Cloaking and Redirection: A Preliminary Study

Baoning Wu and **Brian D. Davison**
Computer Science & Engineering
Lehigh University
{baw4,davison}@cse.lehigh.edu

Abstract

Cloaking and redirection are two possible search engine spamming techniques. In order to understand cloaking and redirection on the Web, we downloaded two sets of Web pages while mimicking a popular Web crawler and as a common Web browser. We estimate that 3% of the first data set and 9% of the second data set utilize cloaking of some kind. By checking manually a sample of the cloaking pages from the second data set, nearly one third of them appear to aim to manipulate search engine ranking.

We also examined redirection methods present in the first data set. We propose a method of detecting cloaking pages by calculating the difference of three copies of the same page. We examine the different types of cloaking that are found and the distribution of different types of redirection.

1 Introduction

Cloaking is the practice of sending different content to a search engine than to regular visitors of a web site. Redirection is used to send users automatically to another URL after loading the current URL. Both of these techniques can be used in search engine spamming [13, 7]. Henzinger et al. [8] has pointed out that search engine spam is one of the major challenges of web search engines and cloaking is among the spamming techniques used today. Since search engine results can be severely affected by spam, search engines typically have policies against cloaking and some kinds of dedicated redirection [5, 16, 1].

Google [5] describes cloaking as the situation in which "the webserver is programmed to return different content to Google than it returns to regular users, usually in an attempt to distort search engine rankings." An obvious solution to detect cloaking is that for each page, calculate whether there is a difference between a copy from a search engine's perspective and a copy from a web browser's perspective. But in reality, this is non-trivial. Unfortunately, it is not enough to know that corresponding copies of a page differ; we still cannot tell whether the page is a cloaking page. The reason is that web pages may be updated frequently, such as in a news website or a blog website, or simply that the web site puts a time stamp on every page it serves. Even if two crawlers were synchronized to visit the same web page at nearly the same moment, some dynamically generated pages may still have different content, such as a banner advertisement that is rotated on each access.

Besides the difficulty of identifying cloaking, it is also hard to tell whether a particular instance of cloaking is considered acceptable or not. We define the cloaking behavior that has the effect of manipulating search engine ranking results as semantic cloaking. Unfortunately, the various search engines may have different criteria for defining unacceptable cloaking. As a result, we have focused on the simpler, more basic task — when we mention cloaking in this paper, we usually refer to the simpler case of whether different content is served to automated crawlers versus web browsers, but not different content to every visitor. We name this cloaking as syntactic cloaking. So, for example, we will not consider dynamic advertisements to be cloaking.

In order to investigate this issue, we collected two data sets: one is a large data set containing 250,000 pages and the other is a smaller data set containing 47,170 pages. The detail of these two data set will be given in Section 3. We manually examined a number of samples of those pages and found several different kinds of cloaking techniques. From this study we make an initial proposition toward building an auto-

AIRWeb'05, May 10, 2005, Chiba, Japan.

mated cloaking detection system. Our hope is that these results may be of use to researchers to design better and more thorough solutions to the cloaking problem.

Since redirection can also be used as a spamming technique, we also calculated some statistics based on our crawled data for cloaking. Four types of redirection are studied.

Few publications address the issue of cloaking on the Web. As a result, the main contribution of this paper is to begin a discussion of the problem of cloaking and its prevalence in the web today. We provide a view of actual cloaking and redirection techniques. We additionally propose a method for detecting cloaking by using three copies of the same page.

We next review those few papers that mention cloaking. The data sets we use for this study are introduced in Section 3. The results of cloaking and redirection are shown in Section 4 and 5 respectively. We conclude this paper with a summary and discussion in Section 6.

2 Related Work

Henzinger et al. [8] mentioned that search engine spam is quite prevalent and search engine results would suffer greatly without taking measures. They also mentioned that cloaking is one of the major search engine spam techniques.

Gyöngyi and Garcia-Molina [7] describe cloaking and redirection as spam hiding techniques. They showed that web sites can identify search engine crawlers by their network IP address or user-agent names. They also described the use of refresh meta tags and JavaScript to perform redirection. They additionally mention that some cloaking (such as sending search engine a version free of navigational links, advertisements but no change to the content) are accepted by search engines.

Perkins [13] argues that agent-based cloaking is spam. No matter what kind of content is sent to search engine, the goal is to manipulate search engines rankings, which is an obvious characteristic of search engine spam.

Cafarella and Cutting [4] mention cloaking as one of the spamming techniques. They said that search engines will fight cloaking by penalizing sites that give substantially different content to different browsers.

