A Proposal to Prevent Click-Fraud Using Clickable CAPTCHAs

Rodrigo Alves Costa, Ruy J. G. B. de Queiroz
Federal University of Pernambuco
UFPE/CIn – Av. Jornalista Aníbal Fernandes, s/n – Cidade Universitária – 50.740-560
Recife – PE, Brazil
rac2,ruy@cin.ufpe.br

Elmano Ramalho Cavalcanti
Federal University of Campina Grande
UFCG/DSC – Rua Aprígio Veloso, 882 – Bairro Universitário – 58.429-140
Campina Grande – PB, Brazil
elmano@copin.ufcg.edu.br

Abstract—Advertising on the Internet is a key factor for the success of several businesses nowadays. The Internet has evolved to a point where it has become possible to develop a business model completely based on Web advertising, which is important for the consolidation of such a model and the continuity of the Internet itself. However, it is often observed that some content publishers are dishonest and employ automated tools to generate traffic and profit by defrauding advertisers. Similarly, some advertisers use automated tools to click on the ads of their competitors, aiming to exhaust the budget of the competitor’s marketing departments. In this paper, differently of recent click-fraud detection mechanisms, that take place after the fraud has already occurred, we propose an approach for preventing automated click-fraud, based on the use of clickable CAPTCHAs.

Keywords-Web Advertising; Advertising Networks; Click-Fraud; Clickable CAPTCHAs

I. INTRODUCTION

Nowadays, advertising over the Internet is one of the most profitable ways of realizing marketing campaigns aiming to impact a large user base, either for small or large businesses. For instance, it is very common for Internet advertisers (such as eBay) to provide their ads to an advertising representative (for example, ValueClick), to reserve a certain amount of financial resources and to commit to paying a commission to the representative as profit for some previously determined user actions, such as clicking on an ad, performing a purchase or even bidding in an auction available in the representative’s site or service. Then, Internet publishers (e.g. myspace.com), motivated by the commission paid by the advertisers, contact the representatives with the objective to share the profit by displaying the ads also in their Web pages, obtaining part of the commission if the source of the user action is their Web site. The focal point in this relation is the representative, since they act as mediators between the publishers and the advertisers. Once the relationship is generalized to include various publishers and advertisers, it is said that the representative is acting as an advertising network. This server uses its inner routines to choose a set of ads and provides this set to the publisher’s Web page, to be displayed in the user’s browser. If the user clicks an ad in the site of the publisher, this action will be associated with the server of the network advertiser with links being displayed in the publisher’s site. The network advertiser’s server then saves the click in a history record, for later charging, and redirects the user to the web page of the advertiser whose ad was clicked.

Since the publishers profit from the click events on the advertiser’s announcements, it is plausible to observe an incentive for dishonest publishers to increase the number of clicks their sites generate [2]. Moreover, dishonest advertisers can simulate clicks in the ads of their competitors with the objective to drain the funds associated with their marketing budget [3]. The scenario can vary from small sites attempting to generate revenue by continuously clicking the ads displayed in their own website, to larger companies looking to damage the budget of their competitors by hiring huge botnets or human teams to click on the competitors’ ads [4].

The online advertising community rejects these practices, which are all different sorts of click-fraud. Click-fraud results in bad reputation to the advertising networks, and there are several cases in which networks had to pay penalties to advertisers [5]. Such a scenario jeopardizes the credibility and furthermore, the industry of online advertising over the Internet.

Click-fraud has been considered a concern for advertising networks every since their conception [6]. The current numbers involved in click-fraud are difficult to quantify because there are many ways to consider a proportion of invalid clicks from total clicks. A 2005 study widely regarded and cited from MarketingExperiments.com estimated that 29.5% of the clicks in three experimental Google campaigns were fraudulent [7]. More recent statistics from Click Forensics Inc. reveal that 19.1% of paid clicks on web ads on 4Q 2010 and 22.3% on 3Q 2010 were fraudulent, a total increase of 15.3% in that year [8]. Even with such expressive numbers, search companies and their customers continuously claim that the problem is under control. However, online marketing observers believe that click-fraud is associated with revenue losses in the order of...
billion dollars, thus having the potential of destroying the whole industry. Despite of the exact number, click-fraud today is inherent to the online advertising business and, although network advertisers seek to defend themselves in many different ways, fraudsters have become more sophisticated, using increasingly complex programs and scripts able to spoof the source of the clicks and perform the fraud automatically.

