The Journal of Systems and Software 84 (2011) 1734–1745

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Exploiting social networks to provide privacy in personalized web search

Arnau Erola, Jordi Castellà-Roca, Alexandre Viejo*, Josep M. Mateo-Sanz

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

ARTICLE INFO

Article history:
Received 10 June 2010
Received in revised form 25 March 2011
Accepted 4 May 2011
Available online 17 May 2011

Keywords:
Privacy
Private information retrieval
Social networks
Web search

ABSTRACT

Web search engines (WSE) have become an essential tool for searching information on the Internet. In order to provide personalized search results for the users, WSEs store all the queries which have been submitted by the users and the search results which they have selected. The AOL scandal in 2006 proved that this information contains personally identifiable information which represents a privacy threat for the users who have generated it. In this way, AOL released a file containing twenty million queries made by 658,000 persons and several of those users were successfully tracked. In this paper, we propose a P2P protocol that exploits social networks in order to protect the privacy of the users from the profiling mechanisms of the WSEs. The proposed scheme has been designed considering the presence of users who do not follow the protocol (*i.e.*, adversaries). In order to evaluate the privacy of the users, we have designed a new measure (the profile exposure level (PEL)). Finally, we have used the AOL's file in order to simulate the behavior of our scheme with real queries which have been generated by real users. Our tests show that our scheme is usable in practice and that it preserves the privacy of the users even in the presence of adversaries.

1. Introduction

Nowadays, the Internet has 1700 million users (Internet, 2008) and nearly 187 million web pages (Netcraft, 2009). In this way, it has become the largest source of information worldwide. Web search engines (WSEs) – *e.g.*, Google, Bing, etc. – are an essential tool for finding specific data among this incalculable amount of information. WSEs are easy to use and they retrieve search results quickly. Accordingly, it can be argued that these tools have played a crucial role in the World Wide Web's success.

The searching process executed by the WSEs usually provides several result pages related to each query of the users (web pages containing links to the resulting data). The work presented in iprospect (2009) states that 68% of the users of WSEs click a search result within the first page of results. Even more relevant is the fact that 92% of the users click a result within the first three pages of search results. Accordingly, in order to provide a better user experience, WSEs should put the links which are more interesting for the users in the first results pages.

When the WSEs rank the search results according to their relevance to the users, they deal with an important problem: some terms are ambiguous. For example, the word *Java* can refer to the Java programming language or to the island of Java. The concept *disambiguation* represents the process of identifying the correct sense when a certain word has different ones. This process requires the knowledge of: (i) the interests of the user and (ii) the query context. For example, if a certain user is interested in computer science, the WSE will assume she is referring to the Java programming language. In the literature, the disambiguation process is used by schemes of personalized search (PS) (Pitkow et al., 2002; Saint-Jean et al., 2007; Shen et al., 2007; Xu et al., 2007).

The interests of the users are gathered using profiling techniques. Several works that address this topic can be found in the literature. In Sugiyama et al. (2004), the authors use the browsing history. The use of click-through data is proposed in Qiu and Cho (2006). In Speretta and Gauch (2004) and Reiter and Rubin (1998), two schemes which introduce the use of web communities for this purpose are presented. In Teevan et al. (2005), the authors present a client side application which stores the interests of the users. Nevertheless, the use of the queries previously submitted by users (Speretta and Gauch, 2004; Google, 2009) have been proved to be the best approach. This latter mechanism is very effective because it profiles users without their collaboration.

Profiles improve the searching experience. However, in most of the cases, they also contain private information of the users. Some examples of this situation are the following: (i) if a certain user has searched for a certain place, it can be inferred that she lives there and (ii) if she searches a certain disease, it can be deduced that she (or someone close to her) suffers that disease.

* Corresponding author.
E-mail addresses: arnau.erola@urv.cat (A. Erola), jordi.castella@urv.cat (J. Castellà-Roca), alexandre.viejo@urv.cat (A. Viejo), josepmaria.mateo@urv.cat (J.M. Mateo-Sanz).

0164-1212/$ – see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2011.05.009

In a standard WSE scenario, there are two main interactions between entities where the privacy of the users can be jeopardized. These are:

- *User – WSE.* The WSE builds user profiles which are a potential source of private information. This entity can get important benefits from that personal data: the business model of the WSEs is based on advertising. Efficient advertising relies on the data obtained from the users (these data can be public or private), hence, WSEs might not be interested in protecting the privacy of the users. Once the personal data is gathered, users can do nothing to prevent the WSEs from using it for commercial purposes.
- *WSE – External third parties (e.g., advertisers, investors, media, etc.)* WSEs can provide user profiles to external entities (Steel, 2010): they can be purchased by companies or governmental authorities can force the WSE to disclose them. A honest WSE might anonymize those profiles in order to protect the privacy of the users who generated them. Nevertheless, the AOL scandal, where 20 million queries made by 658,000 users were publicly disclosed and certain users who generated them were successfully tracked (Barbaro and Zeller, 2006), proved that WSEs cannot properly protect the privacy of their users.

The users of a WSE only participate in the first interaction. In addition to that, if the user anonymizes her own profile in a proper way, her privacy will be properly protected in both interactions. Therefore, privacy-preserving mechanisms which focus on the *User – WSE* interaction are generally more convenient for the users.

The privacy-preserving mechanisms should prevent the WSEs from profiling users in a detailed way. Note that 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.

2. Background

Privacy concerns have a long history. They already existed in information retrieval from public databases. In this field, when a user submits a query, she is also exposing her interests to the database operator. Since the early 1980', several proposals to hide this personal information have emerged in order to address this situation.

Those proposals can be classified according to the level of privacy that is offered to users: (i) schemes that provide perfect privacy but no personalized service; and (ii) schemes that partially protect the privacy of users and provide a certain degree of personalized service. We next summarize the different existing schemes of each type, starting with the ones that offer perfect privacy.

A PIR (private information retrieval) protocol allows a user to retrieve a certain item from a database without allowing the latter to know which item is being acquired. Trivially, PIR can be achieved sending a copy of the entire database to the user, but this is very inefficient and unfeasible in practice.

The first PIR protocol was proposed in 1997 by Chor et al. (1995, 1998). However, it requires the existence of at least two copies of the same database. Besides, those databases cannot communicate between them. Accordingly, this proposal cannot work in a single server scenario like WSEs.

Two years later, Kushilevitz and Ostrovsky (1997) presented the first single-database PIR scheme. Nevertheless, it requires the cooperation of the database. This fact represents a major drawback which disqualifies it in scenarios where the database is not willing to collaborate. WSEs are an example of this situation: they have no motivation to protect the privacy of users.

Intuitively, it can be assumed that the privacy of the users can be achieved by preventing the WSE from identifying the true source of the query. In this way, the WSE cannot link users with their queries and it cannot create their profiles. A trivial way to provide anonymity to the users who use a WSE is to use dynamic IPs and a web browser without cookies. However, this approach has the following drawbacks:

- The renewal policy of the dynamic IP address is not controlled by the user but the network operator. This operator can always give the same IP address to the same media access control (MAC) address. Nevertheless, certain users require static IP addresses.
- A browser without cookies loses its usability in a high number of web applications. This situation may not be affordable for certain users.

Another straightforward way to conceal the source of a query is using an anonymizing proxy. There are many public proxy servers available on the Internet. When a user wants to submit a query to a WSE, she sends the query to the proxy. Then, the proxy submits the query to the WSE, receives the response and sends it back to the user. The whole process is done anonymously, *i.e.*, the true source of the query remains hidden. However, since all the queries are sent through the same proxy, they can be easily linked together by the proxy itself. An adversary with access to the logs of the proxy could identify the true source of the queries.

This problem can be solved using a group of proxies instead of only one. In this way, Chaum (1981) proposed the use of an anonymity network which consists of several routers that act as anonymizers. The input and the output routers in the network are rotated among them. Therefore, logs from all the routers are necessary in order to link the queries which have been generated by the same user. There are many implementations of this scheme. Probably, the *Tor Project* (Tor, 2009) is the most renowned.

