The current issue and full text archive of this journal is available at
www.emeraldinsight.com/1468-4527.htm

Refereed article received
20 July 2008
Approved for publication
10 March 2009

h(*k*)-private information retrieval from privacy-uncooperative queryable databases

Josep Domingo-Ferrer, Agusti Solanas and Jordi Castellà-Roca
Department of Computer Engineering and Mathematics, Rovira i Virgili University, Tarragona, Spain

Abstract

Purpose – This paper aims to address the privacy problem associated with the use of internet search engines. The purpose of the paper is to propose and validate a set of methods and protocols to guarantee the privacy of users' queries.

Design/methodology/approach – In this paper *h*(*k*)-private information retrieval (*h*(*k*)-PIR) is defined as a practical compromise between computational efficiency and privacy. Also presented are *h*(*k*)-PIR protocols that can be used to query any database, which does not even need to know that the user is trying to preserve his or her privacy.

Findings – The proposed methods are able to properly protect the privacy of users' queries. When internet users apply the protocols, search engines (e.g. Google) are not able to determine unequivocally the real interests of their users. The quality of the results does decrease with the increase in privacy, but the obtained trade-off is excellent.

Practical implications – Current private information retrieval (PIR) protocols suffer from two significant shortcomings: their computational complexity is $O(n)$ where n is the number of records in the database, which precludes their use for very large databases and web search engines; and they assume that the database server cooperates in the PIR protocol, which prevents deployment in real-life uncooperative settings. The proposed protocols overcome both problems.

Originality/value – This is the first set of protocols that offer practical protection for the privacy of the queries that internet users submit to an internet search engine. The proposal has been implemented and it will be released to the general public soon. It will help to protect the right to privacy of millions of internet users.

Keywords Information retrieval, Search engines, Internet

Paper type Research paper

Introduction

Privacy in databases can be viewed as a three-dimensional property (Domingo-Ferrer, 2007), because a distinction can be made between respondent privacy (privacy of the respondents to whom the database records correspond), owner privacy (privacy of proprietary datasets used to conduct joint research between several data-owning

The authors are with the UNESCO Chair in Data Privacy, but the views expressed in this article are those of the authors and do not necessarily reflect the position of UNESCO nor commit that organisation. Thanks go to Susana Bujalance for her help in implementing the *GooPIR* prototype and to Úrsula González-Nicolás for her help in computing the outcomes of quality measures. This work was partly supported by the Spanish Government through projects CONSOLIDER INGENIO 2010 CSD2007-00004 "ARES" and TSI2007-65406-C03-01 "E-AEGIS", and by the Government of Catalonia under grant 2005 SGR 00446.

Online Information Review
Vol. 33 No. 4, 2009
pp. 720-744

1468-4527
DOI 10.1108/14684520910985693

organisations) and user privacy (privacy of the queries submitted by database users). In the context of interactively queryable databases and in particular, internet search engines, the most rapidly growing concern is user privacy. This is especially so after scandals like the August 2006 disclosure by the AOL search engine of 20 million queries made by 658,000 users (AOL, 2006). It is estimated that over 223 million data records of US residents have been exposed due to security breaches since January 2005 (Lane *et al.*, 2008), so an attacker could access user profiles stored in the databases of search engines.

There are personal and corporate motivations for requiring user privacy:

- *Private life*. Most users feel uncomfortable about being profiled by a search engine or a database.
- *Industrial life*. If a company is querying a database containing patents or stock market quotations, the database manager can partially infer the company's next technological or financial moves if he or she knows the queries being submitted by the company.

Private information retrieval (PIR) was invented in 1995 by Chor *et al.* (1995, 1998) with the assumption that there are at least two copies of the same database, which do not communicate with each other. In those same papers, Chor *et al.* showed that single-database PIR (that is, with a single copy) does not exist in the information-theoretic sense. However, two years later, Kushilevitz and Ostrovsky (1997) presented a method for constructing single-database PIR based on the algebraic properties of the Goldwasser-Micali public-key encryption scheme (Goldwasser and Micali, 1984). Subsequent developments in PIR are surveyed in Ostrovsky and Skeith (2007).

In the PIR literature the database is usually modelled as a vector. The user wishes to retrieve the value of the i-th component of the vector while keeping the index i hidden from the database. Thus it is assumed that the user knows the physical address of the sought item, which might be too strong an assumption in many practical situations. Keyword PIR (Chor *et al.*, 1997; Kushilevitz and Ostrovsky, 1997) is a more flexible form of PIR – the user can submit a query consisting of a keyword and no modification of the structure of the database is needed.

We claim that the PIR protocols proposed so far have two fundamental shortcomings that hinder their practical deployment:

(1) The database is assumed to contain n items and PIR protocols attempt to guarantee maximum privacy, that is, maximum server uncertainty on the index i of the record retrieved by the user. Thus the computational complexity of such PIR protocols is $O(n)$, as proven by Chor *et al.* (1995, 1998). Intuitively, all records in the database must be "touched"; otherwise, the server could rule out some of the records when trying to discover i (an implicit standard assumption in PIR is that no side information is available to the server allowing it to rule out a touched record). For large databases, an $O(n)$ computational cost is unaffordable (Beimel *et al.*, 2004).

(2) It is assumed that the database server cooperates in the PIR protocol. However, it is the user who is interested in his or her own privacy, whereas the motivation for the database server is dubious. Actually, PIR is likely to be unattractive to

most companies running queryable databases, as it limits their profiling ability. This probably explains why no real instances of PIR-enabled databases seem to exist.

If one wishes to run PIR against a search engine, there is another shortcoming beyond the lack of server cooperation – the database cannot be modelled as a vector in which the user can be assumed to know the physical location of the keyword sought. Even keyword PIR does not really fit, as it still assumes a mapping between individual keywords and physical addresses (in fact, each keyword is used as an alias of a physical address). A search engine allowing only searches of individual keywords stored in this way would be much more limited than real engines like Google or Yahoo.

TrackMeNot (Howe and Nissenbaum, 2009) is a practical system based on the principle that a browser extension installed in the user's computer hides the user's actual queries in a cloud of automatic ghost queries submitted to popular search engines at different time intervals. While useful and affordable on a small scale, if the use of TrackMeNot became generalised, the overhead introduced by ghost queries would significantly degrade the performance of search engines and communications networks. Also, the submission timing of automatic ghost queries may be distinguishable from the submission timing of actual queries, which could provide an intruder with clues to identify the latter type of queries.

In this paper the notion of $h(k)$-PIR is introduced as a pragmatic approach to private information retrieval from real search engines. Given a user's query consisting of one or several target keywords, the basic idea is to achieve some level of privacy by adding $k - 1$ bogus keywords to the user's actual target keywords. In this way, the database or search engine cannot determine with certainty which is (are) the user's target keyword(s). The higher the number $k - 1$ of bogus keywords and the more similar the frequency of the target and the bogus keywords, the more privacy. Then a protocol is proposed that enables a user to perform $h(k)$-PIR against a web search engine or database server that is uncooperative in preserving the privacy of user queries. By "uncooperative", we mean that the engine or server does not offer any specific PIR functionality (this is usually the case in reality); however, we assume that the server is semi-honest, in the sense that the server may be interested in user profiling but will correctly answer any user query.

$h(k)$-Private information retrieval

Defining privacy in terms of maximum index uncertainty is impractical if the database consists of billions of web pages indexed by a search engine. Even if the stored web pages could be modelled as a vector with n records, n would be so large that the $O(n)$ computational complexity of a PIR query would be unaffordable.

Privacy requirements must be relaxed. A possible relaxation is $h(k)$-PIR, whose idea is to ensure that the uncertainty of the database or search engine about which is the actual user target keyword is no less than a lower bound expressed in terms of a privacy parameter k. In what follows, Shannon's entropy[1] will be used to quantify this lower bound on query uncertainty, because it is the most common uncertainty measure.

