems provided neither more user-friendly interface for visually accessible users, nor more robustness against bots [11].

2.3 Image-based CAPTCHA systems

Image-based CAPTCHA systems emerged in efforts to replace text-based CAPTCHA systems which were growing more complex for humans to solve easily. Security is not the only concern in a good CAPTCHA design. All CAPTCHA systems are a form of HIP (Human Interactional Proofs) and require the users involvement. This also makes usability a key issue in CAPTCHA design. Tygar et. al. [17] propose the following requirements for a good CAPTCHA system:

- The task should be easy for humans.
- The task should be difficult for computer algorithms.
- The database should be easy to implement and evaluate.

The general basis of image-based CAPTCHA is that images contain more information than texts. It is intuitive for human to catch visual cues but hard for AI algorithms to do visual recognition. ESP-PIX [3] presents a set of images and asks the user to choose a word from a list of words that describes all images. This approach suffers from two drawbacks, i.e., it still depends on text to convey meaning and since all words are written in English, and the user’s success depends on his/her proficiency in English (or any other particular language it migrates to). It is not only language dependent but also hard to operate; a user needs to scan through the whole list of words to find the most proper answer. SQ-PIX [7] also presents user with an image set, but asks the user to select an image of a given object name, and also trace the object in the image. This is also language dependent and the act of tracing around an object with a pointer operated from a hand-held device like a mouse cannot be assumed to be easy for all users.

Google’s image CAPTCHA “what’s up” [23] asks the user to adjust the orientation of an image. This system is language independent, but the adjustment requires a lot of attention and subtle mouse (or other hardware) movement. Some images also have ambiguity as it can be correctly oriented in multiple ways.

Microsoft’s Asirra [20] utilizes an existing database on petfinder.com and presents the user with images of cats and dogs and asks the user to identify all images of cats out of 12 pets. This platform is language independent, and requires user to scan through 12 images and click 6 times on average to be correct. Figure 3 shows a sample Assira challenge.

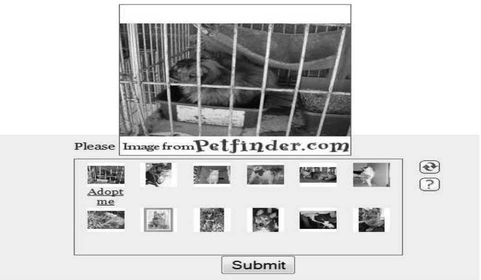


Figure 3: An Assira challenge: A user is always required to select all cats from images of cats and dogs.

Asirra partners with petfinder.com and gets access to their huge database of cats and dogs. But the inherent difficulty for the bot boils down to only classifying each image in either of the two classes: cats and dogs. This makes Assira more vulnerable to machine learning attacks [22]. SEMAGE on the other hand has a two-factor design where in order to pass a challenge the user is required to recognize each image and then understand and identify the semantic relationship between a subset of them. Assira only requires the user to solve the first level (i.e., image recognition). Utilizing the semantic correlation between images to create more secure and user-friendly challenges makes SEMAGE more robust.

3. SEMAGE DESIGN

We propose SEMAGE, “**SE**mantically **MA**tching Ima**GE**s”, a novel image-based two-factor CAPTCHA system which is built upon the idea of semantic relationship between images. The use of semantic meaning of a query has already been applied in other fields like web search [24]. We formulate definitions for semantic similarity of images and design a system that uses these concepts to develop a user-friendly and robust CAPTCHA system.

3.1 Intuitive Idea

All image-based CAPTCHA systems have two main components: a database of images and a “concept” which uses the database to create challenges. The inherent concept may be as simple as PIX [8] which displays different images of the same object from the database and asks the users to assign an appropriate label or a complex one like Cortcha [32] which uses the database to create inpainted and candidate images and asks the users to place the correct candidate image in the inpainted image.

The idea behind SEMAGE is to use semantic relationships among images as the concept and keep the task of the user to simply identify the semantically similar/related images. The semantic relationship is a concrete description which would bind the similar images. The freedom of choosing the semantic relationship for one's application and database gives it the much required customization flexibility. For example, for an electronic e-commerce site, SEMAGE challenge could be formed from the images of the products (an ipod, a zune, tv, heater, refrigerator etc) where the concept would be to ask the users to choose products which do the same thing (ipod and zune in this case, both portable music devices).

SEMAGE presents a set of candidate images with a subset of them sharing an implicit connection or relationship with each other. The challenge for the users is to correctly identify all images in the semantically related subset.

3.2 Defining the Semantic Relationship

We now present the conditions for choosing the "semantically similar" relationship which forms the 'concept' for challenge creation. A "semantic label" could be a term or a relationship which identifies/labels the object. Semantic labels can be directly used to label the database for challenge creation. Let $SL(x)$ denote the function that returns the semantic label of an object x. We consider two images to be "Semantically Matching" if they satisfy any of the following conditions:

- Condition I: if both images can be identified with the same semantic label. Given two images A and B, they are said to be semantically related if $SL(A) = SL(B)$. For example, an image of a computer and a television set can be defined with a semantic label(SL) 'electronics'.
- Condition II: both images can be classified under the same semantic label. Given two images A and B, they are semantically related if $\exists T\, s.t.\, SL(A) \subset T \,\&\, SL(B) \subset T$, where T denotes some semantic label. For example an image of a lion and a deer can be classified under the semantic label 'four legged animals'. Similarly, an image of a television set and a computer can be classified under the semantic label 'electronics'.
- Condition III: when both images put together they express a uniquely identifiable concept. Given two images A and B and some semantic label C that denotes a set of requirements, A and B are said to be semantically matching if $\{A \cup B\} \models C$ where "$\models$" denotes that the left hand side satisfies the requirements of right hand side. For example, an image of a printer and paper can be defined with a identifiable concept 'printing' which becomes the semantic label.

