# Uncovering Cloaking Web Pages with Hybrid Detection Approaches

Jun Deng
College of Information Science and Engineering
Hunan University, Changsha, China
djcafe@126.com

Hao Chen
College of Information Science and Engineering
Hunan University,Changsha, China
haochen@aimlab.org

Jianhua Sun
College of Information Science and Engineering
Hunan University, Changsha, China
jhsun@aimlab.org

***Abstract*—Web search cloaking, used by spammers for the purpose of increasing the visiting rates of their website, is a challenging spamming technique to search engines. Existing cloaking detection systems have some shortcomings: the accuracy of their algorithms is not high enough; the types of cloaking techniques that be detected are limited. In this paper, we present a new system to attack these two problems. To improve the detection accuracy, our algorithm combines text, tag and URL based method. For the purpose of detecting more types of cloaking techniques, our system works as follows: driving a real browser to execute scripts in web pages; crawl a page for the second time by modifying the referrer field of our HTTP headers; obtaining search engine's cached page for further comparison. We apply our system to 104,800 URLs extracted from Yahoo. Results show that our system can gain a high accuracy: precision at 94.52% and recall at 98.57%. More types of cloaking techniques are successfully detected by our system.**

***Keywords-Cloak; SEO; search terms; Cloaking Techniques; similarity detection algorithm***

## I. INTRODUCTION

As the rapid development of the internet, a great deal of information and web pages are difficult to search without search engines. Search engines, in addition to providing results for one or several search queries become web surfers' first choice when seeking information on the web. A web page in the top result of a search engine could attract more visitors, and hence a high amount of revenue associated with internet sales. Search engine optimization (SEO) is the process of affecting the search results of a website or a web page in search engines.

There are two broad categories of SEO techniques. A technique that conforms to the search engines' guidelines and involves no deception is considered white hat. Black hat SEO are techniques that improve ranking in disapproved ways by search engines, or involve deception. Cloaking is an SEO technique which returns a different page depending on whether the page is being requested by a human visitor or a search engine. Web server can recognize that if a visitor if from normal browser or search engine spider based on the IP addresses or User-Agent header of the visitor. Benign content will be delivered to search engine crawlers and scam content to normal visitors. More introductions of these techniques will be seen in Section II.

In this paper, we investigate different techniques that are used in cloaking and the cloaking phenomenon. We initiatively designed a system by combining three methods to detect cloaked page and the cloaking technique it used. Our detection algorithm is a combination of text, tag, URL-based methods. JavaScript Redirection-Cloaking which is not mentioned in most of the previous work can also be detected using our system. The rest of this paper is structured as follows. Section II provides background on cloaking and the related work. Section III describes our detecting system in detail, followed by our experimental results in Section IV. Conclusions are drawn in Section V.

## II. BACKGROUND AND PREVIOUS WORK

In this section, we provide a brief overview of the fundamentals of search engines and SEO techniques. Cloaking technique will be described in more detail in this section. We then summarize the previous work that inspired our detection approach.

### *A. Search Engine And SEO*

Search engines employ crawlers (also called spiders) to retrieve web pages. A search engine crawler will follow every link on a site automatically. Then each page is indexed based on keywords which are retrieved from its content. When a user enters a query into a search engine, the engine checks its index and returns a list of ranked web pages based on their relevance to the search query provided by the user.

Optimizing a website by editing its content, HTML and associated coding to increase its relevance to specific keywords is used constantly by search engine optimizers to gain the site's ranking in the search result. Some of these

978-0-7695-5066-4/13 $26.00 © 2013 IEEE
DOI 10.1109/ISCBI.2013.65

techniques are designed to manipulate page ranking algorithms without regard to customer interests. Cloaking is one of the most prevalent SEO techniques that deceive search engine crawlers to make search engine display a search page which may be totally different from the desired one.