Definition 1: h(k)*-private information retrieval*
Given a non-negative integer k and a function $h(\cdot)$ such that $h(k) \geq 0$, a query protocol against a queryable database or a search engine provides $h(k)$-private information retrieval, or $h(k)$-PIR for short, for a user query q_0 if any intruder (including the database server or the search engine) views the user query as a random variable Q_0, whose Shannon's entropy satisfies $H(Q_0) \geq h(k)$.

In plain words, the above definition means that the query appears to the intruder as a variable Q_0, which can take several possible values, so that the intruder cannot unequivocally determine which was the value q_0 actually submitted by the user.

Example 1. If in the intruder's view there are k queries that are equally likely candidates to be the target query q_0 submitted by the user, then $H(Q_0) = \log_2 k$ and the protocol provides $\log_2 k$-privacy (the highest possible privacy with k candidates). If the probabilities of the k candidates being q_0 are different, $H(Q_0) < \log_2 k$ and therefore $h(k)$-privacy with $h(k) < \log_2 k$ is achieved. If the protocol is such that the intruder knows for sure which of the k candidates is q_0, then the protocol provides only 0-privacy, that is, no privacy at all. Obviously, for any non-negative k' and any function $\hat{h}(\cdot)$ such that $0 \leq \hat{h}(k') \leq h(k)$, a $h(k)$-private protocol is also $\hat{h}(k')$-private.

The computational complexity of protocols implementing $h(k)$-PIR can be as low as $O(k)$, whatever the size of the database. The user can camouflage the target query within a set of k equally likely or almost equally likely queries (the target query plus $k - 1$ bogus queries); then the user submits the set of queries and finally filters out the results relevant to the target query.

Note that the relaxation introduced by $h(k)$-PIR with respect to PIR is somewhat parallel to the one introduced by k-anonymity (Samarati and Sweeney, 1998) with respect to total anonymity.

Single-user *h*(*k*)-PIR from a privacy-uncooperative queryable database

In this section we start with the assumption that queries consist of a single keyword – the query output is expected to consist of the database records or web URLs containing that keyword. (An extension for multi-keyword queries is given later in this paper.) A relatively simple way to approach $h(k)$-PIR has been suggested at the end of the previous section – $k - 1$ bogus keywords are added to the target query before submitting it. Adding means OR-ing the target keyword with the bogus keywords. Some considerations are in order here:

(1) It will be assumed that keywords belong to some designated language (e.g. English and/or a domain-specific language).

(2) A public reference thesaurus with a large number N of words and proper nouns in the designated language will be used to draw bogus keywords from. The thesaurus should provide the keywords along with their relative frequencies. There are several alternatives to obtain such a thesaurus:

- Query logs are possibly the best option to build the thesaurus. Keyword frequencies are taken based on the keyword appearances in the log. A good example of a query log is the above-mentioned search data released by AOL (2006).
- Online encyclopaedias, newspapers or other text collections are good alternatives for building the list of keywords in the thesaurus when no query

log is available for the designated user language. Keyword frequencies can be taken based on the keyword appearances in the text collection or from the frequency output by a search engine (e.g. Google) when the keyword is looked up.

- For queries in English, one can also follow the procedure suggested by Staddon *et al.* (2007): use the British National Corpus (BNC) (Leech *et al.*, 2001) to stem lexical keywords (e.g. the keywords "use", "used", "uses" and "using" would all be mapped to the stem "use"); use the BNC again to associate to each keyword the frequency of the corresponding stem.

(3) OR-ing bogus queries to the target query results in some overhead, because the output returned by the search engine needs to be filtered out to suppress the portion related to the bogus queries.

(4) For privacy reasons, bogus query addition and output filtering must be done locally at the user's computer.

As a first approach, a standalone user can run the following algorithm.

Protocol 1 (Naïve(q_0, k*))*

(1) N: To mask the real target keyword q_0, locally sample the thesaurus to randomly draw $k-1$ bogus keywords $q_1, \quad q_2, \quad \Lambda, \quad q_{k-1}$.

(2) Submit the query:

$$q_{\pi(0)} OR_{q\pi(1)} OR \Lambda_{q\pi(k-1)}$$

where π is a random permutation of the set $\{0, 1, \Lambda, k-1\}$.

(3) Once the query results are received, keep those results where the keyword q_0 appears and discard the rest.

Note that the random permutation π in Protocol 1 is needed to hide the place of q_0 in the submitted query.

After masking q_0 with the bogus keywords $q_1, \quad q_2, \quad \Lambda, \quad q_{k-1}$, the database, search engine or, more generally, any intruder observing the masked query is uncertain about which one among the k keywords is the real target keyword q_0. In fact, one can say that an intruder views the target keyword as a random variable Q_0 with possible outcomes $q_0, \quad q_1, \quad \Lambda, \quad q_{k-1}$. The intruder's uncertainty about the actual value of the target keyword can be measured as the entropy of Q_0 – the more similar the frequencies of the q_i's, the higher the entropy, which makes sense because the intruder's uncertainty is higher (which is good).

This is formalised by the following lemma.

Lemma 1. Let q_0 be the target keyword a user is interested in, and let $f(q_0)$ be its relative frequency in the reference thesaurus. Let $q_1, \quad \Lambda, \quad q_{k-1}$ be $k-1$ bogus keywords with relative frequencies $f(q_1), \quad \Lambda \quad f(q_{k-1})$, respectively, used to construct the privacy-protected query of Expression (1). Assuming that the relative frequencies in the reference thesaurus represent well the relative frequencies of keywords in the natural or domain-specific language used in the database queries, the privacy of the target query can be measured as the entropy:

$$H(Q_0) = -\sum_{i=0}^{k-1} g(q_i) \log \quad g(q_i)$$

where Q_0 is a random variable representing the intruder's view of the target keyword and:

$$g(q_i) := \frac{f(q_i)}{\sum_{j=0}^{k-1} f(q_i)}$$

Proof. Assume that an intruder tries to find the target keyword that is masked in the privacy-preserving submitted query of Expression (1). The intruder views this target keyword as a random variable Q_0 with k possible values $q_{\pi(0)}, q_{\pi(1)}, \Lambda, q_{\pi(k-1)}$ with known probabilities $g\left(q_{\pi(i)}\right) \in [0,1]$ for $i = 0$ to $k-1$. Therefore, the uncertainty about the specific outcome of Q_0, that is, the privacy of q_0, coincides with the entropy of Q_0, that is:

$$H(Q_0) = \sum_{i=0}^{k-1} \left(q_{\pi(i)}\right) \log \quad g\left(q_{\pi(i)}\right) = -\sum_{i=0}^{k-1} g(q_i) \log \quad g(q_i)$$

Protocol 1 has at least two shortcomings:

It does not guarantee any lower bound for the privacy $H(Q_0)$, which in the worst case could be as low as 0. Therefore, a priori the protocol can only guarantee trivial 0-PIR (of course, after a particular set of bogus keywords have been chosen, the actual privacy offered is $H(Q_0)$).

If the same keyword q_0 is queried several times, it will be masked with different random bogus keywords each time, which will make it easy for an intruder to re-identify q_0. In a large thesaurus with n keywords, the probability of a keyword belonging to two different random samples of k keywords is negligible (see Lemma 2 in the Appendix), so a repeated keyword is most likely the target keyword q_0.

The user can circumvent the above shortcomings by running the following modified protocol.

Protocol 2 (Enhanced($\mathrm{q}_0$upwd, k, ε*))*

(1) Let *PRNG* be a cryptographically secure pseudo-random number generator. Seed *PRNG* with the hash of the concatenation of the target keyword q_0 and a user password *upwd*.

(2) Let the keywords in the thesaurus be ranked by increasing relative frequency.

(3) Let $f(q_0)$ be the relative frequency of the target keyword q_0.

(4) Adjust *PRNG* to uniformly generate real numbers in the interval $\left[\max\left(0, f(q_0) - \varepsilon\right), f(q_0)\right]$.

(5) Let $a :=$ *PRNG*.