None of the above papers discuss how to detect cloaking, which is one aspect of the present work. In one cloaking forum [14], many examples of cloaking and methods of detecting cloaking are proposed and discussed. Unfortunately, generally these discussions can be taken as speculation only, as they lack strong evidence or conclusive experiments.

Najork filed for patent [12] on a method for detecting cloaked pages. He proposed an idea of detecting cloaked pages from users' browsers by installing a toolbar and letting the toolbar send the signature of user perceived pages to search engines. His method does not distinguish rapidly changing or dynamically generated Web pages from real cloaking pages, which is a major concern for our algorithms.

3 Data set

Two data sets were examined for our cloaking and redirection testing. For convenience, we name the first data as HITSdata and the second as HOTdata.

3.1 First data set: HITSdata

In related work to recognize spam in the form of link farms [15], we collected Web pages in the neighborhood of the top 100 results for 412 queries by following the HITS data collection process [9]. That is, for each query presented to a popular search engine, we collected the top 200 result references, and for each URL we also retrieved the outgoing link set, and up to 100 incoming link pages. The resulting data set contains 2.1M unique Web pages. From these 2.1M URLs, we randomly selected 250,000 URLs. In order to test for cloaking, we crawled these pages simultaneously from a university IP address (Lehigh) and from a commercial IP address (Verizon DSL). We set the *user-agent* from the university address to be `Mozilla/4.0 (compatible; MSIE 5.5; Windows 98)` and the one from the commercial IP to be `Googlebot/2.1 (+http://www.googlebot.com/bot.html)`. From each location we crawled our dataset twice with a time interval of one day. So, for each page, we finally have four copies, two of which are from a web browser's perspective and two from a crawler's perspective. For convenience, we name these four copies as $B1$, $B2$, $C1$ and $C2$ respectively. For each page, the time order of retrieval of these four copies is always $C1$, $B1$, $C2$ and $B2$.

3.2 Second data set: HOTdata

We also want to know the cloaking ratio within the top response lists for hot queries.

The first step is to collect hot queries from popular search engines. To do this, we collected 10 popular queries of Jan 2005 from Google Zeigeist [6], top 100 search terms of 2004 from Lycos [10], top 10 searches for the week ending Mar 11, 2005 from Ask Jeeves [3], and 10 hot searches in each of 16 categories ending Mar 11, 2005 from AOL [2]. This resulted in 257 unique queries from these web sites.

The second step is to collect top response list for these hot queries. For each of these 257 queries, we retrieved the top 200 responses from the Google search engine. The number of unique URLs is 47,170. Like the first data set, we downloaded four copies for each of these 47,170 URLs, two from a browser's perspective and two from a crawler's perspective. But all these copies are downloaded from machines with a university IP address. For convenience, we name these four copies $HC1$, $HB1$, $HC2$ and $HB2$ respectively. This order also matches the time order of downloading them.

4 Results of Cloaking

In this section, we will show the results for the cloaking test.

4.1 Detecting Cloaking in HITSdata

Intuitively, the goal of cloaking is to give different content to a search engine than to normal web browsers. This can be different text or links. We use two techniques to compare versions retrieved by a crawler and a browser — we consider the number of differences in the terms and links used over time to detect cloaking.

As we mentioned earlier in Section 1, calculating the difference between pages from the browser's and crawler's viewpoints is not strong enough to tell whether the page does cloaking. Our proposed method is that we can use three copies of a page $C1$, $C2$ and $B1$ to decide if it is a cloaking page. The detail is that for each URL, we first calculate the difference between $C1$ and $C2$ (for convenience, we use NCC to represent this number). Then we calculated the difference between $B1$ and $C1$ (for convenience, we use NBC to represent this number). Finally if NBC is greater than NCC, then we mark it as a cloaking candidate. The intuition is that the page may change frequently, but if the difference between the browser's copy and the crawler's copy is bigger than the difference between two crawler copies, the evidence may be enough that the page is cloaking.

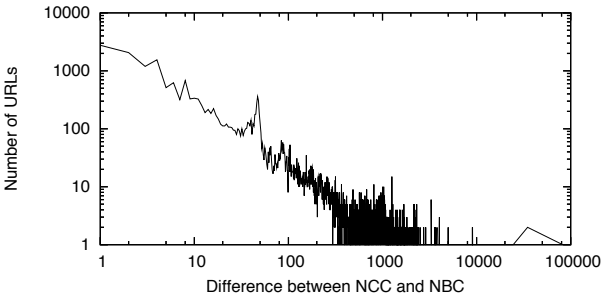


Figure 1: Distribution of the difference of NCC and NBC.

