Privacy-Preserving Query Log Mining for Business Confidentiality Protection

BARBARA POBLETE
Yahoo! Research Chile
MYRA SPILIOPOULOU
Otto von Guericke University
and
RICARDO BAEZA-YATES
Yahoo! Research Spain

10

We introduce the concern of confidentiality protection of business information for the publication of search engine query logs and derived data. We study business confidentiality, as the protection of nonpublic data from institutions, such as companies and people in the public eye. In particular, we relate this concern to the involuntary exposure of confidential Web site information, and we transfer this problem into the field of privacy-preserving data mining. We characterize the possible adversaries interested in disclosing Web site confidential data and the attack strategies that they could use. These attacks are based on different vulnerabilities found in query log for which we present several anonymization heuristics to prevent them. We perform an experimental evaluation to estimate the remaining utility of the log after the application of our anonymization techniques. Our experimental results show that a query log can be anonymized against these specific attacks while retaining a significant volume of useful data.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Search process*; H.2.7 [**Database Management**]: Database Administration—*Security, integrity, and protection*

General Terms: Algorithms, Experimentation, Security

Additional Key Words and Phrases: Privacy preservation, queries, query log publication, Web sites

ACM Reference Format:

Poblete, B., Spiliopoulou, M., and Baeza-Yates, R. 2010. Privacy-preserving query log mining for business confidentiality protection. ACM Trans. Web 4, 3, Article 10 (July 2010), 26 pages.
DOI = 10.1145/1806916.1806919 http://doi.acm.org/10.1145/1806916.1806919

Authors' addresses: B. Poblete, Yahoo! Research, Santiago, Chile; email: bpoblete@yahoo-inc.com; M. Spiliopoulou, Faculty of Computer Science, Otto von Guericke University, Magdeburg, Germany; R. Baeza-Yates, Yahoo! Research, Barcelona, Spain.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2010 ACM 1559-1131/2010/07-ART10 $10.00
DOI 10.1145/1806916.1806919 http://doi.acm.org/10.1145/1806916.1806919

1. INTRODUCTION

Web query logs contain rich information about users' activities, personal preferences, and interests. Query log analysis can provide deep insight into human behavior and serve as a foundation to improve the user experience on the Web. This includes, among other things, satisfying users' information needs more effectively and facilitating social interaction. In the same manner, query log analysis can also reveal information that users consider private and therefore inappropriate for publication. Due to this, a large amount of research has been directed towards protecting *user* information through log anonymization and privacy-preserving data mining. In this work we analyze a new challenge which, in our opinion, deserves equal consideration. This is the protection of confidential *business* information, including both institutions (such as companies) and people in the public eye (such as political leaders). Our definitions, observations, and findings are focused on company confidentiality protection, but extend to people who stand in the public (Internet) light.

We study the issue of confidentiality protection from the perspective of privacy preservation, and consider privacy-preserving data mining methods for the protection of confidential business information. The concept of privacy itself generates much debate. On one hand, some might argue that the term "privacy" can be applied only to humans and not to institutions. On the other hand, there is no unanimous agreement about which information should be considered "private." For example, some users choose to publish personal information such as pictures, videos, and their phone number, while others keep this information private and under no circumstances want it to become public. In the same way, an institution can consider a certain piece of information confidential, while another organization publishes this data without hesitation. From this point of view, the problem of protecting confidential business data is no different from that of preserving private user data: they both require a definition of what is confidential/private, and they both need methods to prevent privacy breaches. In this work we define which information we will consider private/confidential and analyze attacks that can lead to confidentiality breaches along with possible solutions. Consequently we use the terms "private" and "confidential" interchangeably.

The scope of our work is confidentiality protection for query log publishing and query log mining. Our task is to prevent the disclosure of previously unknown data about a particular institution through the publication of an anonymized query log. Our study applies to services offered by third-party search engine log owners, such as reports on Web site usage, and information provided by keyword suggestion tools for search engine ad placement. We believe that entities who provide this type of services should be aware of the *real* amount of data they are disclosing. This is also important for query log owners, such as search engine providers, who depend on Web sites to deliver high quality content. For example, many search engines play a *neutral* role in the assessment of Web sites and would be set back if important sites decide to forbid access to their crawlers due to confidentiality concerns.

To illustrate the need to protect business confidential information, we provide the following made-up, but realistic example.

Example: Disclosing confidential company data. We consider two car selling companies X, Y that use their sites as a promotional channel. Company X's product, $car(X)$, is a direct competitor of company Y's product $car(Y)$: The two cars have comparable functionalities, prices and they are designed for the same target group. Now assume that Y wants to know the impact of X's Web site on the sales of $car(X)$ and assume that X considers this information confidential; this supposition is valid since X has chosen *not* to make this information public.

Company Y applies Web mining on their own site $site(Y)$ and obtains a fairly complete characterization of the user groups accessing it. However, they cannot extrapolate this information on X's Web site, $site(X)$, because they do not know anything about the traffic on that site. Moreover, $site(X)$ is designed differently and many of its contents are irrelevant to Y.

Web analysis also returns statistics about accesses from search engines. Y has learned that $p\%$ of the pageviews on $car(Y)$ lead to a purchase, and $s\%$ of these pageviews come from search engine S. Y can thus compute the contribution of S to the sales of $car(Y)$. This is valuable for the relationship of Y to S. But it still cannot be used to compute the number of sales of $car(X)$.

Finally, Y knows the queries or keywords used in S to access $site(Y)$. This set, $queries(Y)$, reflects the *position* of $car(Y)$ in the Web, according to the perception of users. Company Y can use these queries to enhance their marketing campaign and their Web site. By running each of these queries, they can also find whether $car(X)$ is ranked higher or lower than $car(Y)$ in S. But they cannot make conclusions about how many users have accessed $car(X)$ via S—mainly because $site(X)$ contains words that are different from those used in $site(Y)$.

Now, assume that S decides to publish its query log periodically, using a similar anonymization as AOL, discussed in Arrington [2006]. Thanks to this, company Y can acquire new information about $car(X)$: They simply have to extract all of the entries in the query log for $site(X)$, this is straightforward. Then, Y can collect $queries(X)$ by filtering the subset of requests that show clicks on pages featuring $car(X)$. Even in the case that the URLs in the query log are truncated at the Web site name, it is not a problem to find the full URLs, as explained in Adar [2007]. To find the relevant URLs for $car(X)$, the URLs extracted from the query log for $site(X)$ are inspected online, this gives the number of pageviews from S to $car(X)$. Since Y knows the contribution of pageviews from S to the sales of $car(Y)$ and since X, Y address the same target group, Y can therefore make an approximation of the sales of $car(X)$ induced by S. From this approximation, Y can estimate the total number of sales of $car(X)$, which allows Y to have a point of comparison with its competitor which is a good indicator of its performance in the Web.

Additionally, Y can compare the queries from $queries(Y)$ and $queries(X)$ to learn new things about their own online presence. For example, they can

see which queries exist in *queries*(X) but not in *queries*(Y) (i.e., *queries*(X) − *queries*(Y)) and decide to purchase search engine advertising for these queries (*keyword stealing*). Also, they can use other Web site optimization techniques based on these new queries, for example, placing them as keywords and anchor text on their Web site.

Furthermore, if the query log has been additionally protected by fully anonymizing URLs, then Y can use their own *queries*(Y) in the published log to find the anonymized identifier for *site*(X). This is a more complex deanonymization process which is described in detail in the following sections.

The example just given, although simple, differs in two ways from previously known privacy breaches in query logs [Barbaro and Zeller 2006]: First, information disclosure is achieved by *combining* data sources, that is, a published query log, a private Web server log and publicly accessible Web sites. As we will see in Section 2, privacy preservation methods concentrate in protecting a single type of data source, the query log, rather than a combination of data from multiple *independent* sources. Second, the background knowledge used to disclose confidential company data is not of arbitrary nature: The adversary uses data that is very similar in content and format to that to be disclosed. This makes the confidentiality preservation slightly easier to define but no less challenging to achieve.

The threat of confidential information disclosure is not limited to business institutions that use the Web for marketing and sales. In our example, one may replace companies X and Y with politicians who are candidates for the same region and use their Web sites to inform, conduct polls and to discuss with citizens.