(6) If $a + \varepsilon > 1$ or there are less than k keywords in the thesaurus with relative frequencies in the interval $[a, \quad a + \varepsilon]$, then signal failure and exit the protocol.

(7) Let $M_1 \geq k$ be the number of keywords in the thesaurus with relative frequencies in $[a, \quad a+\varepsilon]$. Let M_2 be the rank in the ranked thesaurus of the least frequent of the M_1 keywords. Adjust *PRNG* to generate pseudo-random numbers uniformly in the interval $[M_2, M_2 + M_1 - 1]$ (i.e. the interval containing the ranks of the M_1keywords).

(8) To mask the target keyword q_0, call the *PRNG* generator $K-1$ times to draw $K-1$ bogus keywords $q_1, \quad q_2, \quad \Lambda, \quad q_{k-1}$ by (pseudo-)randomly sampling without replacement from those keywords in the thesaurus with relative frequencies in $[a, \quad a+\varepsilon]$.

(9) Submit the query

$$q_{\pi(0)} \quad OR \quad q_{\pi(1)} \quad OR\Lambda \quad OR \quad q_{\pi(k-1)}$$

where π is a random permutation of the set $\{0, 1, \Lambda, k-1\}$.

(10) Once the query results are received, keep those results where the keyword q_0 appears and discard the rest.

The complexity of Protocol 2 is clearly linear in k: each bogus keyword must be generated (Step 8) and the results related to the target keyword must be filtered out of the overall results for the k submitted keywords (Step 10). Some explanations of the rationale of the Protocol follow:

- For a certain user with a user password *upwd* and a specific target keyword q_0 the bogus keywords are always the same in successive queries. This prevents an intruder from finding q_0 as the only repeated keyword in successive queries by the same user.
- The user password *upwd* is used to customise the bogus keywords generated for a given target keyword q_0. This has a privacy-preserving effect. Indeed, if the bogus keywords only depended on the target keyword, a user u_1 who saw a query submitted by another user u_2 whose k keywords are identical to those in a past query of his or her own would know that the target keyword in u_2's query is the same as in his or her own past query.
- Protocol 2 finds $k-1$ bogus keywords whose relative frequencies are similar to the relative frequency of q_0. This guarantees the lower bound for the privacy $H(Q_0)$ stated in Theorem 1 below.
- Frequencies for the bogus items are taken in an interval $[a, \quad a+\varepsilon]$ containing $f(q_0)$. Using an interval centred in $f(q_0)$ would be more intuitive, but certainly weaker – an intruder attempting to re-identify q_0 from the protected query could estimate q_0 as the median keyword among those in Expression (3).

The following theorem proves that nontrivial $h(k)$-privacy is guaranteed *a priori* by Protocol 2. For a certain target keyword q_0 to be protected, the lower privacy bound $h(k)$ can guide the data protector in choosing ε and the number $k-1$ of bogus keywords in order to achieve a certain privacy level.

Theorem 1
Let q_0 be the target keyword protected using Protocol 2 to generate the privacy-protected query in Expression (3). Then the privacy $H(Q_0)$ of q_0 can be lower-bounded as:

$$H(Q_0) \geq h(k)$$
$$= -\log_2\left(dB^2 + (1 - dB - (k - d - 1)A)^2 + (k - d - 1)A^2\right)$$

where:

$$A = \frac{a_{\min}}{ka_{\min} + (k-1)\varepsilon},$$

$$B = \frac{a_{\min+\varepsilon}}{ka_{\min} + \varepsilon}$$

with:

$$a_{\min} = \max(0, f(q_0) - \varepsilon)$$

and d is the greatest integer such that $(k - d)A + dB \leq 1$.

Proof. Lemma 1 above can be used to write:

$$H(Q_0) = -\sum_{i=0}^{k-1} g(q_i) \log \quad g(q_i)$$

Using Jensen's inequality on the right-hand side of Equation (4), we get:

$$H(Q_0) \geq \log\left(\frac{1}{\sum_{i=0}^{k-1} (g(q_i))^2}\right)$$

The lower bound in Equation (5) is minimum when:

$$\sum_{i=0}^{k-1} (g(q_i))^2$$

is maximum. Let $a \in \left[\max\left(0, f(q_0) - \varepsilon\right), f(q_0)\right]$ be the pseudo-random number computed at Step 5 of Protocol 2 such that $f(q_i) \in \left[a, a + \varepsilon\right] \subseteq \left[0, 1\right]$ for all $i = 0$ to $k - 1$. Then we have $g(q_i) \in \left[A, B\right] \subseteq \left[0, 1\right]$ where:

$$A := \frac{a}{ka + (k-1)\varepsilon}$$

and:

$$B := \frac{a+\varepsilon}{ka+\varepsilon}.$$

On the other hand:

$$kA = \frac{ka}{ka+(k-1)\varepsilon} < 1$$

$$kB = \frac{ka+k\varepsilon}{ka+\varepsilon} > 1$$

So Lemma 3 (see Appendix) can be used to find that the maximum of Expression (6) is reached for $g(q_i) = \Lambda = g(q_d) = B, g(q_{d+1}) = 1 - dB - (k-d-1)A$ and $g(q_{d+2}) = \Lambda = g(q_k) = A$ or any permutation of those assignments, where $d < k$ is the greatest integer such that $(k-d)A + dB \leq 1$. The corresponding maximum value of Expression (6) is:

$$dB^2 + (1 - dB - (k-d-1)A)^2 + (k-d-1)A^2$$

The value of Expression (7) varies depending on A and B. Actually, it is maximum when A is as small as possible and B is as large as possible, which for fixed ε and k happens when a is minimum, that is, for $a_{\min} = \max(0, f(q_0) - \varepsilon)$.

An extension for multi-keyword queries

Protocol 2 can easily be extended to handle multi-keyword queries by making the following adaptations:

- Replace the target keyword q_0 with a Boolean expression E_0 involving several keywords.
- Replace $f(q_0)$ with the relative frequency $f(E_0)$ of E_0, computed using the standard algebra employed to find the probability of a Boolean expression of events; in particular, if keywords can be assumed to be independent and E_0 is the logical AND of several keywords, $f(E_0)$ can be computed as the product of the relative frequencies of the keywords.
- Instead of bogus keywords $q_1, \quad q_2, \quad \Lambda, \quad q_{k-1}$, use bogus expressions $E_1, \quad E_2 \quad \Lambda, \quad E_{k-1}$ whose relative frequency lies in the interval $[a, a+\varepsilon]$.
- Submit the query:

$$E_{\pi(0)} OR \quad E_{\pi(1)} \quad OR \quad \Lambda \quad OR \quad E_{\pi(k-1)}$$

- Once the query results are received, keep those results where E_0 appears and discard the rest.

The *GooPIR* prototype and empirical results

A prototype called *GooPIR* has been developed in Java JDK 6.0 Standard Edition to implement the scheme described in Protocol 2. The prototype accepts queries consisting of single keywords or queries consisting of a logical AND of several keywords (with the limitation that independence between the keywords must be a

plausible assumption). GooPIR locally masks the target keyword(s), submits the masked query to the Google search engine and then locally filters the results relevant to the target keyword(s). A screenshot of the prototype can be seen in Figure 1. The figure presents the search results for the keyword "University".

By clicking on the "Details" button, the following information about the execution of Protocol 2 is displayed:

- The target query $q_0 =$ "University" and its relative frequency $f(q_0) = 0.00648$ in the reference thesaurus (see below for details of how the thesaurus was obtained);
- The width $\varepsilon = 0.001$ of the frequency interval the bogus keywords are drawn from;
- The interval $\left[\max\left(0, f(q_0) - \varepsilon\right), f(q_0)\right] = \left[0.00548, 0.00648\right]$, called "Alpha interval" in the prototype;
- The pseudo-random value $a = 0.00619$, called "Alpha" in the prototype;
- The search interval $\left[a, a + \varepsilon\right] = \left[0.00619, 0.00719\right]$;
- The number $M_1 = 34$ of candidate bogus keywords with frequencies within the search interval;
- The list of $k - 1 = 10$ bogus keywords used for masking, as well as their frequencies;
- The entropy of the masked entry, that is, $H(Q_0) = 3.4582$ bits (for the parameters used, the a priori lower bound of the entropy guaranteed by Theorem 1 is 3.42412 bits).