### B. Types Of Cloaking Method

David Y. Wang [1] classified different cloaking methods as follows: Repeat Cloaking, User Agent Cloaking, Referrer Cloaking, and IP Cloaking. However, there exists a JavaScript Redirection Cloaking method they did not cover. Here, we have a brief description of these 5 types of cloaking techniques:

- IP Cloaking: The cloaking web sites determine the identity of the visitor using the IP address of the requester. They can easily distinguish all search engine requests with the mapping between IP addresses and search engines.
- User-Agent Cloaking: As Search engines have their own special User-Agent String (e.g., Google-bots' UA: Googlebot/2.1(+http://www.googlebot.com/bot.html )), web servers could use the User-Agent field from the HTTP request header to distinguish the identity of the request.
- Referrer Cloaking: Web site could discern which URL the user click through to reach their site by examining the Referrer field of HTTP headers.
- Repeat Cloaking: A site determines whether the visitor has visited the site before by storing state on either the server side or the client side, such as using cookies.
- JavaScript Redirection Cloaking: As search engine crawlers could not emulate a real browser which could execute JavaScript code in web page, some web sites embed JavaScript redirection code in the page to redirect users to another web site.

### C. Previous Work

Henzinger et al. [2] considered cloaking as one of the major search engine spam techniques. They gave a suggestion of detecting cloaking pages by crawling the same page twice. Najork [3] proposed a method of detecting cloaked pages from browsers by installing a toolbar. The toolbar would send the signature of user perceived pages to search engines. Baoning Wu and Brian D. Davison [4] proposed a method of detecting cloaking pages by calculating the difference of three copies of the same page. They continued their research on detecting semantic cloaking [5]. Chellapilla and Chickering [6] detected syntactic on the most popular and monetizable search terms. They showed that monetized search terms had a higher prevalence of cloaking than popular terms. Referrer cloaking was first studied by Yi-Min Wang and Ming Ma [7]. They found a large number of referrer cloaking pages in their work. Jun-Lin Lin [8] used tag-based method to detect cloaking page and gave a summary of different detecting algorithms among the previous work.

The most recent valuable work on cloaking detection is done by David Y. Wang, et, al [1]. They extended their previous efforts to examine the dynamics of cloaking over five months, identifying when distinct results were provided to search engine crawlers and browsers. They used text-based method and tag-based method to detect Cloaking page. However, JavaScript redirects were not able be handled by their crawler.

## III. System Overview

As shown in figure 1, our detection system consists of two main components. The data crawling component performs three operations sequentially: first, getting hot trend key words on the web; second, getting search result URLs from search engines; third, crawling HTML contents of each URL from search engines' view and normal users' view respectively. The similarity detection component of our system detects the page contents using text, tag and URL based methods. We choose Yahoo as our target search engine. Study of David Y.Wang [1] shows that yahoo has the average count of cloaked page in its search results, while Google has the most of cloaked pages compared to Yahoo and Bing. A detailed description of each part of our system will be seen following.

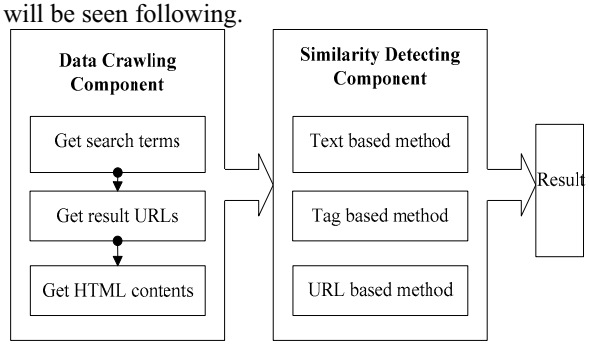


Figure 1. System Architeture.

### A. Crawling Data

The data crawling component consists of three parts: Collecting hot key words; Submitting each term to yahoo to obtain URLs of each term in Yahoo's search result; Crawling HTML contents.