The article is organized as follows. In Section 2, we discuss related work on privacy and privacy preservation with anonymization methods. In Section 3, we specify the types of adversaries expected in a business privacy breach scenario, we introduce the general setting for adversarial activities against a business or a public person's Web site and illustrate with three concrete attacks. Section 4 presents a query log anonymization method that is based on the removal of information disclosing queries. In the same Section, we estimate the information loss produced by our method. Section 5 describes our experiments on the anonymization of a real query log. Finally in Section 6, we summarize our findings and provide an agenda for further research on this subject.

This article is an extended version of Poblete et al. [2008]. Next to expanding the scope of our work and deepening the study on different attacks, we study the loss of information produced by anonymization and the side effects on the "utility" of the remaining data [Verykios et al. 2004].[1] Additionally, the topic of business confidentiality has been mentioned earlier in our survey [Baeza-Yates et al. 2010], which presents an overview of privacy issues in query log analysis.

[1]Utility in this case refers to the usefulness of the remaining data for the extraction of patterns and not for the exploitation of private information.

2. RELATED WORK

Regardless of the extensive research on privacy preservation, the term "privacy" itself is not unanimously defined. An overview of different privacy definitions can be found in Clifton et al. [2002]. For the purpose of our study, we make a distinction between two broad categories of privacy preservation methods in the context of data analysis. The first category involves "anonymization" methods, which prevent the identification of single individuals (or of some of their features) in a database. The second category involves "cryptographic" methods or protocols for learning a statistical model in a collaborative way, *without* disclosing data that is private to the agents involved. Our work adheres to the first category.

A seminal solution to the *anonymity preservation* challenge has been proposed by Sweeney [2002] and studied intensively since. Sweeney introduced the concept of *k-anonymity*, which ensures that each information request contains at least k (anonymized) individuals with the same values, so that it is not possible to identify one individual in particular.

Query log publishing and sharing for *academic research* corresponds to a valid and extremely important need of the scientific community. Without this, only companies that own logs will be able to conduct any usage-related scientific experiments. Additionally, there are several other important "query log retention rationales," discussed in the survey article presented by Cooper [2008]. These retention rationales include: improvement of ranking algorithms, improvement of language-based features, query refinement and personalization, combating fraud and abuse, and sharing data for marketing and other commercial purposes.

User privacy in search engine query logs has become a subject of research very recently, among other things, in response to the privacy breaches detected in the anonymized query log published by AOL [Arrington 2006]. This query log was made public for research, but due to insufficient anonymization it ended up revealing user-private information. Following this motivation, Kumar et al. [2007] propose query tokenization for query log anonymization and apply a secure hash function upon each token. However, they show that even this anonymization does not guarantee privacy and explain how statistical techniques on a reference log can still be used to disclose private information.

Jones et al. [2007] provide a detailed description of a data analysis process that leads to information disclosure in a query log. They show how the combination of simple classifiers can be used to map a series of user queries into a gender, age, and location, showing that this approach remains very accurate even after personally identifying information has been removed from the log. They emphasize that a user can be identified by a real-life acquaintance; this type of person has background knowledge on the user (e.g., location, age, gender or even access the user's browser) and can use it to disclose the activities of the user in the log.

Adar [2007] elaborates on vulnerabilities in the AOL log and shows that traditional privacy preservation methods do not transfer directly to query logs.

Adar also points out that k-anonymity is too costly for query log anonymization, because this type of dataset changes very rapidly. Two user anonymization methods are proposed, whose goal is to balance the achieved privacy and the retained *utility*, i.e. the usability of the anonymized log for statistical analysis. Additionally, the survey by Baeza-Yates et al. [2010] discusses challenges of query log privacy preservation and their relation to k-anonymity.

The term *utility* refers to the *data utility* of the anonymized log for the purposes of non-adversarial information acquisition. Verykios et al. [2004] count utility as one of the important features for the evaluation of privacy preserving algorithms, next to the *performance* of the algorithm, the *level of uncertainty* with which the sensitive information can be predicted and the *resistance* to different data mining techniques. In particular, we model the utility of our log anonymization approach based on the information loss, as explained in Section 5. More information on general privacy preservation for data publishing is gathered in the survey of Chen et al. [2009].

All of the previously mentioned studies on query log privacy concentrate on the protection of user privacy. To the best of our knowledge, we are the first to study the dangers of a privacy breach in an independent area. The disclosure of confidential information about companies can be achieved by exploiting query keywords, clicked URLs and their rank positions in combination with the contents of the companies' Web sites. This information is not necessarily related to the disclosure of data about users who issued queries and visited URLs. Therefore, advances towards protecting user privacy do not guarantee business privacy. From now on, we will prefer the term *confidential*, over the term *private*, to refer to the protection of non-public information about an institution.

3. FRAMEWORK FOR BREACHES ON CONFIDENTIAL INFORMATION

Our model for confidentiality breaches through adversarial attacks covers the following aspects: (i) types of adversaries, which are defined in terms of their goals and the amount of information they own; (ii) different sources of information that can be combined and exploited; (iii) vulnerabilities found in the query log; and (iv) several attacks that can be performed by adversaries to obtain confidential information.

3.1 Defining Privacy for Businesses

Before specifying which information is considered *private* or *confidential* for a company on the Web, we first elaborate on the notion of private information for users on the Web. Many people have a personal Web site or Web page in which they publish a large extent of information about themselves. Other users are very careful about not revealing personal information. It is generally understood that what a user publishes is a matter of his or her own judgment. For example, some people put pictures of themselves on the Web, while others do not. Some employees may add a link from their company's Web page to their personal Web page and vice versa. Some users may list their hobbies in their page at their employer's site and some students may do alike in their

university's Web page. Hence, what a person considers private varies from person to person.

Similarly, many institutions have Web sites. What an institution chooses to makes public in their Web site varies substantially. For example, some companies publish the number of hits on their Web site and others do not. Some companies maintain official blogs for interaction with their customers and with any interested users, and others may publish frequently asked questions. In relation to our example in Section 1, many companies decide to provide detailed periodic reports of their sales per channel, as a refinement of periodic reports that they publish. On the other hand, other companies share these figures only with their employees. For instance, whether a company decides to publish statistics about failed sales opportunities (contacts with customers that failed to result in a sale) is purely a matter of institutional judgment.

In this study, we define as "private" or "confidential" *all* of the information about a company which (a) is not published data and (b) cannot be concluded trivially by combining publicly available information. In a similar way, we say that data or information about a company becomes *public* when (i) it is published by the institution itself or (ii) it is published by some entity that is legally authorized (or obliged) to do so. According to these definitions, a person's office phone number is public if the person itself publishes it or if their employer does it (and is authorized to). The number of hits on a company's Web site is public if the company itself publishes them or someone with appropriate authority does. Such an authorization for publishing may be given by the institution itself; for example, a company may authorize a third-party to publish their annual sales in the Web as part of a service agreement.

Following the previous definitions, a *confidentiality breach* occurs if private information is disclosed in *one* of the following ways: (i) Nonauthorized publication of private information, including the publication of anonymized information that can be de-anonymized, or (ii) nontrivial combination of public and private information, where the latter also includes the use of personal expertise. We concentrate on the first case and focus on confidentiality breaches that occur through the deanonymization of an anonymized query log (or part of it). However, as we will see, such a breach usually involves the combination of multiple sources, both private and public ones, as mentioned in the second case.

According to this definition, the publication of the anonymized AOL log [Arrington 2006] is a confidentiality breach. Referring to our example in Section 1, the disclosure of the sales for company *X*'s car *is* a confidentiality breach *unless X* has authorized the search engine to publish their query related traffic data.

3.2 Adversaries

Monitoring competitors' activities within a business domain (or market) is a legitimate and necessary task for a company. Decision makers use tools such as SWOT analysis on their businesses (SWOT stands for Strengths, Weaknesses,

Opportunities, and Threats) and Knowledge Maps [Zack 1999] (which positions a company regarding its knowledge assets and expertise). These types of analysis require an understanding of the competitors in the business. The term *Competitive Intelligence (CI)* is used to describe the activities undertaken by a company to acquire information about its competitors. This information is important for the company's own strategic decisions. For a short overview on CI we refer the reader to the study of Vedder et al. [1999]. The role of public sources for CI and their analysis through data mining is discussed by Zanasi [1998] and Vedder et al. [1999].