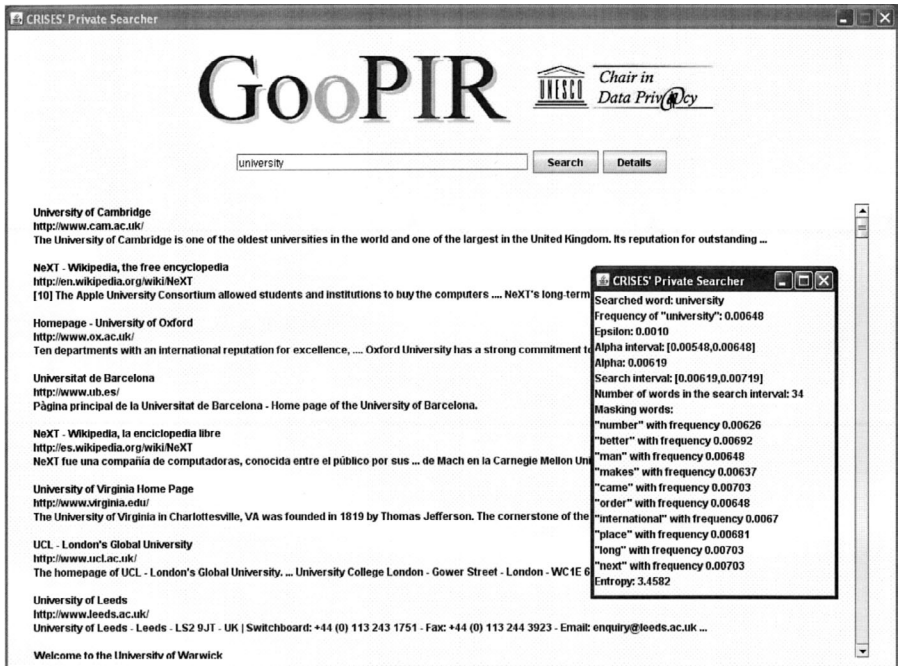

Figure 1.
The main page of the *GooPIR* prototype

GooPIR can work with any reference thesaurus (which may include proper nouns, non-linguistic expressions, etc.). The results reported in this section were obtained using a reference thesaurus constructed as follows:

- An initial thesaurus of 216,551 English words without relative frequencies was obtained from: www.ojohaven.com/fun/search.html
- Prepositions, articles and conjunctions were eliminated from the thesaurus.
- All news articles dated October 2007 available from the "Print Edition" section of the electronic newspaper WikiNews (http://en.wikinews.org/wiki/Wikinews:Print_edition) were converted from PDF to ASCII text.
- GooPIR assigned to each word in the thesaurus its relative frequency of appearance in the news articles described above. The resulting $n = 9096$ EMBED English words with non-zero frequency were taken as the reference thesaurus with frequencies.

Table I reports additional empirical results for keywords in several frequency ranges and for several values of ε and k. For each choice of frequency, ε and k, the following information is given: search interval $[a, a+\varepsilon]$, number M_1 of candidate bogus keywords with frequencies in the search interval, entropy of the masked query and a priori lower bound for the entropy according to Theorem 1. Note that one gets a trivial a priori lower bound equal to zero whenever $f(q_0) \leq \varepsilon$ (in this case $a_{\min} = 0$ and the lower bound is zero too). For conciseness, the bogus keywords used for masking have been omitted; however, their sequence is always the same for a certain target keyword in successive queries (regardless of k). For example, for $q_0 =$"theater" with the parameters in the table, we get $q_1 =$ "vocally" (with frequency $f(q_1) = 0.0001$), $q_2 =$ "murderers" (with frequency $f(q_2) = 0.0001$), $q_3 =$"dollar" (with frequency $f(q_3) = 0.00098$), $q_4 =$"biodiversity" (with frequency $f(q_4) = 0.00043$), etc.

Quality assessment of the results

Offering privacy to users by hiding the target keyword among other bogus keywords would not be attractive if the toll for privacy preservation was a substantial alteration of the query results. In order to assess quality, we have compared the query results obtained with and without our method.

The following notation is used in this section:

B: U: The set of URLs obtained with Google.

$\hat{U}$: The set of URLs obtained with GooPIR.

N_{URL}: The cardinality of U and $\hat{U}$ (the first N_{URL} URLs for both outputs are considered).

We propose three simple quality measures. The first measure is the percentage of coincidences between U and $\hat{U}$, that is:

$$QM_1 = 100 \times \frac{Card\left(U \cap \hat{U}\right)}{N_{URL}}$$

where $Card(\cdot)$ is used to denote set cardinality.

Frequation range	Keyw. frequation	Keyw.	ε	Search interval	#Keyw interval	k	Entropy	$h(k)$ *a priori* lower bound
[0,0.001]	0.0001	Theater	0.001	[0,0.001]	7616	2	1	0
						5	1.67967	0
						10	2.71341	0
			0.005	[0,0.005]	8756	2	1	0
						5	1.71143	0
						10	2.80877	0
[0.004, 0.005]	0.00406	Sort	0.001	[0.00341, 0.00441]	124	2	0.99889	0.97182
						5	2.31882	2.26197
						10	3.31871	3.24021
			0.005	[0.00085, 0.00585]	1522	2	0.91151	0
						5	2.2051	0
						10	3.05604	0
[0.008, 0.009]	0.00868	Edition	0.001	[0.00866, 0.00966]	25	2	0.99997	0.99462
						5	2.32181	2.31083
						10	3.32132	3.30899
			0.005	[0.0086, 0.0136]	73	2	0.98355	0.78135
						5	2.31389	1.82720
						10	3.31241	2.70861

Table I.
GooPIR empirical results for keywords in several frequency ranges and several values of ε and k

The second measure is the average distance between coincidences, that is:

$$QM_2 = \frac{\sum_{u \in U \cap \hat{U}} \left| r_U(u) - r_{\hat{U}}(u) \right|}{Card\left(U \cap \hat{U}\right)}$$

where $r_U(u)$ is the rank of u within U and $r_{\hat{U}}(u)$ is the rank of u within $\hat{U}$. The third measure is the variance of the rank differences between the results, which are both in U and $\hat{U}$, that is:

$$QM_3 = \frac{\sum_{u \in U \cap \hat{U}} \left(\left| r_U(u) - r_{\hat{U}}(u) \right| - QM_2 \right)^2}{Card\left(U \cap \hat{U}\right)}$$

The outcome of the above measures varies depending on the relative frequency of the searched keyword. To study the influence of the frequency on the quality of the results, we have defined the following cases:

- Low-frequency keywords, whose frequency lies in the first quartile.
- Medium-frequency keywords, whose frequency lies in the second or third quartiles.
- High-frequency keywords, whose frequency lies in the fourth quartile.

The following algorithm has been used to assess the quality of our results for different keyword frequencies (the number of queries was $NQ = 10$ and the number of query results was $N_{URL} = 25$).

