Single-Party Private Web Search

Alexandre Viejo, Jordi Castellà-Roca, Oriol Bernadó, Josep M. Mateo-Sanz
Universitat Rovira i Virgili, UNESCO Chair in Data Privacy
Departament d'Enginyeria Informàtica i Matemàtiques
Av. Països Catalans 26, E-43007 Tarragona, Spain
Corresponding author email: alexandre.viejo@urv.cat

***Abstract*—Web search engines profile their users by storing and analyzing their past searches. Profiles reflect the interests of the users and enable web search engines to offer a better service. In this way, search results are personalized to fulfill the expectations of each individual user. Nevertheless, this service is not provided without cost. User profiles contain information that can be considered private and personal. This represents a serious privacy threat which must be addressed. Several privacy-preserving techniques which try to prevent this situation can be found in the literature. In this paper, we focus on those that work directly in the computer of the users without requiring any external entity. More specifically, we propose a new single-party scheme that addresses the trade-off between privacy and quality of service but it does not require any change at the server side. The performance of this new method has been evaluated using real search queries extracted from the AOL's files. The results achieved show that our proposal works as expected and it can be considered a proper option for those users who are concerned about their privacy.**

I. Introduction

The Internet is probably the most important source of knowledge of the present time. The virtual world offers a huge volume of information which grows every day. In order to find specific data among that miscellany, web search engines (WSEs) like Google or Bing are widely used.

In a usual web search process, users submit their queries to a WSE which in turn process them and returns their related search results. During this process, the WSE stores all those queries together with the identity (IP address, cookies...) of the users who sent them. According to [1], Google anonymizes IP addresses and cookies present in its search engine logs after 9 and 18 months respectively.

A. Cooper in [2] describes seven reasons to store query logs. They are next described:

- *Improving ranking algorithms.* WSEs use the retrieved search results which have been selected by users more frequently to improve the ranking algorithms and, hence, the quality of their future results [3], [4].
- *Language-based application.* User queries provide a lot of information about how people use language. In this way, query logs can be useful to improve characteristics based on the language like correcting queries with spelling mistakes.
- *Query refinement.* This process focuses on suggesting new queries to the user which can provide better search results than the original query submitted by her. Suggestions can be offered while the user is typing her query or also after retrieving poor search results for a certain query submission. In any case, suggestions are presented according to past queries.
- *Personalization.* WSEs provide search results using several result pages (web pages containing links to the resulting data). 68% of the users click a search result within the first page of results and 92% click a result within the first three pages [5]. Therefore, WSEs must put the links that are more interesting for the users in the first result page. *Personalization* is the process that allows WSEs to ascertain the interests of each user. It is based on using past queries to build user profiles. In this way, if a certain user has searched "solar system" before "Mercury", the WSE will assume that "Mercury" refers to the planet Mercury and not to the element in the periodic table. Therefore, the WSE will put the results that correspond to the planet Mercury in the first pages.
- *Combating fraud and abuse.* Query logs can help WSEs to detect and act against different kinds of abuses like web spam. Also, they can be analyzed to prevent dishonest parties from performing frauds based on fake clicks to advertisements or other similar attacks.
- *Sharing data for academic research.* Records containing queries from real users are interesting for several research fields ranging from technical topics (*e.g.,* language processing or information retrieval) to social ones based on analyzing human behaviours.
- *Sharing data for marketing and other commercial purposes.* Query logs enable WSEs to sell personalized advertising [6]; car companies, for example, place display advertising shown only to people who have submitted car-related terms to the WSE. In addition to that, a WSE that offers a good service is likely to increase its number of users. More users implies more advertising revenues [7]. Therefore, the use of any previous technique that improves user experience (*e.g., Improving ranking algorithms*, *Language-based application*, *Query refinement* and *Personalization*) has a direct implication in the economical benefits of WSEs. Another source of economical benefits is the sale of users' profiles to law enforcement agencies. Requests for information have become so common that most of the big Internet companies have a formal process for what is often called *subpoena management*. For example, AOL handles nearly 1000 requests each month for information in criminal and civil

cases [6]. Facebook receives between 10 and 20 requests each day [7]. In 2009, the Yahoo Compliance Guide for Law Enforcement was disclosed. This document specifies that Yahoo charges the government about 30$ to 40$ for the contents, including e-mail, of a subscriber's account and 40$ to 80$ for the contents of a Yahoo group [8].