Monitoring the activities of companies is not a task performed only by competitors. For example, investment consultants and institutions that perform market studies are agents that regularly collect data about companies' activities and extract knowledge from it. They gather, among other things, risk and growth indicator values.

It must be noted that information acquisition about a company's activities is a legal and well-expected operation for both types of agents (competitors and third parties) as long as this information is obtained through legitimate means. This also holds for the analysis of publicly available documents, such as those that appear in a company's Web site, and knowledge extraction from a published query log. For example, submitting artificial queries to a search engine with the purpose of identifying them later in the published query log is an activity whose legitimacy is less clear.

We consider two types of agents that are interested in extracting confidential information from a Web site.

—*Competitor adversary*. This agent tries to disclose confidential information about its own competitors. Usually, this agent already has background knowledge about the market share, product portfolio and its competitors' activities (such as research, marketing, and alliances). This background knowledge can be combined with an anonymized public query log. Furthermore, an important data source that this adversary can exploit to disclose confidential competitor information is the private log of its own Web site. As discussed in Section 3.5, this private log can be juxtaposed to the public query log for deanonymization purposes.

—*General adversary*. This agent tries to collect confidential information from arbitrary Web sites, without having a particular *target* site in mind. This adversary is the counterpart of a company that collects publicly available information to perform market studies, investment consulting, or search engine optimization for Web sites.

We refer with the term *adversary* to all of the agents that attempt to disclose confidential Web site information through the combination of data from a published query log and other sources. We use the term *attack* to refer to a sequence of activities that result in the disclosure of confidential information. We stress that our use of both terms, despite their apparent negative connotation, do not imply on their own an illegitimate action. Consequently attacks performed by adversaries are not necessarily illegal in this context.

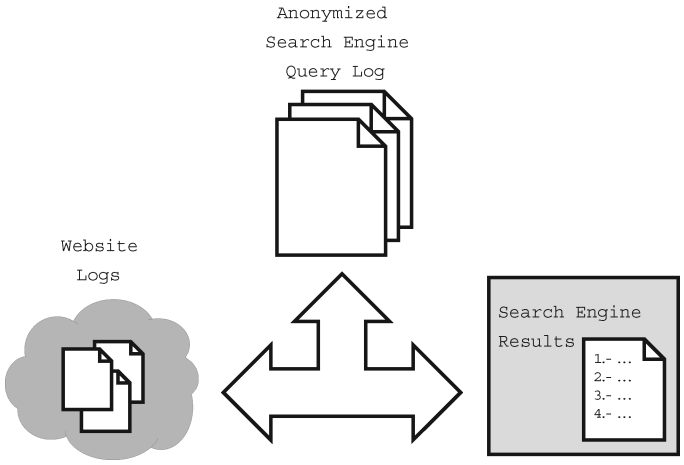

Fig. 1. Data sources that can be exploited in confidentiality breaches.

3.3 Data Sources for Attacks on a Query Log

Our framework for attacks on a public query log considers three main data sources, shown in Figure 1.

—*Anonymized query log*. This is a search engine query log which is made publicly available. It contains confidential information related to people and institutions and is anonymized according to a particular anonymization scheme.
Log releases can take place periodically, this is the case we focus on. We denote the release published at time point i as L_i. The released logs do not need to be consecutive, nor need to be of the same size.

—*Search engine query results*. These are live results that the adversary agent obtains by issuing queries on-line on the search engine.
If the adversary intends to exploit live results, then it must issue queries periodically, so that a set of live results R_i is available for the log L_i at the moment it is released. If the queries are not known in advance but are obtained through inspection of L_i, then there is always a time gap between the live results and the query log. However, most of the queries can be predicted beforehand using the adversary's background knowledge, or leaned fairly quickly when the log is released. Even with a large delay between the search results and the query log, many of the most relevant results are preserved.

—*Web site log*. This is the log S_i of the adversary's Web site (registered by their Web server), or of some other site that is available to the adversary. This log contains private information, to which the adversary has authorized access. Since Web servers record data continuously, the log S_i can be always aligned with L_i.

3.4 Vulnerabilities in the Query Log

In our model for confidentiality breaches we assume that the query log has the following signature, which is very similar to that of the AOL log:

```
ANONuserId, query, timestamp, ANONclickedURL
```

3.4.1 *Basic Anonymization.* We assume that the query log contains one entry for each URL (or document) that is returned as a result for a query *and* then clicked by the user. The prefix `ANON` indicates that the field is anonymized by default. Therefore, the user identifier and the clicked URL are anonymized, while the query keywords and the timestamp of the request are not. Although more fields can be anonymized in this signature this reduces the usefulness of the log for analysis. In this work, we assume that the anonymization of a query log entry does not go beyond these already masked fields. Nevertheless, we apply further anonymization steps to the log as a whole.

Concerning the field `ANONclickedURL`, we note that there are many ways of anonymizing a URL. In the AOL dataset [AOL], URLs are truncated to the site's name (i.e., up to the first "/"). Instead, we assume that the `ANONclickedURL` consists of the *anonymized* identifier of the site and the *anonymized* identifier of the document *within the site*. This means that two documents `doc1.html` and `doc2.html` of site `www.somesite.com` would be anonymized into `anonsiteA.X` and `anonsiteA.Y` respectively, where `anonsiteA` is the anonymized ID of the site and `X,Y` are the anonymized identifiers of the two documents. This provides a rudimentary protection of the identity of the sites and their contents while allowing analysis which differentiates among different sites and considers the size of a site. Therefore, the new signature of the query log for our analysis is:

```
ANONuserId,query,timestamp,ANONsite.ANONclickedURLinSite.
```

3.4.2 *Vulnerable Queries.* For query log anonymization, we concentrate on the identification and removal of what we consider to be *vulnerable queries*. We consider *vulnerability* in the context of k-anonymity [Sweeney 2002]. This means that attacks are successful if they acquire less than k objects from the log and that anonymization must ensure that queries associated with less than k objects do not appear in the published dataset. The value of k is determined by a human expert. In Section 5, we study the impact of the values of k on the size and the amount of information retained in the query log. We consider two types of vulnerable queries.

—*Overrestrictive queries*. These are queries that return less than k documents in the search engine. These queries present a privacy breach because their set of anonymized URLs can be mapped to k, or less, *real* URLs. To deal with this issue, one may either eliminate these queries or generalize them. Generalization can be implemented, for example, by replacing the query with more abstract terms and merging all of the results that satisfy these terms. The initial approach taken in this study is to solve this vulnerability by eliminating these queries from the log (known as *suppression* [Chen et al. 2009]).

—*Well-targeted queries*. These are queries that contain the target URL, or at least the site of the URL, as a keyword. In this case, these queries are a subcategory of *navigational queries* discussed by Broder [2002]. The user knows some part of the target URL, for example the site name and/or some part of the URL's name. The user submits this information in the search engine to obtain the exact URL of the target page, essentially using the search engine as a bookmark manager. A navigational query which fully discloses a site is obviously vulnerable. For example, the query term `amazon.de` points to a specific site in most major Web search engines.

From the viewpoint of privacy preservation, a navigational query that does not fully disclose the site may still be vulnerable. This is the case if the distribution of clicks among the returned URLs is highly skewed towards the same site. For example, if most of clicks to the query `london transport` were found to go to the site `http://www.tfl.gov.uk/` (the homepage of London transport), then the query would be considered vulnerable.[2]

The reason for the vulnerability of well-targeted queries is that an adversary can use background knowledge to de-anonymize the URLs in the log. This background knowledge can be acquired in two ways: First, the adversary may use another third-party published query log,[3] where the sites are not properly anonymized, and derive the click distribution from that log. Second, the adversary may issue the vulnerable query in the live search engine, from which the published query log comes from, and study the distribution of returned (not clicked) URLs among the sites. Although not as accurate as the first approach, this distribution might allow an approximation of the unknown click distribution. These adversarial approaches are *probabilistic background knowledge* or *similar data* attacks [Chen et al. 2009]. A solution for this kind of vulnerability can be the removal of these queries or anonymize their keywords.

