Co-Utile P2P Anonymous Keyword Search

Josep Domingo-Ferrer, Sergio Martínez, David Sánchez, and Jordi Soria-Comas

Abstract Web search engines have become an essential tool to find online resources. However, their use can reveal very detailed information about the interests of their users. In this chapter, we detail co-utile protocols for exchanging queries between users in an anonymous way. Specifically, by exchanging queries with other users, the query profile associated to each user is masked and approximates the average group interests. However, the query exchange protocol has a cost. To thwart free-riders (who take advantage of the protocol to mask their profile but are reluctant to help others) we employ a co-utile reputation management system. The theoretical analysis is complemented with empirical results obtained from an implementation in a simulated multi-agent environment, which illustrates how co-utility can make cooperation self-enforcing and improve the agents' welfare.

Josep Domingo-Ferrer
Universitat Rovira i Virgili, Dept. of Computer Science and Mathematics, UNESCO Chair in Data Privacy, Av. Països Catalans 26, E-43007 Tarragona, Catalonia, e-mail: josep.domingo@urv.cat

Sergio Martínez
Universitat Rovira i Virgili, Dept. of Computer Science and Mathematics, UNESCO Chair in Data Privacy, Av. Països Catalans 26, E-43007 Tarragona, Catalonia, e-mail: sergio.martinezl@urv.cat

David Sánchez
Universitat Rovira i Virgili, Dept. of Computer Science and Mathematics, UNESCO Chair in Data Privacy, Av. Països Catalans 26, E-43007 Tarragona, Catalonia, e-mail: david.sanchez@urv.cat

Jordi Soria-Comas
Universitat Rovira i Virgili, Dept. of Computer Science and Mathematics, UNESCO Chair in Data Privacy, Av. Països Catalans 26, E-43007 Tarragona, Catalonia, e-mail: jordi.soria@urv.cat

1 Introduction

Web search engines (WSEs) have become an essential tool to find online resources. However, their use can reveal very detailed information about the interests (profiles) of their users. The users' defense consists in making their profile diverse enough so that the WSE cannot determine their interests. In essence, to mask their profiles, the users must hide the real queries they are interested in among a set of queries about diverse topics they are not interested in.

The users can perform the hiding on their own, by generating random fake queries and submitting them along with their real queries [3, 9, 12, 15]. Such a standalone approach has several downsides: on the one hand, fake queries overload the WSE, so, if every agent followed this approach, the WSE performance would strongly degrade; on the other hand, it is not so easy to produce fake queries that are plausibly indistinguishable from real ones [10].

Relying on a set of peer agents (as proposed in [11, 2]) is an alternative that is free from the previous downsides. Consider several agents with similar privacy interests. To mask her profile, an agent can ask another agent to submit her query to the WSE rather than submitting it herself. Also, the agent can submit to the WSE real queries originated by other agents. In this way, a rational cooperation between peer agents emerges [4, 5] and all of them manage to blur their profile of interests without having to submit fake queries.

In this chapter, we analyze this cooperation under the co-utility framework [6, 7] (see also the first chapter of this book). We propose co-utile protocols for exchanging queries between users in an anonymous way. Specifically, by exchanging queries with other users, the query profile associated to each user is masked and approximates the average group interests. However, the query exchange protocol has a cost for the agents submitting the queries to the search engine. To thwart free-riders (who take advantage of the protocol to mask their profile but are reluctant to help others), we employ a co-utile reputation management system [8] (also described in the second chapter of this book), so that both the query submission protocol and the reputation management mechanism are self-enforcing and mutually beneficial for the involved peers.

The theoretical details of the protocols are complemented with empirical results obtained from an implementation in a simulated multi-agent environment, which illustrates how co-utility can make cooperation self-enforcing and improve the agents' welfare in this specific scenario.

2 Single-hop query submission game with two agents

Within the anonymous keyword search scenario, the primary utility agents want is functionality: they want to get their queries answered. A secondary utility is privacy: they may also wish their interest profiles to stay private w.r.t. the WSE. Note that any WSE can be expected to try to profile its users (typically for marketing purposes) according to the topics mentioned in the queries they perform. We will call atomic utilities the two components of an agent's utility, namely functionality and privacy. These two atomic utilities are not easily reconciled, because an agent alone cannot optimize both of them simultaneously.

The first scenario/game we analyze, which we call *single-hop query submission*, is restricted to two agents: the initiator of the query (I) and the responder to a query submission request (R). The initiator I is interested in functionality (having her query answered) and privacy w.r.t. the web search engine (keeping her interests private). If the new query diversifies her interest profile (for example, because she had never asked a similar query), I is likely to submit it herself to the WSE; otherwise, I will ask R to submit the query on her behalf. The responder R is only concerned about his privacy (whether I's query is useful to blur R's profile w.r.t. the WSE).

The interaction among I, R and the WSE is as follows:

- I chooses between submitting the query to the WSE herself or forwarding the query to R for submission.
- Upon query reception, R chooses between submitting the query to the WSE (and returning the results to I) or declining to submit it.

These interactions are illustrated in Figure 1. Since (S) and (F,D) do not have any effect on the privacy or the utility of the responder, we assume that the responder's payoff in these protocols (u^R_S and u^R_{FD}, respectively) is 0. It is also reasonable to assume that $u^I_{FS} > u^I_{FD}$, as both (F,S) and (F,D) are equivalent in the initiator's eyes regarding privacy w.r.t. the WSE, but (F,S) is better in terms of functionality.

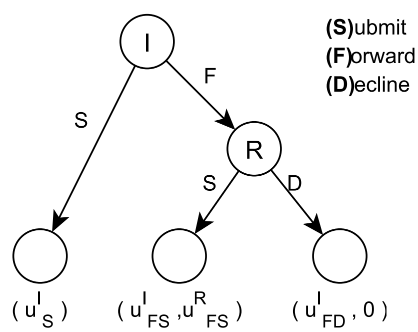

Fig. 1 Tree showing the possible interactions in the single-hop anonymous query submission game between two agents: initiator (I) and responder (R). To keep the figure compact, we only represent the payoffs of the agents active in each interaction: for interaction (S), the payoff of any agent other than the initiator is 0; for the other interactions, only the payoffs of the initiator and the active responder are represented, the rest of agents' payoffs being 0.