According to that, building proper user profiles is essential for WSEs. In turn, identifying which queries have been submitted by each user is a process of paramount importance for generating those profiles. WSEs use different methods to gather this information:

- The IP address of the computer which is used to submit the queries pseudo-identifies the owner. Nevertheless, the use of *dynamic* IP addresses makes identification difficult.
- Browser cookies identify uniquely the browser of a certain user [9].
- Browser search bars (*e.g.*, Google Toolbar) identify users using the two methods above: they send to the WSEs certain user information including IP addresses and some cookies. In the case of Google, the *Google Privacy Policy* [1] states that this data is retained in Google's server logs and protected according to their general Privacy Policy.
- Browser version and configuration data that is transmitted to websites upon request act as a *device fingerprinting* that unequivocally identifies the browser of a certain user with a probability of 99.1% [10].

It has been discussed that these techniques do not directly identify individuals but, instead of that, they successfully pseudo-identify them [11].

Even though pseudo-identifiers do not directly reveal the real identity of users, they can be used to achieve that. This can be done as follows:

- An *Internet Service Provider* (ISP) can connect the IP address linked to a bunch of queries with the complete name of the user who submitted them.
- Let us assume a user who logs in to an account associated to the WSE and submits queries. The WSE is able to connect these queries with that account. This account probably contains the real name of the user who owns it. An example of this point is *Google Web History* [12].
- Let us imagine a user who submits a query about personal information which identifies him uniquely: her name, national ID, etc. This kind of queries connects easily the user's pseudo-identifier with her real identity [13].
- Even though a single query might not reveal the real identity of a certain user, the aggregation of several queries might cause this situation. An example of this situation is the case of Thelma Arnold, user of the AOL's WSE, who was identified by her searches submitted over a three-month period. All these queries were hidden behind a pseudonym to protect the real identity of the user. However, the aggregation of hundreds of queries was enough to identify and profile her [14].

As explained previously, user profiles contain the interests of the users. Nevertheless, they also provide sensible information like diseases, sexual tendencies, economical status, etc. Since pseudo-identifiers may retrieve the real identity of the users and their profiles contain really sensible and personal information, it can be argued that the usual process of building and storing user profiles performed by WSEs poses a serious privacy threat to users.

In this way, the AOL scandal, where 20 million queries made by 658000 users were publicly disclosed [14], proves that WSEs are not the proper parties to protect the privacy of the users. Instead of that, users themselves should be responsible for preserving their own privacy. According to that, they should use privacy-preserving mechanisms strictly controlled by them in order to prevent WSEs from profiling them in a *detailed* way.

Profiles are needed in order to provide an efficient service to users. Thus, there is a trade-off between the privacy level achieved and the quality of the service. If the user desires a high degree of privacy, she will probably receive a deficient service. If the user desires an accurate service, her privacy will probably be jeopardized. In addition to that, different users may have different perceptions of the value of their own privacy. Therefore, each one requires a personalized level of privacy/quality of service.

Privacy-preserving schemes that try to hide the personal information of the users who submit queries to a WSE fall in two main categories: *multi-party* and *single-party*. Multi-party protocols require the collaboration of external entities to generate and/or submit queries. From a general point of view, these external entities can be: (i) anonymizing proxies (or a network of proxies like in the *Tor Project* [15]) which act as intermediate nodes that receive the user queries and forward them to the WSE; and (ii) a group users who form an anonymizing network where each user submits queries generated by other users. As a result of that, individual users are hidden inside *group identities*. User-based schemes mainly differ in the way the users are grouped together: [16], [17] present schemes that group users dynamically, on the other hand, [18], [19] use a static network of users (*i.e.,* an already established social network).

In general, multi-party schemes suffer from some major shortcomings:

- *Slow response time*. The use of external entities which act as intermediaries adds a certain overhead to the process of submitting a query to a WSE. More specifically, performing a query search using the Tor network requires on average 10 seconds [20]. Note that a direct query (without any privacy-preserving mechanism) to Google has a response time of 300 ms [16]. Regarding the user-based schemes, [16] has been tested in real conditions (*i.e.* users connected through the Internet) and it provides a response time of 5.2 seconds with groups of three users.
- *System availability*. These schemes depend on external entities to submit queries to the WSE, hence, their availability also depends on the availability of these entities.
- *Liability for query content*. In the User-based schemes,

users submit queries which have been generated by others, hence, some participants might be uncomfortable with certain query contents which can even be illegal.

On the other hand, single-party schemes work directly in the computer of the user to be protected without requiring any external entity [21], [22]. As a result, these proposals achieve the best response time when submitting queries to the WSE and they are ideally suited to provide a personalized level of privacy/quality of service. According to that, in this paper we focus on this kind of schemes. The next section addresses the work done in this field.