The main drawback of this approach is that the process of submitting a query to the WSE and receiving the answer through an anonymous channel is very time-consuming. The authors in Saint-Jean et al. (2007) used the anonymous network Tor with paths of length two (note that the default length is three) and submitting a query was, on average, 25 times slower than performing a direct search. In addition to that, anonymous channels are not enough to preserve the privacy of the users. They only take care of the data transport, hence users should use specific programs to hide identifying information. This information can be obtained from the cookies, the HTTP headers or from active components of the websites.

This situation is usually solved using the anonymity network in combination with an HTTP filter like Privoxy (Privoxy, 2009). This tool attempts to delete all the unnecessary information that users submit to the WSE. As a result of this process, the disclosure of personal information is reduced but it does not improve the response time of a query submission. In addition to that, Privoxy is a general-purpose filter and it does not remove active components like JavaScript or ActiveX. FoxTor (Foxtor, 2009) and TorButon (Tor, 2009) are two FireFox plug-ins that combine a Tor network with a Privoxy filter.

An alternative method for providing privacy is based on obfuscating the profiles by means of noise. Submitting false random queries is an intuitive way of achieving this. Schemes that follow this approach provide non-anonymous privacy in the sense that a certain user can be identified but her interests remain hidden. The authors in Shen et al. (2007) state that this approach can be considered as a way to get k-anonymity. The author in Sweeney (2002) explains that a release provides k-anonymity protection if

the information for each person contained in the release cannot be distinguished from at least $k-1$ individuals whose information also appears in the release. Applied to the WSE scenario, a protection level comparable with k-anonymity is achieved if a query of a certain user cannot be distinguished from at least $k-1$ queries generated by other users. This means that k different queries have the same probability of being the real one.

The main difficulty in generating false random queries is selecting the query terms properly. Some proposals choose the false query terms outside the interests of the user (Elovici et al., 2002a,b; Domingo-Ferrer et al., 2009). These proposals try to confuse the WSE. On the other hand, schemes presented in Kuflik et al. (2003) and Trackmenot (2009) select the false query terms from the topics which are relevant to the user. This latter approach focuses on the general areas of interest of the user. The specific ones are avoided.

In the literature, there are two main works based on these two approaches:

- TrackMeNot Trackmenot (2009) uses the periods when the activity of the users is low to submit random queries to the WSEs. In this way, the system does not affect the normal work of the users. However, sending fake queries increases the network traffic and overloads the WSEs. As a result of that, this scheme protects the privacy of the users but it also reduces the network and WSEs performance. In addition to that, this behavior introduces a serious privacy threat: for each user, the WSE is able to divide all her queries depending on whether or not they have been submitted during working hours. Probably, all the queries which have been submitted out of the working hours have been sent by TrackMeNot. The period of time between the submission of two different queries can also be used to the same purpose: it can be assumed that when the users are working, they do not submit only one query but several in a short period. Note that, the working period of a certain user can be inferred from her time zone and, in turn, her time zone might be deduced from her IP address. Therefore, users can only protect themselves against this situation by hiding their IP addresses using anonymizing techniques like the ones which have been previously described. However, as stated before, these mechanisms also suffer from some important drawbacks that disqualify them in certain scenarios.
- GooPIR (Domingo-Ferrer et al., 2009) submits a bunch of queries that contain fake words together with the authentic term. The WSE cannot know which words are fake and which are not. In this way, the profile of the user is obfuscated and her privacy is protected. This proposal uses a Thesaurus in order to decide which words can be added to each search. Thus, GooPIR can only submit words. Full sentences are not addressed (note that sentences cannot be formed by random words).

A method that only exposes a part of the user profile is presented in Xu et al. (2007). This scheme extracts the interests of the user from her browsing history and emails. All the collected information is organized following a hierarchical tree where the leaves are the specific interests. Only the general interests are shown to the WSE. The authors argue that this behavior provides a fair search quality and a certain level of privacy to the user. Nevertheless, the WSE can still create a user profile with her general interests. In addition to that, this proposal requires certain modifications at the server side, hence it is not suitable for all the WSEs.

All the above proposals use network resources in order to preserve the privacy of the users. However, there are other methods where users collaborate in order to reach the same purpose.

Crowds (Reiter and Rubin, 1998, 1999) is a system based on the concept of *users blending into a crowd*. In this scheme, a user tries to hide her actions within the actions of many others. Like Tor, it is based on the mixing system of Chaum (1981), but in this case, the users also act as routers. This proposal works as follows: a certain user who wants to submit a query can send it directly to the WSE or she can forward it to one of her *neighbours* in the crowd. A neighbour who receives a query can submit it to the WSE or she can forward it again to one of her own neighbours. A query is forwarded between users until someone submits it to the WSE. There are some shortcomings in this framework proposal:

- Like in the anonymity networks, Crowds only protects the data transport. Users are responsible for hiding their private information.
- Personalization is only possible if the members of the crowd share the same interests.
- The authors argue that Crowds can scale without limits: the load on each user stays approximately constant as the crowd size grows, however this can not be guaranteed. Moreover, The structure of the crowd must be maintained and this task is costly.
- This proposal requires a central node which manages the crowd (users joining/leaving the crowd).
- In order to keep a certain user concealed, her queries have to be uniformly distributed among the rest of the users (Sweeney, 2002). This scheme does not address this point.

The authors in Domingo-Ferrer and Bras-Amorós (2008) present another proposal where the profile of a certain user is hidden within a group profile. This scheme is based on symmetric key cryptography and a P2P network. A group of users share memory sectors which are used to store and read the queries and their answers. There is no connection between the users. The authors argue that a wiki-like collaborative environment can be used to implement a shared memory sector. This scheme has the following drawbacks: (i) there is no study about the memory-space requirements of this proposal; (ii) users must scan their shared memory sectors at regular intervals, this introduces a significant overhead to the network; and (iii) this scheme achieves a response time of 5.84 s without considering the network time. Thus, the final response time is expected to be clearly above 5.84 s but the exact value is not specified by the authors.

In Castellà-Roca et al. (2009), the useless user profile (UUP) protocol is proposed. In this scheme each user who wants to submit a query will not send his own query but a query of another one instead. Users do not know which query belongs to each user, hence the privacy of the users is preserved. Confidentiality is achieved by means of certain cryptographic tools (a threshold cryptosystem and a ciphertext re-masking operation). This scheme has been tested in a real environment and it provides an overhead of 5.2 s with a group of three users and a key length of 1024 bits. The shortcomings of this proposal are the following:

- This scheme requires a central node which creates the groups of users. The process of creating groups introduces a significant delay because it requires a large number of users in order to provide an acceptable response time.
- It does not consider the trade-off between the privacy level achieved and the quality of the service.

Viejo and Castellà-Roca (2010) proposes a scheme with the same principle as Crowds (Reiter and Rubin, 1998): users submitting queries on behalf of other users. An attractive feature of this system is that an existing social network (*e.g.*, Facebook) can be used as a peer community. This fact makes its deployment quite straightforward. Another benefit from the use of social networks is that users who share the same group are intended to be friends in real life. This implies that they are likely to share similar interests, hence, the distorted profiles still allow the users to get a proper service from the WSE (Mislove et al., 2006). Besides, this contribution improves for-

mer proposals in terms of query delay. Nevertheless, this scheme presents some drawbacks that we next summarize:

- This scheme relies on two functions in order to work properly. The first function estimates the profile exposure level and selects who is the user that must send a certain query in order to preserve the privacy of the users. The second function evaluates the selfishness of the users and punishes the users who do not collaborate with the group. The more effective these functions are, the better the system behaves. However, the authors do not provide a deep study of these functions and they leave its design as future work. Therefore, achieving better implementations of both functions is of paramount importance.
- The authors have simulated their scheme and they have demonstrated its functionality in the environment they proposed. Nevertheless, this scheme should be analyzed when dealing with real data in order to gauge the real privacy level achieved by this solution. In addition to that, some measurement functions are needed to evaluate the levels achieved by the proposed protocol in terms of: privacy, protection against selfish users, usability and quality. Without a standard measurement function, it is not possible to argue whether a certain scheme works properly or not.