Even after the elimination of over-restrictive and well-targeted queries, the query log may still be vulnerable. Next, we describe two attacks against an anonymized query log: The first attack exploits properties of *pairs of queries* and calls for a more elaborate log cleaning method, proposed in Section 4. The second attack exploits the adversary's own Web site log, that is, a source of background knowledge that is beyond the control of the owner of the query log.

3.5 Attacks on an Anonymized Query Log

We introduce a first attack on an anonymized query log, assuming the data sources presented in Section 3.3. This attack juxtaposes the published query log $L \equiv L_i$ with the query results of the live search engine $R \equiv R_i$. Then we present two variations of this *linking* attack based on the incorporation of additional background knowledge or data from the adversarial side. In these attacks we assume that over-restrictive and well-targeted queries have already been eliminated from the log.

[2]The click distribution can be built by analyzing the query log.

[3]The published AOL query log, taken offline and which should not be used for analysis, may be misused by a malicious party to reconstruct the click distribution.

ATTACK 1:

(1) Define a set of queries $setQ$
which are known to return URLs of the target competitor's Web site in high-ranking positions.

(2) Submit $setQ$ to the on-line search engine,
collect the results acquiring the $liveResults(q)$ set for each query $q \in setQ$.

(3) Find $setQ$ in the anonymized query log L and
obtain for each $q \in setQ$ the anonymized identifiers of its clicked URLs $clickedURLs(q)$.
The next task is to map anonymized identifiers to the live results.

(4) Build the matrix of intersecting queries M:
the cell $M[i, j] = clickedURLs(q_i) \cap clickedURLs(q_j)$ for each $q_i, q_j \in setQ$.

(5) Clean the symmetric matrix M
by removing all but each upper part and by eliminating cells with empty intersection.
The result is a list of lists $\mathcal{M}$ where $\mathcal{M}[i]$ is the list of nonempty intersections on q_i and queries with index larger than i. Then, $\mathcal{M}[i][j] = clickedURLs(q_i) \cap clickedURLs(q_j)$ for $j > i$.

(6) For each pair of queries (q_i, q_j), $j > i$ with $\mathcal{M}[i][j] \neq \emptyset$
compute the list element $\mathcal{L}[i][j] = liveResults(q_i) \cap liveResults(q_j)$.

(7) Traverse the lists $\mathcal{M}, \mathcal{L}$ and
juxtapose them to deanonymize URLs:
—If $|\mathcal{M}[i][j]| = |\mathcal{L}[i][j]| = 1$,
then the anonymized identifier $ANONu$ constituting $\mathcal{M}[i][j]$ corresponds to the deanonymized URL u that constitutes $\mathcal{L}[i][j]$.
Remove u from all entries in $\mathcal{L}$ and $ANONu$ from all entries in $\mathcal{M}$.
—If $|\mathcal{M}[i][j]| > 1$,
then proceed to the next j and then to the next i.

3.5.1 *ATTACK 1: Combining Queries.* This attack exploits pairs of queries which have a nonempty intersection between their *clicked* result sets in the query log. For some pairs of queries q, q' it is likely that a search engine returns overlapping sets of URLs, that is, $results(q) \cap results(q') \neq \emptyset$. For example, this could be the case for different queries that contain one or more common keywords. However, these queries become vulnerable *only* when $clickedURLs(q) \cap clickedURLs(q') \neq \emptyset$. Our experimental evaluation shows that this situation is rare but not negligible.

First we assume that the agent launching the attack is an *competitor adversary* who tries to extract confidential information about some specific Web sites. Later we show how this attack can be extended to a *general adversary*.

The competitor adversary attack is defined in ATTACK 1. This attack combines the background knowledge of the competitor adversary about queries that retrieve target pages, the published query log and the live search engine. The adversary collects live URLs from the search engine (Step 2) and locates the anonymized URLs retrieved by the same queries from the published query log (Step 3). Then, the queries with nonempty intersection of clicked results are identified (Steps 4 and 5) and juxtaposed to the displayed results from the live search engine (Steps 6 and 7).

Step 7 of ATTACK 1 repeatedly juxtaposes the intersection of anonymized clicked results to the intersection of live results for each pair of queries. As soon as an intersection has only one element it can be de-anonymized. Then, the URL and its anonymized identifier are removed from the lists, which in turn become smaller. The attack stops when no more removals take place.

The success of ATTACK 1 depends on satisfying the first condition in Step 7. If there is no pair of queries, whose intersection of clicked results contains only one record, then the attack cannot start. If the counterpart intersection of displayed live results is much larger, then the attack may fail. A couple of remarks are due in this context: (1) We make the assumption that the likelihood of finding *one* pair of queries with a single common URL is not negligible, this is verified by our experimental results in Section 5. Even if the condition does not hold for a *pair* of queries, it may hold when combining 3, 4, 5, ... queries. As soon as the attack starts, URLs may be disclosed. (2) A set of displayed results is usually much larger than the set of clicked results. However, the adversary may exploit publicly available knowledge about the clicking habits of search engine users and reduce the sets of displayed results only to the first few entries for each query. As before, the adversary may combine three or more queries to reach an intersection that has only one element.

A variation of this attack can also be performed by a *general adversary*. This agent has no background knowledge to perform Step 1, but is still able to identify all pairs of queries that have a nonempty intersection of clicked results (Step 4). Then, these queries can form a (fairly large) set of queries $setQ$, which is launched against the live search engine to build the sets of displayed results. Once these sets are built, the mission-critical Step 7 of the attack can be launched.

3.5.2 *ATTACK 1a: Exploiting a Private Web Site Log.* According to the data sources shown in Figure 1, additionally an adversary can exploit the private Web server log of a given site. An competitor adversary is likely to own this type of log—the log of its own Web site. This extra source of information can be used to enhance ATTACK 1.

A site's Web server registers all of the requests sent to the site. A typical a log entry contains, at least, the target URL, the time of the request, the user agent that generated it, and the IP address of the user. Furthermore, most Web site logs also include the *referrer URL*, that is, the URL from which the request to the site was initiated. If the referrer is a search engine's result page, then

the referrer URL contains the keywords of the query used to retrieve the target URL from the results.

This data can be used to deanonymize information from the published query log. In particular, the adversary can find the queries and clicked URLs recorded by its own Web server. Then, they can match the timestamps in the server to the timestamps in the published query log and finally match the anonymized identifiers of the clicked URLs to their own URLs. This can be done easily even if the times on both logs are not synchronized, for example, using two consecutive request with different queries to the Web site.

The "benefit" of matching a Web site log with the published query log is a twofold: First, the adversary can attempt to reengineer the algorithm used for URL anonymization. Second, the adversary use the already de-anonymized URLs in Step 7 of ATTACK 1, increasing the probabilities of disclosing its competitors anonymized URLs. This attack can become a greater threat when many Web sites collude joining their query logs.

To avoid the consequences of ATTACK 1a one more constraint should be placed in the anonymization process of the query log: The results displayed by the search engine for any given query must contain URLs of at least k different sites, so that k-anonymity can be pursued.

3.5.3 *ATTACK 1b: Exploiting Disclosed User Data.* Many adversarial attacks on anonymized query logs, discussed in related work, aim towards identifying the actions of a particular user. We study a variation of ATTACK 1 in which a search engine user can be exploited to disclose confidential information in the query log. In particular, if the adversary identifies a (single) user in the query log it can then acquire knowledge about the results clicked by this user. With this information the adversary can trace the user and its clicked URLs in the published query log and map the anonymized URLs to the real ones. Similarly to ATTACK 1a, these de-anonymized URLs can be used in Step 7 of ATTACK 1.

In periodical releases of an anonymized query log, this attack can be automated with an agent that submits queries to the search engine regularly. This user is then traced when the query log is published.

This attack can be prevented by avoiding the identification of a particular user with conventional privacy preserving methods for query log anonymization. As pointed out in Section 2, there are already efforts towards this problem.

4. QUERY LOG ANONYMIZATION

We propose a heuristic method for the anonymization of a query log. It is meant to reduce the vulnerabilities identified in Section 3.4 and prevent the attacks described in Section 3.5. Our approach is based on the removal of the objects that cause the vulnerabilities, that is, of vulnerable queries and queries that generate nonempty result intersections. We study the resulting data after the anonymization process to measure the amount of retained information in the log.