Given the above assumptions about the payoffs of the agents under different protocols, we can determine the possible co-utile protocols in the game tree shown in

Figure 1. Since the payoff of the responder when he does not participate in the protocol (that is, when the initiator submits the query to the WSE herself) is 0, (F,D) is not a co-utile protocol because the responder's payoff is the same as in (S). The only possible co-utile protocols are:

- (F,S), which is co-utile if the responder's payoff is strictly greater than in (S), that is $u^R_{FS} > u^R_S = 0$, and (F,S) is self-enforcing, which occurs if $u^R_{FS} > u^R_{FD}$ (the responder improves her privacy by submitting the query, which happens only when the query diversifies R's query profile) and $u^I_{FS} > u^I_S$ (directly submitting the query decreases the initiator's privacy). Notice that, under the previous conditions, (F,S) is also Pareto-optimal.
- (S), which is co-utile (actually just utile) when any of the following conditions is satisfied: $u^I_S > \max\{u^I_{FS}, u^I_{FD}\} = u^I_{FS}$ (by directly submitting her query, the initiator improves her privacy); or $u^I_S = u^I_{FS}$ and $u^R_{FS} = 0$ (the initiator does not decrease her privacy by directly submitting her query and the responder does not increase his privacy by submitting the initiator's query).

The following proposition summarizes the previous discussion.

Proposition 4.1. *Protocol (F,S) is co-utile when $u^R_{FS} > 0$ and $u^R_{FS} > u^R_{FD}$ and $u^I_{FS} > u^I_S$. In fact, under these conditions (F,S) is strictly co-utile.*

While we have been able to obtain Proposition 4.1 without specifying the form of the agents' utility functions, we need to do so in order to compute the agents' payoffs.

The initiator is interested in getting her query answered by the WSE and keeping her interests private w.r.t. the WSE. Thus, the initiator's utility under a protocol P can be expressed as $u_I(P) = u_I(f_I(P), p_I(P))$, where $p_I(P)$ quantifies the effect of the query submission on the initiator's privacy and $f_I(P)$ is an indicator reflecting whether protocol P gives the expected functionality to the initiator:

$$f_I(P) = \begin{cases} 0 & \text{agent} I \text{does not get the expected functionality;} \\ 1 & \text{otherwise.} \end{cases}$$

As to the responder, his only interest is his privacy w.r.t. the WSE; therefore, his utility can be written as $u_R(P) = u_R(p_R(P))$, where $p_R(P)$ quantifies the effect of the query submission on the responder's privacy.

To measure the privacy of an agent w.r.t. the WSE, we look at the agent's query profile, Y, which is the distribution of topics of the queries submitted so far by the agent. The query profile characterizes the exposure level (that is, privacy loss) of the agent's interests towards the WSE: if an agent's profile just contains one or a few dominant topics, her/his preferences are obvious; on the other hand, if it contains a variety of topics all appearing with the same frequency, the agent's preferences remain uncertain. We measure the agent's privacy level by computing the Shannon's entropy (that is, uncertainty) of her/his profile. Thus, the effect of a query q can be measured as the variation of the profile entropy caused by q:

$$p_I(P) = H(Y_I \cup \{q\}) - H(Y_I);$$

$$p_R(P) = H(Y_R \cup \{q\}) - H(Y_R).$$

Even if the Shannon's entropy of an empty profile is not defined mathematically, by convention we set it to $\log_2 \tau$, which is the maximum value of the entropy of a distribution over τ possible topics; the reason is that an empty profile discloses no information about the interests of the agent. For non-empty profiles, the maximum entropy/privacy $\log_2 \tau$ is reached if all the topics in the profile appear with the same frequency; the rationale is that such a flat profile does not disclose any particular preferences by the agent.

3 Single-hop query submission game with multiple responders

So far, we have considered only two agents (initiator and responder). While this simplification is useful to illustrate how the anonymous query submission game works, it also strongly limits cooperation and therefore the privacy gain of the agents: the two agents will cooperate to submit each other's queries only if their interests are complementary. To overcome this limitation, more agents are needed. Here we assume that there is one initiator (I) and n responders ($R_1, \ldots, R_n$).

The interactions and utilities of the agents are the same as in the previous case; the difference is that, when forwarding the query to another agent, the initiator has to select one among the n responders. Figure 2 illustrates the possible interactions in the scenario with multiple responders. Notice that action F_i has no effect on the privacy or the functionality of responder R_j, with $j \neq i$. Thus, it is reasonable to assume that the utility of R_j, with $j \neq i$, is zero under F_i. Also, (S) has no effect over any of the responders, so that the utility of R_i under (S) should be zero for all i. Finally, when responder R_i declines the submission, the effect on the privacy and the functionality of all responders should also be zero.

If we analyze the conditions under which co-utile protocols exist in this game, we have:

Proposition 4.2. *Protocol (F_i, S) is co-utile when $u^{R_i}_{F_iS} > 0$ and $u^{R_i}_{F_iS} > u^{R_i}_{F_iD}$ and $u^I_{F_iS} > u^I_S$.*

Unlike in the game with only two agents, none of the protocols (F_i, S), for $i = 1, \cdots, n$, is strictly co-utile because each responder maximizes his utility at a different terminal node.

Since the initiator is unaware of the state of the other agents, she cannot anticipate whether a specific responder will submit or decline her query. In a game with incomplete information as this one, the initiator must check her prior knowledge on the other agents and choose the responder that she expects to be most likely to submit her query. However, following a protocol that is expected to be co-utile is not a satisfactory solution if the expectations are not fulfilled and the query is declined (in this case the protocol actually turns out to be not co-utile).

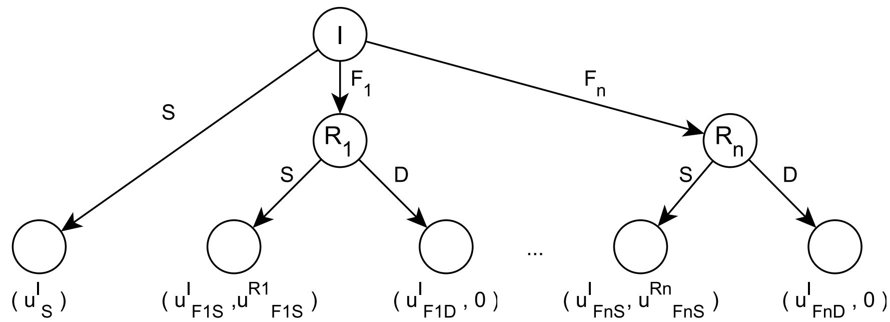

Fig. 2 Tree showing the possible interactions in the single-hop anonymous query submission game with n responders. To keep the figure compact, we only represent the payoffs of the agents active in each interaction.

Also, it does not make sense for the initiator to stop when the query submission is declined by the selected responder. After a refusal, the initiator updates her knowledge about the other agents (by excluding the declining responder) and tries again. This trial and error process has a drawback: it can defer the query submission too long. To prevent this, the initiator should include time among the factors that shape utility, more precisely as part of the functionality atomic utility: the initiator does not only want her query answered, she wants to get the answer within a certain time interval. To account for time, we assume that the response from the WSE is immediate when the initiator submits the query herself, while forwarding it to another agent for submission and getting the answer back implies some delay.