2.1. Research questions

The review of the current proposals in the literature shows that the work presented in Viejo and Castellà-Roca (2010) offers unique and very interesting features:

- It uses existing social networks in order to provide already-generated groups of users.
- These fixed groups are made of friends in real life. Therefore, the users of a certain group are very likely to share similar interests. As a result, (Viejo and Castellà-Roca, 2010) generates a distorted profile which is a trade-off between the privacy level achieved and the quality of the service.
- It outperforms the rest of proposals in the literature in terms of query delay. A system that provides a small query delay is more likely to be used by the users.

Nevertheless, the following research questions appear when considering this scheme:

- The privacy level achieved by the users of this proposal depends on the function that calculates the probability of submitting a query. Can this function be re-designed to improve the current results?
- Mechanisms to measure the privacy level achieved by the users are needed in order to compare different proposals. Is there a standard measure that can be used for this purpose?
- The simulations which are shown in Viejo and Castellà-Roca (2010) have been performed using synthetic queries (queries which are generated at random by a computer) and each user is always submitting the same one towards the WSE. Will the use of real queries (queries which are generated by humans) influence the behavior of this scheme in terms of privacy protection?

2.2. Contribution and plan of this paper

In this paper, we propose a collaborative system to preserve the privacy of WSEs users. Specifically, we propose a new version of the scheme presented in Viejo and Castellà-Roca (2010). Our contributions are summarised next:

- The function used to decide which user must submit a certain query to the WSE has been studied and re-designed. As a result, the privacy level achieved by the users has improved.
- A new measure to estimate the privacy achieved by the users, the *profile exposure level (PEL)*, is proposed.
- For the first time – to the best of our knowledge – the tests have been performed using real data extracted from the AOL file (Aol, 2006). In this way, the correct behavior of the proposed system has been tested with queries which have been generated by real users.

These changes improve the privacy achieved by the users in the previous version while keeping its usability. The protocol submits standard queries to the WSE, so it requires neither changes at the server side nor the collaboration of the server with the users. The system is based on the two following assumptions: (i) due the flat-rate broadband connection proliferation, users (their computers) are most of the day connected to the Internet; and (ii) users are organized in social networks.

This paper is organized as follows: Section 3 presents our proposal in detail. The measurement methods which have been used are explained in Section 4. Section 5 presents the simulation results and they are compared with the results achieved by Viejo and Castellà-Roca (2010). Finally, some conclusions are given in Section 6.

3. Protocol for protecting the privacy of the users

The proliferation of flat-rate broadband connections enables users to be most of the day connected to the Internet. More specifically, the use of mobile devices with communication capabilities (*e.g.*, iphone, htc, blackberry, etc.) allows users to be practically always online receiving the latest tweets or facebook messages among others. This fact paves the way for new applications like Peer-to-Peer (P2P) (Anderson, 1996). In a P2P environment, users collaborate between them to perform a particular service. Crowds (Reiter and Rubin, 1998) and P2P-PIR (Domingo-Ferrer and Bras-Amorós, 2008) are two examples where the user profile is obfuscated within a group profile.

These schemes do not overload the network but require the maintenance of the network structure. In order to solve this threat, existing structures are used: *Social Networks*. More specifically, the social network concept, which is explained in Domingo-Ferrer (2007), is followed. This approach does not require the existence of a central node. Therefore, users are connected directly between them or by means of other users who act as intermediaries. Besides, these social networks are not open to examination (a user only knows her direct connections in the network).

The users of a social network are connected to similar users, with common interests (Mislove et al., 2006). Accordingly, a group of users which is maintained by a social network can be used to create a usable user profile. Note that WSEs need usable profiles in order to provide a proper service. Using the proposed protocol the WSE obtains a profile of each user but this profile is not a detailed one. In addition to that, our work is based on social networks that allow users to know the number of neighbours that a certain neighbour has (only the number of connections, not the identities behind them). This helps to equitably distribute all the queries across the network.

Next, it is briefly described how the protocol works: a user U generates queries that she can either directly submit to the WSE or she can send to a neighbour (a neighbour is a direct relationship in the social network). A neighbour that receives a query can submit it directly to the WSE or she can forward it again to one of her own neighbours. This process is repeated until someone submits

the query to the WSE. The profile of U is distorted by the queries that she submits to the WSE but that are generated by other users of the social network.

The entire process is run in background mode without the interaction of the user. As stated before, this proposal relies on two functions that evaluate the profile exposure level and the selfishness of the neighbours and they automatically regulate the acceptance, forwarding or submission of queries. Note that, requiring the interaction of the users for these purposes would slow down the search process considerably and, hence, the system's usability would be jeopardized.

Ideally, all users should behave properly. Nevertheless, this cannot be guaranteed in a real environment. Therefore, two types of users can be defined:

- *Selfish user.* A selfish user is a user who does not follow the proposed protocol. When a selfish user wants to submit a query to the WSE, she sends the query to a neighbour who submits it on her behalf. However, when a selfish user receives a query from another user, she systematically discards it and she does not answer.
- *Honest user.* An honest user is a user who follows the proposed protocol. When an honest user wants to submit a query to the WSE, she decides if she submits it directly or if she forwards it to a neighbour who submits it on her behalf. When the honest user receives a query from another user, she decides if she submits the query to the WSE or if she forwards the query to another user.

A selfish behavior prevents the queries from being distributed equitably among the group. This situation can jeopardize the privacy of the honest users. Thus, we propose a mechanism to prevent users from behaving in that way.

3.1. The protocol in detail

Supposing that a certain user U_i is a member of a social network and that she has k neighbours (direct connections) $\{N_1, \ldots, N_k\}$, U_i knows the number of connections of each of her neighbours. With all this information, U_i can calculate the *sending probability* P_s for each neighbour (see Section 3.2 for more details about this probability). Besides, U_i keeps a measure about the selfishness level of each neighbours (see Section 3.5 for more details about the function that evaluates the selfishness and its parameters).

When U_i wants to submit a query q to the WSE, she runs the following protocol:

(1) U_i executes the *user selection function* Ψ $(U_i,N_1,\ldots,N_k)$ which returns a sorted vector ν of users belonging to the group $\{U_i,N_1,\ldots,N_k\}$. U_i will use ν to decide whether she submits q to the WSE or she sends q to a neighbour N_j (see Section 3.4 for details about the ordering of ν and the operations performed by Ψ).

Let U_j be the j-th user belonging to the vector ν. U_i can carry out two actions according to the value of U_j:

(a) If $U_j = U_i$, then U_i submits q to the WSE.
(b) If $U_j \in \{N_1,\ldots,N_k\}$, then U_i sends q to U_j. U_j can accept or reject q, hence there are two possible behaviors:

- If U_j rejects q then U_i updates negatively the parameters that measure the selfishness of U_j (see Section 3.5).
- If U_j accepts q then U_i initializes a timer. If the response from U_j arrives before the end of this timer, U_i updates positively the parameters that measure the selfishness of U_j. Otherwise, U_i updates negatively the parameters that measure the selfishness of U_j (see Section 3.5).

This process is repeated until someone accepts q. In case that all the neighbours reject q, U_i submits her query by herself.

(2) U_j executes the *selfishness function* Υ (U_i) in order to accept q or not. Therefore, there are two possible situations:

- If U_j accepts q, then U_j is the new responsible for submitting q. Thus, U_j repeats the first step of the protocol replacing the function Ψ with Ψ_f (*i.e.*, U_i executes Ψ_f). Let Ψ_f be the function that decides whether U_j has to submit q to the WSE or not. If Ψ_f answers negatively to that question, U_j attempts to forward q to a random neighbour. The aim of Ψ_f is to distribute equitably the queries among U_j and her neighbours (see Section 3.3 for more details about it). When U_j receives the answer to q (*i.e.*, from the WSE or from a neighbour), she returns it to U_i.
- If U_j rejects q, then U_j ends her participation and U_i updates negatively the parameters that measure the selfishness of U_j.

Note that, when user U_j receives a query from user U_i, she does not know whether U_i has generated the query or she has only forwarded it. Therefore, the query content cannot be surely linked to U_i and, hence, her privacy is protected.