Algorithm 1 (Assessing the quality of GooPIR results)

(1) INPUT: Number of queries (*NQ*), Frequency (*F*, Number of results N_{URL}
(2) $i \leftarrow 0$
(3) While ($i < NQ$):
   - $W \leftarrow SelectRandomKeyword(F)$
   - $\hat{U} \leftarrow$Send (W) to *GooPIR*
   - $U \leftarrow$ Send (W) to Google
   - $\hat{U} \leftarrow$GetTheFirstResults($\hat{U}, N_{URL}$)
   - $U \leftarrow$GetTheFirstResults(U, N_{URL})
   - $M_1 \leftarrow$Compute $QM_1\left(\hat{U}, U\right)$
   - $M_2 \leftarrow$Compute $QM_2\left(\hat{U}, U\right)$
   - $M_3 \leftarrow$Compute $QM_3\left(\hat{U}, U\right)$
   - $i := i + 1$
(4) Return set of measurements for QM_1, QM_2andQM_3

Tables II-IV show the outcome of QM_1, QM_2 and QM_3 for ten randomly selected keywords with respectively low, medium and high frequencies. It can be observed that

Keyword	k = 2			k = 6			k = 10		
	QM_1	QM_2	QM_3	QM_1	QM_2	QM_3	QM_1	QM_2	QM_3
Domain	96	1.63	5.55	84	4.81	11.66	80	4.70	12.96
Anonymity	28	1.29	3.57	28	1.29	4.90	28	1.57	6.95
Fuzzy	84	1.33	3.23	52	3.85	7.14	24	3.50	5.90
Swim	80	4.55	9.84	76	5.53	9.37	40	6.50	16.94
Sourcing	64	6.25	6.07	24	5.67	4.27	8	4.00	0.00
Prejudice	92	3.39	15.07	68	6.18	11.90	56	7.43	21.65
Newsroom	80	3.20	7.43	72	6.06	30.41	72	4.94	21.00
Controllers	76	2.21	2.84	48	2.92	1.72	32	2.63	1.98
Orders	40	3.90	7.88	32	4.13	13.27	20	2.60	10.30
Hammer	92	3.13	7.48	88	2.91	4.09	80	4.20	4.17
	73.2	3.09	6.90	57.2	4.33	9.87	44	4.21	10.19

Table II.
Quality analysis for some low-frequency keywords

Keyword	k = 2			k = 6			k = 10		
	QM_1	QM_2	QM_3	QM_1	QM_2	QM_3	QM_1	QM_2	QM_3
Highlighting	80	1.65	1.19	20	0.80	0.70	16	0.75	0.92
Militia	64	7.50	24.27	72	5.06	15.00	56	4.36	10.40
Expansion	96	1.42	5.21	60	3.80	22.17	56	3.07	19.61
Contributions	68	5.35	19.74	32	5.25	32.21	32	5.13	32.98
Chevy	60	3.27	14.21	40	6.70	39.57	40	7.10	40.32
Libido	64	1.38	0.78	32	2.00	3.43	24	1.17	2.17
Oliver	80	2.20	7.12	60	4.00	21.86	56	2.50	16.27
Statistics	88	2.73	1.06	88	2.14	4.89	88	4.27	14.68
Presidency	68	5.00	10.63	52	6.92	22.91	40	7.20	20.84
Researchers	28	5.00	32.00	16	0.25	0.25	16	0.25	0.25
	69.60	3.55	11.62	47.20	3.69	16.30	42.40	3.58	15.84

Table III.
Quality analysis for some medium-frequency keywords

Keyword	k = 2			k = 6			k = 10		
	QM_1	QM_2	QM_3	QM_1	QM_2	QM_3	QM_1	QM_2	QM_3
Conversation	60	3.60	4.97	52	3.92	4.08	24	3.50	7.90
Reporter	64	8.06	11.13	48	9.50	18.82	40	11.30	19.79
Prepared	32	2.25	3.93	24	2.67	5.87	12	1.33	2.33
Documentary	84	2.24	2.79	32	4.25	16.50	16	5.00	16.67
Calls	32	2.38	5.98	16	2.75	0.92	8	4.00	0.00
Choices	64	4.00	9.07	80	4.80	16.59	40	5.10	11.66
Bid	44	7.45	39.07	24	5.67	61.87	20	5.20	78.20
Structure	84	1.62	1.65	68	1.82	4.78	40	3.90	10.32
Finally	68	2.12	2.74	28	3.00	2.33	8	2.00	0.00
Opinions	80	1.05	0.79	76	3.11	9.88	76	3.11	9.77
	61.20	3.48	8.21	44.80	4.15	14.16	28.40	4.44	15.66

Table IV.
Quality analysis for some high-frequency keywords

the quality of the results decreases when the number $K-1$ of bogus keywords increases (see Figure 2). However, the average of QM_1 (percentage of coincidences) is higher than 40 per cent even when the number of bogus keywords is high (e.g. $K=10$ keywords). In addition, the average of QM_2 (rank distance) is always lower than QM_3. As could be expected, the average of QM_3 (rank variance) tends to grow with k. It is apparent from Figure 2 that the lower the frequency the better the quality.

The above measures show that, overall, the search results obtained with *GooPIR* are quite good. We next focus our analysis on the search results most relevant to the user, namely those results which appear first, that is, with lowest ranks. We have defined blocks of five results each:

(1) Block 1: Results 0 to 4
(2) Block 2: Results 5 to 9
(3) Block 3: Results 10 to 14
(4) Block 4: Results 15 to 19
(5) Block 5: Results 20 to 24

Tables V-X show some blockwise quality measurements of the *GooPIR* results for low, medium and high frequencies, respectively. We have observed that the average of QM_1 (average percentage of coincidences) in the first block is very high (i.e. between 65 per cent and 86 per cent) and the quality of the results tends to decrease in blocks with higher ranks. In addition, the influence of the frequency seems to be the same as in the non-blockwise analysis – better quality is obtained for lower frequencies.

Conclusions and future work

In this paper the concept of $h(k)$-private information retrieval or $h(k)$-PIR has been introduced as a pragmatic approach to offer users some query privacy in real databases and search engines. Then two protocols for keywords (a naïve one and an

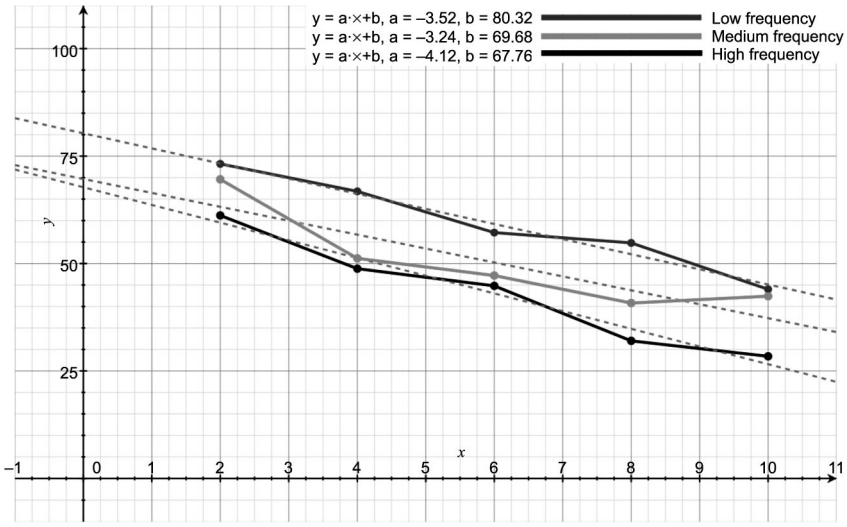

Figure 2.
Average percentage of coincidences for different values of k and different frequencies