A. Previous work on single-party privacy-preserving schemes

A trivial way to provide anonymity when submitting search queries is to use dynamic IPs and a plain web browser without cookies. This method hides the real identity of the user while her queries remain unmodified. As a result, queries submitted in different sessions cannot be grouped together and personalized search is unable to working properly. Even though this is a major drawback, this proposal has some additional flaws:

- The renewal policy of dynamic IP addresses is not controlled by users but the network operator. This operator can always give the same IP address to the same Media Access Control (MAC) address. In this case, the anonymity of the users will not be protected. In addition to that, certain users require static IP addresses, hence this approach is not suitable for them.
- A browser without cookies losses its usability in a high number of web applications. This situation may not be affordable for certain users.

A different single-party approach to solve the privacy problem in WSEs is based on submitting random queries together with real ones. TrackMeNot [23], [21] was the first scheme built from this idea. This proposal generates dynamic queries using RSS feedback which are periodically submitted to the WSE. The RSS feedback can come from blog entries or news headlines among others and are selected in a random way. This implies that the control of the user over the content of the fake queries submitted to the WSE is very limited and the resulting profile may be almost random. This behaviour clearly jeopardizes the quality of service.

GooPIR [22] is another scheme based on submitting random queries, however, it is worth to mention that this method only covers one-word queries. In this way, GooPIR submits a unique query to the WSE that contains fake words together with the authentic one (which was the original user query). This behaviour obfuscates the user's profile because the WSE cannot know which words are fake and which are not. GooPIR uses a Thesaurus in order to decide which words can be added to the search. Specifically, the Thesaurus retrieves the frequency of appearance in general texts of the authentic user query and, then, fake terms with similar frequencies are selected. This procedure does not consider the interest related to each term and, hence, the resulting profile may be almost random too.

A proposal more focused on the trade-off between privacy and quality of service is presented in [24]. The authors of this work provide a tool that enables users to automatically build their own *public* profiles following a hierarchical organization which is based on specific interests. Two parameters for specifying privacy requirements are proposed to help the user to choose the content and degree of detail of the profile information which is exposed to the WSE. Search queries submitted to the WSE are modified according to these requirements. The main problem of this approach is that it requires deep modifications at the server side: a search engine wrapper is developed at the server side to incorporate a partial user profile with the results returned from the search engine. Results from both sources are combined and the customized results are delivered to the user by the wrapper. It can be argued that well-known WSEs will hardly devote resources to undergo these internal modifications in order to preserve the privacy of their users.

In conclusion, it is necessary to design a hybrid approach that: (i) enables users to construct their own *public* profiles using a hierarchical organization of interests that provides the required trade-off between privacy and quality of service which can be decided by the user herself; and (ii) it does not require any change at the server side.

B. Contribution and plan of this paper

In this paper, we propose a new scheme for private web search that works directly in the computer of the user to be protected without requiring any external entity (it is single-party). The main advantage of our system is that it addresses the trade-off between privacy and quality of service but it does not require any change at the server side. The performance of this new method is evaluated using real search queries submitted by real AOL users. Those queries have been extracted from the AOL's files [25].

The paper is organized as follows. Section II explains the new scheme. In Section III the system's performance is evaluated from different points of view (*e.g.,* privacy, quality of service, etc) in order to guarantee its deployment and usefulness. Finally, Section IV gives some concluding remarks.

II. Our proposal

Our proposal uses a similar approach to GooPIR [22] or TrackMeNot [23], [21] in the sense that m fake queries are generated and submitted at the same time (or with a very slightly delay, depending on the implementation) together with the authentic one and no changes are required at the server side. Nevertheless, as explained previously, GooPIR and TrackMeNot generate fake searches using a procedure that does not consider similarities between the interests related to fake and original queries. Therefore, the quality of service achieved is not high. Our scheme generates fake queries using a *knowledge base* to semantically interpret the original interests and control the *distance* between fake interests (related to fake queries) and authentic ones (related to original queries).

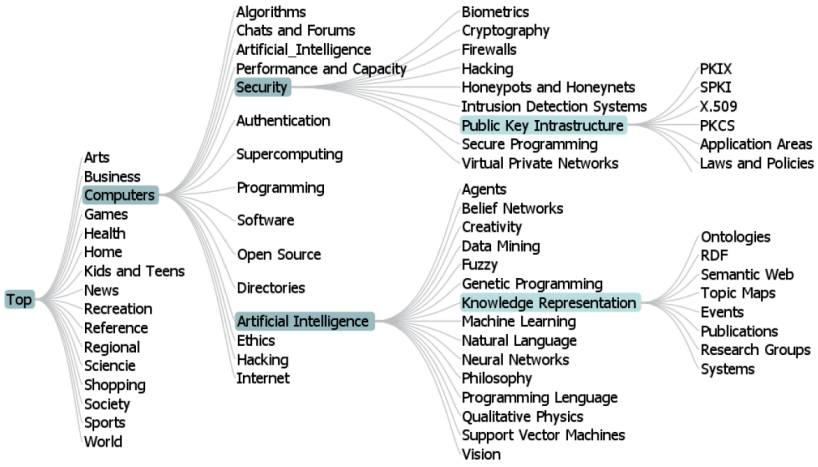


Fig. 1. Snapshot of the ODP hierarchy of categories.