We used two methods to calculate the difference between pages — the difference in terms used, and the difference in links provided. We describe each below, along with the results obtained.

4.1.1 Term Difference

The first method for detecting cloaking is to use term difference among different copies. Instead of using all the terms in the HTML files, we used the "bag of words" method for analyzing the web pages, i.e., we parse the HTML file into terms and only count each unique term once no matter how many times this term appears. Thus, each page is marked by a set of words after parsing.

For each page, we first calculated the number of different terms between the copies $C1$ and $C2$ (designated NCC, as described above). We then calculated the number of different terms between the copies $C1$ and $B1$, (designated NBC). We then select pages that have a bigger NBC than NCC as candidates of cloaking. For this data set, we marked 23,475 candidates of the original 250K data set.

The distribution of the difference of these 23,475 pages forms a power-law-like distribution, shown in Figure 1.

To check what threshold for this difference between NCC and NBC is a good indication for real cloaking, first, we put the 23,475 URLs into ten different buckets based on the difference value. The range for each bucket and the number of pages within each bucket are shown in Table 1.

Then, from each bucket we randomly selected

Bucket ID	RANGE	No. of Pages
1	$x <= 5$	8084
2	$5 < x <= 10$	2287
3	$10 < x <= 20$	1938
4	$20 < x <= 40$	2065
5	$40 < x <= 80$	2908
6	$80 < x <= 160$	1731
7	$160 < x <= 320$	1496
8	$320 < x <= 640$	912
9	$640 < x <= 1280$	1297
10	$1280 < x$	757

Table 1: Buckets of term difference

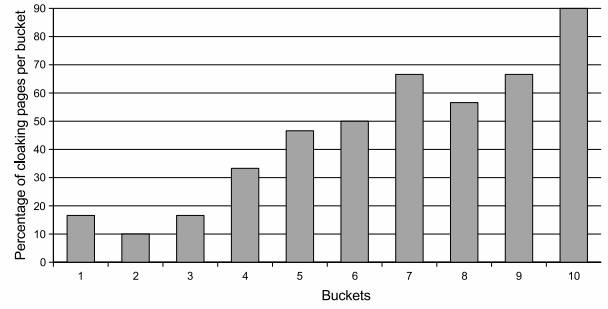


Figure 2: The ratio of syntactic cloaking in each bucket based on term difference.

thirty pages and checked them manually to see how many from these thirty pages are real syntactic cloaking pages within each bucket. The result is shown in Figure 2.

The trend is obvious in Figure 2. The greater the difference, the higher proportion of cloaking that is contained in the bucket. In order to know which is the optimal threshold to choose, we calculated the precision, recall and F-measure based on the range of these buckets. For these three measures, we follow the definitions in [11] and select α to be 0.5 in the F-measure formula to give equal weight to recall and precision. Precision is the proportion of selected items that the system got right; Recall is the proportion of the target items that the system selected; F-measure is the measure that combines precision and recall. The results of these three measures are shown in Table 2. If we choose F-measure as the criteria, buckets 4 and 5 have the highest value. Since the range of bucket 4 and 5 is around 40 in Table 1, we can set the threshold to be 40 and declare that all pages with the difference above 40 to be categorized as cloaking pages. In that case, the precision and

Threshold	PRECISION	RECALL	F value
1	0.355	1.000	0.502
5	0.423	0.828	0.560
10	0.480	0.799	0.560
20	0.534	0.758	0.627
40	0.580	0.671	0.622
80	0.633	0.498	0.588
160	0.685	0.388	0.496
320	0.695	0.262	0.380
640	0.752	0.196	0.311
1280	0.899	0.086	0.157

Table 2: F-measure for different thresholds based on term difference.

recall are 0.580 and 0.671 respectively.

From Figure 2, we can make an estimation of what percentage of our 250,000 page set are cloaking pages. Since we know the total number of pages within each bucket and the number of cloaking pages within the 30 manually checked pages from each bucket, the estimation of total number of cloaking pages is the product of the number of pages within each bucket and the ratio of cloaking pages within the 30 pages. The result is 7,780, so we expect that we can identify nearly 8,000 cloaking pages (about 3%) within the 250,000 pages.

4.1.2 Link Difference

Similar to term difference, we also analyzed this data sets on the basis of link differences. Here link difference means the number of different links between two corresponding pages.