*1) Collecting Hot Search Terms:*

We collect the most popular key-words searched every day on the web. We use jsoup-1.6.3——a Java library providing a convenient API for extracting and manipulating data——to extract search terms from the websites ( Google, Yahoo, BUZZ, AOL, et). They are then stored into database(MySQL-5.5.22).

*2) Crawling URLs:*

In this step, We submit each term to Yahoo, and extract the top 200 search result URLs, Yahoo cache URLs and the result position, using HtmlUnit-2.11, a GUI-Less browser

for Java program providing an API that allows us to invoke pages, fill out forms, click links, etc. As many of the URLs are known good domains, we maintain a list of known good domains and remove the URLs match the white list. Up to this point, we've got 68,282 URLs after removing known good URLs (about 48,800) and duplicate ones (about 7,718) from 104,800 URLs.

HtmlUnit we use in the step of crawling URLs can be used to simulate the behavior of a configured browser, but the JavaScript libraries it supports are limited. In order to detect all the JavaScript Redirection Cloaking, we choose SWT (Standard Widget Toolkit) to detect JavaScript redirection. For the consideration of time delay, our system adopts multi-thread technique which could greatly reduce the running time. We record the URL before redirection, and also the URL after redirection. If the URL redirects to another page with different URL address, we mark it as redirected. We find that some URLs are redirected to different files in the same web sites, while some are redirected to different domains.

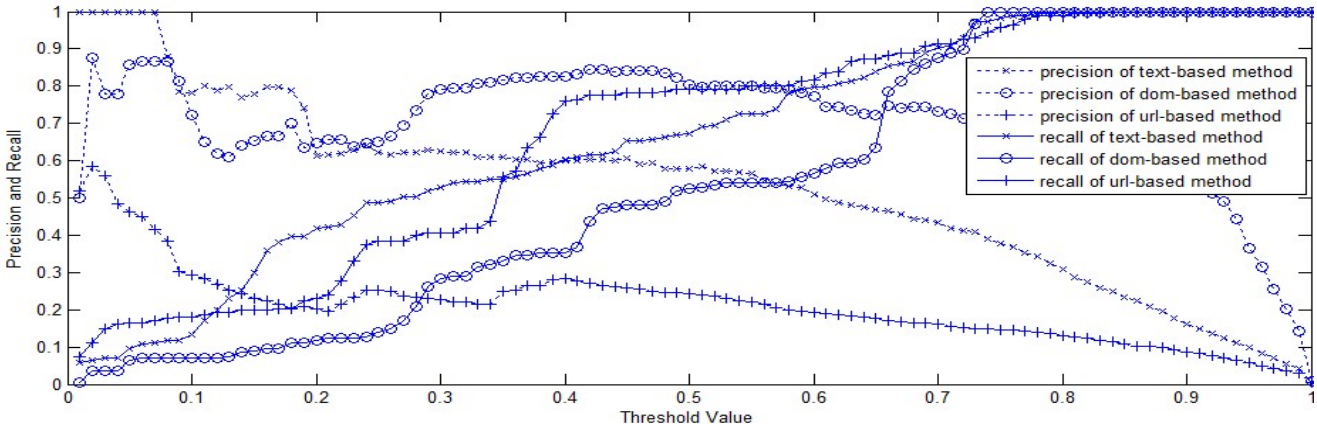


Figure 2. Precision and recall under different threshold values.

*3) Crawling HTML Contents:*

As another goal of this paper is to detect the cloaked methods used, we obtain the HTML content of each URL for 4 times. First, we crawl the page content as a normal user visiting the page directly which we mark as U1. In this step, one more crawl would occur if the URL is a JavaScript redirection URL. We crawl the content of the URL before redirection, also with the content of the URL after redirection for a further detection to see if the JavaScript redirection is a benign or scam one. We mark the HTML content after JavaScript Redirection as JV. Second, we make our crawler mimic a user clicking through Yahoo's search result to visit the page. We mark this version as U2. Third, we crawl the content of Yahoo's cache which is marked as S1. Then we mimic Yahoo's robot by modifying the User Agent field of the request header (this can be done by jsoup-1.6.3) to crawl the content of the URL, this version is marked as S2.