The requirements for a "semantic relationship" gets more generic and the semantic correlation increases as we move from Condition I to III. In order to form a SEMAGE challenge, the images have to be chosen such that only one subset meets any one of the above conditions with preference given to the least generic label. That is, if a set of images contain images that satisfy more than one of the above conditions, the least generic matching is the solution required to pass the challenge. Thus, given a set of images where a small subset of images is of fishes and the rest of the images are of other unique animals, the solution to the challenge would be selecting all images of fishes.

The mechanism may seem complicated but as we show below, a system designed to create challenges where all solutions satisfy only one chosen condition is relatively easy to implement. Also the user study in Section 5 supports our claim that such a system is intuitive and easy for the normal user to solve. The important thing after one has decided upon the "semantic relationship" is to label the images accordingly. We discuss database generation in Section 3.4.

3.3 Challenge Creation

We develop a simple algorithm to create SEMAGE challenges. First we present the definitions and requirements of the involved parameters as follows.

Let n be the number of images in the challenge and m be the number of similar/related images. Let U be the superset of all image sets in the database. Each challenge set is denoted as S where $|S| = n$. There exists a 'semantically similar' subset of images R such that every image in R has the same semantic label, i.e., $\forall\, r_i,\, r_j \in R,\, SL(r_i) = SL(r_j) \,\&\, |R| = m$. A set of images D with $|D| = n - m$, and each image in D has a different semantic label than R. Also $\forall\, d_i,\, d_j \in D,\, SL(d_i) \neq SL(d_j) \neq SL(R)$. This ensures that all the images in the subset D have a different semantic label so that the images in subset R remain the unambiguous semantically related set. Now each challenge set becomes $S = R \cup D$.

We now present a simple algorithm to implement the challenge set as shown in Algorithm 1 . The database consists of a collection of semantically labeled images. The algorithm starts with empty sets R and D. We then pick a semantic label at random from the database and populate R with images having the picked semantic label. Then we populate D with images such that each image has a different semantic label than any of the images chosen previously in D and R. The number of images in the R and D depends on the values of n and m and is customizable. The images in set $R\, and\, D$ are then presented in a random tabular order to the user.

Algorithm 1 : An algorithm to generate SEMAGE challenges from a labeled database

```
$R \leftarrow \phi$
$D \leftarrow \phi$
$A \leftarrow$ Pick an Semantic label at random
while $|R| \neq m$ do
    $X \leftarrow$ (pick a unique image with label $A$)
    $R = R \cup X$
end while
$Y \leftarrow \phi$
while $|D| \neq (n - m)$ do
    $Z \leftarrow$ Pick a label at random which is not $A \cup Y$
    $Y \leftarrow Y \cup Z$
    $D = D\cup$ (pick a unique image with label $Z$)
end while
$S \leftarrow R \cup D$
Randomize(S)
```

3.4 Database

Populating the database is a major issues with all image-based systems. Unlike text CAPTCHAs which can use any random combination of characters in the challenge creation, images in SEMAGE owing to the requirement of semantic similarity have to be carefully selected. One may always use freely available image search services like google image search to find relevant images. For our implementation, we developed a semi-automated mechanism that populates the database by crawling the Internet. One can also consider taking frames from movies and short videos. Both of the above approaches can be considered as semi-automatic and require some manual work to weed out irrelevant images. The drawback of such methods is that an attacker can venture to spend enough time and manual work to reproduce the whole database.

SEMAGE, however, due to its inherent design offers an way of database creation for web sites, such as e-commerce sites, which already have a image database. Web vendors in e-commerce usually have multiple images of the same product (such as pictures from different angles), multiple styles of the same product (same product of different color, size, packages), and multiple products of the same category. Images are tagged with the product information, and product info is categorized into different classes. Multiple relations can be established among these images and used as the 'semantic context'. With the abundance of existing tagging information, we can implement the 'challenge creation' algorithm by adding simple logical changes. Furthermore, some databases actually have implemented more sophisticated relations such as 'similar products' as a recommendation for users when they browse certain products, thus more sophisticated 'semantic relationships' can be formed based on such information. Using these images not only adds to the security of the database, but also serves as a good form of advertisement.

4. SEMAGE ANALYSIS

4.1 Design Analysis

4.1.1 Usability

Usability with security is the primary focus of SEMAGE. The images contain content that cognitively make sense to the users, and are easy to discern. By drawing on human's vast storage of common-sense knowledge, our design helps user spend minimum effort solving the challenge. Moreover, it fits the way a human thinks - it is natural for humans at first sight to see what an image is about, much better than dealing with any details (orientation, certain feature image, etc.). Establishing relationships among objects is another ability humans are natural at, and humans almost automatically dissolve any ambiguity they need to resolve. For example, if a red car is presented with other colored cars, human immediately notice the color difference. However, if the same red car is presented with red buckets, red clothes etc. humans notice the difference in object category. For a computer, both of the steps pose a difficult AI problem. It first needs to do image recognition to determine what the image contains, and tag the image in a pre-determined category. To solve the 'relationship' answer, the computer would not only need vast correctly labeled database, but also complex AI intuition. This creates a great gap in the difficulty level for humans and bots.