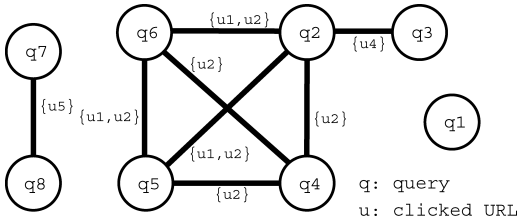

Fig. 2. Example graph representation of a query log.

4.1 Heuristics for Log Anonymization

Our approach is based on a graph representation of the query log, in which all instances of a query are modeled as a node and two nodes are connected (by an undirected edge) when the intersection of their clicked URLs sets is not empty. In other words, a node represents the aggregation of all the occurrences in the query log of a particular query, and the clicked results are all of the URLs clicked by users for instances of that query. This type of query log graph is discussed in Baeza-Yates [2007]. A simple toy example of this model is presented in Figure 2.

The conditions needed to prevent ATTACK 1 can be formalized into a well-defined optimization problem on the query log graph. The idea is to disconnect the graph while trying to preserve the most *important* nodes. We associate the importance of a node with the frequency of the query, as discussed in Section 4.2. We model node importance as a weight, so our problem translates into that of preserving the *maximum weighted graph* or the *maximum (weighted) independent set*. This is an NP-hard problem, therefore we use a heuristic approach to solve it. We begin by defining a density measure for the graph, then we solve a baseline case in which all node weights are equal to 1 and later we introduce variations based on the relevance of each node.

Definition 1 (*Graph Density*). Let $G(V, E)$ be the graph of a query log, where V is the set of queries in the log and E is the set of edges. Two nodes $v, v' \in V$ are connected with an undirected edge $(v, v') \in E$ if and only if $clickedURLs(v) \cap clickedURLs(v') \neq \emptyset$.

The *density* of the graph is the likelihood of finding an edge among any two nodes and it is defined as:

$$Density(G) = \frac{2 \times |E|}{|V| \times (|V| - 1)};$$

that is, the graph density is based on the cardinality of the set of edges and the set of nodes.

We use a greedy heuristic to solve the baseline case according to the concepts of graph *density* and the *degree* of each node. This heuristic is defined as follows:

Graph Disconnection Heuristic:

Input: Graph of the query log $G(V, E)$

Output: Disconnected graph

(1) Sort V on node degree.
(2) Identify the node with the highest degree, v_{max} and the set of its neighbors $X := \{x|(x, v_{max}) \in E\}$.
(3) Remove v_{max}, i.e. set $V := V \setminus \{v_{max}\}$
(4) Remove the edges involving v_{max}, i.e. set $E := E \setminus \{e|e = (x, v_{max})\}$.
(5) Recalculate the degree of all nodes in X.
(6) Recompute $Density(G)$.
(7) If $Density(G) \neq 0$ then go to Step 1, else finish.

The graph disconnection heuristic eliminates the query node with the highest degree first, that is, the one involved in the most nonempty intersections of clicked results. This generates the removal of many edges. The heuristic proceeds gradually, until the graph is fully disconnected. This heuristic can be relaxed to stop when the value of Density reaches a preestablished minimum threshold.

4.2 Variations on the Graph Disconnection Heuristic

This basic heuristic takes only the connectivity of a query-node into account. However, since one goal of the anonymization is to allow statistical analysis over the anonymized published log, it is reasonable to model the "importance" of the eliminated queries and to remove less important queries first. In this work we will consider that the importance or *weight* of a query can be represented by its frequency or its number of clicked documents. However, the weight can represent any other measure of the significance of the queries.

This leads to two further variations of the basic graph disconnection heuristic, each of which sorts nodes based on a different property (Step 1 of the heuristic).

—*Method 1*. The property used for sorting is the degree of the nodes; this is the basic heuristic.
—*Method 2*. The property used for sorting is the degree multiplied by the inverse frequency of the query in the log, i.e. $\frac{degree(v)}{frequency(v)}$ for $v \in V$.
—*Method 3*. The sorting property is the degree multiplied by the inverse number of clicked documents for the query, i.e. $\frac{degree(v)}{clicks(v)}$.

In Method 2, $frequency(v)$ is the number of times that any instance of query v has been submitted to the search engine. In Method 3, $clicks(v)$ is the number of times any document in $clickedURLs(v)$ was clicked as a result of v.

4.3 Extending towards *K*-Anonymity

The graph disconnection heuristic and its variations focus on eliminating queries that share URLs among their clicked results. Although the heuristic disconnects the graph, anonymity is not yet guaranteed. In particular, the heuristic does not consider the number of results displayed by the search engine for each query. This number is important to prevent attacks to over-restrictive queries. Furthermore, if all URLs returned for a query belong only to one site or just a few sites, then the adversary may be able to disclose them as discussed in ATTACK 1a.

The previous observations can be mapped into requirements for k-anonymity. In particular, a query must display at least k results in the search engine, which come from at least K different sites ($k \geq K$).

Thus, the complete anonymization process encompasses the removal of (a) all vulnerable queries, according to Section 3.4, (b) all queries that return less than k documents and (c) those returning documents from less than K sites and (d) all queries that contribute to a nonzero density of the query graph.

This anonymization process inevitably incurs the loss of data and of potentially useful information. We model *information loss* as the decrease in the volume of the query log with respect to (i) the number of queries and (ii) the number of clicks. In the following section, among other things we study how different values of k and K influence the retained volume of the query log and how the log shrinks with each of the variations of the graph disconnection heuristic.

5. EXPERIMENTS

We study the behavior of the graph disconnection heuristic on a real query log. The goal of the experimental evaluation is to gain insight on the process of log anonymization and the information loss that it causes. First we present the log used for the experiments and report some of its characteristics. Then, we study the behavior of the three graph disconnection variants presented in Section 4. Finally, we elaborate on the effects of pursuing k-anonymity, that is, of eliminating queries that are associated with less than k clicked documents or less than K sites. In this evaluation we do not analyze the specific case of well-targeted queries, since we consider that it requires more extended evaluation. Nevertheless, we consider all of the other previously mentioned attacks.

5.1 The Dataset

We performed our experiments on a query log sample from the Yahoo! UK & Ireland search engine. The sample contains consecutive request registered by the search engine for a certain period of time.[4] For the purpose of this evaluation we did not use the raw log data but rather worked with its graph representation. Since our goal is to study log anonymization through graph disconnection heuristics this graph is appropriate for this task.

[4]The extension and date of the log is omitted to preserve the confidentiality of the data.

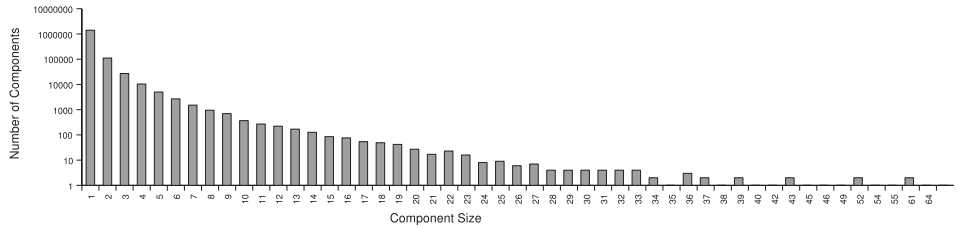

Fig. 3. Size distribution for the connected components of the query log graph.

The query log graph representation is an application of the graph models presented in Baeza-Yates and Tiberi [2007] and can be computed fast. The final graph contains over 3 million nodes and its computation took approximately 2 hours on a dual core AMD Opteron™ Processor 270 with 6.7GB of RAM.[5] The density of this graph is low to begin with, equal to 0.000089.

First, we identified all of the connected components in the graph and computed their size distribution, as shown in Figure 3. Without considering components with only one element, we find that there is a large connected component that includes 70% of the connected nodes. The second largest component found is only 0.01% of the size of the largest one. Also, more than 80% of the clicks to documents in the log proceed from queries in the large component. The density of this component is 0.00075. We focus on studying the effects of the attacks on the largest component, since it involves most of the connected elements in the graph, which are the most vulnerable to attacks. Also the effects on the largest component represent an approximation of the worst case of log volume loss. Since one of the worst scenarios of anonymization, according to our heuristics, is for a query log graph that is completely connected.

