Computer Networks 67 (2014) 26–42

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Distributed system for private web search with untrusted partners

Cristina Romero-Tris *, Jordi Castellà-Roca, Alexandre Viejo

Departament d'Enginyeria Informàtica i Matemàtiques, UNESCO Chair in Data Privacy, Universitat Rovira i Virgili, Av. Paisos Catalans 26, E-43007 Tarragona, Spain

ARTICLE INFO

Article history:
Received 10 May 2013
Received in revised form 21 November 2013
Accepted 24 March 2014
Available online 30 March 2014

Keywords:
Privacy
Cryptography
Web search engines
Distributed system
Private information retrieval

ABSTRACT

Web search engines (WSEs) allow information retrieval from the Internet, a really useful service which is not provided without cost: users' queries and related information (e.g., query time, browser type, etc.) are stored and analyzed in the WSE database. The stored logs may contain sensitive information (e.g., health issues, location, religion, etc.) and identifiers (e.g., full name, IP address, cookies, etc.), which poses a serious threat to users' privacy. In the literature, there are several proposals that try to address this situation. In general, current schemes consider the WSE as the only adversary, and do not address the presence of other attackers or, if addressed, the introduced query delay is unaffordable in real environments. In this paper, we propose a distributed protocol, where a group of users collaborate to protect their privacy in front of WSEs and dishonest users, while introducing a reasonable delay. The performance of the new scheme is evaluated in terms of privacy level and delay. The former is analyzed using a set of query logs belonging to real users and provided by AOL. The latter involves the implementation, deployment and evaluation of the protocol in a real environment.

1. Introduction

The Internet is a huge repository of data that contains information provided by a lot of sources covering a wide range of topics. Web search engines (WSEs) like Google or Bing are tools that allow information retrieval from this collection of documents. They index billions of web pages, so that users can find the information that they search through the use of keywords. During this process, WSEs record all the submitted queries and their related information. In Google's Privacy Center [1], it is stated that Google servers automatically record requests made by users. These "server logs" include user's query, IP address, browser type, browser language, date and time of the request and a reference to one or more cookies that may uniquely identify the user's browser.

Once collected, WSEs process and analyze the "server logs" in order to build *users' profiles*, and use them with different purposes. For example, WSEs employ the profile of a certain user in order to personalize her search results. This improves the user's experience because personalized results are more likely to match her interests. Nevertheless, WSEs also exploit *users' profiles* in other activities that do not help users: for example, selling profiles to third parties (*e.g.,* advertisers, media, etc.) represents a large source of income for WSEs [2]. This threatens users' privacy because queries may contain identifiable personal information (*e.g.,* names, Social Security numbers, geolocation data, etc.), and other sensitive information (*e.g.,* queries about health, sexual orientation, politics, religion, etc.). Accordingly, building profiles presents a trade-off between the improvement of users' experience and the threat to their privacy.

* Corresponding author. Tel.: +34 652968327.
E-mail addresses: cristina.romero@urv.cat, cristina.romerot@estudiants.urv.cat (C. Romero-Tris).

http://dx.doi.org/10.1016/j.comnet.2014.03.022
1389-1286/

There are some proposals in the literature [3,4], that describe how WSEs can *anonymize* users' profiles. However, these techniques are applied in the server side and do not allow individuals to control how their information is managed. Consequently, they must rely on the honesty of WSEs and their ability and interest in applying these techniques. Nevertheless, some incidents in the past years have shown that WSEs cannot be trusted in this matter. In 2006 AOL released a file with twenty million queries generated by approximately 658,000 of its users [5]. This incident had serious consequences since the data was not correctly anonymized, and some users were successfully identified. Moreover, in 2006, Google suffered a subpoena where the Justice Department of the U.S.A. required this company to provide millions of records with users' queries [6].

These facts show the risks related to performing unprotected web searching and the necessity of proposing alternatives to prevent the WSEs from acquiring sensitive information. One way to classify the existing alternatives is between *standalone* and *distributed*. The former approach allows that one user alone protects her privacy in front of the WSE. The latter requires that a group of users or entities collaborate in order to protect the privacy of each member of the group.