In addition, SEMAGE provides an easy-to-operate interface for users to indicate correct answers. Only a few mouse clicks is required to pick up the correct images, this makes SEMAGE to be a good choice of touch-based systems and smart-phones where typing is more difficult. This is much easier than tracing an outline of objects (as in SQ-PIX [7]) and typing in letters from a keyboard, especially on mobile devices.

4.1.2 Language Independence

Our design utilizes the fact that a picture transcends the boundaries of languages. Some CAPTCHA systems also use semantic clues, such as ESP-PIX [3]. However it asks the user to find the right word among a list of English words that describes the content of the image. This limits the audience to people with decent proficiency in the language. Our design is language independent and can be used by people across the world. This is especially beneficial for people who are not comfortable using English as a daily language.

4.1.3 Customization Flexibility

Our design offers several ways to customize the challenge on content, security level and usability level. The image database can be customized to suit the needs and style of the hosting website. For example, for special interest groups, the database can be objects of the theme of the group, such as movie screenshots for a movie rental site or specific products for an e-commerce site. This provides possibility of advertisement of content or fun in the traditionally boring test of CAPTCHA.

It is also easy for web administrators to customize on the security level. The administrator can decide on the size of the candidate image pool, and the size of the correct answer set. For a scheme that present n candidate images and ask the user to pick up k matching images, the success rate of random guessing is $1/C(n, k)$. The increase of the size of answer set does not necessarily decrease the chance of success of a random guessing success when n is small, but as n increases, the probability of a random guess attack goes down. As for the user experience, the time users spent on the CAPTCHA task increases as the size of candidate image pool increase, but the effect of an increased size of answer set on users time is not obvious. We think the optimum choice of n and k might depends on particular content of the images used, and a specialized user study can be conducted if such data is desired.

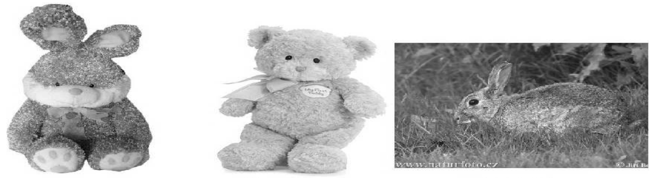
4.2 Security Analysis

We consider an adversary model wherein a bot has access to the unlabeled and uncategorized database of images from which we form our challenges. It is to be noted that given ample time and resources some of the attacks discussed below could succeed but taking a long time defeats the primary purpose of the bot. Our goal as in any CAPCTHA system is to make current attacks as difficult as possible, so that any successful attack would need a major step forward in technology. We now identify and analyze possible ways of attacks against our system and how it fares against them.

4.2.1 Attacks using machine learning techniques

Similar techniques used to attack Asirra [22] could be used to attack our system too. The attack on Assira was an attack on the first level of our model namely simple "image recognition". In essence, attackers try to get a certain number of correctly labeled images, and train on several different classifiers, either based on color information or texture information. However, solving a SEMAGE challenge not only requires image recognition but also identifying the "semantic relationship". The identification of "semantic relationship" among images is an unsolved AI problem. Moreover, even if the semantic correlation is weak and the semantic label is just the object name, SEMAGE accommodates much more object classes than Asirra (which had only 2), and the attacker will need to build many more types of classifiers accordingly.

Now let us consider a very simple example of "semantic relationship", e.g., "real and cartoon" images of the same animal (as used in Section 5). The color and texture data between a cartoon specie and real animal specie varies much more than in between cartoons and real animals, as illustrated in Figure 4. While attackers might attempt to train classifier of real animal and cartoon animal independently, the performance decreases as the number of classifiers increase which could be very complex. Thus the success rate of attacks using this sort of algorithm is likely to be very low.



(a) A Rabbit (b) A Bear (c) A Rabbit

Figure 4: Example limitations of the texture-based machine learning attack; (a) shares more commonality with (b) than with (c) , while (a) and (c) are of the same type (rabbit).

Attacks using template fitting techniques: In image recognition, one developed area is to fit objects into (visual) feature templates. For example, a chair can be identified if given the template of 'four legs and a horizontal top'. Accordingly, for a rabbit, the feature should probably be 'upwards pointing long ears'. However, it is much harder to define 'long' than 'upwards'. A deer, with pointy upward ears would be classified into the 'rabbit' template. Furthermore, not all objects have such uniquely identifiable simple feature.

4.2.2 *Random guess attack*

For a SEMAGE scheme that presents n candidate images and asks the user to select k matching images, the success rate of random guessing is $1/C(n, k)$. As shown in Figure 5, choosing a low value of n and k could make the system more vulnerable to random guess attacks. On the other hand a low n, k makes the system more user friendly and less frustrating for the user. Our implementation for the user study uses low n, k values making it more susceptible to random guess attacks. In case of a low n, k system, multiple rounds of SEMAGE could constitute one challenge; such technique is already in use in current systems such as reCAPCTHA. By choosing a relatively low n, k value, we sacrifice a bit of security against random guess attacks for usability. We do so because we can make up for the relatively high susceptibility of SEMAGE to random guess attacks and deter brute force attackers by enhancing SEMAGE with Token Buckets [20] system. Assira needs more images in each challenge set to be secure because of the limited set of differentiating classes of objects (two to be precise, just cats and dogs) whereas there can be theoretically thousands of differentiating classes in our SEMAGE implementation. The added security provided by SEMAGE's two-factor design allows us to use a low n, k system without sacrificing security much.