First we calculated the link difference between the copy of $C1$ and $C2$ (termed LCC).We then calculated the link difference between the copy of $C1$ and $B1$ (termed LBC). Finally we marked the page that have a higher LBC than LCC as cloaking candidates. In this way, we marked 8,205 candidates. The frequency of these candidates also approximates a power-law distribution like term cloaking. It is shown in Figure 3.

As with term difference, we also put these 8,205 candidates into 10 buckets. The range and number of pages within each bucket is shown in Table 3.

From each bucket, we randomly selected 30 pages and checked manually to see how many of them are real cloaking pages. The result is shown in Figure 4.

It is obvious that the most of the pages from bucket 4 or above are cloaking pages. We also calculated the F values for these thresholds corresponding to the

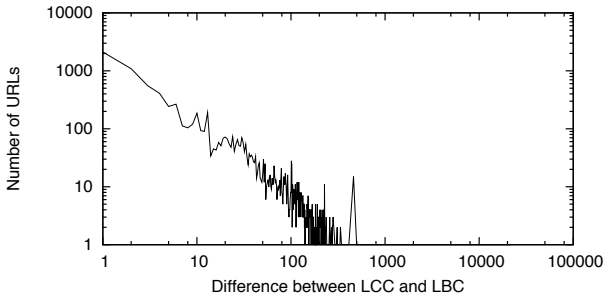


Figure 3: Distribution of the difference of LCC and LBC.

Bucket ID	RANGE	No. of Pages
1	$x <= 5$	4415
2	$5 < x <= 10$	787
3	$10 < x <= 20$	746
4	$20 < x <= 35$	783
5	$35 < x <= 55$	441
6	$55 < x <= 80$	299
7	$80 < x <= 110$	279
8	$110 < x <= 145$	182
9	$145 < x <= 185$	100
10	$185 < x$	173

Table 3: Buckets of link difference

range of each bucket. The result is shown in Table 4. We can tell that 5 is an optimal threshold with the best F value.

Since the number of pages having link difference is smaller than the ones having term difference in reality, fewer cloaking pages can be found by using link difference alone, but are more accurate.

Threshold	PRECISION	RECALL	F value
1	0.479	1.000	0.648
5	0.727	0.700	0.713
10	0.822	0.627	0.711
20	0.906	0.520	0.660
35	0.910	0.340	0.496
55	0.900	0.236	0.374
80	0.900	0.167	0.283
110	0.900	0.104	0.186
145	0.878	0.060	0.114
185	0.866	0.038	0.072

Table 4: F-measure for different thresholds based on link difference.

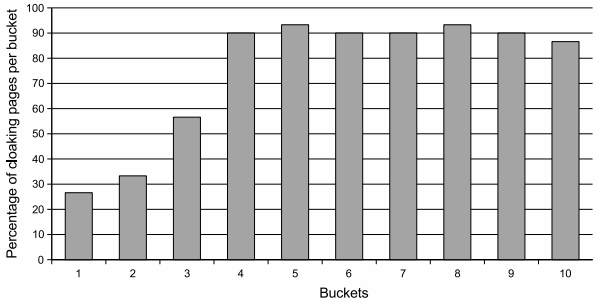


Figure 4: The ratio of syntactic cloaking in each bucket based on link difference.

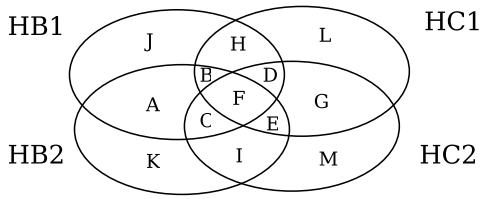


Figure 5: Intersection of the four copies for a Web page.

4.2 Detecting Cloaking in HOTdata

Based on the experience of manually checking for cloaking pages for the first data set, we attempted to detect syntactic cloaking automatically by using all four copies of each page.

4.2.1 Algorithm of detecting cloaking automatically

Our assumption about syntactic cloaking is that the web site will send something consistent to the crawler but send something different yet still consistent to the browser. So, if there exists such terms that only appear in both of the copies sent to the crawler but never appear in any of the copies send to the browser or vice versa, it is quite possible that the page is doing syntactic cloaking. Here when getting the terms out of each copy, we still use the "bag of words" approach, i.e., we replace all the non-word characters within an HTML file with blank and then get all the words out of it for the intersection operation.