Each new trial implies a different game, in which the utilities are updated with respect to the previous trial game: not only the declining responder in the previous game is excluded, but the initiator's utility is decreased to reflect the passing of time. To prevent the query submission from being deferred too long, there must be a time t such that, regardless of privacy, getting the query answered at time t is always preferable to getting it answered at time $t+1$. Thus, at time t, (S) is preferred by the initiator to (F_i,S) for any i.

Figure 3 depicts the initial game tree for a scenario with an initiator and two responders, where the initiator includes time as part of functionality. Figure 4 shows the updated game tree assuming that the initiator has played F_1 and responder R_1 has declined the submission. This is reflected in the initiator's payoffs, which are reduced by one due to the passed time: the utilities $\hat{u}^I$ in Figure 4 are smaller than the utilities u^I in Figure 3.

Let us now examine how functionality can account for passing time. Clearly, functionality can no longer be a binary function in $\{0,1\}$. Let us assume it takes values in $[0,1]$, so that the actual value measures the initiator's satisfaction at obtaining the query answer at time t:

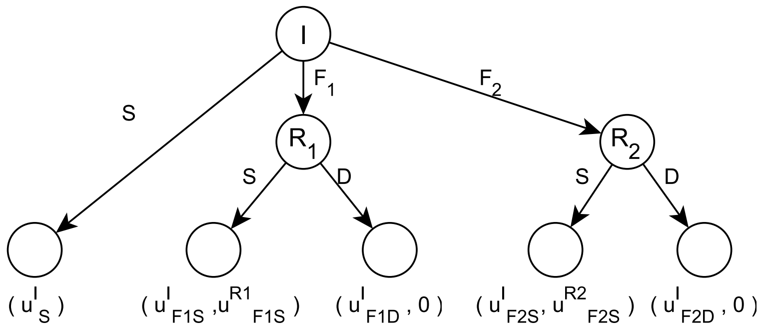

Fig. 3 Tree of interactions in the case of an initiator and two responders at time $t = 0$. To keep the figure compact, we only represent the payoffs of the agents active in each interaction.

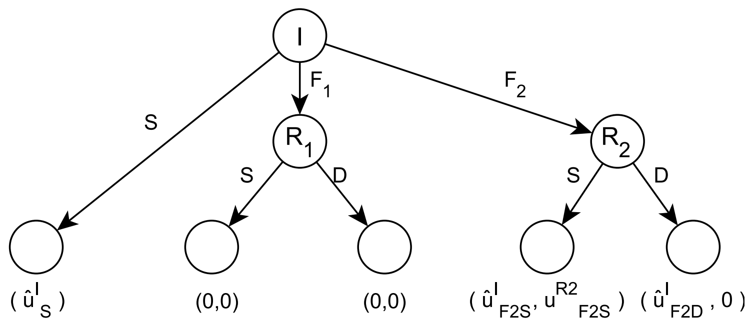

Fig. 4 Updated tree of interactions in the case of an initiator and two responders at time $t = 1$, assuming that responder R_1 declined to submit the initiator's query. To keep the figure compact, we only represent the payoffs of the agents active in each interaction. Due to the elapsed time, it holds that $\hat{u}^I_S \leq u^I_S$, $\hat{u}^I_{F_2S} \leq u^I_{F_2S}$ and $\hat{u}^I_{F_2D} \leq u^I_{F_2D}$, where u^I_S, $u^I_{F_2S}$ and $u^I_{F_2D}$ are the utilities in Figure 3.

$$f_I(P;t) = \begin{cases} 0 & \text{the initiator does not get the query answered at time } t; \\ \alpha_t \in [0,1] & \text{satisfaction level when getting the answer at time } t. \end{cases}$$

A stricter way to define $f_I(P;t)$ in order to prevent too long a deferral of query submission is to set a timeout, T that, if reached, makes (S) the best choice for the initiator; that is, the query answer is only useful if it comes no later than T. To model this behavior, we can set the functionality to $-\infty$ when time T is reached; furthermore, to abstract from the initiator's beliefs, we can consider the functionality to be constant across time until T and to become $-\infty$ thereafter:

$$f_I(P;t) = \begin{cases} 0 & \text{if } t < T \text{ and the initiator does not get the query answered;} \\ 1 & \text{if } t < T \text{ and the initiator gets the query answered;} \\ -\infty & \text{if } t > T. \end{cases}$$

With the above functionality, the initiator's utility does not change for times until T, so that the reasonable way to act for her is to keep trying different responders until either the query is submitted to the WSE or time T is reached. If T is reached, then the initiator submits the query herself. Although Expression (3) is reasonable for the initiator's functionality atomic utility, the actual expression depends on the initiator's preferences: if the functionality gradually decreases with time, the beliefs about the other agents' preferences become more relevant, as it may be better for the initiator to submit the query herself, even before T, if she believes the other agents are likely to decline.

4 Multi-hop query submission game

The single-hop query submission protocol discussed above is effective at protecting the privacy of agents against the WSE. However, since the initiator directly forwards her query to the responders, she also discloses her interests to those responders. Thus, if agents are not only interested in protecting their privacy toward the WSE, but also against their peers, the single-hop protocol is not co-utile anymore. Actually, collaborating with peer agents to avoid being profiled by the WSE can hardly be satisfactory if it has the side effect of being profiled by one's peers.

To realize co-utility when agents want privacy against the WSE and the other agents, a more complex game needs to be designed whereby the identity of the query initiator is not only hidden to the WSE, but also to all other agents. We can achieve this by extending the actions available in the single-hop game as follows: an agent that receives a query to be submitted may now submit it, decline it or *forward it to another peer*. Specifically, forwarding the query to another agent is an alternative to declining it in case the responder does not wish to submit the query to the WSE (because doing so does not increase his privacy). The result of allowing query forwarding is that a responder agent who receives a query no longer knows whether he is receiving it from the initiator or from another responder. Figure 5 illustrates the tree of interactions of this *multi-hop* game in the case of three agents.

If, in the single-hop game with multiple responders, it could take too long for a query to be answered (in case many responders needed to be tried), in the multi-hop game the situation can be even worse. If agents keep forwarding the query without submitting it, the submission can be deferred indefinitely. Like in the previous game, including time in the initiator's utility is a way to deal with this potential problem. Thus, similarly as we did in the single-hop game, to abstract from the initiator's beliefs about the other agents, we assume that the initiator is indifferent to the query response time as long as the answer is received before T; when time T is reached, her utility becomes $-\infty$, which makes direct submission by the initiator the most preferable protocol.