3.2. Sending probability P_s

Each user assigns to each one of her neighbours a sending probability P_s in order to equitably distribute her queries throughout the network.

Let U_i be a user $\{N_1,\ldots,N_k\}$ be her group of neighbours and $\{H_1,\ldots,H_k\}$ be the number of neighbours that each neighbour of U_i has (*i.e.*, H_j is the number of neighbours of N_j). Note that k is the number of neighbours of U_i. U_i assigns a certain P_s to each neighbour, so that the probability that U_i sends a query q to her neighbour N_j is proportional to H_j:

$$P_s(N_j) = \frac{H_j}{\sum_{f=1}^{k} H_f + 1}$$

3.3. Query forward function Ψ_f

Let U_i be a user and $\{N_1,\ldots,N_k\}$ be her group of neighbours. Function Ψ_f determines whether U_i should submit the query q to the WSE or she should forward it to one of her neighbours. The selection of a certain user within $\{U_i,N_1,\ldots,N_k\}$ is equiprobable:

$$\Psi_f = \frac{1}{k+1}$$

3.4. User selection function Ψ

Considering a certain user U_i who wants to submit a query q. U_i has the following list of neighbours: $\{N_1,\ldots,N_k\}$. U_i executes Ψ in order to decide which user within $\{U_i,N_1,\ldots,N_k\}$ should submit q to the WSE. U_i is perfectly concealed when all the members of $\{U_i,N_1,\ldots,N_k\}$ have submitted the same number of queries generated by U_i (Viejo and Castellà-Roca, 2010). If U_i achieves this requirement, she will be hidden among the group $\{U_i,N_1,\ldots,N_k\}$. Thus, she will achieve a privacy level comparable to k-anonymity (Sweeney, 2002).

The system uses the probability P_s to equitably distribute the queries in a path of length two, *i.e.*, the source of the queries submits to the WSE the same number of queries as her neighbours and as the neighbours of her neighbours. This is possible because the number of neighbours and the number of neighbours of these neighbours are known. Since the complete topology of the social network is unknown, it is not possible to control the distribution of queries in paths longer than two. Therefore, more distant neighbours will submit a small quantity of queries to the WSE. As a result, the true source of the queries is hidden among a group with a path length of two.

Nevertheless, if the group is known, then, the WSE can obtain certain information. For example, consider a group of users $\{U_i,N_1,\ldots,N_k\}$ where each one has always submitted the same query. In this extreme situation, the WSE can know the structure of the group and it can be also capable of determining the direct and indirect (path of length two) connections that a certain user holds with the other members of the group. If the WSE gathers all this information, it is straightforward to find the centroid of the users who have sent the same number of queries and hence to identify the source of the queries.

This weakness is solved by modifying P_s in order to obfuscate the users within $\{U_i,N_1,\ldots,N_k\}$ adding more distant neighbours. In this way, the system creates a vector which contains the users $\{U_i,N_1,\ldots,N_k\}$. Each user in this vector appears repeated as many times as the number of neighbours she has. U_i appears only once. This vector is duplicated a random number of times. Finally, an interval of vector elements are deleted at random. The final vector which is obtained through this process is the new P_s.

Let θ be the interval of users which are pruned in order to create variance in the system. The variance prevents the WSE from calculating a group centroid. Let U_i be a certain user and $\{U_i,N_1,\ldots,N_k\}$ be the group of users which have submitted queries from U_i to the WSE. The group $\{U_i,N_1,\ldots,N_k\}$ is sorted in ascending order by the number of queries from U_i which they have submitted. If during several executions of the protocol the position of U_i in this group is the same, or very similar, the WSE could be able to identify the source of the query. Therefore, θ is randomly chosen between two percentages (these values have been calculated empirically and are justified in Section 5) and it can be recalculated as often as desired.

3.5. Selfishness function Υ (N)

Let U be a user that receives a query q from her neighbour N. U executes Υ in order to decide whether to accept q or not. The aim of Υ is to punish the users who behave in a selfish manner.

Initially, U assigns to each neighbour a probability p of accepting queries from that source. Value p is set to 1.0 (100% of probability) for each neighbour. For notation purposes, $p_{U,N}$ represents the probability of U to accept a query from N.

Assuming that U receives a query q from user N, U accepts q with probability $p_{U,N}$:

- If U accepts q, the following actions are done:
  - N increments her own probability of accepting queries from U:

    $p_{N,U} = p_{N,U} + (2 \cdot \zeta)$

    where ζ is a constant defined in the system. Value ζ is calculated empirically and justified in Section 5.
  - U decrements her probability of accepting queries from N:

    $p_{U,N} = p_{U,N} - \zeta$

  Both operations are done to incentivize users to accept queries from their neighbours. A certain user U who forwards several queries to the same neighbour N without accepting queries in exchange (selfish behavior) will finally get a $p_{N,U} = 0$. If the same situation happens with all her neighbours, U will be isolated and forced to submit all her queries to the WSE by her own. An isolated user is able to improve her situation by accepting queries from her neighbours. By decreasing ζ for a reject and increasing $2 \cdot \zeta$ for an accept, the protocol punishes the users that systematically reject all queries instead of the users that accept and reject queries in an even way.
- If U rejects q, the following happens:
  - N decrements her own probability of accepting queries from U:

    $p_{N,U} = p_{N,U} - \zeta$

We have simulated our scheme for several values of ζ to check out which one better isolates the selfish users of the system. The best results were achieved with $\zeta = 0.03$ (see Section 5 for details about the choice of this value). Therefore, this is the value used in the rest of this paper.

3.5.1. Coprivacy

The proposed scheme protects the privacy of the users in communities where most of the users are honest. If there is a large quantity of selfish users, the privacy of the honest users would be jeopardized (see Section 5 for more details about this). On the other hand, selfish users lose their privacy in both scenarios. Besides, an honest user alone cannot protect her own privacy. Therefore, honest users do not have any motivation to leave the network, they need each other. Regarding the selfish users, if they leave the social network, the system will work better. If they do not leave the network, they will be isolated by the honest users using the selfishness function Υ.

The situation where users collaborate in a system to preserve their privacy is named *coprivacy* and it was defined in Domingo-Ferrer (2010). A protocol is "coprivate" if the best option for a player to preserve her privacy is to help another player in preserving her own privacy.

4. Evaluation

All the schemes which have been surveyed in Section 2 expose the features that should be fulfilled by any system which provides privacy to WSE's users. These features are: privacy, protection against selfish users, usability and quality. Evaluating a system is based on assessing these four characteristics. In our case, the optimal values for ζ and θ are the ones that offer better results for these features.

To the best of our knowledge, there is no standard method to evaluate these four attributes. Accordingly, we propose the following ones which are later used to analyze our system:

- *Privacy*. The user profile must be kept hidden from the WSE. We propose the profile exposure level (PEL) which uses mutual information (see Section 4.1) in order to measure the level of exposure of the user profile.
- *Protection against selfish users*. If a user behaves in a selfish way she must be penalized. In our simulations, we use different values of ζ in order to find which one performs better against selfish users.
- *Usability*. This feature is crucial for the system. A completely secure scheme that suffers from high query delay will not be used by most of the users (see Section 4.3). In our work, the usability is evaluated using the average response time of a query.
- *Quality*. The quality refers to the similarities between the responses which are gathered using the proposed system and the responses which are received submitting the queries directly to the WSE. Our protocol submits original queries to the WSE (queries are not modified in any way). Thus, this feature is not considered in our evaluation.

4.1. Mutual information

Given two random discrete variables X and Y, that have sample spaces Ω_X and Ω_Y respectively, it can be considered that:

(1) The probability function of the variable X defined by $p(x)$ when, for all $x \in \Omega_X$, $p(x) = P(X = x)$.
(2) The probability function of the variable Y defined by $p(y)$ when, for all $y \in \Omega_Y$, $p(y) = P(Y = y)$.
(3) The joint probability function of the variables X and Y defined by $p(x, y)$ when, for all $x \in \Omega_X$ and $y \in \Omega_Y$, $p(x, y) = P(X = x, Y = y)$.
(4) The probability function of the variable X conditioned on the variable Y defined by $p(x/y)$ when, for all $x \in \Omega_X$ and $y \in \Omega_Y$, $p(x/y) = P(X = x/Y = y)$.