To easily describe our algorithm, the intersection of four copies are shown as a Venn diagram in Fig-

Bucket	RANGE	No.	Accuracy
1	$x <= 1$	725	40%
2	$1 < x <= 2$	540	30%
3	$2 < x <= 4$	495	30%
4	$4 < x <= 8$	623	40%
5	$8 < x <= 16$	650	90%
6	$16 < x <= 32$	822	100%
7	$32 < x <= 64$	600	100%
8	$64 < x <= 128$	741	100%
9	$128 < x <= 256$	420	100%
10	$256 < x$	1120	100%

Table 5: Buckets of unique terms in area A and G

Thresholds	PRECISION	RECALL	F value
0	0.647	1.000	0.785
1	0.703	0.965	0.813
2	0.766	0.952	0.849
4	0.836	0.940	0.885
8	0.902	0.881	0.891
16	0.922	0.756	0.831
32	0.960	0.599	0.738
64	0.979	0.470	0.635
128	0.972	0.358	0.523
256	1.000	0.267	0.422

Table 6: F-measure of different threshold

ure 5. We use capital letters from A to M to represent each intersection component of four copies. For example, the area L contains content that only appears in $HC1$, but never appear in $HC2$, $HB1$ and $HB2$; area F is the intersection of four copies, i.e., the content that appears on all of the four copies. The most interesting components to us are areas A and G. Area A represents terms that appear on both browsers' copies but never appear on any of the crawlers' copies, while area G represents terms that appear on both crawlers' copies but never appear on any of the browsers' copies.

So our algorithm of detecting syntactic cloaking automatically is that for each web page, we calculate the number of terms in area A and the number of terms in area G. If the sum of these two numbers is nonzero, we may mark this page as a cloaking page.

There are false negative examples for this algorithm. A simple example is that suppose there is a dynamic picture on the page, every time the web server will randomly select one from 4 JPEG files (a1.jpg to a4.jpg) to serve the request. It happens that a1.jpg is sent every time when our crawler visits this page, but a2.jpg and a3.jpg are sent when our browser visit this page. By our algorithm, the page will be marked as cloaking, but it can be easily verified that this is not the case. So, again we need a threshold for the algorithm to work more accurately.

For the 47,170 URLs, we found 6466 pages that have the sum of number of terms in area A and G greater than 0. Again, we put them into 10 buckets, as shown in Table 5. The third column is the number of pages within this bucket.

From each bucket, we randomly selected 10 pages and manually checked to see whether this page is real syntactic cloaking. The accuracy is shown in the fourth column in Table 5. We also calculated the F-measure, the results are shown in Table 6.

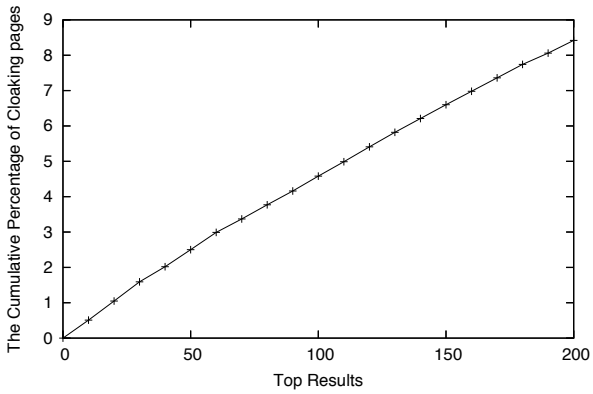
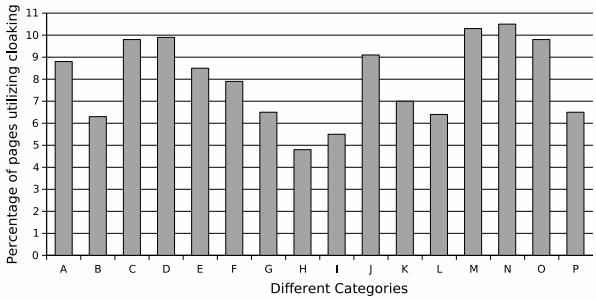


Figure 6: Percentage of syntactic cloaking pages within google's top responses.

Since the 4th and 5th bucket have highest F value in Table 6, we choose the threshold to be the range between bucket 4 and bucket 5, i.e., 8. So, our automated cloaking algorithm is revised to only mark pages with the sum of area A and G greater than 8 as cloaking pages. So, for our second data set, all pages in bucket 5 to bucket 10 are marked cloaking pages. Finally, we marked 4,083 pages out of the 47,170 pages, i.e., about 9% of pages from the hot query data set are syntactic cloaking pages.