We obtain these versions of each URL: U1, U2, S1, S2 and Js (if there is a JavaScript Redirection). In our crawling step, 53,909 URLs are requested successfully. These five versions of each URL will be processed in our next detection step. The similarity between U1 and S1, U2 and S1, U1 and S2, U2 and S2, JV and S1 are respectively detected by the detection component of our system which will be described in part B of this section in more details.

## *B. Detection Algorithms*

Inspired by previous work [1, 11], to enhance the accuracy of our cloaking detection, our detection algorithm contains three parts: detection of text similarity, detection of tag similarity and detection of URL similarity. Text-shingling algorithm which is used by David Y. Wang [1] to filter out nearly identical pages was first introduced by Andrei Z. Border [9]. Considering of the efficiency and accuracy of our detection system, we adopt text shingling for our first step to detect text similarity between two pages. As for tag based method, Jun-Lin Lin [8] present three methods of using differences in tags to determine whether a URL is cloaked. They consider every copy of a URL as a multi set or a sequence of tags, and execute set operations. However, the sequence of tags cannot represent the whole structure of the page. For higher detection accuracy, we use a recursive algorithm based on optimal free matching for children trees. As for URL similarity detection, it is a comparison of two URL lists. We detail the detection algorithms in the following.

*1) Text Based Algorithm:*

Each document is divided to k-word sequences of adjacent words. A contiguous subsequence is called a shingle. Given a document D, we associate with it its w-shingling. For instance, the 4-shingling of document D (a, rose, is, a, rose, is, a, rose) is the bag {(a, rose, is, a), (rose, is, a, rose), (is, a, rose, is), (a, rose, is, a), (rose, is, a, rose)}. Then, we compute a 64-bit checksum of each shingle using Rabin's fingerprinting algorithm [13, 14]. Let's define S(D,w) as the smallest set of the shingles' fingerprints, while w indicates w-shingling which is defined by ourselves (the value of w is 8 in our system). The resemblance of document A and B is defined as

$$r_w(A,B) = \frac{|S(A,w) \cap S(B,w)|}{|S(A,w) \cup S(B,w)|}. \quad (1)$$

Two documents are considered equal if r=1, which means that they contain the same set of shingles.

For computing the similarity between document A and B, here are the steps using shingling algorithm in our system:

*a)* Extract all the texts from each page forming a text string;

*b)* Divide the text string into 8-shingling;

*c)* Each shingle is converted to corresponding hash value using Rabin's fingerprinting algorithm [13, 14];

*d)* Choose the smallest n hash values ( H(A) and H(B));

*e)* Compute the resemblance of document A and B:

$$r(A,B) = \frac{H(A) \cap H(B)}{H(A) \cup H(B)}. \quad (2)$$

*2)* Tag Based Algorithm:

*a)* Parse the two pages to DOM trees, as $T_1$ and $T_2$;

*b)* Compare the root node of $T_1$ and $T_2$. If they are different, set the similarity result parameter S as zero, and stop. If they are the same, go to step (c);

*c)* Count the node number of $T_1$ and $T_2$, as n1 and n2. If $n_1$=1 or $n_2$=1, then set S=2/($n_1$+$n_2$); or if not, go to step (d) to count the similarities of each sub tree;

*d)* For each sub tree $T_{1i}$ in $T_1$, compared it with all the sub trees in $T_2$, count the similarities $S(T_{1i}, T_{2j})$, set the biggest value to $P_i$,

$$P_i = \underset{j}{Max}\{S(T_{1i}, T_{2j})\}. \quad (3)$$

$$P_j = \underset{i}{Max}\{S(T_{1i}, T_{2j})\}. \quad (4)$$

*e)* Then compute all the sub trees' reference value $R_i(ST_1,ST_2)$ according to this formula:

$$R_i(ST_i, ST_j) = \frac{\sum_{i=1}^{m}\{n_{1i} * p_i\} + \sum_{j=1}^{n}\{n_{2j} * p_j\}}{n_1 + n_2 - 2}. \quad (5)$$