In this work, we use as knowledge base the *Open Directory Project (ODP)* hierarchy of categories because it is the largest, most comprehensive human-edited directory of the Web, constructed and maintained by a vast community of volunteer editors [26]. The purpose of ODP is to list and categorize web sites. Manually created categories are taxonomically structured and populated with related web resources. Nowadays, it classifies almost 5 million web sites in more than 1 million categories.

The workflow of the new method, in a nutshell, is as follows: for each original user query, the ODP retrieves a certain *ODP category*. The ODP is organized following a tree of categories from more general to more specific (see Figure 1). From the retrieved category which is linked to the original user interest, the ODP can provide m fake categories but still related to the original one (*e.g.,* more specific, more general, another child of the same parent category, etc). The distance between the original category and the fake ones might be selected by the user according to her desired level of privacy/quality of service. Then, from each fake category, a fake query is generated by acquiring related terms from the ODP. As a result, m fake queries are built. Finally, all the queries (authentic and fake) are randomly ordered and submitted to the WSE simultaneously. This is done to prevent the search engine from ascertaining which one is the correct by observing their order of arrival.

For example, let us assume that we want to obfuscate a one-word query like "tomato" considering $m = 1$. First, ODP retrieves its category "Top / Cooking / Soups and Stews / Fruit and Vegetable / Tomatoes". Then, ODP retrieves a fake category related to the original one, let us assume that this fake category is "Top / Cooking / Soups and Stews / Fruit and Vegetable / Carrots". From this category, ODP acquires the term "carrots" that is selected as our fake query. Finally, both queries are submitted to the WSE but only the search results linked to "tomato" are kept and presented to the user.

In the next subsection this process is explained with more detail.

A. Query obfuscation process in detail

Let us consider that a user query q is a concatenation of n terms t_i:

$$q = t_1||t_2||\ldots||t_n$$

In order to obfuscate q, the following steps are executed:

1) Each t_i is assigned to an ODP category χ_i. To do that, t_i is queried to the ODP and from the retrieved categories, the more relevant is selected. The number of web sites linked to each category is the value used to ascertain their relevance. As a result of this process, the system obtains:

$$\begin{aligned} t_1 &\rightarrow \chi_1 \\ t_2 &\rightarrow \chi_2 \\ &\vdots \\ t_n &\rightarrow \chi_n \end{aligned}$$

2) In this step, the proposed system specializes the most general category of those retrieved in the last step. Very general categories are not desirable because they may not classify q properly. An accurate classification enables the system to generate better fake queries. A category that contains a small number of web sites is assumed to be very specific while a category related to a lot of web sites is expected to be very general. Let us assume that the number of web sites related to a certain term t_i is γ_i. Note that γ_i is directly retrieved from ODP.

   First of all, the terms are ordered according to γ. The term with the highest number of web sites (the term that falls in the most general category) is selected and it is mixed with the term related to the lowest number of web sites (the term that falls in the most specific category). Let us assume that

those terms are t_j and t_z respectively (this implies that $\gamma_j > \gamma_z$). Both terms are mixed together using the operator "AND" (note that a query to ODP like "x AND y" retrieves web sites that contain both x and y), hence, the resulting term is: $t_r = t_j$ AND t_z. Then, ODP is used to assign a category to t_r as done in step-1.

As explained above, term t_r requires web results related to both t_j and t_z, therefore it is expected that $\gamma_r < \gamma_j$. However, if the resulting t_r is very specialized, γ_r might be zero. If $\gamma_r = 0$ (this means that this term does not appear in any web site), it probably means that t_r does not make sense and, hence, it should not be used. As a result, if $0 < \gamma_r < \gamma_j$, t_r replaces t_j and t_z; the corresponding category χ_r is stored; and the step-2 ends here. Else, the next term with the highest γ is mixed with t_j and ODP is queried again. This process is executed until $0 < \gamma_r < \gamma_j$ or all the terms $t_1, \ldots, t_n$ have been tested. If this last case occurs, the initial categories obtained in step-1 are used.