4.2.2 Distribution of syntactic cloaking within top rankings

Since we have identified 4,083 pages that utilize cloaking, we can now draw the distribution of these cloaking pages within different top rankings. Figure 6 shows the cumulative percentage of cloaking pages within the Top 200 response lists returned by google. As we can see, about 2% of top 50, about 4% of top 100 URLs and more than 8% of top 200 URLs do utilize cloaking. The ratio is quite high and the cloaking



A. Autos
B. Companies
C. Computing
D. Entertainment
E. Games
F. Health
G. House
H. Holidays
I. Local
J. Movies
K. Music
L. Research
M. Shopping
N. Sports
O. TV
P. Travel

Figure 7: Category-specific Cloaking.

may be helpful for these pages to be ranked high.

Since we retrieved top 10 hot queries from each of 16 categories from AOL, we can consider the topic of the cloaking pages. Intuitively some popular categories, such as sports or computers, may contain more cloaking pages in the top ranking list. So we also calculated the fraction of cloaking pages within each category. The results are shown in Figure 7. Some categories, such as *Shopping* and *Sports*, are more likely to have cloaked results than other categories.

4.2.3 Syntactic vs Semantic cloaking

Not all syntactic cloaking is considered unacceptable to search engines. For example, a page sent to the crawler that doesn't contain advertising content or a PHP session identifier which is used to distinguish different real users is not a problem to search engines. In contrast to acceptable cloaking, we define semantic cloaking as cloaking behavior with the effect of manipulating search engine results.

To make one more step about our cloaking study, we randomly selected 100 pages from the 4,083 pages we have detected as syntactic cloaking pages and manually checked the percentage of semantic cloaking among them. In practice, it is difficult to judge whether some behavior is harmful to search engine rankings. For example, some web sites will send login page to browser, while send full page to crawler. So, we end up with three categories: acceptable cloaking, unknown and semantic cloaking.

From these 100 pages, we classified 33 pages as semantic cloaking, 32 as unknown and 35 as acceptable cloaking.

4.3 Different types of cloaking

In the process of manually checking 600 pages for the above sections, we found several different types of cloaking.

4.3.1 Types of term cloaking

We identified many different methods of sending different term content to crawlers and web browsers. They can be categorized by the magnitude of the difference.

We first consider the case in which the content of the pages sent to the crawler and web browser are quite different.

- The page provided to the crawler is full of detail, but the one to the web browser is empty, or only contains frames or JavaScript.
- The web site sends text page to the crawler, but sends non-text content (such as macromedia Flash content) to web browser.
- The page sent to the crawler incorporates content, but the one sent to the web browser contains only a redirect or 404 error response.

The second case is when content differs only partially between the pages sent to the crawler and the browser and the remaining content is identical, or one copy has slightly more content than the other.

- The pages sent to the crawler contain more text content than the ones to web browser. For example, only the page sent to the crawler contains keywords shown in Figure 8.
- Different redirection target URLs are contained in the pages sent to the crawler and to the web browser.
- The web site sends different titles, meta-description or keywords to the crawler than to web browser. For example, the header to browser uses "Shape of Things movie info at Video Universe" as the meta-description, while the one to the crawler uses "Great prices on Shape of Things VHS movies at Video Universe. Great service, secure ordering and fast shipping at everyday discount prices."
- The page sent to the crawler contains JavaScript, but no such JavaScript is sent to the browser, or the pages have different JavaScripts sent to the crawler than to web browser.

game computer games PC games console games video games computer action games adventure games role playing games simulation games sports games strategy games contest contests prize prizes game cheats hints strategy computer games PC games computer action games adventure games role playing games Nintendo Playstation simulation games sports games strategy games contest contests prize prizes game computer games PC games computer action games adventure games role playing games simulation games sports games strategy games contest contests prize prizes.

Figure 8: Sample of keywords content only sent to the crawler.

- Pages to the crawler do not contain some banner advertisements, while the pages to web browser do.
- The *NOSCRIPT* element is used to define an alternate content if a script is not executed. The page sent to web browser has the *NOSCRIPT* tag, while the page sent to the crawler does not.

4.3.2 Types of link cloaking

For link cloaking, we again group the situations by the magnitude of the differences between different versions of the same page. In one case, both pages contain similar number of links and the other is that both pages have quite different number of links.