*f)* Return the similarity of two root trees,

$$S(T_1,T_2) = \frac{2}{n_1+n_2} + \frac{n_1+n_2-2}{n_1+n_2} * R(ST_1, ST_2). \quad (6)$$

*3)* URL Based Algorithm:

For document A and B, we extract all the URLs from each page (we define U(A) as the URL-set of document A, U(B) as the URL-set of document B). We compute the URL similarity of document A and B:

$$r(A,B) = (U(A) \cap U(B)) / (U(A) \cup U(B)). \quad (7)$$

In which, $U(A) \cap U(B)$ indicates the amount of the same URLs in two sets.

For each pair of Html documents, we gain three values between 0 and 1 which respectively represent the similarity of texts, tags and URLs. A large value close to 1 indicates that the two documents are very similar. In contrast, the two documents are totally different if the value is 0. Using our detecting algorithm, we process all Html document pairs (U1-S1, U1-S2, U2-S1, U2-S2, JV-S1) of each URL to gain their similarities which then are stored into a database for further making a distinction between different cloaking techniques.

### *C. Setting threshold Value*

We randomly select 10% URLs from all the 53909 URLs as sample URLs. There are about 4,000 URLs among the sampled URLs are benign as the two versions of these URLs are nearly the same. We mark all these URLs as good ones. We manually check each of the remained URLs (about 1000). After marking each URL in the sample set, we divide the similarity values into 100 intervals (0.01, 0.02, 0.03…0.98, 0.99 and 1) to check which interval could gain the highest accuracy.

In this study, precision presents the percentage of URLs marked as cloaked among all the URLs that be detected as cloaked, and recall presents the percentage of URLs that be detected as cloaked among all the URLs marked as cloaked. If a page's similarity between the user's view and search engine's view is smaller than the threshold, we consider it as cloaked. We first try to set the threshold value among all 100 intervals. Then we check the detected results to see if this threshold value could gain high in both precision and recall. Figure 2 shows the precision and recall under different threshold values. According to Figure 2, a larger threshold value give rise to higher recall but lower precision, while smaller threshold value results in higher precision but lower recall.

First, we filter out surely benign URLs whose text similarity value is larger than the threshold that is set using the method mentioned above. Then we further filter out surely benign URLs according to Tags similarity. At last the cloaked page could be detected out if their URL similarity is under the threshold of URL based method.

## IV. SYSTEM EVALUATION

In this section, we evaluate our system from two aspects: the accuracy of our detecting algorithm and the types of cloaking techniques we can detect. While evaluating the

accuracy of our detecting algorithm, we compare our algorithm with text based method, tag based method and URL based method. Results show that our detection algorithm gain both higher precision and recall than the other methods. We detect more cloaking techniques including JavaScript Redirection Cloaking and IP Cloaking which are barely achieved in previous work. User Agent Cloaking and Referrer Cloaking can also be detected.

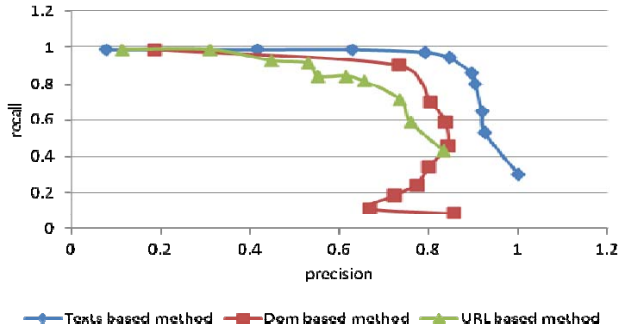


Figure 3. Precision and recall of different detecting methods.