On the other hand, in order to remain perfectly private w.r.t. other peers in the network (or any external observer), agents will also be interested to add uncertainty to the profile that results from the whole set of queries they send (those forwarded

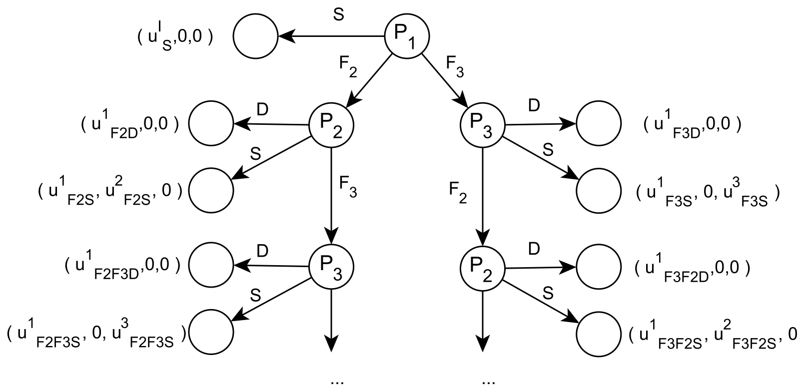

Fig. 5 Tree showing the possible interactions of the multi-hop query submission game with three agents

to other agents or submitted to the WSE). Note that this assumes the worst case in which a network adversary is capable of collecting all queries initiated, forwarded or submitted by each agent. We denote the profile that results from accumulating these queries by Z and, like for Y, we use Shannon's entropy to measure how disclosive it is. By incorporating this additional privacy concern into the privacy atomic utility, we have the following updated expressions that also consider the variation of the entropy that q produces in Z when q is submitted to the WSE or forwarded to another agent:

$$p'_I(P) = p_I(P) + H(Z_I \cup \{q\}) - H(Z_I);$$
$$p'_R(P) = p_R(P) + H(Z_R \cup \{q\}) - H(Z_R).$$

From a co-utility perspective, the possibility of forwarding the query to another agent masks the initiator of the query, thereby preserving her privacy versus the other agents. On the other hand, the privacy concerns of agents w.r.t. external observers make forwarding preferable to declining to any responder agent: forwarding agents also increase the privacy of their external profiles (Z) because the forwarded queries add uncertainty to the queries they really originate. Thus, in a multi-hop protocol with several forwarding hops ending in a submission, not only the initiator and the submitter take their best possible actions, but the forwarders also do. In this sense, the protocol is strictly co-utile (if we consider only the agents participating in it).

5 On the availability of agents with appropriate types

The above co-utile protocols are viable provided that agents with appropriate preferences/types are available. In this section, we discuss how these types are, at the end, the most common ones in a real scenario. To do so, we analyze the system in the most extreme cases.

The first extreme case happens when all agents have empty profiles towards the WSE and/or towards all the other agents because they have not (directly) submitted any query so far. We call this the "cold start" case. As discussed in Section 2, an empty profile offers perfect privacy (to which we assigned by convention maximum entropy in Section 2) because it does not disclose any preference of the agent. According to this, one may assume that agents will always prefer to rely on others to submit their queries and will decline submitting queries of others because this would decrease the entropy of their profiles towards the WSE. This "free-riding" attitude results in all requests for query submission being declined. However, this deadlock cannot be sustained very long: if the initiator has a query that does not make anyone else more private (and, thus, no one wants to forward it, let alone submit it) and if the response time is part of the initiator's utility, she will eventually be forced to submit the query herself. In this way, the initiator's profile stops being empty and its entropy falls sharply (the agent's query is completely exposed); thus, the initiator becomes highly motivated to submit queries from other agents in order to flatten her profile and thereby increase its entropy.

A similar analysis and conclusion apply to the other extreme case that we dub "happy laziness". This situation happens when most of the agents have a non-empty flat profile and, because of this, they prefer not to submit or forward queries from others. "Happy laziness" however comes to an end as soon as some agents initiate new queries: since they cannot find anyone willing to forward or submit them, they are forced to submit them themselves. As in the "cold start" case, this action unflattens their profiles and makes them willing to collaborate with others to increase again the entropy of their profiles.

6 Anonymous query submission considering agents' reputation

In the protocols above, the utility of the initiating agent $\mathscr{P}_i$ (i.e., the one who wants to have a query q answered), can be defined as the aggregation of the functionality and privacy atomic utilities, which are a function of the query q:

$$u_i(q, f_i(\cdot), Y_i, \alpha_i^H) = f_i(q) + \alpha_i^H \cdot \Delta(H(Y_i), q). \tag{1}$$

The first term, $f_i(q)$, represents the functionality that the agent gains from having q answered, which is 1 if the query is answered and 0 otherwise. The second term quantifies the privacy gain or loss w.r.t. the WSE as a result of the submission of the query to the WSE (by $\mathscr{P}_i$ herself or by another agent). Specifically, Y_i is $\mathscr{P}_i$'s profile of submitted queries (which is what the WSE sees on $\mathscr{P}_i$); it is represented as a distribution of topics extracted from the queries submitted so far by agent $\mathscr{P}_i$. From $\mathscr{P}_i$'s perspective, Y_i characterizes the exposure level (i.e., privacy loss) of the agent's interests towards the WSE. The function H is Shannon's entropy and, thus,

$H(Y_i)$ measures the uncertainty (i.e., privacy) of the profile. Note that the highest privacy level is reached when the profile is empty, because no information about the profile has been disclosed. For non-empty profiles, the maximum entropy/privacy is reached if all topics appear with the same frequency, because the agent's interests remain unspecific. The $\Delta(H(Y_i),q)$ function represents the variation (positive or negative) of the agent's privacy in terms of the profile's entropy $H(Y_i)$ as a result of $\mathscr{P}_i$ submitting herself the query q to the WSE; if the query is finally submitted by any other agent, $\Delta(H(Y_i),q)=0$, because the WSE will not learn anything new about the initiator.

Finally, the α_i^H coefficient normalizes the scales of functionality and privacy (so that these heterogeneous features can be coherently compared) and also weights their relative importance to agent $\mathscr{P}_i$. By properly configuring this coefficient, we can make functionality dominate (which is a natural choice).