Keyword	QM_1	QM_2	QM_3
$k = 2$, Block 1			
Domain	100	0.00	0.00
Anonymity	80	0.00	0.00
Fuzzy	80	1.50	3.00
Swim	100	1.80	3.70
Sourcing	40	5.00	32.00
Prejudice	80	5.25	30.92
Newsroom	100	2.80	6.20
Controllers	40	3.50	12.50
Orders	60	1.33	0.33
Hammer	80	1.50	1.00
	76.00	2.27	8.96
$k = 2$, Block 2			
Domain	80	3.00	0.00
Anonymity	40	2.00	0.00
Fuzzy	60	1.67	0.33
Swim	60	7.33	46.33
Sourcing	80	3.50	0.33
Prejudice	100	1.40	0.80
Newsroom	100	1.20	0.20
Controllers	80	1.00	1.33
Orders	20	2.00	–
Hammer	100	6.40	21.80
	72.00	2.95	7.90
$k = 2$, Block 3			
Domain	80	6.00	0.00
Anonymity	20	6.00	–
Fuzzy	60	5.33	0.33
Swim	100	6.00	1.50
Sourcing	40	7.00	2.00
Prejudice	40	7.00	2.00
Newsroom	80	7.00	22.67
Controllers	80	3.25	1.58
Orders	40	5.00	8.00
Hammer	80	2.50	0.33
	62.00	5.51	4.27
$k = 2$, Block 4			
Domain	100	3.60	18.80
Anonymity	0	–	0.00
Fuzzy	100	0.40	0.30
Swim	60	4.33	2.33
Sourcing	80	7.75	0.92
Prejudice	100	6.20	34.20
Newsroom	80	5.75	13.58
Controllers	80	1.75	0.25
Orders	60	6.00	12.00
Hammer	100	3.00	0.00
	76.00	4.31	8.24
$k = 2$, Block 5			
Domain	100	0.40	0.30
Anonymity	0	–	0.00
Fuzzy	80	0.75	0.25
Swim	80	5.75	0.25
Sourcing	80	8.00	0.00
Prejudice	100	2.00	0.50
Newsroom	60	5.00	0.00
Controllers	100	3.80	0.70
Orders	20	4.00	–
Hammer	80	2.25	0.25
	70.00	3.55	0.25

Table V.
Quality analysis for $k = 2$ of the URLs returned by *GooPIR* for low-frequency keywords

Keyword	QM_1	QM_2	QM_3
$k = 6$, Block 1			
Domain	80	1.00	0.67
Anonymity	80	0.00	0.00
Fuzzy	80	0.75	0.92
Swim	100	2.40	9.80
Sourcing	0	–	0.00
Prejudice	80	2.50	4.33
Newsroom	100	2.00	8.00
Controllers	20	0.00	–
Orders	60	1.00	1.00
Hammer	80	0.75	0.92
	68.00	1.16	2.85
$k = 6$, Block 2			
Domain	60	4.67	40.33
Anonymity	40	1.50	0.50
Fuzzy	40	2.50	0.50
Swim	20	12.00	–
Sourcing	60	4.00	1.00
Prejudice	60	3.33	0.33
Newsroom	40	4.50	0.50
Controllers	80	2.50	0.33
Orders	20	3.00	–
Hammer	100	4.00	12.50
	52.00	4.20	7.00
$k = 6$, Block 3			
Domain	80	6.00	0.00
Anonymity	20	6.00	–
Fuzzy	60	5.33	0.33
Swim	100	6.00	1.50
Sourcing	40	7.00	2.00
Prejudice	40	7.00	2.00
Newsroom	80	7.00	22.67
Controllers	80	3.25	1.58
Orders	40	5.00	8.00
Hammer	80	2.50	0.33
	62.00	5.51	4.27
$k = 6$, Block 4			
Domain	100	6.60	17.30
Anonymity	0	–	0.00
Fuzzy	80	6.50	1.67
Swim	60	5.33	9.33
Sourcing	20	8.00	–
Prejudice	60	10.33	10.33
Newsroom	60	8.33	54.33
Controllers	60	4.00	0.00
Orders	20	6.00	–
Hammer	100	3.80	0.20
	56.00	6.54	11.65
$k = 6$, Block 5			
Domain	100	5.20	0.20
Anonymity	0	–	0.00
Fuzzy	0	–	0.00
Swim	100	7.00	0.50
Sourcing	0	–	0.00
Prejudice	100	8.00	1.50
Newsroom	80	9.25	53.58
Controllers	0	–	0.00
Orders	20	11.00	–
Hammer	80	3.00	0.67
	48.00	7.24	5.64

Table VI.
Quality analysis for $k = 6$ of the URLs returned by *GooPIR* for low-frequency keywords

Keyword	QM_1	QM_2	QM_3
$k = 2$, Block 1			
Highlighting	100	0.80	1.70
Militia	60	1.00	3.00
Expansion	100	1.40	3.80
Contributions	80	1.75	0.25
Chevy	100	1.60	0.80
Libido	100	0.80	0.70
Oliver	100	0.00	0.00
Statistics	60	1.00	0.00
Presidency	80	1.75	1.58
Researchers	80	1.25	1.58
	86.00	1.14	1.34
$k = 2$, Block 2			
Highlighting	40	1.50	0.50
Militia	40	11.50	84.50
Expansion	100	4.80	5.70
Contributions	80	5.25	11.58
Chevy	60	0.67	1.33
Libido	100	1.80	1.20
Oliver	100	1.00	0.00
Statistics	80	3.00	1.33
Presidency	20	2.00	–
Researchers	0	–	0.00
	62.00	3.50	11.79
$k = 2$, Block 3			
Highlighting	80	1.50	0.33
Militia	40	8.00	0.00
Expansion	100	0.40	0.80
Contributions	40	13.50	0.50
Chevy	60	2.00	0.00
Libido	20	2.00	–
Oliver	100	3.00	11.50
Statistics	100	2.80	0.20
Presidency	80	7.50	20.33
Researchers	20	4.00	–
	64.00	4.47	4.21
$k = 2$, Block 4			
Highlighting	100	2.00	1.50
Militia	100	9.80	8.70
Expansion	100	0.00	0.00
Contributions	100	6.80	11.70
Chevy	20	2.00	–
Libido	60	1.33	0.33
Oliver	40	2.50	0.50
Statistics	100	3.20	1.70
Presidency	80	6.50	3.67
Researchers	20	14.00	–
	72.00	4.81	3.51
$k = 2$, Block 5			
Highlighting	4	2.50	0.33
Militia	80	7.25	17.58
Expansion	80	0.25	0.25
Contributions	40	1.00	0.00
Chevy	60	10.33	1.33
Libido	40	1.50	0.50
Oliver	60	6.33	1.33
Statistics	100	3.00	0.00
Presidency	80	5.00	2.67
Researchers	20	12.00	–
	56.40	4.92	2.67

Table VII.
Quality analysis for $k = 2$ of the URLs returned by *GooPIR* for medium-frequency keywords

Keyword	QM_1	QM_2	QM_3
$k = 6$, Block 1			
Highlighting	100	0.80	0.70
Militia	60	0.00	0.00
Expansion	100	3.80	63.20
Contributions	80	0.75	0.25
Chevy	100	2.80	6.70
Libido	80	1.25	1.58
Oliver	80	0.50	0.33
Statistics	60	0.67	0.33
Presidency	40	3.50	0.50
Researchers	80	0.25	0.25
	78.00	1.43	7.38
$k = 6$, Block 2			
Highlighting	0	–	0.00
Militia	80	5.25	8.92
Expansion	80	3.00	16.00
Contributions	20	3.00	–
Chevy	40	3.50	4.50
Libido	80	2.75	4.92
Oliver	60	1.00	0.00
Statistics	80	2.00	0.00
Presidency	20	2.00	–
Researchers	0	–	0.00
	46.00	2.81	4.29
$k = 6$, Block 3			
Highlighting	0	–	0.00
Militia	80	5.00	2.00
Expansion	60	4.67	1.33
Contributions	40	12.50	0.50
Chevy	0	–	0.00
Libido	0	–	0.00
Oliver	60	4.00	12.00
Statistics	100	2.00	0.00
Presidency	80	9.00	22.00
Researchers	0	–	0.00
	42.00	6.19	3.78
$k = 6$, Block 4			
Highlighting	0	–	0.00
Militia	100	7.80	26.70
Expansion	40	3.50	0.50
Contributions	20	11.00	–
Chevy	0	–	0.00
Libido	0	–	0.00
Oliver	40	5.50	0.50
Statistics	100	2.00	4.00
Presidency	80	8.25	34.25
Researchers	0	–	0.00
	38.00	6.34	7.33
$k = 6$, Block 5			
Highlighting	0	–	0.00
Militia	40	5.50	0.50
Expansion	20	5.00	–
Contributions	0	–	0.00
Chevy	60	15.33	2.33
Libido	0	–	0.00
Oliver	60	10.67	33.33
Statistics	100	3.40	17.80
Presidency	40	6.00	32.00
Researchers	0	–	0.00
	32.00	7.65	9.55