The mutual information, $I(X, Y)$, of two random variables X and Y is a measure that allows us to evaluate the information that each variable provides about the other. $I(X, Y)$ shows the amount of uncertainty in X which is removed by knowing Y. Mathematically, this is expressed as:

$$I(X, Y) = H(X) - H\left(\frac{X}{Y}\right)$$

where $H(X)$ is the entropy of the variable X and $H(X/Y)$ is the conditional entropy of the variable X given the variable Y. $H(X/Y)$ is defined as the uncertainty about X which still remains after Y is known. These entropies are expressed as:

$$H(X) = -\sum_{x} p(x) \cdot \log_2 p(x)$$

$$H\left(\frac{X}{Y}\right) = -\sum_{x,y} p(x, y) \cdot \log_2 p\left(\frac{x}{y}\right)$$

From the previous expressions we can obtain the following one:

$$I(X, Y) = -\sum_{x} p(x) \cdot \log_2 p(x) + \sum_{x,y} p(x, y) \cdot \log_2 p\left(\frac{x}{y}\right) = \sum_{x,y} p(x, y) \cdot \log_2 \frac{p(x/y)}{p(x)}$$

Being $p(x, y) = p(x/y) \cdot p(y)$, the former expression develops to:

$$I(X, Y) = \sum_{x,y} p\left(\frac{x}{y}\right) \cdot p(y) \cdot \log_2 \frac{p(x/y)}{p(x)}$$

In our scheme, X represents the queries which are originally generated by the user, *i.e.*, the queries she wants to submit. The variable Y represents the real queries that the user finally submits to the WSE.

Notation:

$\Omega_X = \{x_i\}_{i=1}^{m}$, set with the elements of X (without repetitions).

$\Omega_Y = \{y_i\}_{i=1}^{r}$, set with the elements of Y (without repetitions).

$C_X = \{c_{x_i}\}_{i=1}^{m}$, set with the cardinal of each element of X. $C_Y = \{c_{y_i}\}_{i=1}^{r}$, set with the cardinal of each element of Y.

$M = \sum_{i=1}^{m} c_{x_i}$, total number of elements of the set X, counting repetitions.

$R = \sum_{i=1}^{r} c_{y_i}$, total number of elements of the set Y, counting repetitions.

Calculation of $p(x)$ and $p(y)$:

Let us suppose that the probability of each element of X and Y is proportional to its cardinal. Then, $p(x)$ and $p(y)$ are computed as follows:

$$P(X = x_i) = \frac{c_{x_i}}{M}, \quad 1 \le i \le m.$$

$$P(Y = y_i) = \frac{c_{y_i}}{N}, \quad 1 \le i \le r.$$

Calculation of $p(x/y)$:

$P(X = x_i/Y = y_j)$ is calculated for each pair x_i, y_j where $1 \le i \le m$ and $1 \le j \le r$.

Fixing $y_j \in Y$, the possible situations and their calculations are:

(1) $y_j \notin X$. There is no information, *i.e.*, the probability is randomly assigned among the elements of X proportionally to the cardinal.

$$P\left(X = \frac{x_i}{Y} = y_j\right) = \frac{c_{x_i}}{M}, \ 1 \le \text{and} \le m.$$

(2) $y_j \in X$. Then there exists a $x_k \in X$ so that $x_k = y_j$.

(a) If $c_{y_j} \le c_{x_k}$, it is assumed that y_j *comes* from x_k and not from any other $x \in \Omega_x$.

$$P\left(X = \frac{x_k}{Y} = y_j\right) = 1$$

$$P\left(X = \frac{x_{k'}}{Y} = y_j\right) = 0, \quad 1 \le k' \le m, k \ne k'$$

(b) If $c_{y_j} > c_{x_k}$, it is assumed one of the following statements: (i) y_j *comes* from x_k and not from any other $x \in \Omega_x$, with the probability proportional to the cardinal of x_k; (ii) there is no information with probability proportional to the difference of the cardinals.

$$P\left(X = \frac{x_k}{Y} = y_j\right) = \frac{c_{x_k}}{c_{y_j}} + \frac{c_{y_j} - c_{x_k}}{c_{y_j}} \cdot \frac{c_{x_k}}{M}$$

$$P\left(X = \frac{x_{k'}}{Y} = y_j\right) = \frac{c_{y_j} - c_{x_k}}{c_{y_j}} \cdot \frac{c_{x_{k'}}}{M}, \quad 1 \le k' \le m, k \ne k'$$

4.2. Profile exposure level (PEL)

Let Ω_X be the real queries of the user and Ω_Y be the queries sent to the WSE. Note that, Ω_X and Ω_Y can be seen as two random variables X and Y, respectively, which can take so many values as different queries they have and with probability proportional to the number of repetitions. Accordingly, we define the profile exposure level (PEL) as follows:

$$PEL = \frac{I(X, Y)}{H(X)} \cdot 100$$

where $H(X)$ is the entropy of the original set of queries and $I(X, Y)$ is the mutual information between X and Y. *PEL* measures the percentage of the user information that is exposed when Y is disclosed. Thus, the user information is calculated as the entropy of X, and the mutual information gives a measure of the information that Y provides about X (*i.e.*, if Y is known, how much does this reduce the uncertainty about X?).

4.3. Usability measure

The international standard ISO/IEC 9126 (Software engineering-Product quality) (Iso/iec, 2001), defines usability as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use. More specifically, we focus on whether the users can easily accomplish intended tasks at their desired speed or not.

In the proposed scheme, a task is considered to be the process of submitting a query and receiving the answer. The period of time needed to execute this process is named *response time*. Systems based on Tor networks obtain large response times. For instance, the authors in Saint-Jean et al. (2007) got an average response time of 10 s. We argue that this time is too long for a web search system to be usable. Therefore, the objective is to reduce it as much as possible.

The period of time between the submission of a query to the WSE and the reception of the answer can be decomposed as follows:

(a) The query goes from the user to the WSE.
(b) The WSE processes the query and generates an answer.
(c) The answer goes from the WSE to the user.

In our scheme, there is an extra step where the message is forwarded between a certain number of users before reaching the WSE. The answer is returned through the reverse path. Therefore, the average response time of a query is:

$$\text{Response time} = (2 \cdot \#hops \cdot latency) + time_{WSE}$$

Where:

- $time_{WSE}$ is the time needed by the WSE to answer the query. On average, this time is 400 ms (Viejo and Castellà-Roca, 2010).
- *latency* is the round-trip time (RTT) between two peers. The study presented in Choffnes and Bustamante (2008) determines that the average latency between two random users in a worldwide P2P network is 530 ms.
- *hops* is the average number of hops that a query performs before reaching the WSE. This value has been obtained from the simulations.

5. Simulations

The evaluation of the proposed system has been done simulating various social networks between 1000 and 10,000,000 users. In these networks, each user is directly connected with a number of users between 1 and 10 following a power-law distribution (Liben-Nowell, 2005).

5.1. Tests

The evaluation process includes two different types of tests:

- The first type checks the equitable distribution of messages around the network. Each user generates a unique query and sends it many times. This is the worst case possible because all the queries which are submitted by the same user are equal and different from the queries sent by other users. As a result, these queries can be easily linked together. Nevertheless, if the system works correctly, the user who has generated all these queries remains hidden among the set of users who have submitted them. These kind of tests use synthetic queries (queries generated at random by a computer). Besides, the results of these tests provide the optimal values for ζ (see Section 3.5).

Table 1
Average position, variance and deviation obtained with different θ intervals.