Next, we analyzed the distribution of the node degrees in the large component. If the number of edges per node were to follow a power law, then this would indicate that the edges can be reduced quickly by node removal: The graph would become disconnected very fast by removing a few high-degree nodes [Albert et al. 2000]. However, as shown in Figure 4, the degrees do not follow a power law, which does not guarantee that quick graph disconnection can be achieved.

5.2 Disconnecting the Graph

We applied the three variations of the basic heuristic presented in Section 4 and studied the decrease in the size of the largest component in the log. We consider the reduction of the volume of the log as an initial approximation of the information loss induced by the anonymization heuristics.

We show the effects of the gradual removal of nodes on the volume of *retained queries* (Figure 5) and on the *remaining document clicks* (Figure 6). In this experiment we define as:

—*retained queries* the sum of the frequencies (in the query log) of the queries represented by the remaining nodes, and

[5]The algorithm uses only one of the CPUs and less than 4GB of RAM.

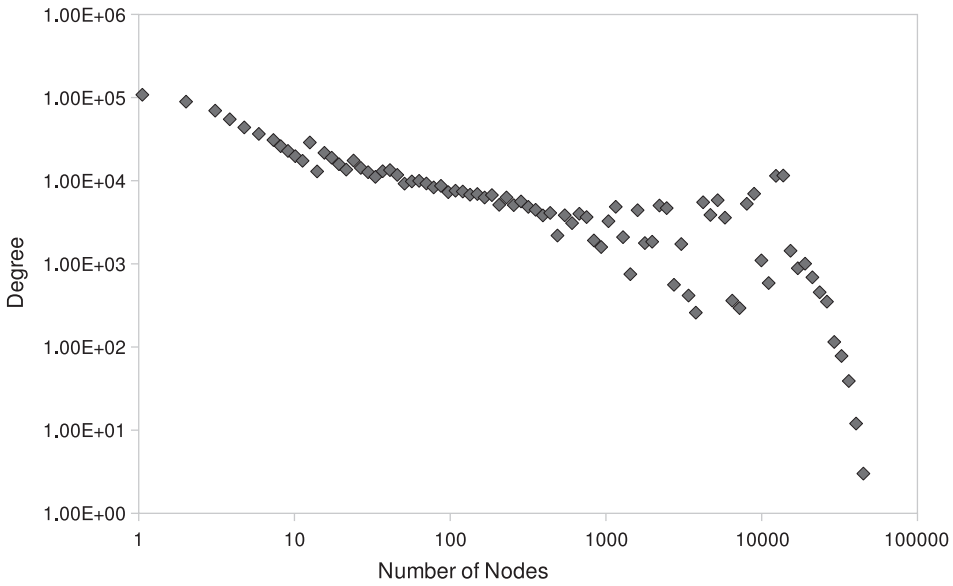

Fig. 4. Degree distribution in the largest connected component of the graph.

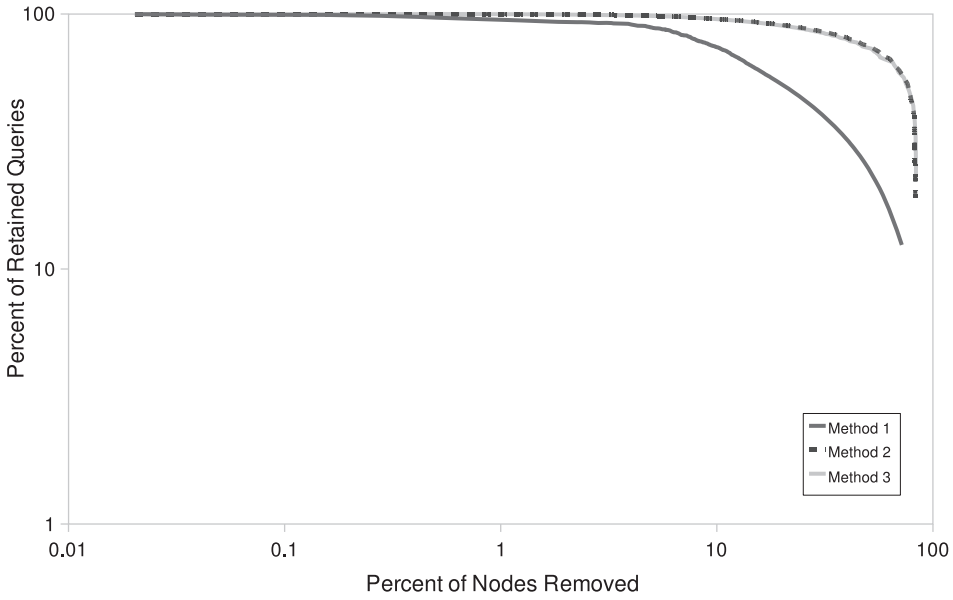

Fig. 5. Percent of retained queries during graph disconnection.

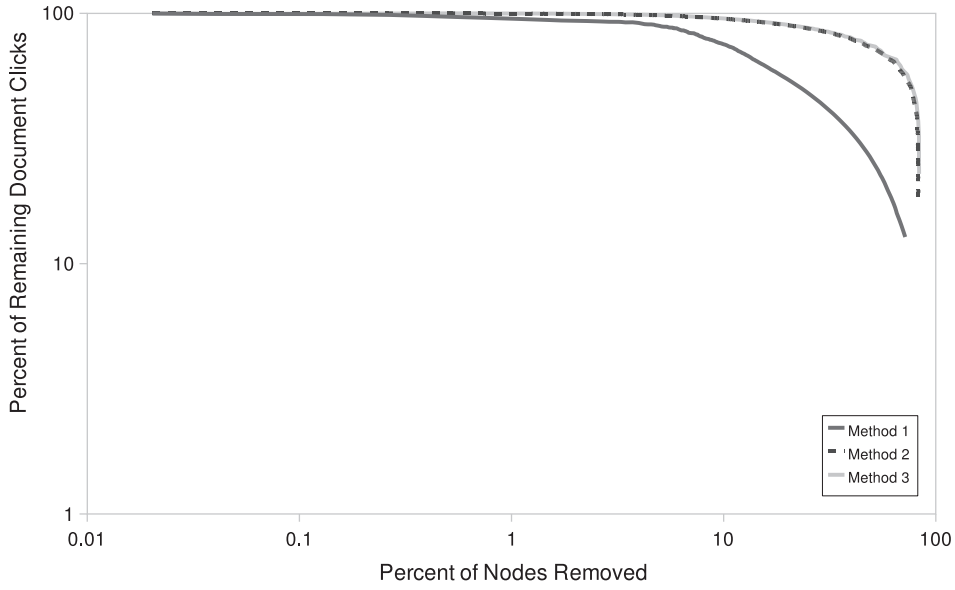

Fig. 6. Percent of remaining document clicks during graph disconnection.

—*remaining document clicks* as the sum of the clicks to documents from the queries retained in the log.

In Figure 7, we show the decrease in the number of retained edges as the percentage of removed nodes approaches 100%. The number of retained edges does not serve as a utility indicator, since all edges must be removed anyway. It rather indicates the speed at which the graph gets disconnected. Between 70% and 80% of the query nodes need to be removed in order eliminate all of the edges. The effect of the removal of nodes on the value of the density is presented in Figure 8. The complete disconnection of the graph component dramatically affects the number of nodes remaining. Nevertheless, the value of the density in the large component can be decreased to under 0.000038 removing less than 10% of the nodes. This value significantly decreases the number of edges, to only a 6.3% of their original value, while preserving over 95% of the retained queries and document clicks, according to Methods 2 and 3 in Figures 5 and 6.

The results show that Methods 2 and 3 perform better than Method 1. These methods consider both the degree of a node and its inverse weight, thus favoring the elimination of highly connected infrequent queries. As we can see in Figure 5 and Figure 6, Method 2 and 3 remove less queries and clicks than Method 1. So, they produce an anonymized log with larger volume.