Table VIII.
Quality analysis for $k = 6$ of the URLs returned by *GooPIR* for medium-frequency keywords

Keyword	QM_1	QM_2	QM_3
$k = 2$, Block 1			
Conversation	60	0.33	0.33
Reporter	40	0.50	0.50
Prepared	60	0.67	0.33
Documentary	100	0.40	0.30
Calls	100	2.60	9.80
Choices	80	1.50	3.67
Bid	100	2.80	4.70
Structure	80	1.00	0.00
Finally	100	0.80	1.70
Opinions	100	0.60	0.80
	82.00	1.12	2.21
$k = 2$, Block 2			
Conversation	60	2.33	2.33
Reporter	60	6.67	5.33
Prepared	60	2.33	4.33
Documentary	100	2.20	4.70
Calls	40	2.50	0.50
Choices	80	2.25	0.92
Bid	20	6.00	–
Structure	100	1.00	0.00
Finally	60	1.67	4.33
Opinions	80	1.00	0.00
	66.00	3.11	2.49
$k = 2$, Block 3			
Conversation	60	4.00	0.00
Reporter	20	10.00	–
Prepared	40	4.50	0.50
Documentary	60	1.67	0.33
Calls	20	1.00	–
Choices	60	6.67	16.33
Bid	0	–	0.00
Structure	100	2.60	5.80
Finally	80	3.00	0.00
Opinions	80	0.50	0.33
	52.00	3.77	2.91
$k = 2$, Block 4			
Conversation	80	5.50	0.33
Reporter	100	10.00	0.00
Prepared	0	–	0.00
Documentary	60	3.00	0.00
Calls	0	–	0.00
Choices	60	4.67	0.33
Bid	20	11.00	–
Structure	80	1.50	0.33
Finally	100	3.00	3.00
Opinions	80	2.25	0.25
	58.00	5.11	0.47
$k = 2$, Block 5			
Conversation	40	6.00	0.00
Reporter	100	9.60	0.30
Prepared	0	–	0.00
Documentary	100	4.00	0.00
Calls	0	–	0.00
Choices	40	7.50	4.50
Bid	80	12.75	45.58
Structure	60	2.00	0.00
Finally	0	–	0.00
Opinions	60	1.00	1.00
	48.00	6.12	5.14

Table IX.
Quality analysis for $k = 2$ of the URLs returned by *GooPIR* for high-frequency keywords

Keyword	QM_1	QM_2	QM_3
$k = 6$, Block 1			
Conversation	60	1.33	0.33
Reporter	20	0.00	–
Prepared	40	0.50	0.50
Documentary	60	0.67	0.33
Calls	40	2.50	0.50
Choices	80	5.00	87.33
Bid	80	1.50	1.00
Structure	80	0.50	0.33
Finally	40	3.00	8.00
Opinions	100	3.40	35.80
	60.00	1.84	14.90
$k = 6$, Block 2			
Conversation	40	3.00	8.00
Reporter	60	5.67	4.33
Prepared	40	2.00	2.00
Documentary	40	3.50	4.50
Calls	20	2.00	–
Choices	80	2.75	0.92
Bid	0	–	0.00
Structure	40	0.00	0.00
Finally	40	4.00	0.00
Opinions	80	1.50	0.33
	44.00	2.71	2.23
$k = 6$, Block 3			
Conversation	60	4.33	0.33
Reporter	20	11.00	–
Prepared	40	5.50	0.50
Documentary	40	6.50	0.50
Calls	20	4.00	–
Choices	60	5.00	3.00
Bid	20	7.00	–
Structure	60	4.00	1.00
Finally	60	2.33	1.33
Opinions	60	2.67	1.33
	44.00	5.23	1.14
$k = 6$, Block 4			
Conversation	80	5.50	1.00
Reporter	80	11.50	0.33
Prepared	0	–	0.00
Documentary	20	12.00	–
Calls	0	–	0.00
Choices	100	5.40	2.30
Bid	0	–	0.00
Structure	80	3.75	7.58
Finally	0	–	0.00
Opinions	60	5.00	1.00
	42.00	7.19	1.36
$k = 6$, Block 5			
Conversation	20	6.00	–
Reporter	60	13.33	0.33
Prepared	0	–	0.00
Documentary	0	–	0.00
Calls	0	–	0.00
Choices	80	5.75	4.25
Bid	20	21.00	–
Structure	80	0.50	0.33
Finally	0	–	0.00
Opinions	80	3.25	2.25
	34.00	8.31	0.90

Table X.
Quality analysis for $k = 6$ of the URLs returned by *GooPIR* for high-frequency keywords

enhanced one) have been presented that do not assume any cooperation from the database in protecting the privacy of user queries; the second one offers nontrivial $h(k)$-PIR and can be extended to handle multi-keyword queries. The proposed protocols rely on a public thesaurus of keywords labelled with their relative frequencies. We have also presented empirical results obtained with a prototype implementing the enhanced protocol to provide keyword $h(k)$-PIR when querying internet search engines.

However simple, the presented approach to obtain some user privacy when querying a privacy-uncooperative database server or search engine seems to be new in the literature.

Future work may include:

- Adding semantic diversity requirements to the frequency requirements when choosing the bogus keywords used to mask a given target keyword. A possible measure of semantic diversity is the hierarchical nominal variance defined in Domingo-Ferrer and Solanas (2008).
- Pre-processing the thesaurus to create clusters of keywords with similar frequency, so that selecting the bogus keywords is faster. Given a target keyword, a entire cluster of thesaurus keywords is taken as bogus keywords. Clusters must consist of $k-1$ (or more) keywords, and can be computed by using microaggregation (Domingo-Ferrer *et al.*, 2008).

Note

1. Given a discrete random variable X and probability function p, the entropy $H(X)$ of X is defined as $H(X) = -\sum_{i=1}^{n} p(x_i) log_2 p(x_i)$, where $x_1, \cdots, x_n$ are the outcomes of X having non-zero probability (see Shannon, 1948). Shannon's entropy measures the uncertainty about the specific outcome that will be obtained when the random variable X is sampled.

References

AOL (2006), "AOL keyword searches" (online), available at: http://dontdelete.com/default.asp (accessed 2 December 2008).

Beimel, A., Ishai, Y. and Malkin, T. (2004), "Reducing the servers computation in private information retrieval: PIR with preprocessing", *Journal of Cryptology*, Vol. 17, pp. 125-51.

Chor, B., Gilboa, N. and Naor, M. (1997), *Technical Report TR CS0917*, Private information retrieval by keywords, Department of Computer Science, Technion, Israel Institute of Technology, Haifa.

Chor, B., Goldreich, O., Kushilevitz, E. and Sudan, M. (1995), "Private information retrieval", in *Proccedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS), 23-25 October, IEEE*, pp. 41-50.

Chor, B., Goldreich, O., Kushilevitz, E. and Sudan, M. (1998), "Private information retrieval", *Journal of the ACM*, Vol. 45, pp. 965-81.

Domingo-Ferrer, J. (2007), "A three-dimensional conceptual framework for database privacy, in Secure Data Management – 4th VLDB Workshop (SDM 2007)", *Lecture Notes in Computer Science*, Vol. 4721, Springer, Berlin and Heidelberg, pp. 193-202.

Domingo-Ferrer, J. and Solanas, A. (2008), "A measure of variance for nominal hierarchical attributes", *Information Sciences*, Vol. 178, pp. 4644-55.

Domingo-Ferrer, J., Sebé, F. and Solanas, A. (2008), "A polynomial-time approximation to optimal multivariate microaggregation", *Computers and Mathematics with Applications*, Vol. 55, pp. 714-32.

Goldwasser, S. and Micali, S. (1984), "Probabilistic encryption", *Journal of Computer and Systems Science*, Vol. 28 No. 1, pp. 270-99.