A SEMAGE system could also be complemented with other techniques such as the Partial Credit Algorithm in [20], which would allow a large n, k and an 'almost right' answer can be defined as missing one image in the answer set. Token buckets [20] can also be implemented to prevent brute-force attackers from making a number of continuous random guess attacks.

4.2.3 *Attack using the static image name in source*

If the source code of the HTML page hosting the challenge uses image names, an attacker could potentially use those names to identify similar images. However, this sort of attack is easily defeated by randomizing the images name in the source. In our system implementation, names of the images in the challenge are in no way exposed to the user. The image names in the html source is randomized when sent to the user.

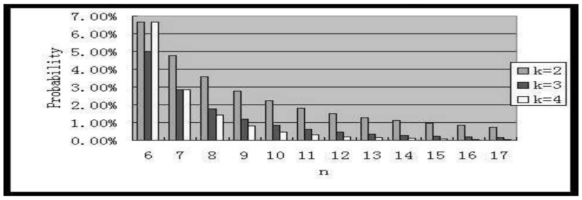


Figure 5: Random guess attack success rate with respect to k and n

4.2.4 *Attack by creating an attack database using the general relationships used in the system*

The attacker might manually identify the general "semantic relationship" used in the system and then search and build an image repository to create an attack database. Using the labeled images of the attack database, a brute force search against the candidate set might yield him a correct 'similar' set. However comparing each image of the challenge with all the images in the attacker's image archive would take lots of time and resources than what would constitute a feasible attack; also this might exceed the maximum time allowed to take a challenge.

4.2.5 *Attack by mining Textual description of images*

Potentially an attacker could use systems such as google's goggle[1], an image based search system, to uncover textual descriptions of the candidate image set and then use the textual descriptions to identify relationships among images. We argue that first of all image recognition or search is still not mature enough for now (very hard problem for unknown images). In addition, identifying relationships among objects even with textual descriptions is a complex AI problem to solve, especially since the correct similar images depend on the semantic context. Such an attack would potentially defeat most present image-based systems such as Assira, PIX, SQ-PIX, but because of the two level design of SEMAGE, the bot would still need to understand and identify the semantic correlation. Having a textual description only possibly solves the problem of image recognition. There may exist images with overlapping descriptions but are not a part of the 'semantic similar' image set in the context. Consider for example a candidate image set wherein the context is identifying 'four legged' animals among images of insect, deer, lion, human, electronics item and other unrelated objects. Now even with accompanying textual descriptions such a relationship is hard for a bot to find and relate to lion and deer.

5. EVALUATION

We conducted a large-scale user study to evaluate the usability of SEMAGE as compared to Assira and reCAPTCHA. For this purpose, we firstly built a website which would present the users with sample SEMAGE challenges.

5.1 Sample Implementation of SEMAGE

In our sample implementation, each challenge consists of a set of images (the number of images is configurable) where a subset of images would share a distinct relationship/feature with each other. The images are furthermore randomly distorted by introducing noise and changing the texture. Our implementation was carried out in PHP with MySQL being used as the database. Figure 6 gives a high level design of the implementation.

[1] http://www.google.com/mobile/goggles/

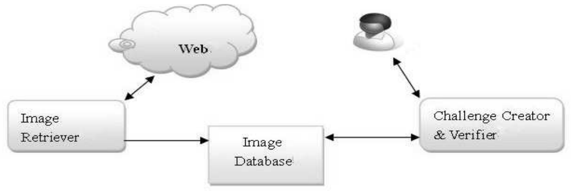
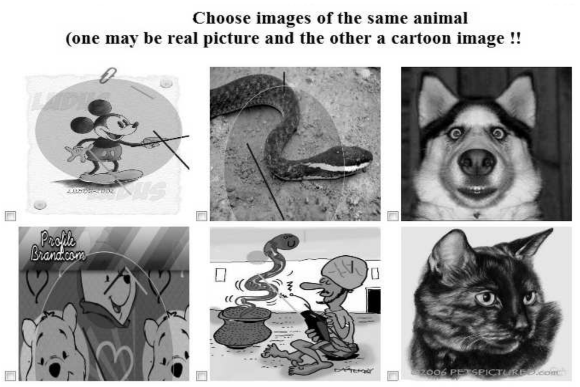


Figure 6: Overall Implementation Illustration



Choose images of the same animal
(one may be real picture and the other a cartoon image !!

Figure 7: Screenshot of sample SEMAGE implementation with Image 2 and 5 being similar, both snakes.