For the first situation, examples found include:

- There are the same number of links within the page sent to the crawler and web browser, but the corresponding link pairs have a different format. For example, the link to web browser may contain a PHP session id while the link to the crawler does not. Another example is that the page to the crawler only contains absolute URLs, while the page to the browser contains relative URLs that are in fact pointing to the same targets as the absolute ones.
- The links in the page to the crawler are direct links, while the corresponding links within the page to web browser are encoded redirections.
- The links to web browser are normal links, but the links to the crawler are around small images instead of texts.
- The website shows links to different style sheets to web browser than to the crawler. For example, the page to the crawler contains "href=/styles/styles_win_ie.css", while the page to the browser contains "href=/styles/styles_win_ns.css".

In some cases, the number of links within the page to the crawler and the page to the web browser can be quite different.

- More links exist in the page sent to the crawler than the page sent to web browser. For example, these links may point to a link farm.
- The page sent to web browser has more links than the page sent to the crawler. For example, these links may be navigational links.
- The page sent to the browser contains some normal links, but in the same position of the page sent to the crawler, only error messages saying "no permission to include links" exist.

From the results shown within this section, it is obvious that cloaking is not rare in the real Web. It happens more often for hot queries or popular topics.

5 Results of Redirect

As we have discussed in Section 1, redirection can also be used as a spamming technique. To get an insight into how often the redirection appear and distribution of different redirect methods, we use the HITSdata set mentioned in Section 3. We don't use all four copies but only compare two copies for each page: one from the simulated browser's set (*BROWSER*) and the other from the crawler's set (*CRAWLER*).

5.1 Distribution

We check the distribution of four different types of redirection: HTTP 301 Moved Permanently and 302 Moved Temporarily responses, the HTML meta refresh tag, and the use of JavaScript to load a new page.

In order to know the distribution of above four different redirects, we tabulated the number of appearances of each type. For the first two types, the situation is simple: we just count the pages with response status of "301" and "302". The last two are more complicated; the HTTP refresh tag does not necessarily mean a redirection and JavaScript

TYPE	CRAWLER	BROWSER
301	20	22
302	56	60
Refresh tag	4230	4356
JavaScript	2399	2469

Table 7: Number of pages using different types of redirection.

is even more complicated for redirection purpose. For the first step, we just count the appearance of "<meta http-equiv=refresh>" tag for the third type and count the appearance of "location.replace" and "window.location" for the fourth type. The results for this first step are shown in Table 7.

To get a more accurate number of appearances of the HTTP refresh tag, we examined this further. In reality, the *Refresh* tag may just mean refreshing, not necessarily to redirection to another page. For example, the *Refresh* tag may be put inside a *NOSCRIPT* tag for browsers that do not support JavaScript. To estimate the number of real redirection by using this refresh tag, we randomly selected 20 pages from the 4,230 pages that use the refresh tag and checked them manually. We found that 95% of them are real redirection and only 5% are inside a *NOSCRIPT* tag. Besides, some pages may have identical target URL as themselves in the *Refresh* tag to keep refreshing themselves. We also counted these numbers. There are 47 pages out of 4,230 pages within the *CRAWLER* data set and 142 pages out of 4,356 pages within the *BROWSER* data set that refresh to themselves.

We did one more step for the 4,214 (4,356 - 142) pages that are pages using *Refresh* tag and refresh to a different page. Usually there is a time value assigned within the refresh tag to show how long to wait before refreshing. If this time is small enough, i.e., 0 or 1 seconds, users can not see the origin page but are redirected to a new page immediately. We fetched this time value for these 4,214 pages and draw the distribution of different time values from the range of 0 seconds to 30 seconds in Figure 9. More than 50% of these pages refresh to a different URL after 0 seconds and about 10% refresh after 1 second.

To estimate the real distribution of the JavaScript refresh method, we randomly selected 40 pages from the 2,399 pages that have been identified as candidates for using JavaScript for redirection in the first step. After manually checking these 40 files, we found the 20% of them are real redirections, 32.5% of them are conditional redirections, and the rest are not for redirection purpose, such as to avoid showing the page within a frame.

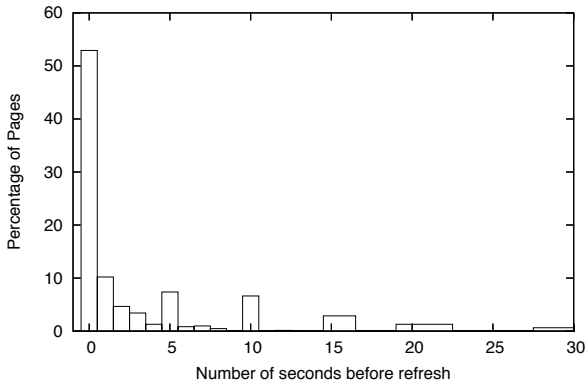


Figure 9: Distribution of delays before refresh.