The majority of academic research and publications in this area investigate the fraud of the publisher, since it could be generalized to the advertiser fraud, and discuss the fraud detection through various methods, such as: the cryptographic approach [9], data analysis techniques [10], fraud detection tools [11], and traffic analysis and brute force algorithms [1]. All of these methods are detection techniques and treat the fraud after it has occurred. As stated previously, programs have become increasingly complex and such has the resolution of the problem involving fraud detection. For these reasons, we propose a different approach, whose objective is the prevention of the click-fraud, such that the detection is in theory no longer required, but ideally complementary.

The proposed scheme in this paper introduces an entity which we call the Certifier, responsible for providing credentials to customers after they have responded to a test. Such credentials will enable the network advertiser to distinguish valid clicks, performed by humans, from ordinary clicks originated in overall traffic. The certifier is then an attester of the click who strengthens the heuristic of isolation of fraudulent clicks.

The technique hereby proposed could very well be leveraged as a complementary technique to the existing click filtering and detection mechanisms. In sections II and III, we will provide a detailed presentation of the prevention methodology we propose. In section IV, a brief analysis of related work is realized. Finally, in section V, conclusions and further work will be discussed.

II. CAPTCHAs

In recent years, a considerable number of public services in the web have been trying to prevent abuse from automated programs by requiring from the users the resolution of a challenge in the format of a “Turing test” [12], popularly known as CAPTCHA, “Completely Automated Public Turing test to tell Computers and Humans Apart”. In these services, the users are only able to start using the service after providing a successful answer to the test. The main advantage of leveraging such tests is that, theoretically, the tests are easily generated and answered by humans. All CAPTCHAs possess some sort of secret information which is initially only known by the challenger but not by the agent being challenged.

The scheme necessary to implement our solution is based on the authentication of users after resolution of a CAPTCHA, aiming to distinguish human users from computer bots. Once the CAPTCHA is successfully solved, the user receives a ticket in the form of a cookie [13]. For security reasons, the cookie has an expiration date and time, after which the user needs to answer a new test. The ticket and all its related information could be stored locally as a user session variable or could be implemented differently, in a central authority that would keep record of valid active tickets.

Therefore, the scheme to be studied here has two distinct aspects:

Certifier: an architectural element of extreme importance in this scheme, that runs as a web service, and is responsible for identifying and tagging the computers of human users, after the correct response of the CAPTCHA. The technical topology of the Certifier depends on how the ticket will be stored: if the local storage option is selected, a simple cookie-based authentication method should be utilized; if the storage in a central authority is the selected option, it will be necessary to implement a service to act as an authenticating unit, that could work as a web service in one of the servers of the advertising network.

Ticket authentication: this scheme enables the validation of clicks originated by users who have not yet produced history of network traffic. In this approach, the click is not considered valid unless it is accompanied by a ticket (that is, a cookie). Thus, it is possible to detect multiple requests from the same source through ticket presentation. In traditional approaches, detection of traffic from similar locations can become a big problem, often depending on mapping the origin of the transmitted data, via a mapping mechanism such as IP lookup or analysis of navigation impression through session identifiers.

At this point we would like to reinstate that this approach is not the ultimate defense against click-fraud. Such as [4], this paper presents an approach that can be used as complementary defense mechanism that should be leveraged in parallel to other, regular, detection methodologies, and is expected to contribute to the security against click-fraud research community by presenting a new point of view in attempting to prevent the fraud from occurring instead of simply distinguishing valid clicks from non-valid clicks, after the fraud has already been committed.