3) At this point, there is an original category χ_i for each original term t_i (or combination of terms). In this step, the proposed system obtains the fake categories that are used to generate the fake queries. For each original χ_i the process is executed as follows (see Figure 2):

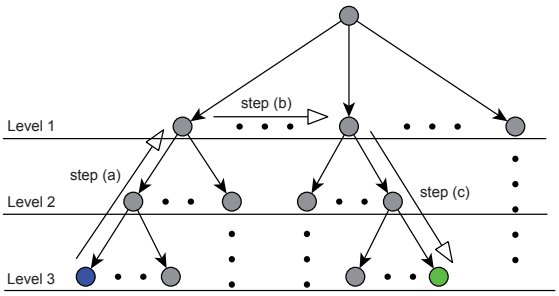


Fig. 2. Selecting a fake category from the original one using the ODP hierarchy.

   a) ODP categories are organized following a tree topology with an average depth of 7. First of all, the proposed system uses the position of χ_i in that tree as a reference and ascends towards the root of the tree k levels. This value might be fixed by the user to obtain fake categories more or less close to the original ones.
   b) From the upper category, one of its child nodes is selected at random. Note that the subtree that contains χ_i is not considered among the candidates.
   c) In the new subtree of categories that has been selected, the system descends a random value of levels to retrieve the fake category χ_i^c that will be used. This behaviour implies that m different fake categories $\chi_i^1, \ldots, \chi_i^m$ obtained from the same original χ_i do not follow any pattern and, hence, a WSE will not be able to identify them. Note that the system retrieves one fake category χ_i^c for each fake query that must be generated. The number of fake queries to be generated (parameter m) can be fixed by the user.
   As a result, the system obtains a group of m alternative (and fake) categories related to each original one. This is reflected as follows:

$$\begin{aligned}\chi_1 &\rightarrow (\chi_1^1, \chi_1^2, \ldots, \chi_1^m)\\ \chi_2 &\rightarrow (\chi_2^1, \chi_2^2, \ldots, \chi_2^m)\\ &\vdots\\ \chi_n &\rightarrow (\chi_n^1, \chi_n^2, \ldots, \chi_n^m)\end{aligned}$$

4) Now, from each fake category χ_i^c, the system retrieves a fake term t_i^c. For example, assuming that χ_j^j refers to the ODP category "Top / Recreation / Autos / Makes_and_Models / Audi", its related fake term t_j^j will be "Audi". As a result of this process, the system gets the following groups of fake terms:

$$\begin{aligned}(\chi_1^1, \ldots, \chi_1^m) &\rightarrow (t_1^1, t_1^2, \ldots, t_1^m)\\ (\chi_2^1, \ldots, \chi_2^m) &\rightarrow (t_2^1, t_2^2, \ldots, t_2^m)\\ &\vdots\\ (\chi_n^1, \ldots, \chi_n^m) &\rightarrow (t_n^1, t_n^2, \ldots, t_n^m)\end{aligned}$$

5) In this step, the m fake queries are generated using the fake terms retrieved previously. Each fake query q^c contains n fake terms t_i^c and only one fake term is selected from each group of alternatives. According to that, the set of m fake queries is computed as follows:

$$\begin{aligned}q^1 &= (t_1^1 || t_2^1 || \ldots || t_n^1)\\ q^2 &= (t_1^2 || t_2^2 || \ldots || t_n^2)\\ &\vdots\\ q^m &= (t_1^m || t_2^m || \ldots || t_n^m)\end{aligned}$$

6) In the final step, all the queries (m fake queries plus the original one) are submitted to the WSE. Each query is sent in an individual way to the WSE like TrackMeNot [23], [21], however, our scheme submits them at the same time (TrackMeNot periodically sends queries through all the day) and the total number of queries is fixed by the user. Note that the $m+1$ queries are arranged in random order before being sent to the WSE.

III. Evaluation

In order to evaluate the performance of the proposed system, it has been implemented and the following points have been analyzed:

- *Quality of the fake queries*. The authors in [27] and [28] argue that it is possible to distinguish user queries from machine-generated queries. More specifically, [28] develops a classifier which is very accurate in identifying TrackMeNot queries with a mean of misclassification around 0.02%. Therefore, it is important to guarantee that the fake queries generated by our scheme cannot be easily differentiated from authentic ones (at least from the point of view of a computer). This point is evaluated using two tests: (i) the ODP hierarchy is used to know the depth in the category tree which is reached by all the terms (fake and authentic). Then, those depths are compared in order to find any significant variation; and (ii) in both authentic and machine-generated queries, terms have different appearance probabilities. Thus, we use the Mann-Whitney test [29] to compare the probability distributions of the set of authentic terms and the set of fake ones and ascertain if they are indistinguishable from that point of view.
- *Privacy*. The *Profile Exposure Level (PEL)* [19] is used to compute the level of privacy protection achieved by our scheme. PEL measures the percentage of user information which can be gathered from an obfuscated user profile.
- *Fake query generation time*. This is the time required by the system to generate m fake queries. If this time is too high, users might be reluctant to use the proposed mechanism.
- *Quality of service*. As explained previously, submitting queries that reflect fake interests reduces the quality of user profiles and, hence, the quality of the search results presented by WSEs.