θ Interval	Average position	Variance	Deviation
0–0%	5.81	8.34	2.89
80–40%	4.36	11.54	3.39
60–40%	4.46	11.37	3.37
80–20%	4.51	10.71	3.27
80–60%	4.51	12.5	3.53
40–20%	4.49	10.13	3.18
30–10%	4.42	10.12	3.18
30–20%	4.46	10.21	3.19
20–10%	4.3	9.67	3.11

- The second type of tests use real queries (queries generated by humans) in order to evaluate the privacy level achieved by the users. These queries were extracted from the AOL file (Aol, 2006). This file shows which queries were submitted by each AOL user (note that the real identity of each AOL user is not disclosed, only her queries). In this way, in our tests, a certain simulated user gets the personality of a certain AOL user. Therefore, the simulated user only sends the queries which were generated by her assigned AOL user. Note that each AOL user submitted a different number of queries, hence, each simulated user also submits a different number of queries. Therefore, the evaluation process considers simulated users who send hundreds of queries together with simulated users who submit only a few. Their privacy level may vary according to the particular number of queries that each one sends.

5.2. Privacy

Prior to evaluate the privacy offered by our proposal we need to determine the interval θ of users that are pruned in order to introduce variance into the system. We run 1.000 tests using social networks of 1000 users. Table 1 shows the average position, variance and deviation of the users with 10 neighbours for different intervals of θ. The highest variance is obtained by the 80–60% interval. This is the interval which has been used in the rest of the simulations.

Fig. 1 shows the average position and the deviation of the users according to the number of neighbours that they have. For a good comparison, we have included the ideal average position. It can be observed that the average position does not differ much while the position of the users within the group obtains a large deviation.

5.2.1. Simulation results from scenarios without selfish users

The optimal system behavior is achieved when all the users follow the protocol. We performed tests with 1000 users from the AOL files (Aol, 2006) to determine the level of privacy that is achieved in this situation. The mutual information (see Section 4.1) returns the entropy reduction in bits. This information is not useful without the initial uncertainty, hence, we use as a measure the percentage of information that involves the mutual information about the initial entropy. Hereafter, we consider that a user is exposed when this percentage is less than 40%, *i.e.*, above 60% of the initial entropy.

Table 2 shows the average number of exposed users according to the number of neighbours they have. We have completed the table with the number of users who have an uncertainty percentage above 70% and 80%.

Generally, the users with fewer connections are the most exposed ones. However, there are some users with many connections who also expose their profile to the WSE. This case can occur when the number of queries of these users is small or when their neighbours have sent them a small number of queries.

In Table 3 it can be seen that nearly 90% of the users expose less than 20% of their profile. Moreover, taking into account that a user

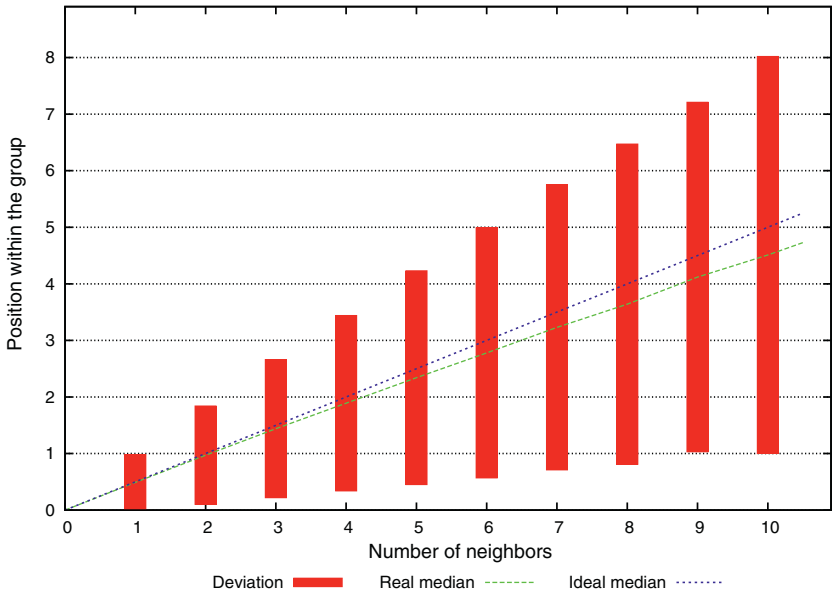

Fig. 1. The ideal and simulated average position and the deviation for each number of neighbours.

Table 2
Number of exposed users according to various percentages of exposure.

# Neighbours	# Exposed 60%	# Exposed 70%	# Exposed 80%
1	15.4	15.1	4.7
2	12.6	12	4.7
3	5.2	4	1.2
4	4.67	4	2
5	1	1	1
6	1.3	0	0
7	1.2	0	0
8	0	0	0
9	0	0	0
10	0	0	0
	41.37	36.1	13.6

is exposed when at least 60% of her profile has been revealed, more than 95% of the users preserve their privacy.

5.2.2. Simulation results from scenarios with selfish users

Before evaluating the privacy offered by the system in front of selfish users, we have to determine the ζ value which must be applied. The optimal value is important in order to prevent honest users from jeopardizing their privacy. We have run 100 tests over social networks of 1000 users. 10% of these users were set to behave selfishly. Each user submits her own query 100 times.

Table 4 shows the percentage of honest users and selfish users that expose their profile for several ζ values. It can be observed that the number of honest users who are exposed grows when ζ is increased. Thus, the optimal penalty value is 0.03. This is the value used in the rest of this paper.

In order to evaluate the privacy offered by the system in environments with selfish users, we run 100 test over various social networks of 1000 users using real data.

Fig. 2 presents the percentage of exposed users for different percentages of selfish users. Note that, when there are more selfish users, the number of exposed honest users also grows.

With the above tests we have also calculated the number of queries that a selfish user submits to the WSE in a system's execution with real data or synthetic data. Fig. 3 shows the results achieved according the number of neighbours.

Table 3
Percentage of similarity between the mutual information and the initial entropy for the users.

% Similarity	# Users	% Users
10	813.1	81.31
20	82	8.2
30	22.2	2.22
40	20.1	2.01
50	11.3	1.13
60	11	1.1
70	14.7	1.47
80	12.9	1.29
90	10.1	1.01
100	2.6	0.26

5.3. Usability

In Section 4.3, we have defined the term *usability*. In this section, our target is to obtain the average number of hops. We have performed several simulations with social networks of 10,000 users. In each simulation, each user generates 1000 queries. The results show that the average number of hops is 3.59. Therefore, the average response time of a query is $(2 \cdot 3.59 \cdot 530\,\text{ms}) + 400\,\text{ms} = 4205.4\,\text{ms}$.

This time is slightly worse than the 3914 ms obtained by Viejo and Castellà-Roca (2010). However, the privacy level achieved by our scheme is significantly better. The controlled distribution of queries allows the system to hide the source user among her social network group with a path of length two.

In Table 5, it can be observed that the new system achieves larger deviations and average positions closer to the ideal than the previous one.

Table 4
Percentage of exposed users for different ζ values.

ζ	% Exposed honest users	% Exposed selfish users
0.00	2.4	2
0.01	2.3	5
0.02	2.2	93
0.03	2.2	100
0.04	2.4	100
0.06	2.56	100
0.08	2.89	100
0.10	2.89	100

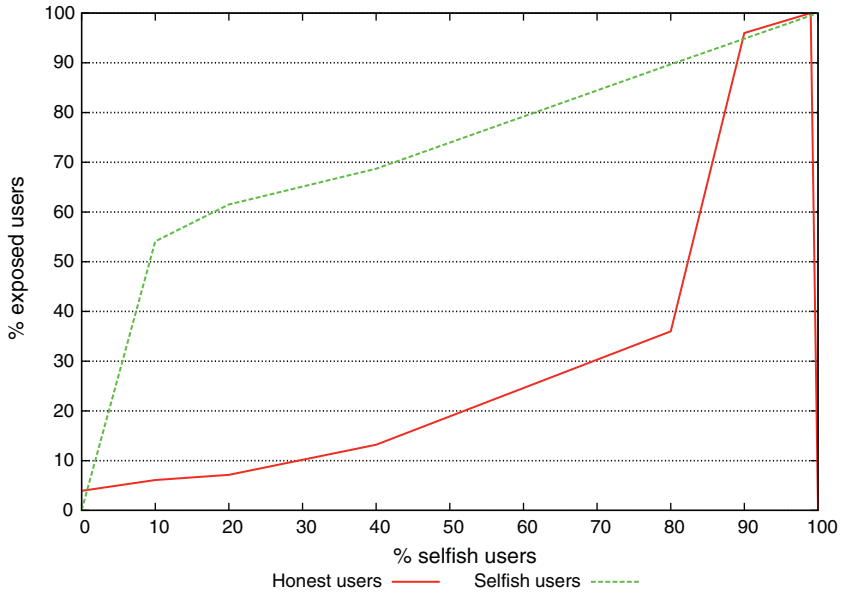

Fig. 2. Percentage of exposed users for different percentages of selfish users.