A. Classes of CAPTCHA

According to Elson et al [14], it is possible to define two distinct classes of CAPTCHAs. The Class I CAPTCHAs are composed by those in which the secret information is merely a random number, used by a public algorithm to originate a challenge, in what can be seen as analogous to a public key crypto-system. The Class II CAPTCHAs, on the other hand, besides a random number, also use a secret database with a high degree of entropy, analog to a one-time-pad crypto-system.

The Class I CAPTCHAs are advantageous in many ways: they can be written in a simple program with few lines of code, do not require the storage of any secret information and can generate an infinite set of unique challenges. However, its greatest achievement – a distorted text recognition challenge – evidences small differences in the resolution success rates from either humans or computers. Programs implementing Optical Character Recognition (OCR) algorithms have been increasingly competitive with humans in recognizing distinct characters, fact that has motivated researchers to augment the difficulty by segmenting the characters in an image in several
distinct regions [15]. The main problem is that, when increasing the difficult to protect from OCR algorithms, the difficulty also affects humans, and can potentially generate CAPTCHAs too hard to resolve, which would eventually discourage users from using the service. Because of this, major Internet commercial sites continue to use challenges that are clearly segmented and therefore of easy resolution by OCR algorithms (such as the one displayed in Picture 1).

![Figure 1. Simple CAPTCHA (register.com)](image1)

It is important to observe that the web page owners have apparently decided that the success of a user when navigating through a CAPTCHA depends not only upon the fact that he is able to resolve the challenge, but also on their willingness to do so. In fact, relatively simple CAPTCHAs are sufficient to disencourage a substantial amount of customers to unsign from using an online service [14].

Class II CAPTCHAs are able to surpass the weaknesses described above. Since they are not restricted to generating challenges from algorithms with a low level of entropy, they can explore a significantly larger set of human skills, like the recognition of details from photographic images captured from the real world. Such challenges evidence very different success rates between humans and computers, basically due to two reasons: besides being a lot harder for a computer to interpret the contents of an image, it is possible to generate image-based challenges less tedious for humans without losing their efficiency in blocking unwanted programs or scripts.

Therefore, the methodology here proposed involves a type of Class II CAPTCHAs, the clickable CAPTCHA, and their application to prevent click-fraud.

### B. Clickable CAPTCHA

A critical problem in building a Class II CAPTCHA is populating the database with a sufficiently large set of classifiable data with a high level of entropy. To resolve this problem, the ASIRRA project [14] proposes an alignment of interests to generate a deal in which the population of the database is fed externally, from a database hosted by a partner who continuously manually categorizes new data.

ASIRRA is a CAPTCHA that requests the users to classify pictures of dogs and cats, and its major strength comes from a partnership with the site Petfinder.com, the largest Internet service in which users can adopt abandoned pets [14]. Petfinder has a database of about three million images of cats and dogs, categorized by human volunteers in the United States and Canada. The deal consists in Petfinder providing access to the ASIRRA project to this database. In exchange, ASIRRA adds an “Adopt Me” link underneath each photo, helping Petfinder promote its mission of exposing animals for adoption. Clearly this is a mutually beneficial partnership and promotes the social roles of increasing computer security and animal welfare.

![Figure 2. Example of a clickable CAPTCHA (sample from http://research.microsoft.com/asirra)](image2)

There are numerous advantages of leveraging a clickable CAPTCHA like ASIRRA. First of all, it is of very easy resolution by human users. For instance, according to Elson et al [14], a human user can answer such a CAPTCHA in less than 30 seconds 99.6% of times, whereas computers would have an insignificant success rate of 1/54,000. Besides that, considering the overall user experience, humans tend to consider the resolution of such challenge way more interesting than the traditional text-based CAPTCHA.

Other than the usability aspect, we can also enumerate the additional following advantages in adding a clickable CAPTCHA like ASIRRA to a service:

- Humans can solve it quickly and accurately;
- In contrast to several CAPTCHAs, which are abstract or subjective, the challenges in ASIRRA are concrete, harmless and do not require any sort of specialized or cultural knowledge. This makes such a CAPTCHA less frustrating to users;
- Promotes an additional social role other than increasing network security: find a home to homeless pets.