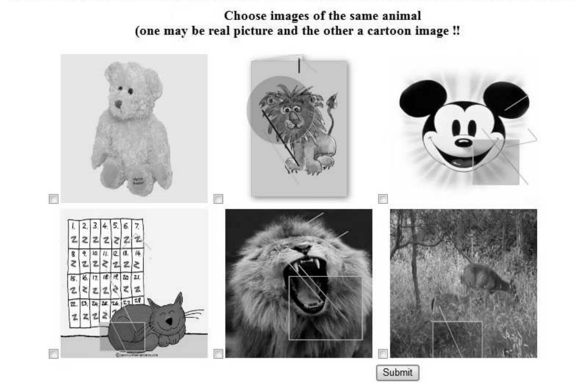
Choosing the "semantic relationship": In our particular implementation, the challenge set consists of real and cartoon images of animals with the relationship defining the 'similar' subset being "real and cartoon images" of the same animal. The advantages of choosing the 'real and cartoon' relationship to define "semantic relationship" between images are as follows:

- The relationship between real and cartoon images of the same animal in most cases is subtle and variable. The reason is that the animals may completely differ in visual characteristics such as size, shape and outline in real and cartoon representations.
- Humans with inherent capability to relate visibly dissimilar objects would be able to pass the challenge easily whereas the current state-of-the-art bots cannot. We test this assumption of ours in the user study we conduct, discussed in details in Section 5.
- Generating a large database is easier. A simple search for an animal on images.google.com yields millions of entries, hence we have a fast and easy way to build up a large database.

Figure 7 shows a sample SEMAGE challenge of our simple implementation. The total number of images in one challenge is six with the "semantically similar" set of two images, one a real image and the other a cartoon image of the same animal.

Database Generation: The first step for SEMAGE implementation after defining the semantic relationship between the "similar" images is database generation. An image search and download tool was implemented shown as Image Retriever in Figure 6, which searches and downloads the required images from the web. The tool would take in the search keywords (to search for real or cartoon images of the animals), image dimensions, and number of images to download and the label tags. It then automatically downloads the images and stores in the database. A simple search for an animal on images.google.com yields millions of entries, hence we have a fast and easy way to build up a large database. In reality, since the automated search does not always yield relevant results, we manually weed out the irrelevant images from the collection.

Dynamic Noise Addition: To make machine learning attacks based on image classifiers difficult, we randomly introduce noise in the images of the challenge set at each challenge creation phase. We introduce noise in the form of random shapes and color scale alteration in the image with the help of the ImageMagick library [5]. The position of inserting the random shapes varies from the center of the images to its edges. Also scale of color adjustment is also randomly varied to prevent the bot classifiers from easily weeding out the noise. Such random noise introduction makes sure that each image appears with different noise levels. Figure 8 shows a SEMAGE challenge after the introduction of noise.



Choose images of the same animal
(one may be real picture and the other a cartoon image !!

Figure 8: Example of noise addition in our implementation. Here we can clearly see noise but still identify Image 2 and 5 being similar, both lions. The changes in color scale are not visible due to the black and white nature of images.

Interface: As shown in Figures 8 and 7, each challenge appears as a tabular strip of images. The title of the tabular strip presents the challenge and then the user needs to click on the similar images and press submit to send the response to the server for verification. We experimented with different layouts, e.g., the images being apart from one another, images in a single straight strip, and found that it is much easier to identify similar images if they are bunched together in a tabular format.

5.2 User Study Methodology

A comprehensive IRB approved user study was then conducted to gather data about how user-friendly SEMAGE is, which is one of the most essential criterion for a CAPTCHA to be deployed in real systems. We also incorporated reCAPTCHA, a text-based system and Asirra, an image-based system from Microsoft in the user study to carry out a comparative analysis. Both Asirra and ReCAPTCHA are available as a free web service allowing us to easily integrate them in our study. The volunteers took the study remotely and were given a brief 1-page pictorial description of what they need to do to pass a challenge for all the systems. We logged the time taken to complete each challenge as the difference in time between when the test first appears on the screen and the time user clicks on the 'submit' button to submit his attempt. The users were let known of whether they passed or failed the previous challenge before presenting a new one.

A total of 174 volunteers took the study and the population was a mix of graduate and under-graduate students. The subject pool was diverse with most of the users from a non-computer science discipline, with a mix of native and non-native English speakers. The subject pool consisted of 66 females and 108 males. The subject pool were in no way made aware of the fact that SEMAGE is our system. We collected the time taken by each user to complete a challenge for each of the system as described earlier. We monitor the time taken for all attempts irrespective of whether it was successful or not. We also collected numbers of successful and failed attempts to solve a challenge.

5.3 User Study Layout

The user study was carried out via a website with the following sections:

- An initial questionnaire asking the users to rate their familiarity with CAPTCHAs, proficiency in English language and other demographic questions such as sex and age range.
- A 1-page pictorial description of EMAGE, Assira and reCAPTCHA, showing users how to solve each challenge.
- 5 different challenges from SEMAGE.
- 5 different challenges from Asirra.
- 5 different challenges from ReCAPTCHA.
- A final short questionnaire asking users to rate SEMAGE for fun factor and ease of use as compared to Assira.

We believe a pictorial description of each of the systems was necessary for fair usage statistics on the image recognition systems. It was probably a user's first time seeing an image-based CAPCTHA whereas all the users had invariably taken a text-based challenge before. Presenting a brief description of what they need to do to pass a challenge would prepare them with necessary basic information of each system and allow us to collect fair usage data. The study took an average of 8.7 minutes to complete.

We divide the usability evaluation in different sections presented below according to the following metrics:

- How fast can a user complete a challenge?
- How many times does the user pass the challenge successfully?
- Does the user consider the system to be fun and easy?