Sometimes the target URL and origin URL are within the same site, while other times they are on different sites. In order to know the percentage of redirections that redirection to the same sites, we also analyzed our collected data set for this information. Since the JavaScript redirection is complicated, we only count the first three types of redirection here. The sum of the first three types of redirection is 4,306. Within the *CRAWLER* data set, there are 2,328 pages within these 4,306 pages redirecting to the same site, while for the *BROWSER* data set, the number is 2,453.

5.2 Redirection Cloaking

As we have mentioned in Section 4.3, the site may return pages redirecting to different locations in case of different user agents. We consider this redirection cloaking.

We found that there are 153 pairs of pages out of 250,000 pairs that have different response code for a crawler and a normal browser when doing redirecting. Usually these web sites will send 404 or 503 response code to one and send 200 response code to the other. We even found that there are 10 pages that use different redirection method for a crawler and normal web browser. For example, they may use 302 or 301 for the crawler, but use refresh tag with the response code 200 for a normal web browser.

6 Summary and Discussion

Detection of search engine spam is a challenging research area. Cloaking and redirection are two impor-

tant spamming techniques.

This study is based on a sample of a quarter of million pages and top responses from a popular search engine to hot queries on the Web. We identified different kinds of cloaking and gave an estimate of the percentage of pages that are cloaked in the sample and also show an estimation of distribution of different redirect.

There are four issues that we would like to see addressed in future work. The first is that of a bias in the dataset used. Our data sets (pages in or near the top results for queries) do not nearly reflect the Web as a whole. However, it might be argued that it reflects the Web that is important (at least for the purposes of finding pages that might affect search engine rankings through cloaking). The second is that this paper does not address IP-based cloaking, so there are likely pages that do indeed provide cloaked content to the major engines when they recognize the crawling IP. We would welcome the partnership of a search engine to collaborate on future crawls.

The final issue is the bottom line. While search engines may be interested in finding and eliminating instances of cloaking, our proposed technique requires three or four crawls. Ideally, a future technique would incorporate a two-stage approach that identifies a subset of the full web that is more likely to contain cloaked pages, so that a full crawl using a browser identity would not be necessary.

Our hope is that this study can provide a realistic view of the use of these two techniques and will contribute to robust and effective solutions to the identification of search engine spam.

References

[1] AskJeeves / Teoma Site Submit managed by ineedhits.com: Program Terms, 2005. Online at http://ask.ineedhits.com/programterms.asp.

[2] America Online, Inc. AOL Search: Hot searches, Mar. 2005. http://hot.aol.com/hot/hot.

[3] Ask Jeeves, Inc. Ask Jeeves About, Mar. 2005. http://sp.ask.com/docs/about/jeevesiq.html.

[4] M. Cafarella and D. Cutting. Building Nutch: Open source. *ACM QUEUE*, 2, Apr. 2004.

[5] Google, Inc. Google information for webmasters, 2005. Online at http://www.google.com/webmasters/faq.html.

[6] Google, Inc. Google Zeitgeist, Jan. 2005. http://www.google.com/press/zeitgeist/zeitgeist-jan05.html.

[7] Z. Gyöngyi and H. Garcia-Molina. Web spam taxonomy. In *First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2005.

[8] M. R. Henzinger, R. Motwani, and C. Silverstein. Challenges in web search engines. *SIGIR Forum*, 36(2), Fall 2002.

[9] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[10] Lycos. Lycos 50 with Dean: 2004 web's most wanted, Dec. 2004. http://50.lycos.com/121504.asp.

[11] C. D. Manning and H. Schutze. *Foundations of statistical natural language processing*, chapter 8, pages 268–269. MIT press, 2001.

[12] M. Najork. System and method for identifying cloaked web servers, Jul 10 2003. Patent Application number 20030131048.

[13] A. Perkins. White paper: The classification of search engine spam, Sept. 2001. Online at http://www.silverdisc.co.uk/articles/spam-classification/.

[14] WebmasterWorld.com. Cloaking, 2005. Online at http://www.webmasterworld.com/forum24/.

[15] B. Wu and B. D. Davison. Identifying link farm spam pages. In *Proceedings of the 14th International World Wide Web Conference, Industrial Track*, May 2005.

[16] Yahoo! Inc. Yahoo! Help - Yahoo! Search, 2005. Online at http://help.yahoo.com/help/us/ysearch/deletions/.