We can also consider some disadvantages for such a CAPTCHA:

- It needs to be implemented as a centralized and external dependant service, since it consists of a single instance database and algorithm. This CAPTCHA should be implemented as a web service that generates and then verifies CAPTCHAs to all demanding users;
- It depends on the reliability of its database. If a clickfraudster hires cheap manpower to classify its three million images, it would be basically impossible to prevent the attack – this is one reason why the approach we are proposing in this paper needs to be leveraged as a complement to currently existing methodologies in both the industry and academia;
- It still has issues to deal with users with disabilities.
III. ARCHITECTURE

In a traditional scheme, when a user clicks on an ad located in the site of a publisher, the corresponding ad is obtained from the advertiser and the transaction is recorded by the advertising network. After this, the advertising network will charge the advertiser and pay the publisher.

In the model here proposed (cf. Figure 3), there are some additional tasks to be performed: when the user clicks on an ad (for example, in an advertising network banner located on the site of a publisher), it is prompted by the advertising network with a new web page containing a clickable CAPTCHA that needs to be solved. If the challenge is not answered correctly, a new challenge will be proposed, and the ad will not be exhibited. This process will be repeated until the challenge is solved correctly. Once this happens, a ticket certifying that the user is human is embedded by the advertising network in the user browser, the advertising network records the action and the ad is finally displayed to the user.

Whenever a previously authenticated user clicks in any ad, the ticket will be automatically sent to the advertising network, triggering a validation process in its servers. After the validation and confirmation that the ticket sent is valid, the ad is displayed. To mitigate the risk involving a situation in which a user authenticates once and then executes a script that will run overnight from their workstation, the tickets are only valid during a short period of time, after which it is necessary to perform a new authentication.

Figure 3. Proposed Communication Flow for Ticket Authentication

It is important to observe that employing the use of cookies or tickets in the current filtering methods used for click-fraud detection is not indicated. This is because the filtering process is an exclusive one: if one cookie was used to mark and exclude a determined type of malicious user, fraudsters could simply remove the cookies from their web browsers. This is why we consider this methodology “separative”, since it accepts only valid clicks, it may very well make use of cookies. The tickets mark good, trustable users. This is the same reason why the mechanism here proposed is prevention-based instead of detection-based.

IV. RELATED WORK

One of the first projects that realized the need to combat click-fraud through a prevention mechanism was Premium Clicks [16]. Like our work, Premium Clicks sought to hinder click-fraud through the adoption of an affirmative approach to accept only legitimate clicks instead of a click filtering validation mechanism. The difference between our work and Premium Clicks is in the fact that, in Premium Clicks, certification of valid clicks occurs through the sharing of evidences of user legitimacy across multiple Internet sites. After storing information about the navigation pattern of several users in coupons, sites would have a way of not only to distinguish between valid users, bots and scripts, but also to classify amongst valid users those who are likely to spend more money, eventually leading to a click ranking mechanism, after the identification of which clicks are more valuable. This shared information would be responsible for an analogous authentication mechanism to the one we propose in this paper, which, in our case, is performed through the answer of a CAPTCHA.

The Premium Clicks work is extremely interesting for presenting two innovations: first, an authentication mechanism that is transparent to users, since the information would be automatically updated in the users’ browser, and seamlessly shared amongst the different Internet sites participants of the model, such as advertising networks and their publishers. The second innovation is in its very own sharing mechanism: by sharing such information and implementing the validation mechanism including evaluation of the monetary value of each click, the work eventually proposes a new business model for the online advertising industry.