5.4 Timing Statistics

As shown in Table 1, users complete text-based and SEMAGE challenges faster than Asirra. Each user takes an average of 6 seconds more to complete an Assira challenge.

	Semage	**Asirra**	**recaptcha**
Time Taken in seconds	11.64	17.35	11.05

Table 1: Average Time taken per challenge for each of the systems (in seconds)

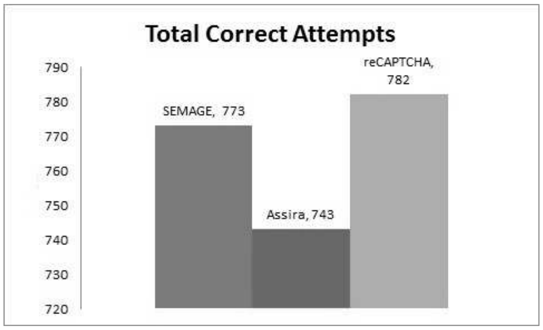
The distribution plots in Figures 9 show that most of the users of SEMAGE finished each challenge in about 11.647 seconds or less, whereas this number is comparatively higher for Asirra with most of the users taking around 17.355 seconds. Consistency and uniformity in majority of the data points of the plots show that the timing average was not largely affected by some isolated outlier cases and it represents the general behavior of the users.

We notice that the average time taken by the users to solve a challenge from SEMAGE is almost the same as that of reCAPTCHA. This is actually surprising. We expected that solving a SEMAGE challenge is much slower than solving a reCAPTCHA challenge because text-based CAPTCHAs have been widely in use for a long time and users have gotten used to them whereas users were seeing our system for the very first time. This encouraging fact suggests that SEMAGE is pretty user-friendly and easy to use.

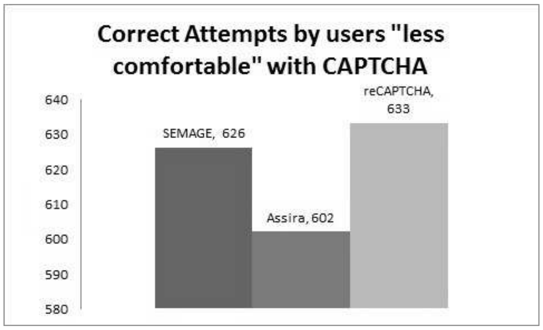
We concede that an Assira challenge consists of more images than a SEMAGE challenge leading to more time spent in completing each challenge. However, Assira needs more images in each challenge set to be secure because of the limited set of differentiating classes of objects (two to be precise, cats and dogs) whereas there can be theoretically thousands of differentiating classes in our SEMAGE implementation. Moreover, presence of just two differentiating given classes should have made the challenge easier for humans as they simply need to place each image in one of the two categories. SEMAGE on the other hand requires the user to relate two or more images, making it potentially more time consuming. However the timing data clearly shows that taking SEMAGE challenges is easier than it seems because of the natural cognitive ability of humans.

5.5 Accuracy Statistics

Simply speaking, the total number of correct attempts for SEMAGE is higher than Asirra, indicating that users are able to correctly solve more challenges of SEMAGE. Figure 10(a) shows a graphical representation of the difference in correct attempts between Assira and SEMAGE. We had also asked the users to rate their familiarity and comfort level with CAPTCHAs on the scale of 1 to 5 (with 5 being very comfortable) in the initial questionnaire. As we see in Figure 10(b), the participants who voluntarily identified themselves as 'less comfortable' (rated 3 or less) with CAPTCHA systems in general also show high accuracy with SEMAGE and reCAPTCHA than with Asirra.



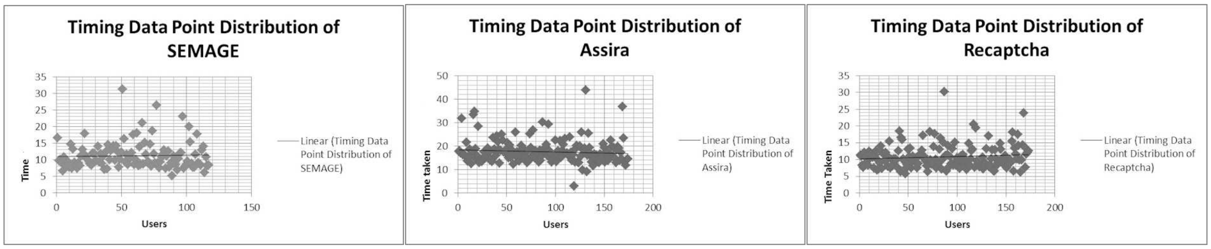
(a) Total correct attempts out of 815 attempts



(b) Total correct attempts from 132 users who rated themselves as less comfortable with CAPTCHA

Figure 10: Accuracy achieved on individual systems

In order for the system to be deployed in the real world, it should have a high 'Correct Attempts ratio' for humans. The 'Correct Attempts ratio' (C.A.R) is simply the number of correct attempts di-



(a) SEMAGE Timing (b) Assira Timing (c) reCAPTCHA Timing

Figure 9: Timing Distribution of each system for all users

vided by the total attempts. It signifies how many times a human passes the challenge. The closer the ratio is to 1, the better the system is in terms of usability.