5.3 The Impact of *K*-Anonymity Enforcement

As pointed out in Section 4.3, queries that *display* less than k results in the search engine violate the k-anonymity requirement for conventional

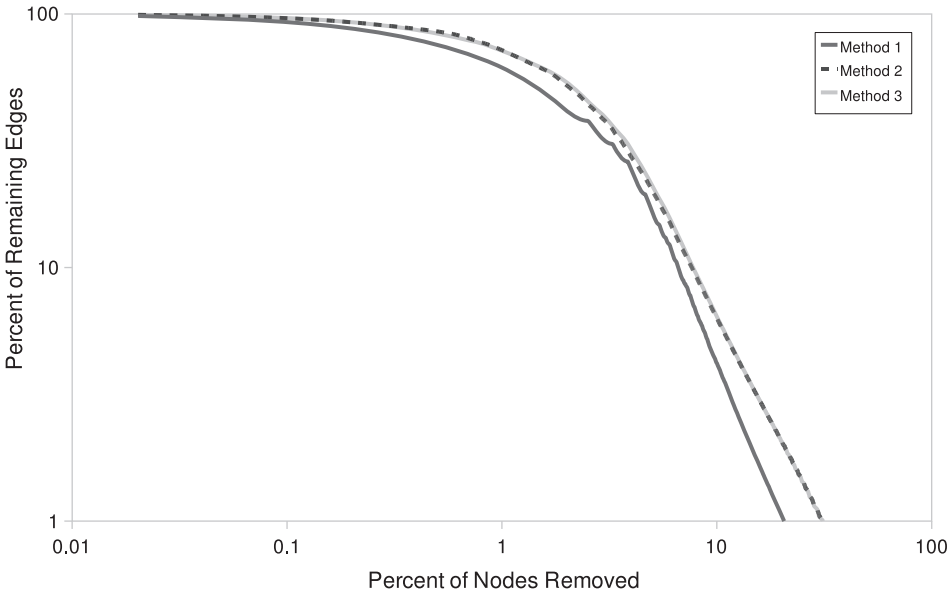

Fig. 7. Percent of retained edges during node removal heuristics.

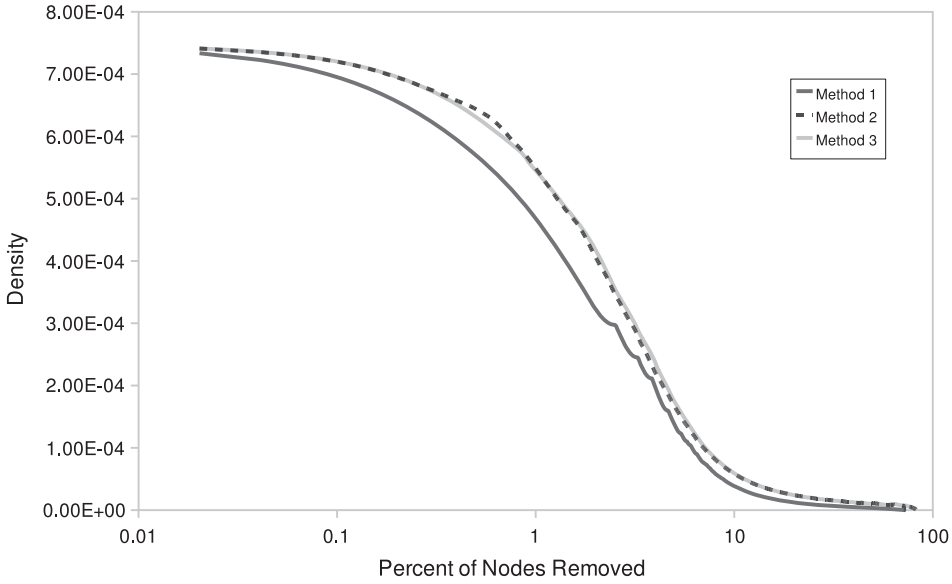

Fig. 8. Density during node removal heuristics.

anonymization. Even if a query returns more than k documents, a confidentiality breach may occur if the documents come only from very few sites. So, we consider anonymity enforcement with respect to k results and to K sites, where $k \geq K$.

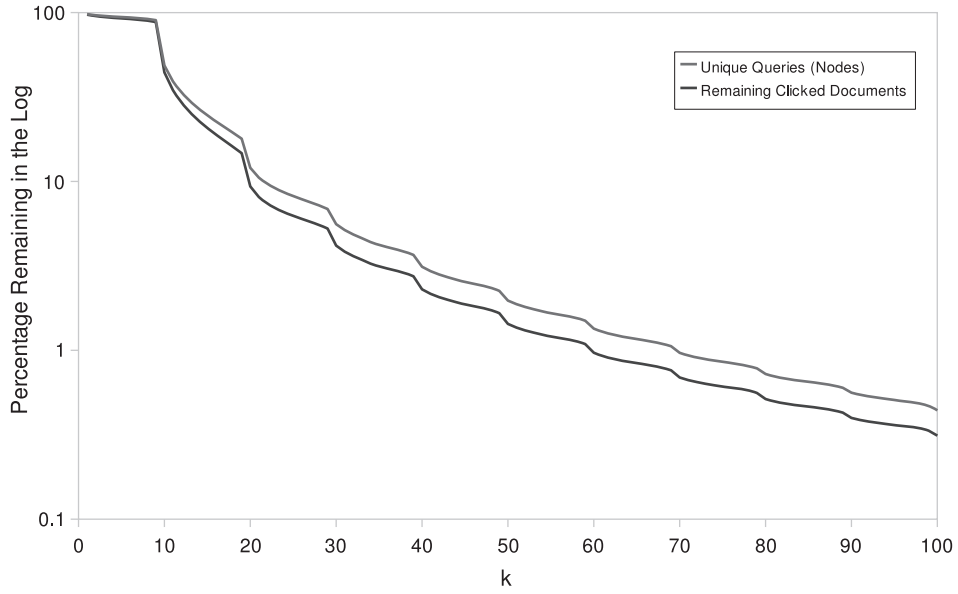

Fig. 9. Log volume decrease when removing queries with k or less results.

Similarly to the graph disconnection methods, we measure the impact of anonymity enforcement on the basis of retained queries and remaining clicked documents. Note that when removing a query, this eliminates all of the instances of that query in the log. This includes all of the documents clicked as a result of this query.

5.3.1 *Removing Queries That Display Fewer than k Results in the Search Engine.* In Figure 9, we show the decrease of the log volume (percentage of retained queries and clicks) as we remove queries with $n < k$ displayed results. We are particularly interested in queries that return fewer than 10 results, because 10 is the default number of URLs for the first page of search engine results: For queries returning more than 10 results, we do not know how many results were available, only the number of results *displayed* or shown to the users. On the other hand, for a query with fewer than 10 results we know that these were the *only* available results, and that they were all displayed to the user in the first results page.

In Table I, we show that the removal of queries with $k \leq 7$ displayed results does not affect the overall volume of the log, because these queries are not frequent nor have many clicks. We can see in the table that the largest portion of the log refers to queries with $k \geq 10$. Therefore enforcing k-anonymity for $k = 7$ would not affect very much the log volume.

5.3.2 *Removing Queries with Results from Less than K Sites.* In Figure 10, we show the decrease of the log volume (percentage of retained queries and clicks) as we remove queries that return results from K or less Web sites.