Although we consider these proposals extremely relevant in academic terms, and also technically challenging, it is unavoidable not to realize some negative aspects in both. While user transparency is a significantly valuable usability aspect of the proposal, when sharing navigation information from their users, the model naturally faces privacy concerns implications that could represent the ultimate challenge in making the model feasible. Regarding the possible change in the business model for online advertising, there is a potential issue regarding the premise of good faith from each of the participants of the model. Some of these participants will certainly be competitors amongst each other. Unfortunately, it is difficult to imagine, in nowadays extremely competitive market, industry competitors sharing reliable information about their best customers and buyers among each other.

About the CAPTCHA authentication mechanism, since the release of the CAPTCHA concept [17], hundreds of different proposals for CAPTCHA implementations have been presented. The great majority of these proposals is for text based Class I CAPTCHAs: the computer generates a challenge
by selecting a sequence of characters, renders them, distorts the image and adds noise. The text based CAPTCHAs are popular because they are simple, small and easy to project and implement. Unfortunately, as previously stated and successfully demonstrated by Smart et al [15], OCR based algorithms have shown to be as efficient as humans in solving them, even if these images have been distortable in a trustworthy manner. In a related work, Mori and Malik [18] have demonstrated that the GIMPY CAPTCHA, proposed by von Ahn [17], could be solved by computers in 92% of the attempts.

Because of these facts and numbers, the recent development of Class I CAPTCHA has been focused in adding even more segmentation to the image. Several works were proposed in this sense, by a number of organizations in the industry. The overall conclusion is that text based CAPTCHA were either unsafe due to being of easy resolution, or too difficult to be tolerated by human users.

This is the scenario where a number of Class II CAPTCHA implementations arise. One of the first proposals was KittenAuth [19], which authenticates users after requesting them to identify pictures of cats. This project is similar to ASIRRA – the difference between both projects is that the database that populates ASIRRA is external, obtained due to an agreement with the site Petfinder.com, as previously explained in this paper.

Regarding click-fraud defense techniques, a more recent work was the Bluff ads [4]. The idea of this project is to insert amongst valid ads a set of ads that has been designed to work as a litmus test. These ads, whose text is totally unrelated to the users search, user profile or user recent activity on the publishers’ site, are to be clickable only by the machines, since they are going to be presented containing specialized text, randomly displayed but not targeted to the specific user profile. If a high number of such ads is clicked, the user is flagged as suspicious, therefore helping in the detection of the click-fraud.

There are a number of other researches and proposals on the click-fraud space, and more will continue to rise, given the relevance of the industry and the presence of the technology in our daily activities.

V. CONCLUSIONS AND FURTHER WORK

In this paper, we presented a click fraud prevention method, an innovation if compared to the great majority of the current click-fraud combat methods, which treat the fraud after its occurrence through the filtering and detection of fraudulent clicks. Contrary to the filtering methods, the approach here presented proposes the use of differentiation tests between humans and computers, through the use of clickable Class II CAPTCHA. The answer to these tests will work as a validation certificate of the clicks which, after considered valid, will be accounted for further charge. In an ideal world, the method here proposed could surpass the current detection mechanisms, however it is specially indicated to be utilized in a complementary fashion.

Regarding the next steps of this project, one of the first decisions that need to be made is to realize which clickable CAPTCHA will be leveraged by the project. Due to the limitations discussed in Section II, the development of a clickable CAPTCHA could be an interesting option. Since the idea of using an external database that grows continuously seems to be an indicated path, further work on this project would be to identify a partner whose role would be similar to Petfinder’s in the ASIRRA CAPTCHA. In fact, we also need to consider leveraging an existing model like ASIRRA itself, so it is necessary to perform a more detailed evaluation on the available CAPTCHA models.

Nothing prevents that, such as in the Premium Clicks project [16], the cookies in our project carry more information about the users such as the frequency in which they perform a purchase after clicking an ad. These tickets could be then classified and certain clicks could be considered more valuable, leading to the proposal of an update in the advertising network business model, as performed in their work. Essentially, however, at this point our primary objective is to develop a new and feasible security mechanism against click-fraud for advertising networks, in which tickets are assigned to valid users after the validation of their response to a clickable CAPTCHA challenge.

REFERENCES


66