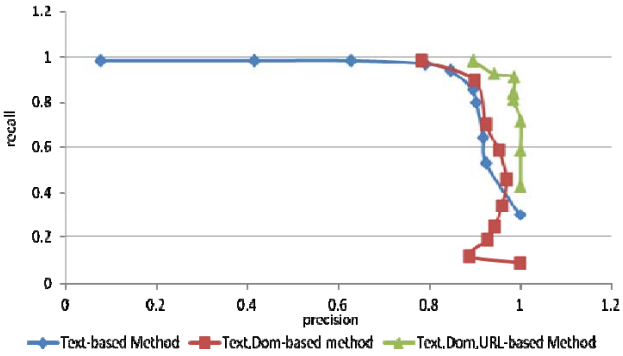


Figure 4. Precision and recall in Text-based method, Text-Dom-based method, and Text-Dom-URL-based mthod.

### A. *Evaluation Of Our Detecting Algorithm*

This section gives a comparison among three methods: text based method, tag based method and URL based method. We also present a comparison among text based method, text-tag based method and text-tag-URL based method (used in our system). The experimental data are the sampled URLs we manually marked (mentioned in part C of section III).

For each method, we define a page as benign if its similarity value is larger than the threshold; otherwise it is marked as cloaked. The detection results are stored into database. Figure 3 shows the precision and recall of each detection method. We can learn from Figure 3 that text based method has better results in both precision and recall, URL based method has the worst results.

As detailed in part B of section III, our detection algorithm combines text, tag and URL based method. The comparison results between these three values and the corresponding threshold values combine to determine the page as cloaked or not. In our system, the filtering step is as follows:

*a)* If the text similarity of two documents is larger than the threshold we set before, we mark it as good, else go to step (b);

*b)* If the tag similarity is larger than the threshold, we mark it as good, else go to step (c);

*c)* If the URL similarity is larger than the threshold, we mark it as good, else we consider it as cloaked, and modify the data base.

Figure 4 shows a comparison of accuracy between Text-based method, Text-Dom-based method, and Text-Dom-URL-based method. We can see that the Text-Dom-URL-based method gain both the highest precision and recall. Our experimental results show precision at 94.52% and recall at 98.57%.

### B. *Cloaking Techniques Detected By Our System*

It is important to point out that all the cloaked pages we detected are first time cloaking, as we didn't request the URL for a second time as the same user in order to strike a balance between performance and accuracy of our system, which means that the Repeat-Cloaking type is out of our consideration.

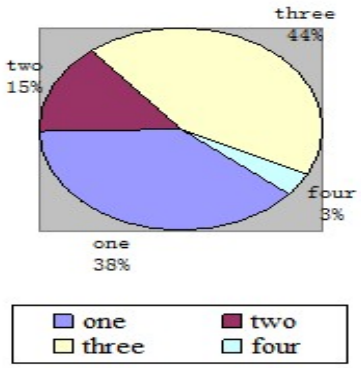


Figure 5. Precision and recall in Text-based method, Text-Dom-based method, and Text-Dom-URL-based mthod.

Results show that our system can be used to detect different types of cloaking techniques including User Agent Cloaking, Referrer Cloaking, IP Cloaking and JavaScript Cloaking. Using our system, we can also learn which type of cloaking technique is used by the corresponding cloaked web site.

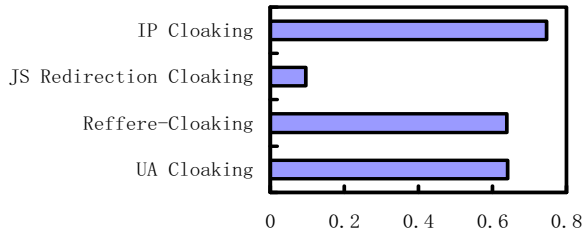


Figure 6. Distribution of combined cloaking techniques.