For the responder $\mathscr{P}_j$, the only utility that matters is his own privacy w.r.t. the WSE and how it is modified by submitting the query q of the initiator $\mathscr{P}_i$. Thus, to derive the utility for $\mathscr{P}_j$, we set $f_j(q)=0$ in Expression (1) and we obtain the following simplified expression:

$$u_j(q,Y_j)=\Delta(H(Y_j),q). \tag{2}$$

Co-utility in the scenario depicted above may be hampered by additional hindrances that may arise in a real setting. In particular, submitting a query q may result in a negative cost (e.g., spent bandwidth or money) to the agents. In this case, the utility functions for the initiator and responder agents need to incorporate this cost $o(q)$ as an additional negative atomic utility:

$$u_i(q,f_i(\cdot),o_i(\cdot),Y_i,\alpha_i^H,\alpha_i^o)=f_i(q)+\alpha_i^H\cdot\Delta(H(Y_i),q)-\alpha_i^o\cdot o_i(q); \tag{3}$$

$$u_j(q,o_j(\cdot),Y_j,\alpha_j^o)=\Delta(H(Y_j),q)-\alpha_j^o\cdot o_j(q). \tag{4}$$

For the initiator agent, the cost $o_i(q)$ constitutes an additional incentive to prefer forwarding the query (which results in $o_i(q)=0$) rather than submitting it herself. For the responder agent, if the privacy gain $\Delta(H(Y_j),q)$ brought to him by the query q does not dominate (considering the weighting coefficient) the cost $o_j(q)$ of the query, the outcome of the utility function is negative and submitting the query "for free" is not rational for him anymore. Thus, in this case, the protocol is no longer self-enforcing, let alone co-utile.

In order to neutralize this negative payoff, agent reputations can be used to ensure collaboration in a way that the costs associated to query submission are fairly distributed throughout the network.

Specifically, we propose to compute the reputation t_i of an agent $\mathscr{P}_i$ as the difference between the cost incurred by $\mathscr{P}_i$ when submitting queries on behalf of other agents and the cost incurred by other agents as a result of submitting $\mathscr{P}_i$'s queries. Assuming the cost of a query submission is uniform for all agents (which implies that agents calculate and evaluate reputations in a similar way), if an agent $\mathscr{P}_i$ has a reputation t_i greater than 0, it means that she has helped others more than what others have helped her.

To evaluate whether the collaboration record of an initiator agent $\mathscr{P}_i$ compensates the cost of helping her, a responder agent $\mathscr{P}_j$ can take into account the reputation t_i of the initiator agent. In our case study, this yields:

$$u_j(q, o_j(\cdot), Y_j, t_i, \alpha_j^o, \alpha_j^t) = \Delta(H(Y_j), q) - \alpha_j^o \cdot (o_j(q) - \alpha_j^t \cdot t_i). \quad (5)$$

According to this expression and leaving aside the privacy gain, $\mathscr{P}_j$ will get a positive utility outcome if the reputation t_i of the initiator compensates the negative cost of the query $o_j(q)$. Likewise, $\mathscr{P}_j$ will decide to refuse submitting queries from any agent $\mathscr{P}_i$ whose reputation (multiplied by the corresponding weight) is less than the relative cost of the query $o_j(q)$. Since the cost is an unbounded value and the reputation is a normalized value, the α_j^t coefficient is necessary to make both values comparable.

Note, however, that the relative importance given to privacy with respect to the cost/reputation difference (weighted by the α_j^o coefficient) may even compensate a negative difference, because an agent may still help a peer with low reputation if the former attaches a lot of importance to her own privacy.

Reputations in [8] are biased towards positive values; that is, in general, initiator agents only get help from other peers if they have already helped other agents (i.e., submitted their queries) more than what others have helped them. In this way, non-collaborative agents become motivated to help others in order to increase their own reputation beyond the relative cost of their queries; otherwise they will be forced to submit their queries themselves, thereby incurring two negative payoffs: query submission cost and loss of privacy. Since the sum of these negative payoffs will severely decrease the agent's utility, helping others becomes the rational choice (helping others certainly has a cost, but it also increases both the reputation and the privacy of the helper).

Thanks to the compensation of costs via reputation, the above reputation-based query submission protocol is strictly co-utile in spite of the query costs. To see this, note that, if the responder $\mathscr{P}_j$ agrees to submit, it is because this is the action among submitting or refusing the initiator's query that brings the maximum utility to $\mathscr{P}_j$. In this case, the outcome for the initiator $\mathscr{P}_i$ is also maximum (because she avoids the privacy loss and the cost of submitting her query).

Ideally, to balance the costs and the social welfare of all agents, these should achieve and maintain a bounded reputation t_{target} slightly greater than 0 that is sufficient to get help from other peers. In this respect, as discussed in [8], agents can view their own reputations as atomic utilities: they should measure the reputation gain that each action of the protocol brings them and how far their actual reputation is from the target reputation t_{target} they want to achieve. Thus, the aggregated utility functions are as follows:

$$\begin{aligned} &u_i(q, f_i(\cdot), o_i(\cdot), Y_i, \alpha_i^H, \alpha_i^o, \alpha_i^{t'}) = \\ &= f_i(q) + \alpha_i^H \cdot \Delta(H(Y_i), q) - \alpha_i^o \cdot o_i(q) + \alpha_i^{t'} \min((t_{i,1} - t_{i,0}), (t_{target} - t_{i,0})); \end{aligned} \quad (6)$$

$$u_j(q, o_j(\cdot), Y_j, \alpha_j^o, \alpha_j^t, \alpha_j^{t'}) = \\ = \Delta(H(Y_j), q) - \alpha_j^o \cdot (o_j(q) - \alpha_j^t \cdot t_i) + \alpha_j^{t'} \cdot \min((t_{j,2} - t_{j,1}), (t_{target} - t_{j,1})), \quad (7)$$

where α_*^H, α_*^o are the weights attached by agents to privacy and cost, respectively, α_*^t is the weight attached to the reputation of other agents, $\alpha_*^{t'}$ is the weight attached to one's own reputation, $t_{i,0}$ is $\mathscr{P}_i$'s reputation before starting the protocol, $t_{i,1}$ is $\mathscr{P}_i$'s reputation after requesting and obtaining $\mathscr{P}_j$ to submit q, $t_{j,1}$ is $\mathscr{P}_j$'s reputation before accepting to submit q on $\mathscr{P}_i$'s behalf and $t_{j,2}$ is $\mathscr{P}_j$'s reputation after submitting q. Note that $\mathscr{P}_i$ and $\mathscr{P}_j$ can accurately anticipate $t_{i,1}$ and $t_{j,2}$, respectively (they do not need to use estimates); this is because in our case study reputation is objective, based on help provided and received.

Considering reputations as atomic utilities as in Expressions (6) and (7) further reinforces the co-utility of the protocol, because now responder agents are additionally motivated to collaborate (in order to increase their own reputations) and initiator agents need to make sure they have high enough reputations to afford their queries to be submitted by other peers (which decreases their reputation). Moreover, as discussed in [8], viewing reputations as utilities also contributes to a fair allocation of query submission costs among the peers.

6.1 Other benefits of co-utile reputation enforcement

Co-utile protocols are viable given the availability of agents with appropriate types/preferences. In this section we illustrate in our case study how the use of reputation makes the protocol realistically self-enforcing by widening the range of agent types and situations in which the query submission protocol remains co-utile.