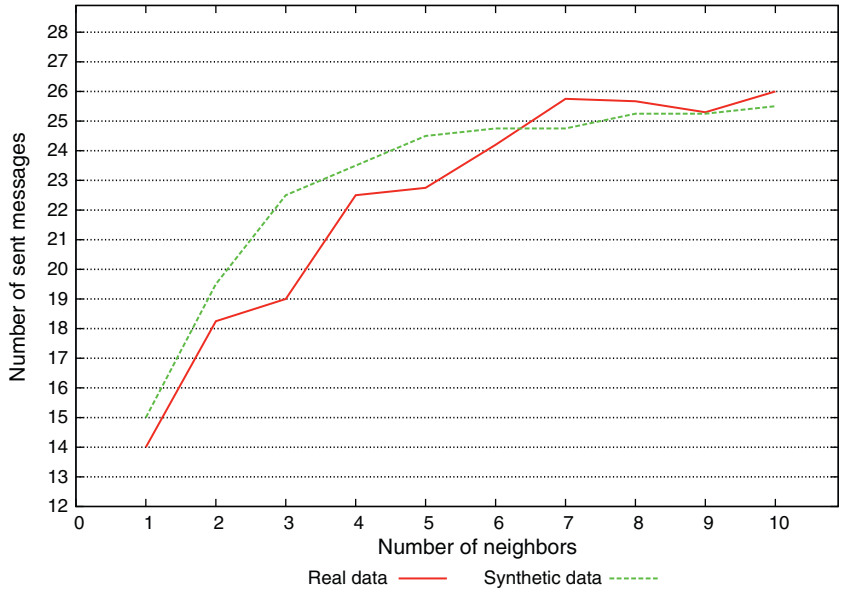

Fig. 3. Average number of queries which are submitted by selfish users.

As we have mentioned before, proposals based on Tor have an average response time of 10,000 ms with paths of length two. The UUP (Castellà-Roca et al., 2009) achieves a response time of 5200 ms. This is a 20% worse than the time attained by our proposal. In addition to that, our scheme improves the quality of the search results because it considers the trade-off between the privacy level achieved and the quality of the service.

Table 5
Average position and deviation obtained with our system and its previous version Viejo and Castellà-Roca, 2010.

Category	Average position	Previous average position	Deviation	Previous deviation
1 neighbour	0.49	1.01	0.49	0.01
2 neighbours	0.97	1.08	0.87	0.08
3 neighbours	1.44	1.68	1.22	0.72
4 neighbours	1.89	2.11	1.55	0.95
5 neighbours	2.34	3.23	1.89	1.21
6 neighbours	2.78	3.95	2.21	1.55
7 neighbours	3.23	5.15	2.52	2.15
8 neighbours	3.64	6.32	2.83	2.41
9 neighbours	4.12	7.30	3.09	2.40
10 neighbours	4.51	7.00	3.51	2.91

Regarding the response time of a direct WSE search, note that this search method does not protect the privacy of the users. The privacy-protection process represents a cost for the users. They should consider whether their privacy deserves this cost or not.

5.4. Discussion

In Viejo and Castellà-Roca (2010), the privacy of a certain user U who is generating certain queries is obtained by concealing her into a group Y of k users. This group is formed by the users who have submitted the queries generated by U. The users in Y are ordered according to the number of queries that each one has submitted to the WSE.

The authors of this work state that the WSE can identify U if her position in the ordered group Y is always the same. According to that, U should ideally be situated in the middle of Y, but the deviation of this position should be high. A high deviation implies that it is difficult for the WSE to ascertain the exact position of U in Y. Thus, the WSE cannot identify U.

In our proposal, the privacy level is not evaluated using these requirements. We have defined and used the PEL instead. Therefore, in order to compare both protocols we have calculated the average position and deviation achieved by our system and we have compared those results with the results attained by the former protocol (Viejo and Castellà-Roca, 2010). Table 5 shows that the new proposal obtains larger deviations and average positions which are closer to the ideal ones (see Fig. 1). Thus, the privacy level achieved by our scheme is significantly better.

There is a lack of standard measures to evaluate the privacy achieved by the users in this type of systems so we have proposed a new measure: the profile exposure level (PEL). PEL measures the percentage of the personal information that is exposed when a certain user submits her queries to the WSE. This measure can be used by any entity that knows which queries have been generated by the user and which queries have been submitted by the user. Therefore, each user can compute her own PEL because she knows this information.

As explained previously, we have used real queries (queries generated by humans) in our tests. These queries have been obtained from the AOL file. Former proposals in the literature used synthetic queries (queries generated by computers) to test their behavior. From the validity point of view, our approach is more accurate and hence the results of our simulations are more trustworthy.

Nevertheless, our proposal has been only tested on simulated social networks. Real social networks (*e.g.*, Facebook, Windows Live Messenger, etc.) have not been considered in our evaluation process. Therefore, we cannot assume that the results would be exactly the same in a real network. However, we consider that the behavior should be quite similar since the simulation process has been realized trying to replicate a real environment.

6. Conclusions and further research

The queries that a user submits to a WSE can disclose a lot of information about her. Some disclosure incidents have proved that we cannot trust in the WSEs in order to keep our personal data safe. Accordingly, users should preserve their privacy on their own.

We have described in this paper a new version of the protocol presented in Viejo and Castellà-Roca (2010) which preserves the privacy of the users by distorting their profiles. The new version improves the privacy achieved by the users maintaining its usability.

In addition to that, we have proposed a new measure to estimate the privacy achieved by the users, the *profile exposure level (PEL)*. Also, for the first time – to the best of our knowledge – the tests have been performed using real data which was extracted from AOL's files. In this way, the validity of the proposed system has been demonstrated.

We have simulated the performance of our proposal and the results show that it can be successfully deployed in real environments because: (i) it provides an acceptable query delay; (ii) it offers privacy to users who have a reasonable number of direct connections; and (iii) it works properly in scenarios with users who do not follow the protocol.

In the future, we will consider the use of specialized social networks in order to get more homogenous shared interests between the users of the network. This enhancement should improve the quality of the service. Nevertheless, there are some privacy issues that must be investigated.

In addition to that, our scheme has been simulated considering that a certain number of users (specifically, their computers) spend most of the day connected to the Internet. This assumption is quite optimistic and it might not be suitable for some scenarios (*e.g.*, transition countries with underdeveloped technology infrastructures). Accordingly, the proposed scheme should be evaluated in a more dynamic environment. More exactly, we plan to study its behaviour in networks where users become online/offline frequently.

Disclaimer

The authors are with the UNESCO Chair in Data Privacy, but they are solely responsible for the views expressed in this paper, which do not necessarily reflect the position of UNESCO nor commit that organization.

Acknowledgements

This work was partly supported by the Spanish Ministry of Science and Innovation through projects TSI2007-65406-C03-01 "E-AEGIS", CONSOLIDER CSD2007-00004 "ARES", PT-430000-2010-31 "Audit Transparency Voting Process", by the Spanish Ministry of Industry, Commerce and Tourism through projects TSI-020100-2009-720 "eVerification", TSI-020302-2010-153 "SeCloud", and by the Government of Catalonia under grant 2009 SGR 1135.