To evaluate each one of those points, we have used the queries made by 1000 AOL users. This data has been extracted from the AOL's files [25]. Therefore, our scheme is tested obfuscating real queries which were sent by real users. Regarding the parameter k that determines the distance between original and fake categories, in our tests it has been fixed to $k = 3$. We consider that this k value offers a correct trade-off between privacy and quality of distorted user profiles. Note that a small k will generate fake terms very close to the original ones (which implies weak privacy protection), while a large k will produce fake terms far away from the authentic ones (which in turn generates a useless user profile). Assuming that the ODP hierarchy has an average depth of 7, $k = 3$ is an average value which may produce a balanced behaviour. Related evaluation results are next provided and discussed.

A. Quality of the fake queries

As explained above, the first test evaluates whether there are differences between the depths of original and fake categories in the ODP hierarchy or not. Significant differences could be used by WSEs to identify fake queries.

Table I presents the average depth of both types of queries and their variances. This table shows that the average depth is very similar in both cases but it also can be observed that the variance of the fake queries is bigger than the variance of the legitimate ones.

TABLE I
AVERAGE DEPTHS AND THEIR VARIANCE ACHIEVED BY ORIGINAL AND FAKE QUERIES IN THE ODP HIERARCHY OF CATEGORIES FOR DIFFERENT NUMBERS OF FAKE QUERIES (m)

m	Original (av. depth)	Fake (av. depth)	Original (variance)	Fake (variance)
1	7.6	7.8	3.7	5.2
2	7.6	7.7	3.7	5.0
3	7.6	7.6	3.7	5.0
4	7.6	7.6	3.7	4.9
5	7.6	7.5	3.7	4.8

Accordingly, a WSE that runs this test would be able to detect the fake queries which are directly related to *extreme* categories. Nevertheless, it is worth to mention that these queries only represent a small part of the total number of fake queries. More specifically, Table II shows the percentage of fake queries which were detected in our simulations. For all the values of m which were tested, the percentage always remained under 4%. We argue that this percentage is too low and, hence, from the point of view of the category depth, fake queries are well constructed.

TABLE II
PERCENTAGE OF FAKE QUERIES WHICH WERE DETECTED FOR DIFFERENT m VALUES

m	% of queries out of range
1	3.20%
2	3.12%
3	3.87%
4	3.89%
5	3.93%

Regarding the Mann-Whitney test [29], the set of probabilities of the authentic terms is compared with the set of probabilities of the fake terms. Considering that those sets are linked to the random variables X and Y respectively, and that F_X and F_Y represent their distribution functions, the statistical hypothesis test to be done is based on the null hypothesis $H_0 : F_X = F_Y$ and the alternative hypothesis $H_1 : F_X \neq F_Y$. Accepting H_0 implies that there is no statistical difference between both functions and, hence, the sets are indistinguishable. Assuming a significance level $\alpha = 0.05$, H_0 is rejected if $p-value \leq \alpha$. Therefore, we require a large $p-value$ (the closer to 1, the better) in order to not reject H_0.

In this way, Table III shows that, for all the m values tested, $p-value \approx 1$ and, hence, H_0 is clearly not rejected. Therefore, from the point of view of the frequency of appearance of terms, the WSE cannot distinguish the fake queries from the authentic ones. Accordingly, we conclude again that our scheme generates fake queries with good quality.

B. Privacy

In order to measure the privacy level achieved by the users of our proposal, the Profile Exposure Level (PEL) [19] is used. PEL calculates the quantity of information from the real profile which can be extracted from the obfuscated one. This measure is defined as:

TABLE III
RESULTS OF THE MANN-WHITNEY TEST FOR DIFFERENT NUMBERS OF FAKE QUERIES (m)

m	p-value
1	0.99618002
2	0.98116465
3	0.97967428
4	0.97909349
5	0.97877498

$$PEL = \frac{I(\varphi, \varphi')}{H(\varphi)} \cdot 100$$

Where φ is the set of authentic queries, φ' is the set of obfuscated queries, $H(\varphi)$ is the entropy of φ and $I(\varphi, \varphi')$ is the mutual information between both sets. As a result, PEL measures the percentage of user information which is disseminated when φ' is disclosed.