First, let us consider the case in which a new agent with an empty profile (i.e., she has not submitted any query so far to the WSE) joins the network. Recall that an empty profile has maximum privacy because it does not disclose any interest of the agent. Even neglecting the cost associated to the query submission, such a newcomer will always prefer to rely on others to submit her queries and will refuse submitting queries from others because doing so will decrease her privacy. In plain words, the newcomer becomes a rational "free rider".

This is an endemic problem of many P2P co-utile protocols [13, 8] that can be fixed using reputation. With the reputation mechanism, newcomers have zero global reputation, because no one has interacted with them and the distributed reputation calculation is based on the local reputations, which are zero for newcomers. Thus, in the reputation-based protocol discussed in this case study, responder agents will systematically refuse requests for query submission from free riders. Given that the functionality (i.e., to have the query answered) is likely to dominate the other utilities, the free-riding agent has only two options left:

- She submits her query herself. This action, in turn, will create a non-empty profile for the agent and a sharp decline of the profile's privacy because the agent is now completely exposed to the WSE. As a result, the agent will become highly motivated to contribute to the system by submitting other agents' queries in order to flatten her profile again.
- She accepts a query request from another agent. This results both in a cost and in a decrease of her privacy, even though her profile gives a distorted view of her interests towards the WSE because the query is unlikely to match her true interests. However, this action will also increase the agent's reputation, who will now be able to find another peer willing to submit her query.

In both cases, newcomers with empty profiles become collaborative members of the network in a self-enforcing way. Note that newcomers with non-empty profiles (i.e., who have already submitted queries before joining the network) are not problematic at all because they are already motivated to accept queries from other peers to flatten their profiles.

The above discussion can be extended to an extreme scenario in which all agents join the system at once with empty profiles and zero global reputations. One might think that this setting brings the system to a standstill in which any request for query submission by any agent is refused and agents do not want to submit their own queries. However, it is easy to see that, thanks to functionality dominating the other utilities, agents will end up submitting their own queries; in turn, this will instantly motivate them to submit queries from peers, which will also increase their reputations and the chances that other agents accept their requests.

A different scenario which might also seem standstill-prone occurs when agents achieve a non-empty but flat profile (and probably a high reputation) thanks to the number of query requests they have satisfied; at this point, they might be tempted to stop collaborating. In the unlikely case that all agents reach such a state, they will be unable to find peers willing to submit their queries, regardless of the reputations they have; as a result, they will be forced to submit their own new queries themselves, which in turn will unflatten their profiles and make them again willing to collaborate. If only some of the agents reach this "too happy" state, their reputation will decrease if they ask other peers to submit their new queries; when their reputation becomes too low, they will be unable to find any other peer willing to submit their queries, which will force them to submit them themselves; this will make them again willing to collaborate.

Finally, the fact that agents may also consider their own reputations as atomic utilities has two virtuous effects. On the one hand, since the responder increases his reputation as a result of the submission whereas the initiator decreases her reputation by the same magnitude, the reputation management guides the system towards a state in which most of the agents have a similar –target– reputation, which corresponds to a *fair* allocation of costs/negative payoffs. On the other hand, in the basic protocol, initiator agents randomly chose potential responders, who may or may not fulfill their requests according to their –secret– privacy needs (profile). With the reputation mechanism, initiator agents may also take into account the reputation of the potential responders so that, instead of choosing them randomly, they may choose agents with zero or low reputation, because (for an equal secret profile) these are more motivated to help in order to increase their own reputations (and thus be able to get help from others). In this way, the initiator increases the chances of finding a helpful responder.

7 Implementation and experiments

In this section, we empirically study the behavior of the privacy-preserving query submission protocol and the influence of the reputation mechanism to foster collaboration between agents.

The agent profiles w.r.t. the WSE have been characterized by following the state of the art in user profiling [16]: profiles are represented as vectors of normalized weights, each one quantifying the relative interest of the agent in a certain topic. Each topic weight represents the aggregation of the semantics of the user's queries that fall into the topic, which is quantified by means of linguistic and semantic analyses (see details in [16]).

The simulations have been carried out with 900 agents. Each one has a set of queries for which she wishes to get an answer from the WSE. These queries were randomly picked from the query logs of real users available in the AOL data set [1], which consists of logs of the actual queries performed by users of the AOL's WSE during 3 months in 2006. In total, we compiled 20,914 individual queries to be performed by the agents through the life cycle of the system. In our simulations, agents iteratively select a query from their query logs and decide whether or not to follow the anonymous submission protocol to retrieve the result from the WSE according to their states (current profiles of submitted queries) and types (utility functions). When applicable, global reputations are computed after each query submission.

We have set up the initial profile of each agent to follow the distribution of topics defined by the query log associated to each agent; we do this so that the queries initially submitted by each agent reflect her inherent interests. With this configuration, the average entropy of the initial profile of the agents is 2.15.

We evaluated the behavior of the system in the following scenarios:

- *S1*: In the simplest scenario, neither costs nor reputations are considered. Agents only care about the functionality of the query and their own privacy. Their utility functions are those of Expressions (1) and (2). Utility coefficients have been set to make functionality dominate, so that queries will be *always* submitted, either by the initiators or by other agents, regardless of the privacy gain/loss.
- *S2*: In the second scenario, the cost of submitting a query is considered. The utility functions of agents follow Expressions (3) and (4). The cost of a query $o(q)$ for the submitting agent is quantified as +1. We performed several executions by varying the relative importance (weighting coefficient α^o) that agents attach to the submission cost in their utility functions.
- *S3*: In the third scenario, both costs and agent reputations (measured as the difference between costs borne and caused) have been considered. In this case,

responder agents have the utility of Expression (5). Given that reputations are normalized in the [0..1] range and reputation values of all peers add to 1, to properly aggregate costs and reputations, the latter have been de-normalized by setting $\alpha_j^t = \text{number\_of\_peers} = 900$. Again, we performed several simulations by modifying the relative importance of the cost (cost/reputation difference for responder agents).

- *S4*: In the fourth scenario, agents also consider how their own reputation would be affected by the actions they can take (i.e., query forwarding or submission). Initiator agents follow Expression (6) and responder agents follow Expression (7). The target reputation value has been set to the average reputation ($t_{target} = 1/\text{number\_of\_peers} = 1/900$), so that reputations will tend to be balanced and, thus, costs will be fairly allocated. To properly aggregate costs and reputations, we have set the coefficient $\alpha_*^{t'} = \text{number\_of\_peers} \cdot \alpha_*^o$ for each simulation. In this manner, the difference of reputations is de-normalized (so that it can be aggregated with cost values) and given the same weight as agent reputations in scenario *S3*.