Howe, D.C. and Nissenbaum, H. (2009), "TrackMeNot: resisting surveillance in web search", in Kerr, I., Steeves, V. and Lucock, C. (Eds), *Lessons from the Identity Trail: Privacy, Anonymity and Identity in a Networked Society*, Oxford University Press, Oxford.

Kushilevitz, E. and 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 (FOCS), 20-22 October, IEEE*, pp. 364-373.

Lane, J., Heus, P. and Mulcahy, T. (2008), "Data access in a cyber world: making use of cyberinfrastructure", *Transactions on Data Privacy*, Vol. 1 No. 1, pp. 2-16.

Leech, G., Rayson, P. and Wilson, A. (2001), *Word Frequencies in Written and Spoken English: Based on the British National Corpus*, Longman, London.

Ostrovsky, R. and Skeith, W.E. III (2007), "A survey of single-database PIR: techniques and applications, in public key cryptography – PKC 2007", *Lecture Notes in Computer Science*, Vol. 4450, Springer, Berlin and Heidelberg, pp. 393-411.

Samarati, P. and Sweeney, L. (1998), Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression, Technical report, SRI International, Menlo Park, CA.

Shannon, C.E. (1948), "A mathematical theory of communication", *Bell Systems Technical Journal*, Vol. 14 No. 1, pp. 7-13.

Staddon, J., Golle, P. and Zimny, B. (2007), "Web-based inference detection", in *Proceedings of the 16th USENIX Security Symposium, The Advanced Computer Systems Association*, pp. 71-86.

Appendix

Lemma 2

Given a thesaurus with N keywords, the probability that the same keyword belongs to two independent random samples of k keywords drawn from the thesaurus is negligible if $N >> k$.

Proof

Let us compute the probability that there is an empty intersection between two independent random samples of size k drawn from the thesaurus. This can be computed as the number of favourable cases divided by the number of possible cases – the number of favourable cases is the number of different second samples that can be drawn from the thesaurus once the k keywords in the first sample have been removed from it, and the number of possible cases is the number of possible second samples. Therefore:

$$P(nointersection) = \frac{\binom{N-k}{k}}{\binom{N}{k}} = \frac{(N-k)\cdots(N-2k+1)}{N(N-1)\cdots(N-k+1)}$$

Clearly, Expression (8) tends to 1 as N grows, which proves the Lemma. χ

Lemma 3
Let $[A,B] \subseteq [0,1]$ and k be such that $kA < 1$ and $kB > 1$. Let $p_i \in [A,B]$ for $i = 1,\cdots,k$ be variables such that $\sum_{i=1}^{k} p_i = 1$. Then the maximum of:

$$p_1^2 + \cdots + p_k^2$$

is reached for $p_1 = \Lambda = p_d = B, p_{d+1} = 1 - dB - (k-d-1)A$ and $p_{d+2} = \Lambda = p_k = A$ or any permutation of those assignments, where $d < k$ is the greatest integer such that $(k-d)A + dB \leq 1$.

Proof
We proceed by construction. A necessary condition to maximise Expression (9) is to have as many p_i's as possible set at their maximum value B. At most, d variables p_i can be set to B, where d is the greatest integer such that $(k-d)A + dB \leq 1$. (Note that, by construction, $(k-1)A + B \leq 1$, so $d \geq 1$).

Without loss of generality, let $p_1 = p_2 = \Lambda = p_d = B$. Now, there is a "mass" $1 - dB$ with $(k-d)A \leq 1 - dB$ left to be distributed among p_{d+1}, Λ, p_k, with the constraint that $p_j \geq A$ for $j = d+1, \Lambda, k$. The following cases appear:

- $_{k=d+1}$. In this case the maximum is $p_1 = \Lambda = p_d = B$ and $p_k = 1 - dB$. (Note that $p_k = 1 - dB$ is within range. Indeed, for $k = d+1$, by construction d is such that $A + dB \leq 1 < (d+1)B$, so $A \leq 1 - dB < B$).
- $k = d+2$. In this case, the maximum is reached for $p_1 = \Lambda = p_d = B$ and we have to decide about the values for p_{d+1} and p_{d+2}. Both probabilities should add $1 - dB$. So maximising their sum of squares can be regarded as maximising the following univariate function

  $$F(p_{d+1}^2) = p_{d+1}^2 + (1 - dB - p_{d+1})^2 with p_{d+1} \in [A, 1 - dB - A]$$

  continue with list (Including maths) The above concave function is maximised at either extreme value of p_{d+1}^2. Without loss of generality, let us take $p_{d+1} = 1 - dB - A$ and $p_{d+2} = A$. (Note that $p_{d+1} = 1 - dB - A$ is within range. For $k = d+2$, by construction $2A + dB \leq 1 < (d+1)B$, so $A \leq 1 - dB - A < B - A$.)
- $k > d+1$. The maximum is reached for $p_1 = \Lambda = p_d = B$ we have to decide about the values for $p_{d+1}, p_{d+2}, \Lambda, p_k$. To maximise $p_{d+1}^2 + \Lambda + p_k^2$, we have to concentrate as much mass as possible in any single probability and set the rest to their lowest possible value A. Without loss of generality, $p_{d+1} = 1 - dB - (k-d-1)A$ and $p_{d+2} = \Lambda = p_k = A$. (Note p_{d+1} is within range. By construction $(k-d)A + dB \leq 1 < (d+1)B$, so $A \leq 1 - dB - (k-d-1)A < B - (k-d-1)A$.)

About the authors

Josep Domingo-Ferrer is Professor of Computer Science at Rovira i Virgili University of Tarragona, Spain, where he holds the UNESCO Chair in Data Privacy. He received with honors his MSc and PhD degrees in Computer Science from the Autonomous University of Barcelona in 1988 and 1991 (Outstanding Graduation Award). He also holds a MSc in Mathematics. His fields of activity are data privacy, data security and cryptographic protocols. In 2003, he was a co-recipient of a research prize from the Association of Telecom Engineers of Catalonia. In 2004, he got the TOYPS'2004 Award from the Junior Chambers of Catalonia. In 2009, he won the ICREA Academia Research Prize, awarded by the Government of Catalonia to the 40 leading Catalan faculty members in all areas of research. He has authored three patents and over 210 publications, one of which became an ISI highly-cited paper in early 2005. He has been the

co-ordinator of EU FP5 project CO-ORTHOGONAL and of several Spanish-funded and US-funded research projects. He currently coordinates the CONSOLIDER "ARES" team on security and privacy, one of Spain's 34 strongest research teams. He has chaired or co-chaired nine international conferences and has served on the programme committee of over 65 conferences on privacy and security. He is a co-Editor-in-Chief of *Transactions on Data Privacy* and he is an Associate Editor of three international journals. In 2004, he was a Visiting Fellow at Princeton University.

Agusti Solanas is a tenure-track lecturer at the Rovira i Virgili University of Tarragona (URV), Spain. He received his BSc and MSc degrees in Computer Engineering from URV in 2002 and 2004, respectively, the latter with honours (Outstanding Graduation Award). He received a PhD in Telematics Engineering from the Technical University of Catalonia in 2007 with honours. His fields of activity are data privacy, data security, clustering, neural networks and evolutionary computation. He is a participant in the CONSOLIDER-INGENIO 2007 project. He has participated in several Spanish-funded and Catalan-funded research projects. He has authored over 40 publications and he has delivered several talks. He has served as Chair, programme committee member and reviewer for several conferences and journals. Agusti Solanas is the corresponding author and can be contacted at: agusti.solanas@urv.cat

Jordi Castellà-Roca is a tenure-track lecturer at Rovira i Virgili University, Spain. 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 his PhD 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 30 works in international journals, book chapters, and international and national congresses. He is on the advisory board of an international magazine, and he has been a member of the scientific and organising committees of several international congresses. He has participated in Spanish-funded and Catalan-funded research projects. He is the author of six patents, five of them international and in operation. He is a founding partner of three technology companies.