Standalone schemes (e.g. [7] and TrackMeNot [8]) are based on generating a stream of automated queries that are used to hide the real queries of the user. Although they obtain a fast response time, they suffer from other disadvantages: machine-generated queries do not have the same features as human-generated queries. Some proposals like [9,10] show that it is possible to distinguish real queries from queries generated automatically, with a mean of misclassification around 0.02% [9,10].

Consequently, in order to obfuscate a profile, it is better to employ queries generated by real users. This is precisely how distributed systems work, hence, this paper focuses on this kind of systems to achieve user's privacy in WSEs.

1.1. Previous work

As previously mentioned, distributed schemes require the collaboration of external entities. For example, this is the case of using a proxy (*e.g.* Scroogle [11], anonymizer.com [12]) to conceal the source of a query. In this solution, the user first sends her query to the proxy, then the proxy submits the query to the WSE and sends the answer back to the user. Nevertheless, this is not the best solution, since profiling could be done at the proxy and so, instead of trusting the WSE, users are required to trust the proxy.

The use of a group of proxies instead of a single one has been proposed in order to address this problem. Specifically, the Tor Project [13] is the most renowned implementation of this idea. The problem of this approach is that it is not specifically designed for web search, it provides anonymous routing for all purposes. This means that some of the desirable characteristics in web search are lost: with Tor, the WSE gets an empty profile of the user. Although this means the highest level of privacy, the WSE can no longer provide personalized results, and the quality of the user's experience is significantly reduced.

An alternative that solves this problem is the obfuscation of the profile by means of noise. Submitting some of the queries generated by the user mixed with queries generated by other users is an intuitive way of achieving this. This allows the WSE to personalize the search results and improve users' experience better than with the group of proxies, while the true contents of the profile remain indistinguishable. Following this idea, [14–16] work with static groups of users (i.e., a group is initially created and then, the same members participate in every protocol execution). The three schemes follow the same idea: users are put into a large group where they submit queries to a WSE on behalf of other members. When a user wants to make a query, she decides, with random probability, whether to directly submit the query to the WSE or to establish a random path through the network. In the second case, she randomly picks one of her neighbors and forwards the query to her. That neighbor then, either randomly selects one of her neighbors to forward the query, or she submits the query to the WSE. The privacy in this approach relies on the fact that when a neighbor receives a query, it does not know if the sender was the original owner or only a forwarder. The differences between the three schemes focus on the kind of infrastructure used to support the system: while [14] only considers a group of users managed by a central node, [15] is deployed on already developed social networks (*e.g.* Windows Live Messenger, Facebook, etc.), and [16] is an extension of [15], specially designed for social networks where each user knows how many friends each of her friends has. Note that this information allows users to distribute her queries among her friends in a more equitable manner, achieving a higher level of privacy. Nevertheless, distributed protocols that use static groups are very vulnerable to internal adversaries. In this scenario, attackers can exploit their knowledge about the topology, since groups of users are formed once and they never change. The authors in [17] elaborate on this kind of attacks.

On the other hand, there is another kind of distributed systems that work with dynamic groups. These schemes are better suited to address the problem of internal attackers since the group members are different in every execution of the protocol. In this way, the Useless User Profile (UUP) protocol [18] is based on a central node that distributes users into dynamic groups of n members where they randomly exchange their queries. As a result, each user submits a query from one of her partners and not her own and, hence, she obtains a distorted profile. The response from the WSE is broadcast to all the members of the group. Finally, each user selects only her answer and discards the rest. The proposed protocol uses ElGamal encryption together with a rerandomization technique and a joint decryption technique. This combination was previously employed in works such as [19], but applied in different contexts (e.g., data collection from a group of respondents).

The major drawback of the UUP protocol is that it is not secure in presence of malicious internal users. This scheme assumes that the users follow the protocol and that there are no collusions between two entities.