The system behavior has been evaluated according to the following aspects:

- Number of queries that have been submitted by following the co-utile protocol. To put this value in context, we also compute the ratio between this number of queries and the total number of queries that initiator agents decide to forward according to their types, that is, the number of queries that potentially can be submitted in a co-utile way. A rate around 1 would be desirable, since it means that the protocol remains self-enforcing for most situations and agent types.
- The standard deviation of agents' reputations. Since our reputation measures the difference between the cost incurred by an agent when submitting other agents' queries and the cost caused to other agents who agree to submit the agent's own queries, a low deviation from the average reputation (which by definition is $1/\text{number\_of\_peers}$) indicates a fair allocation of costs resulting from agent collaboration across the network.
- The privacy of agents' profiles, measured as the average entropy of their profiles at the end of each simulation.

For Scenario *S1*, Figure 6 shows the evolution (on average over all the agents) of the agents' privacy (Y-axis), as a function of the number of queries that have been submitted so far (X-axis). The horizontal line at the top shows the privacy upper bound (i.e., the maximum entropy of the profiles when all agents achieve a perfectly flat distribution of the 15 topic weights considered in their profiles, which is $\log_2 15 = 3.9$).

We can see that the agent profiles become more private w.r.t. the WSE at each new iteration of the protocol. The entropy $H(Y)$ grows logarithmically and becomes quite flat after 10,000 queries. It is important to note that, in practice, the limited variability of the queries (whose topics are not evenly distributed among the 15 possible topics) makes it difficult to achieve this theoretical upper bound (in which all agent profiles are perfectly flat). The average entropy of the agents' profiles at the end of the simulation is 2.9.

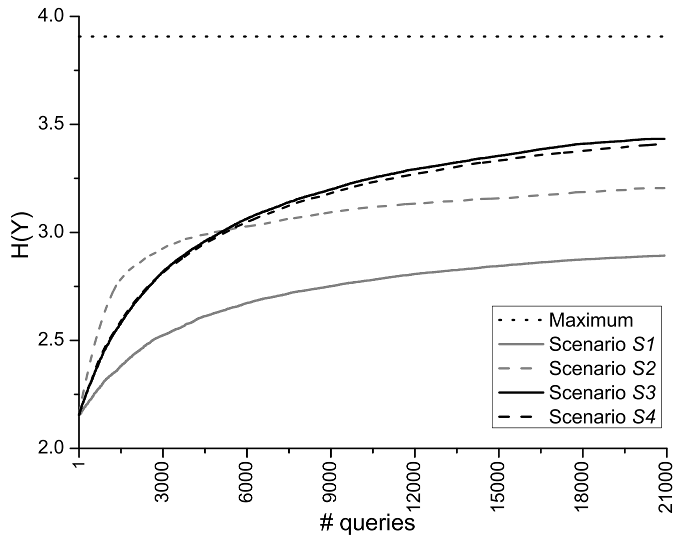

Fig. 6 Evolution of privacy (agents' profile entropy) for all scenarios with a cost coefficient of 0.25 (for *S2*, *S3*, *S4*)

In this scenario, initiator agents preferred to forward 11,544 queries (from the total 20,914 queries), because submitting them directly would have decreased their privacy. From these, 10,806 queries (93.61%) were submitted by following the co-utile protocol; that is, in less than 7% of the cases, no other agent was willing to satisfy the initiator's request and the initiator was forced to submit the query herself (because the functionality utility dominates). This illustrates how, in absence of costs, the game is naturally co-utile for most agent types.

For Scenario *S2*, Table 1 shows the simulation results obtained by varying the relative importance given to the cost of performing a query to the WSE (α_i^o coefficient in Expression (3) for initiators and α_j^o coefficient in Expression (4) for responders).

Table 1 Scenario *S2*: number and proportion of queries submitted to the WSE following the co-utile protocol, and average profile entropy (privacy) at the end of the simulation, for different cost coefficients.

cost coefficient	co-utile queries (proportion)	average profile entropy
4	0 (0)	2.15
1	398 (0.019)	2.62
0.25	2,930 (0.14)	3.20
0.063	8,788 (0.42)	3.47
0.016	15,816 (0.83)	3.35
0.004	14,341 (0.93)	3.05
0.001	11,910 (0.93)	2.95

We can clearly see how the more importance the agents attach to the cost of the queries, the more they tend to refuse submitting queries from other peers (even if refusal implies missing the chance of a privacy gain). As a result, initiators are forced to ask more peers and, if none of the latter is willing to satisfy their requests, initiators will end up submitting their queries themselves. This makes the system not co-utile for many agent types. Only when agents give a small importance to the cost (below 0.016), they follow the protocol in most cases (i.e., for more than 80% of the queries to be submitted). We can also see that, even though the proportion of co-utile queries monotonically increases as the cost coefficient decreases, the actual number of co-utile queries starts to decrease for costs coefficients below 0.016. Indeed, when initiator agents give very little importance to the query costs, they tend to submit themselves the queries that do not harm their privacy, rather than using the protocol just to save submission costs.

The privacy of agents is also significantly affected by the lack of co-utility: profile entropies are hardly improved (from the initial average entropy of 2.15) when cost coefficients are high enough (from 1 to 4) to prevent the system from being co-utile in most cases. Interestingly, profile entropies reach a high value (around 3.2) when just 14% of the queries are submitted by following the co-utile protocol (see the privacy evolution of this case in Figure 6). This shows how even submitting a small amount of *privacy-preserving* queries from other peers has a very positive flattening effect on the responder's profile. In fact, due to the large weight given by the agents to the query costs, the queries they actually submit on behalf of other agents *have to* provide large privacy gains that compensate their submission cost (see Expression (4)).

Table 2 shows the results for Scenario *S3*, in which reputation is also considered by responders as a payoff.

Table 2 Scenario *S3*: number and proportion of queries submitted to the WSE following the co-utile protocol, average profile entropy (privacy), and standard deviation of agents' reputations at the end of the simulation, for different cost coefficients.

cost coefficient	co-utile queries (proportion)	average profile entropy	agents' reputation std. deviation
4	7,898 (0.37)	2.80	0.00047
1	8,252 (0.39)	3.16	0.00044
0.25	13,501 (0.64)	3.44	0.00038
0.063	18,986 (0.92)	3.34	0.00038
0.016	18,257 (0.97)	3.10	0.00037
0.004	14,611 (0.95)	2.98	0.00037
0.001	10,899 (0.93)	2.90	0.00037

In comparison with the previous scenario, Scenario *S3* results in a significantly greater number of co-utile query submissions for the same coefficients of cost (cost-reputation difference, in this case). The co-utile query proportion among the forwarded queries, in fact, reaches and maintains a value above 0.9 for cost coefficients below 0.063; this is is comparable to the proportion of Scenario *S1*, in which costs were not considered at all. The actual number of queries submitted in a co-utile way is also higher than in Scenario S2, especially for high cost coefficients. Reputation deviations tend to stay steady for cost coefficients below 1.