TABLE IV
PEL RESULTS FOR DIFFERENT NUMBERS OF FAKE QUERIES (m)

m	PEL
1	31.14%
2	18.01%
3	12.37%
4	9.51%
5	7.65%

Table IV shows the average results for the 1000 AOL user profiles which were obfuscated using our scheme. It can be observed that the level of exposure decreases when m grows. This is normal because submitting more fake queries implies that the user profile becomes more obfuscated.

The PEL results achieved for all the m values are below 40%. According to [30], a user profile has enough protection when it achieves a PEL of 40% or less. Therefore, from the point of view of the privacy, fake queries are properly contructed in the sense that they successfully hide the real profile.

C. Fake query generation time

This section addresses the time required by our system to generate m fake queries. Note that this only covers the time devoted to generate those queries, the time needed for sending them to the WSE is not considered. In any case, a direct query to Google has a response time of 300 ms [16].

TABLE V
RESPONSE TIMES FOR DIFFERENT NUMBERS OF FAKE QUERIES (m)

m	Response time (ms)
1	108.08
2	159.89
3	209.73
4	260.80
5	311.60

Table V shows that the worst time is achieved when $m = 5$ (as expected) but, even in this case, the generation time is lower than performing a direct search. Assuming that the $5+1$ queries can be submitted to the WSE in parallel, the *total response time* of the proposed system is expected to be less than 700 ms. Even though this is significantly higher than performing a direct search, we argue that this time can be accepted by the users concerned about their privacy.

D. Quality of service

The quality of service offered by a WSE directly depends on the quality of the search results retrieved to users which, in turn, depends on the quality of the user profiles which are used by the WSE. This means that a privacy-preserving tool should obfuscate those profiles using fake interests more or less related to the original ones. Therefore, we perform a test to measure the difference between authentic and fake interests and, hence, ascertain the quality of service that the users who use the proposed privacy-preserving tool can achieve.

To do that, we get the *percentage of preserved categories (PPC)* between original and fake queries. The ODP hierarchy of categories is used and the PPC is computed as follows:

$$PPC = \frac{T}{L} \cdot 100$$

Where L is the number of levels of the original category, L' is the number of levels of the fake category and T is the number of levels which are shared between L and L'. For example, a fake query that obtains a PPC of 100% means that it has not been obfuscated.

TABLE VI
PERCENTAGE OF PRESERVED CATEGORIES (PPC) FOR DIFFERENT NUMBERS OF FAKE QUERIES (m)

m	PPC
1	57.32 %
2	57.18 %
3	57.05 %
4	56.94 %
5	56.83 %

Table VI shows that our proposal preserves a $56,83\%$ of the categories in the worst case ($m = 5$), however, all the results are very close to this one. This means that the proposed scheme obfuscates the user queries but it uses nearby categories in more than half of the cases. Note that those PPC results are linked to the system's parameter k which has been fixed to 3 in our tests.

IV. CONCLUSIONS

In this work we have presented a new single-party scheme designed to preserve the privacy of the users of web search engines. Basically, it generates fake search queries which are sent together with the authentic ones to the WSE.

The main advantage of this new proposal over the current systems found in the literature is that it successfully focuses on the trade-off between privacy and quality of service but it does not require any change at the server side. This latter characteristic is really valuable and represents a significant difference in terms of deployment in real environments.

We have evaluated our scheme in terms of *quality of the fake queries*, *privacy*, *fake query generation time* and *quality of service* using real queries submitted by 1000 real AOL users. The results achieved show that our proposal works as expected and it can be considered a proper option for those users who are concerned about their privacy.

DISCLAIMER AND ACKNOWLEDGMENTS

Authors are solely responsible for the views expressed in this paper, which do not necessarily reflect the position of UNESCO nor commit that organization. This work was partly supported by the Spanish Ministry of Science and Innovation (through projects eAEGIS TSI2007-65406-C03-01, CO-PRIVACY TIN2011-27076-C03-01, ARES-CONSOLIDER INGENIO 2010 CSD2007-00004 and Audit Transparency Voting Process IPT-430000-2010-31), by the Spanish Ministry of Industry, Commerce and Tourism (through projects eVerification2 TSI-020100-2011-39 and SeCloud TSI-020302-2010-153) and by the Government of Catalonia (under grant 2009 SGR 1135).