The user study data shows that our system has a higher C.A.R (0.94) than Asirra (0.91). Users had been familiar with text-based CAPTCHA systems, so we expected them to do very well in the reCAPCTHA system. But again, the difference between SEMAGE and the traditional text-based system is almost negligible. This along with the timing data shows that our system likely has a higher usability factor than the current state-of-the-art image-based system (Asirra).

5.6 Fun Factor and Ease of Use

After the completion of challenges from the three systems, the users were then asked to compare and rate SEMAGE and Assira on the criterion of Fun and Easiness. There were two separate questions: one for Fun factor and the other for Easiness, which asked them to choose a rating as follows:

- 1, if they found Assira to be way more fun or easy
- 3, if they found Assira and SEMAGE to be equal on the Fun or Easiness factors
- 5, if they found SEMAGE to be way more fun or easy
- 2 or 4, if they were slightly inclined towards Assira or SEMAGE, respectively.

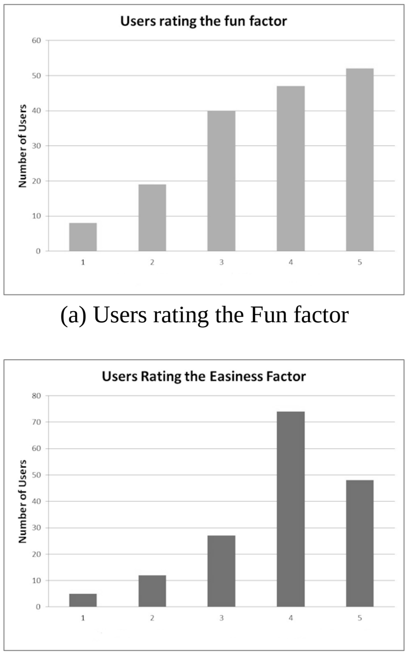
These factors gave us a more subjective indicator of usability. We can clearly see from Figure 11(a) that majority of the users (58.92 %) choose rating 4 and 5 indicating a high fun factor with SEMAGE. Only 16.07% choose rating 1 and 2 indicating Assira was better while the rest considered them to be equally fun. This clearly supports that more users found SEMAGE to be a system that was more fun to solve than Assira.

Figure 11(b) shows the rating distribution for the easiness factor. 72.61% of the users rated 4 and 5 indicating SEMAGE to be easier than Assira. Only 10.72% of the users rated 1 and 2 indicating Assira to be easier while 16.66% of the users rated 3 indicating they considered both systems to be equally easy.

These metrics as well as the timing and accuracy results shown previously clearly demonstrate that SEMAGE is a highly user-friendly CAPTCHA system.

6. LIMITATIONS AND FUTURE WORK

Generating a vast and correct database is always a challenge for image-based CAPTCHA systems. In our simple SEMAGE implementation we crawl the web to automatically gather and label images. However not all images returned by the crawler were relevant, some were even objectionable. We then manually weeded out the irrelevant images. Such manual labor is time consuming and would pose a big problem when the database content is regularly updated.



(a) Users rating the Fun factor

(b) Users rating the Easiness

Figure 11: We asked users to comparatively rate SEMAGE and Assira on the metrics of fun and easiness. Rating 1,2 indicate Assira to be more fun and easy, rating 3 indicate both systems are equal and rating 4,5 indicate SEMAGE to be more fun and easy.

There can also be legal issues in directly using the crawled images.

SEMAGE by the virtue of its design though, does not require the database to be built in such a way. Websites like e-commerce services, movie rental services can easily use the available image database with a suitable "semantic relationship". However, further work is required to create a large, correct database automatically to allow widespread deployment in real world.

In this paper we introduced the concept and technique of creating CAPTCHAs using "semantic relationships" between objects and then implemented a simple system for demonstration. Our naive implementation does *not* reach the full potential of SEMAGE and we plan to build a more robust, high semantic correlation based SEMAGE system as future work.

7. CONCLUSION

In this paper, we present SEMAGE (semantically matching images). The design of this CAPTCHA presents a set of candidate images and asks users to choose a set of images that fit a certain relation. The challenge is layered in that both knowledge about semantic meaning of images and relationship between the subjects of images is required. The challenge comes naturally to humans as it incorporates light-weight visual and cognitive task. However, the layering scheme provides double protection against bot attacks. It is easy to understand and the interaction interface is simple and efficient. CAPTCHA systems constantly seek an optimum trade-off point on security and usability. SEMAGE provides great room for customization by the website administrators. They can customize the number of candidate images and semantically similar images in the challenges to adjust the usability and security level according to the need of particular websites. Moreover SEMAGE can be targeted towards touch-based smart-phones and devices where typing to solve a text-based CAPCTHA is difficult. Website administrators can also determine the content of the image database and cater towards their promotional needs. The database can be populated especially for SEMAGE, or adapted from existing database. E-commerce is one area where SEMAGE database can be easily built and SEMAGE can be utilized for both security and advertisement purposes.

8. REFERENCES

[1] Audio and visual captcha. `http://www.nswardh.com/shout/`.

[2] Breaking text captcha. `http://www.blackhat-seo.com/2008/how-to-break-captchas/`.

[3] Esp-pix. `http://server251.theory.cs.cmu.edu/cgi-bin/esp-pix/esp-pix`.

[4] Gimpy project. `http://www.captcha.net/captchas/gimpy/`.

[5] Imagemagick. `http://www.imagemagick.org/script/index.php`.

[6] recaptcha official site. `reCaptchaOfficialSite:http://www.google.com/reCAPTCHA`.