Finally, Table 3 shows the results for Scenario *S4*, in which the impact of the various available actions on the agent's own reputation is also considered as an atomic utility.

Table 3 Scenario *S4*: number and proportion of queries submitted to the WSE following the co-utile protocol, average profile entropy (privacy), and standard deviation of agents' reputations at the end of the simulation, for different cost coefficients.

cost coefficient	co-utile queries (proportion)	average profile entropy	agents' reputation std. deviation
4	14,968 (0.71)	3.11	0.00017
1	16,201 (0.77)	3.30	0.00017
0.25	17,258 (0.82)	3.41	0.00021
0.063	19,671 (0.95)	3.32	0.00026
0.016	18,307 (0.96)	3.13	0.00028
0.004	14,685 (0.95)	3.01	0.00028
0.001	12,090 (0.94)	2.93	0.00029

In this case, the interest of the agents to increase their own reputation makes them more motivated to accept query requests from other peers, in spite of query costs or initiators' low reputations. As a result, the system becomes co-utile in most cases and for any cost coefficient values. In fact, the proportions of co-utile queries are quite similar to the ones of scenario *S1*, in which no costs were considered, and the actual numbers of queries submitted in a co-utile way are greater than in any of the previous scenarios.

Moreover, in comparison with Scenario *S3*, in Scenario *S4* we obtain lower standard deviations for the reputation values. Agents are motivated to reach the target reputation but not to exceed it. As a consequence, agents with high enough reputations refuse query requests more frequently than agents with low reputations, which yields a fairer allocation of query costs among the peers.

Finally, it is important to note the beneficial effect of reputation (as a cost-neutralizing feature) on the privacy finally achieved by the agents. In comparison with Scenario *S1* (for which we obtained an average profile entropy of 2.9), with the reputation mechanism we were able to reach substantially higher figures for all cost coefficients, as shown in Figure 6. Indeed, due to the incurred costs and reputation differences resulting from forwarding or submitting queries, agents become more selective (in terms of the privacy gains) about which queries they forward or accept. As a consequence, the privacy gains of queries are better allocated among the peers.

8 Conclusions

In this chapter, we have shown how to apply the theory of co-utility to enforce rationally sustainable P2P anonymous keyword search, and we have developed several self-enforcing co-utile protocols for that scenario. Moreover, we have discussed the need of a reputation mechanism to introduce artificial incentives that compensate the effect of negative payoffs. Specifically, by means of the reputation mechanism detailed in [8] and in the second chapter of this book (which is itself a co-utile mechanism), we can bring co-utility to many protocols that are not naturally co-utile due to costs incurred by participants [13, 14, 8].

Acknowledgements Funding by the Templeton World Charity Foundation (grant TWCF0095/AB60 "CO-UTILITY") is gratefully acknowledged. Also, partial support to this work has been received from the Government of Catalonia (ICREA Acadèmia Prize to J. Domingo-Ferrer and grant 2014 SGR 537), the Spanish Government (projects TIN2014-57364-C2-1-R "SmartGlacis", TIN2015-70054-REDC and TIN2016-80250-R "Sec-MCloud") and the European Commission (projects H2020-644024 "CLARUS" and H2020-700540 "CANVAS"). The authors are with the UNESCO Chair in Data Privacy, but the views in this work are the authors' own and are not necessarily shared by UNESCO or any of the funding bodies.

References

1. AOL Search Data Mirrors. Available at `http://gregsadetsky.com/aol-data/` (last accessed: Jan. 17, 2016)
2. Castellà-Roca, J., Viejo, A., Herrera-Joancomartí, J. (2009) Preserving user's privacy in web search engines. Computer Communications 32(13-14):1541–1551.
3. Domingo-Ferrer, J., Solanas, A., Castellà-Roca, J. (2009) $h(k)$-Private information retrieval from privacy-uncooperative queryable databases. Online Information Review 33(4):720–744.
4. Domingo-Ferrer, J., González-Nicolás, Ú. (2012) Rational behavior in peer-to-peer profile obfuscation for anonymous keyword search. Information Sciences 185:191–204.
5. Domingo-Ferrer, J., González-Nicolás, Ú. (2012) Rational behavior in peer-to-peer profile obfuscation for anonymous keyword search: the multi-hop scenario. Information Sciences 200:123–134.
6. Domingo-Ferrer, J., Martínez, S., Sánchez, D., Soria-Comas, J. (2017) Co-utility: self-enforcing protocols for the mutual benefit of participants. Engineering Applications of Artificial Intelligence 59:148–158.
7. Domingo-Ferrer, J., Sánchez, D., Soria-Comas, J. (2016) Co-utility: self-enforcing collaborative protocols with mutual help. Progress in Artificial Intelligence 5(2):105–110.
8. Domingo-Ferrer, J., Farràs, O., Martínez, S., Sánchez, D., Soria-Comas, J. (2016) Self-enforcing protocols via co-utile reputation management. Information Sciences 367(C):159–175.
9. Howe, D. C., Nissenbaum, H. (2009) TrackMeNot: resisting surveillance in web search. In: Lessons from the Identity Trail: Privacy, Anonymity and Identity in a Networked Society. Oxford University Press, pp. 417–436.
10. Murugesan, M., Clifton, C. (2009) Providing privacy through plausible deniable search. In: Proceedings of the SIAM International Conference on Data Mining (SDM 2009). SIAM, pp. 768–779.
11. Reiter, M. K., Rubin, A. D. (1998) Crowds: anonymity for web transactions. ACM Transactions on Information and System Security-TISSEC 1(1):66–92.

12. Sánchez, D., Castellà-Roca, J., Viejo, A. (2013) Knowledge-based scheme to create privacy-preserving but semantically-related queries for web search engines. Information Sciences 218:17–30.
13. Sánchez, D., Martínez, S., Domingo-Ferrer, J. (2016) Co-utile P2P ridesharing via decentralization and reputation management. Transportation Research Part C 73:147–166.
14. Turi, A. N., Domingo-Ferrer, J., Sánchez, D. (2016) Filtering P2P loans on co-utile reputation. In: Proceedings of the 13th International Conference on Applied Computing, AC 2016, pp. 139–146.
15. Viejo, A., Sánchez, D. (2014) Profiling social networks to provide useful and privacy-preserving web search. Journal of the Association for Information Science and Technology 65(12):2444–2458.
16. Viejo, A., Sánchez, D., Castellà-Roca, J. (2012) Preventing automatic user profiling in Web 2.0 applications. Knowledge-based Systems 36:191–205.