REFERENCES

[1] Google Privacy Policy, 2011. http://www.google.com.
[2] A. Cooper, "A survey of query log privacy-enhancing techniques from a policy perspective", *ACM Transactions on the Web*, vol. 2, no. 4. pp. 1–27, 2008.
[3] E. Agichtein, E. Brill, S. Anddumais, "Improving web search ranking by incorporating user behavior information", *Proc. of the 29th annual international ACM conference on Research and development in information retrieval – SIGIR'06*, 2006.
[4] T. Joachims, "Optimizing search engines using clickthrough data", *Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining – KDD'02*, 2002.
[5] iProspect.com, Inc,. "iProspect Blended Search Results Study", 2008. http:// www.iprospect.com
[6] S. Hansell, "Increasingly, Internet's Data Trail Leads to Court", *The New York Times*, February, 2006.
[7] N. Summers, "Walking the Cyberbeat", *Newsweek*, May, 2009.
[8] K. Zetter, "Yahoo Issues Takedown Notice for Spying Price List", *Wired*, December, 2009.
[9] Collusion mozilla add-on, Discover who's tracking you online, 2012. http://www.mozilla.org/en-US/collusion.
[10] P. Eckersley, "How unique is your web browser?", *Proc. of the 10th International Conference on Privacy Enhancing Technologies – PETS'10*, 2010.
[11] S. Ye, F. Wu, R. Pandey, H. Chen, "Noise Injection for Search Privacy Protection", *Technical Report CSE-2008-10 - University of California*, 2008.
[12] Google Web History, 2012. http://www.google.com/history.
[13] R. Jones, R. Kumar, B. Pang, A. Tomkins, "Vanity fair: Privacy in query log bundles", *Proc. of the 17th ACM conference on Information and knowledge management – CIKM'08*, pp. 853–862, 2008.
[14] M. Barbaro and T. Zeller, "A Face Is Exposed for AOL Searcher No. 4417749", *The New York Times*, 2006.
[15] The Tor Project, 2012. http://www.torproject.org
[16] J. Castellà-Roca, A. Viejo, J. Herrera-Joancomartí, "Preserving user's privacy in web search engines", *Computer Communications*, vol. 32, no. 13–14, pp. 1541–1551, 2009.
[17] C. Romero-Tris, J. Castellà-Roca, A. Viejo, "Multi-party private web search with untrusted partners", *Proc. of the 7th International Conference on Security and Privacy in Communication Networks - SecureComm'11*, 2011.
[18] A. Viejo, J. Castellà-Roca, "Using Social Networks to Distort Users' Profiles Generated by Web Search Engines", *Computer Networks*, vol. 54, no. 9, pp. 1343–1357, 2010.
[19] A. Erola, J. Castellà-Roca, A. Viejo, J. Maria Mateo-Sanz, "Exploiting Social Networks to Provide Privacy in Personalized Web Search", *Journal of Systems and Software*, vol. 84, no. 10, pp. 1734–1745, 2011.
[20] F. Saint-Jean, A. Johnson, D. Boneh, J. Feigenbaum, "Private Web Search", *Proc. of the ACM workshop on Privacy in electronic society – WPES'07*, pp. 84–90, 2007.
[21] G. T. Marx, "A Tack in the Shoe: Neutralizing and Resisting the New Surveillance" *Journal of Social Issues*, vol. 59, no. 2, pp. 369–390, 2003.
[22] J. Domingo-Ferrer, A. Solanas, J. Castellà-Roca, "$h(k)$-Private Information Retrieval from Privacy-Uncooperative Queryable Databases", *Journal of Online Information Review*, vol. 33, no. 4, pp. 1468-4527, 2009.
[23] TrackMeNot, 2010. http://mrl.nyu.edu/~dhowe/trackmenot
[24] Y. Xu, B. Zhang, Z. Chen, K. Wang, "Privacy-enhancing personalized web search", *Proc. of 16th international conference on World Wide Web*, pp. 591-600, 2007.
[25] AOL Search Data Mirrors, 2006. http://gregsadetsky.com/aol-data/
[26] Open Directory Project, 2012. http://www.dmoz.org
[27] R. Chow, P. Golle, "Faking contextual data for fun, profit, and privacy", *Proc. of the 8th ACM workshop on Privacy in the electronic society WPES09*, pp. 105-108, 2009.
[28] S. T. Peddinti, N. Saxena, "On the privacy of web search based on query obfuscation: a case study of TrackMeNot", *Proc. of the 10th international conference on Privacy enhancing technologies PETS10*, pp. 19-37, 2010.
[29] F. Wilcoxon, "Individual comparisons by ranking methods", *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.
[30] G. Navarro-Arribas, V. Torra, A. Erola and J. Castellà-Roca, "User kanonymity for privacy preserving data mining of query logs", *Information Processing & Management*, in press.