[7] Sq-pix. `http://server251.theory.cs.cmu.edu/cgi-bin/sq-pix`.

[8] L. v. Ahn. *Human Computation*. Ph. d. dissertation, Carnegie Mellon University, 2005.

[9] H. S. Baird and K. Popat. Human interactive proofs and document image analysis. In *Proceedings of the 5th International Workshop on Document Analysis Systems V*, DAS '02, pages 507–518, London, UK, 2002. Springer-Verlag.

[10] J. P. Bigham and A. C. Cavender. Evaluating existing audio captchas and an interface optimized for non-visual use. In *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, pages 1829–1838, New York, NY, USA, 2009. ACM.

[11] E. Bursztein, R. Bauxis, H. Paskov, D. Perito, C. Fabry, and J. C. Mitchell. The failure of noise-based non-continuous audio captchas. In *Proceedings of 2011 IEEE Symposium on Security and Privacy (Oakland'11)*, 2011.

[12] E. Bursztein, R. Beauxis, H. S. Paskov, D. Perito, C. Fabry, and J. C. Mitchell. The failure of noise-based non-continuous audio captchas. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2011.

[13] E. Bursztein, S. Bethard, C. Fabry, D. Jurafsky, and J. C. Mitchell. How good are humans at solving captchas? a large scale evaluation. In *Proceedings of 2010 IEEE Symposium on Security and Privacy (Oakland'10)*, 2010.

[14] T.-Y. Chan. Using a text-to-speech synthesizer to generate a reverse turing test. *Tools with Artificial Intelligence, IEEE International Conference on*, 0:226, 2003.

[15] K. Chellapilla, K. Larson, P. Simard, and M. Czerwinski. Designing human friendly human interaction proofs (hips). In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '05, pages 711–720, New York, NY, USA, 2005. ACM.

[16] K. Chellapilla and P. Simard. Using machine learning to break visual human interaction proofs (hips). In *In Advances in Neural Information Processing Systems*, pages 265–272, 2005.

[17] M. Chew and J. D. Tygar. Image recognition captchas. In *In Proceedings of the 7th International Information Security Conference (ISC)*, pages 268–279, 2004.

[18] R. Datta, J. Li, and J. Z. Wang. Imagination: a robust image-based captcha generation system. In *Proceedings of the 13th annual ACM international conference on Multimedia*, MULTIMEDIA '05, pages 331–334, New York, NY, USA, 2005. ACM.

[19] A. S. El Ahmad, J. Yan, and L. Marshall. The robustness of a new captcha. In *Proceedings of the Third European Workshop on System Security*, EUROSEC '10, pages 36–41, New York, NY, USA, 2010. ACM.

[20] J. Elson, J. R. Doucerur, J. Howell, and J. Saul. Asirra: A captcha that exploits interest-aligned manual image categorization. In *Proceedings of the 14th ACM conference on Computer and communications security*, CCS '07, pages 366–374, New York, NY, USA, 2007. ACM.

[21] H. Gao, H. Liu, D. Yao, X. Liu, and U. Aickelin. An audio captcha to distinguish humans from computers. In *Proceedings of the 2010 Third International Symposium on Electronic Commerce and Security*, ISECS '10, pages 265–269, Washington, DC, USA, 2010. IEEE Computer Society.

[22] P. Golle. Machine learning attacks against the asirra captcha. In *Proceedings of the 15th ACM conference on Computer and communications security*, CCS '08, pages 535–542, New York, NY, USA, 2008. ACM.

[23] R. Gossweiler, M. Kamvar, and S. Baluja. What's up captcha?: a captcha based on image orientation. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 841–850, New York, NY, USA, 2009. ACM.

[24] R. Guha, R. McCool, and E. Miller. Semantic search. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 700–709, New York, NY, USA, 2003. ACM.

[25] P. Matthews and C. C. Zou. Scene tagging: image-based captcha using image composition and object relationships. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '10, pages 345–350, New York, NY, USA, 2010. ACM.

[26] G. Mori and J. Malik. Recognizing objects in adversarial clutter—breaking a visual captcha. In *In Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2003.

[27] Y. Rui and Z. Liu. Excuse but are you human? In *Proceedings of the eleventh ACM international conference on Multimedia*, MULTIMEDIA '03, pages 462–463, New York, NY, USA, 2003. ACM.

[28] L. von Ahn, M. Blum, N. J. Hopper, and J. Langford. Captcha: Using hard ai problems for security. In *In In Proceedings of Eurocrypt, Vol. 2656*, pages 294–311, 2003.

[29] L. von Ahn, M. Blum, and J. Langford. Telling humans and computers apart automatically. *Commun. ACM*, 47:56–60, February 2004.

[30] J. Yan and A. S. El Ahmad. A low-cost attack on a microsoft captcha. In *Proceedings of the 15th ACM conference on Computer and communications security*, CCS '08, pages 543–554, New York, NY, USA, 2008. ACM.

[31] J. Yan and A. S. El Ahmad. Usability of captchas or usability issues in captcha design. In *Proceedings of the 4th symposium on Usable privacy and security*, SOUPS '08, pages 44–52, New York, NY, USA, 2008. ACM.

[32] 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. In *Proceedings of the 17th ACM conference on Computer and communications security*, CCS '10, pages 187–200, New York, NY, USA, 2010. ACM.