References

Anderson, R., 1996. The eternity service. In: Proc. PRAGO-. CRYPT 96 , pp. 242–252.
Aol, AOL Keyword Searches, 2006. http://dontdelete.com/default.asp.
Barbaro, M., Zeller, T., 2006. A face is exposed for aol searcher no. 4417749, New York Times (August).
Castellà-Roca, J., Viejo, A., Herrera-Joancomartí, J., 2009. Preserving user 1/2s privacy in web search engines. Comput. Commun. 32, 1541–1551.
Chaum, D., 1981. Untraceable electronic mail, return addresses, and digital pseudonyms. Commun. ACM 24 (2), 84–90.
Choffnes, D., Bustamante, F., 2008. Taming the torrent: a practical approach to reducing cross-isp traffic in peer-to-peer systems. In: SIGCOMM '08: Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication , pp. 363–374.
Chor, B., Goldreich, O., Kushilevitz, E., Sudan, M., 1995. Private information retrieval. In: FOCS '95: Proceedings of the 36th Annual Symposium on Foundations of Computer Science.
Chor, B., Goldreich, O., Kushilevitz, E., Sudan, M., 1998. Private information retrieval. J. ACM 45, 965–981.
Domingo-Ferrer, J., 2007. A public-key protocol for social networks with private relationships. In: MDAI '07: Proceedings of the 4th International Conference on Modeling Decisions for Artificial Intelligence , pp. 373–379.
Domingo-Ferrer, J., Bras-Amorós, M., 2008. Peer-to-peer private information retrieval. In: PSD '08: Proceedings of the UNESCO Chair in Data Privacy International Conference on Privacy in Statistical Databases , pp. 315–323.
Domingo-Ferrer, J., Solanas, A., Castellà-Roca, J., 2009. h(k)-private information retrieval from privacy-uncooperative queryable databases. J. Online Inf. Rev. 33, 720–744.
Domingo-Ferrer, J., 2010. Coprivacy: towards a theory of sustainable privacy, PSD2010. LNCS 6344, 258–268.
Elovici, Y., Shapira, B., Maschiach, A., 2002a. A new privacy model for web surfing. In: NGITS '02: Proceedings of the 5th International Workshop on Next Generation Information Technologies and Systems , pp. 45–57.

Elovici, Y., Shapira, B., Maschiach, A., 2002b. A new privacy model for hiding group interests while accessing the web. In: WPES '02: Proceedings of the 2002 ACM Workshop on Privacy in the Electronic Society , pp. 63–70.
Foxtor, 2009. http://www.romanosky.net.
Google personalized search, 2009. http://www.google.com/psearch.
Internet world stats, 2008. http://www.internetworldstats. om/stats.htm.
iprospect.com, inc., 2009. iProspect Blended Search Results Study, http://www.iProspect.com.
Iso/iec 9126–1, 2001. Software engineering-product quality. Part 1. Quality model.
Kuflik, T., Shapira, B., Elovici, Y., Maschiach, A., 2003. Privacy preservation improvement by learning optimal profile generation rate. User Modeling. 168–177.
Kushilevitz, E., Ostrovsky, R., 1997. Replication is not needed: single database computationally-private information retrieval. In: Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science , pp. 364–373.
Liben-Nowell, D., 2005. An algorithmic approach to social networks. Ph.D. thesis. MIT Computer Science and Artificial Intelligence Laboratory.
Mislove, A., Gummadi, K., Druschel, P., 2006. Exploiting social networks for internet search. In: Proceedings of the 5th Workshop on Hot Topics in Networks (HotNets'06).
Netcraft, 2009. http://news.netcraft.com.
Pitkow, J., Schütze, H., Cass, T., Cooley, R., Turnbull, D., Edmonds, A., Adar, E., Breuel, T., 2002. Personalized search. Commun. ACM 45 (9), 50–55.
Privoxy, 2009. http://www.privoxy.org.
Qiu, F., Cho, J., 2006. Automatic identification of user interest for personalized search. In: Proceedings of the 15th International Conference on World Wide Web , pp. 727–736.
Reiter, M., Rubin, A., 1998. Crowds: anonymity for web transactions. ACM Trans. Inf. Syst. Secur. 1 (1), 66–92.
Reiter, M., Rubin, A., 1999. Anonymous web transactions with crowds. Commun. ACM 42 (2), 32–48.
Saint-Jean, F., Johnson, A., Boneh, D., Feigenbaum, J., 2007. Private web search. In: WPES '07: Proceedings of the 2007 ACM Workshop on Privacy in Electronic Society , pp. 84–90.
Shen, X., Tan, B., Zhai, C., 2007. Privacy protection in personalized search. SIGIR Forum 41 (1), 4–17.
Speretta, M., Gauch, S., 2004. Personalized search based on user search histories. In: Proceedings of the International Conference of Knowledge Management-CIKM'04.
Steel, E., 2010. A web pioneer profiles users by name, The Wall Street Journal (October).
Sugiyama, K., Hatano, K., Yoshikawa, M., 2004. Adaptive web search based on user profile constructed without any effort from users. In: Proceedings of the 13th International Conference on World Wide Web , pp. 675–684.
Sweeney, L., 2002. k-Anonymity: a model for protecting privacy. Int. J. Uncertainty Fuzziness Knowledge-based Syst. 10 (5), 557–570.
Teevan, J., Dumais, S., Horvitz, E., 2005. Personalizing search via automated analysis of interests and activities. In: SIGIR '05: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval , pp. 449–456.
Tor project, 2009. http://www.torproject.org.
Trackmenot, 2009. http://www.mrl.nyu.edu/dhowe/trackmenot/.
Viejo, A., Castellà-Roca, J., 2010. Using social networks to distort users' profiles generated by web search engines. Comput. Networks 54, 1343–1357.
Xu, Y., Wang, K., Zhang, B., Chen, Z., 2007. Privacy-enhancing personalized web search. In: WWW '07: Proceedings of the 16th International Conference on World Wide Web , pp. 591–600.

Arnau Erola is a Ph.D. student of computer science at Rovira i Virgili University. He got his B.Sc. in computer engineering from Rovira i Virgili University in 2006. In 2009, he got his Master in Information and Security Engineering from Rovira i Virgili University. His research interests are data privacy and data security.

Jordi Castellà-Roca (Menàrguens, Catalunya, 1975) is tenured assistant professor at Rovira i Virgili University, he is currently a member of the UNESCO Chair in Data Privacy. He got his title of Engineer in computer systems from University of Lleida in 1998, the title of Engineer in computer science from Rovira i Virgili University in 2000 and Ph.D. in computer science from the Autonomous University of Barcelona in 2005. His research focuses on the fields of cryptography (cryptographic protocols) and privacy. He has published over 40 works in international journals, book chapters, international and national congresses. He has participated in 23 Spanish-funded and Catalan-funded research projects. He has been the main researcher in two research projects funded by Rovira i Virgili University, one funded by the Ministry of Science and innovation and two funded by the Ministry of Industry, Tourism and Trade. He has also participated in several transfer projects, and he is the author of six patents, five of them international and in operation. He is a founding partner of three technology companies that have been awarded.

Alexandre Viejo is a tenure-track lecturer at Rovira i Virgili University (Tarragona, Spain). He received his Ph.D. in computer science from Rovira i Virgili University in 2008. He received a Master in telematics engineering from the Technical University of Catalonia (Barcelona, Spain) in 2007. He got his M.Sc. in computer engineering from Rovira i Virgili University in 2005. In 2009, he was a researcher at Humboldt-Universität zu Berlin (Berlin, Germany). His fields of activity are data privacy, data security and cryptographic protocols.

Josep Maria Mateo-Sanz (Tarragona, 1964) is a tenured assistant professor at Rovira i Virgili University. He got the M.Sc. and the Ph.D. in mathematics at University of Barcelona in 1992 and 1998, respectively. His fields of activity are data privacy and data security using statistical techniques. In 2010, he received the Teaching Quality award of the Rovira i Virgili University. In the same year, he received the Jaume Vicens Vives award of the Generalitat de Catalunya to the University Teaching Quality. He has published over 40 works, one of them was ISI Highly Cited Paper at the beginning of 2005. He has also authored more than 50 congress contributions. He has participated in various Spanish, European and North American projects. He has also participated in research contracts with the U.S. Bureau of the Census, Yahoo Inc. and the Statistical Institute of Catalonia, among others. He is currently a member of the CONSOLIDER ARES Group.