We also give a statistical analysis on the prevalence of each cloaking techniques. We find that most of the cloaked pages use a combination of two, three or four cloaking techniques, such as user agent and JavaScript Redirection combined cloaking technique.

Figure 5 presents the percentage of cloaked pages in each type of cloaking techniques. JavaScript Redirection Cloaking technique has the least percentage of about 10% among all the cloaked pages. Most of the cloaked pages identify a normal user and search engine according to the IP address. Figure 6 shows the occurrence of combined cloaking techniques. About 44% of cloaked pages use combination of three techniques, 38% of cloaked pages use only one type of cloaking technique. Only 3% of cloaked pages leverage all the four cloaking techniques.

## V. CONCLUSION

A similarity detection algorithm combing with text, tag and URL based method is proposed in this work, which gain a much higher accuracy compared to other three methods used in other works. More types of cloaking techniques that the cloaked web sites use are successfully detected by our system, including User Agent Cloaking, IP Cloaking, Referrer Cloaking and JavaScript Cloaking.

## ACKNOWLEDGMENT

We thank the anonymous reviewers fot numerous valuable comments. This research was supported by the Nationa Science Foundation of China under grant 61173166.

## REFERENCES

[1] David Y.Wang, Stefan Savage, and Geoffrey M.Voelker, “Cloak and Dagger: Dynamics of Web Search Cloaking”. CCS’ 11, October 17-21, 2011, Chicago, Illinois, USA.

[2] M. R. Henzinger, R. Motwani, and C. Silverstein, “Challenges in web search engines”. SIGIR Forum, 36(2):11-22, Fall 2002.

[3] M. Najork, “System and method for identifying cloaked web servers”, June 21 2005.U. S. Patent number 6,910,077.

[4] Baoning Wu and Brian D. Davison, “Cloaking and Redirection: A Preliminary Study”. In Proceedings of the SIGIR Workshop on Adversarial Information Retrieval on the Web (AIRWeb), May 2005.

[5] Baoning Wu and Brian D. Davison, “Detecting Semantic Cloaking on the Web”. In proceedings of the 15th International World Wide Web Conference, pp. 819-828, May 2006.

[6] Kumar Chellapilla and David Maxwell Chickering, “Improving Cloaking Detection Using Search Query Popularity and Monetizability”. In proceedings of SIGIR workshop on Adversarial Information Retrieval on the Web (AIRWeb), August 2006.

[7] Yi-Min Wang and Ming Ma, “Detecting Stealth Web Pages That Use Click-Through Cloaking”. Technical Report Msr-tr-2006-178. Microsoft Research. December 2006.

[8] Jun-Lin Lin, “Detection of cloaked web spam by using tag-based methods”. Expert Systems with Applications, 36(4):7493-7499, 2009.

[9] Andrei Z. Broder, “On the Resemblance and Containment of Documents”. In Proceedings of the Compression and Complexity of Sequences (SEQUENCES' 97), pp. 21-29, June 1997.

[10] Craig Silverstein, Monkia Henzinger, Hannes Marais, and Michael Moricz, “Analysis of a Very large Web Search Engine Query Log”. ACM SIGIR Forum, 33(1):6-12,1999.

[11] Gusfield, D.(1997). Algorithms on strings, trees and sequences: Computer science and computational biology. Cambridge University Press.

[12] Gyongyi, Z. and Garcia-Molina, H., “Web Spam Taxonomy,” in Proc. International Workshop on Adversarial Information Retrieval on the Web (AIRWeb), 2005.

[13] A. Broder, “Some applications of Rabin’s fingerprinting method”. In R. Capocelli, A. De Santis and U. Vaccaro. Editors,Sequeces II: Methods in Communications, Security, and Computer Science, Springer Verlag, 1993.

[14] M. Rabin, “Fingerprinting by random polynomials”. Report TR-15-81, Center for Research in Computing Technology, Harvard University, 1981.