Table I. Log Volume Decrease when Removing Queries with k or Fewer Results for $k \leq 10$

k Results	Percent of Retained Queries	Percent of Retained Clicks
1	97.9	97.2
2	96.5	95.5
3	95.5	94.1
4	94.6	93.1
5	93.9	92.2
6	93.3	91.4
7	92.7	90.7
8	91.9	89.9
9	90.2	87.9
10	48.6	44.2

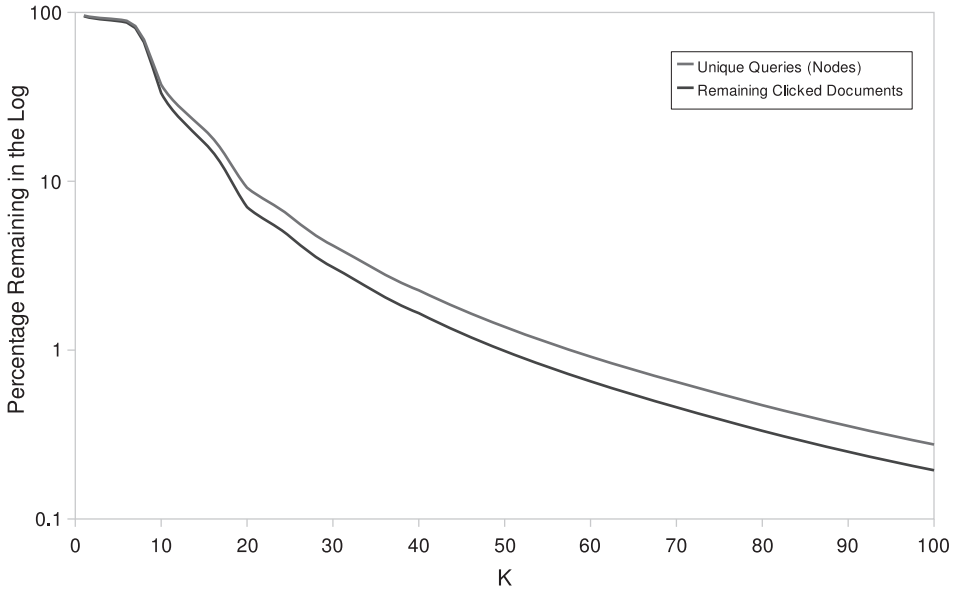

Fig. 10. Log volume decrease when removing queries with K or less sites.

Table II shows in detail how the removal of queries with $K \leq 3$ different Web sites does not affect the overall volume of the log, as these cases are neither frequent nor have many clicks. In Table II we can also see that the largest portion of the log refers to queries with $K \geq 8$.

6. CONCLUSIONS AND OUTLOOK

We have presented a new challenge for privacy preservation in query logs. While most of the research on privacy preservation in this context focuses on protecting private data about users, we show that the protection of confidential information of business institutions is an independent and no less challenging issue. We have formulated this new issue by showing types of adversaries and explaining attacks towards a published, anonymized query log. Then we have

Table II. Log Volume Decrease when Removing Queries Referring to K or Fewer Sites for $K \leq 10$

K Web Sites	Percent of Remaining Queries	Percent of Remaining Clicks
1	95.7	94.9
2	93.8	92.5
3	92.5	91.0
4	91.6	89.9
5	90.8	88.9
6	89.1	87.1
7	82.9	80.8
8	69.0	66.6
9	50.8	47.5
10	37.3	33.3

proposed a heuristic approach that removes those queries from the log, which can be exploited in adversarial activities.

We have tested the three variations of our heuristic approach experimentally with a real query log. The variants that sort queries on connectivity divided by query and document click frequencies yield the best experimental results in preserving the most amount of log volume. However, the complete removal of edges in the largest connected component of the graph still has a striking effect on the remaining log volume. This can be improved if the density of the graph is set to an acceptable threshold, reducing significantly the number of edges in the graph while still preserving most of the volume of the log. Our type of graph disconnection mostly involves the removal of infrequent queries. Since such queries are more likely to point to identifiable people or institutions, and since their contribution to statistical analysis is expected to be rather limited, their removal is justified.

It is difficult to estimate accurately the information loss induced by our anonymization heuristics. As a first approach we use an estimation of the retained log volume based on the remaining queries and clicks to documents. It is likely that the anonymization heuristics and information loss can be optimized for different types of applications improving the utility of the remaining log. So far, the anonymized log allows tasks which study accesses to Web sites but that do not require to reveal the identity of a particular site. Data mining applications which analyze rare and infrequent queries will not be able to perform as well.

The protection of confidential information is of vital importance for companies that use the Web as communication medium and as a marketing and sales channel. Although our approach is a first contribution towards confidentiality preservation, many open issues remain. First of all, we have discussed and alleviated specific types of attacks. Different attacks are thinkable and need to be identified, studied and prevented as well. Furthermore, confidentiality preservation is closely associated to k-anonymity: Despite the efforts on preservation of k-anonymity in query logs, there is yet no anonymization method guaranteeing that private information cannot be disclosed.

A perspective worth studying in this context is the role of generalization. Queries can be replaced with some of their keywords, while keywords can

themselves be replaced with more abstract terms. We would like to study whether generalization can mitigate the vulnerability posed by infrequent queries that share clicked results.

ACKNOWLEDGMENTS

The authors thank Alessandro Tiberi from the University of Rome "La Sapienza" and Claudio Corsi from the University of Pisa for providing tools to generate the graph representation of the query log. Also we thank the following people from Yahoo! Research: Aristides Gionis for many valuable discussions and feedback and Carlos Castillo for his help in the query log data preparation.

REFERENCES

ADAR, E. 2007. User 4xxxxx9: Anonymizing query logs. In *Proceedings of the Workshop in Query Log Analysis: Social and Technological Challenges (WWW'07)*.

ALBERT, R., JEONG, H., AND BARABASI, A.-L. 2000. Error and attack tolerance of complex networks. *Nature 406*, 6794, 378–382.

AOL. AOL Research Web site, no longer online. http://research.aol.com.

ARRINGTON, M. 2006. AOL proudly releases massive amounts of private data. http://www.techcrunch.com/2006/08/06/aol-proudly-releases-massive-amounts-of-user-search-data/.

BAEZA-YATES, R. 2007. Graphs from search engine queries. In *Proceedings of the 33rd International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM'07)*. Springer, 1–8.

BAEZA-YATES, R., JONES, R., AND POBLETE, B. 2010. Issues with privacy preservation in query log mining. In *Privacy-Aware Knowledge Discovery: Novel Applications and New Techniques*, F. Bonchi and E. Ferrari, Eds. Chapman and Hall/CRC Press.

BAEZA-YATES, R. AND TIBERI, A. 2007. Extracting semantic relations from query logs. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

BARBARO, M. AND ZELLER, T. 2006. A face is exposed for AOL searcher no. 4417749. *New York Times*.

BRODER, A. 2002. A taxonomy of web search. *ACM SIGIR Forum 36*, 2, 3–10.

CHEN, B.-C., KIFER, D., LEFEVRE, K., AND MACHANAVAJJHALA, A. 2009. *Privacy-Preserving Data Publishing*. Vol. 2. Now Publishers Inc.

CLIFTON, C., KANTARCIOGLU, M., AND J.VAIDYA. 2002. Defining privacy for data mining. In *Proceedings of the National Science Foundation Workshop on Next Generation Data Mining*.

COOPER, A. 2008. A survey of query log privacy-enhancing techniques from a policy perspective. *ACM Trans. Web 2*, 4.

JONES, R., KUMAR, R., PANG, B., AND TOMKINS, A. 2007. "I know what you did last summer": Query logs and user privacy. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM'07)*. ACM, New York, 909–914.

KUMAR, R., NOVAK, J., PANG, B., AND TOMKINS, A. 2007. On anonymizing query logs via token-based hashing. In *Proceedings of the 16th International Conference on World Wide Web (WWW'07)*. ACM Press, New York, 629–638.

POBLETE, B., SPILIOPOULOU, M., AND BAEZA-YATES, R. 2008. Website privacy preservation for query log publishing. In *Proceedings of the 1st SIGKDD International Workshop on Privacy, Security, and Trust in KDD (PinKDD'07)*. Lecture Notes in Computer Science. vol. 4890. Springer.

SWEENEY, L. 2002. k-anonymity: A model for protecting privacy. *Int. J. Uncert. Fuzz. Knowl. Based Syst. 10*, 5, 557–570.

VEDDER, R. G., VANECEK, M. T., GUYNES, C. S., AND CAPPEL, J. J. 1999. CEO and CIO perspectives on competitive intelligence. *Comm. ACM 42*, 8, 108–116.

VERYKIOS, V., BERTINO, E., FOVINO, I., PROVENZA, L., SAYGIN, Y., AND THEODORIDIS, Y. 2004. State-of-the-art in privacy preserving data mining. *SIGMOD Record 33*, 1, 50–57.

ZACK, M. H. 1999. Developing a knowledge strategy. *California Management Review 41*, 125–145.

ZANASI, A. 1998. Competitive intelligence through data mining public sources. *Compet. Intell. Rev. 9*, 1, 44–54.

Received December 2007; revised July 2008, November 2008, February 2010; accepted April 2010