A Privacy-Preserving Architecture for the Semantic Web based on Tag Suppression

Javier Parra-Arnau, David Rebollo-Monedero, and Jordi Forné

Department of Telematics Engineering, Technical University of Catalonia (UPC),
E-08034 Barcelona, Spain
{javier.parra,david.rebollo,jforne}@entel.upc.edu[⋆]

Abstract. We propose an architecture that preserves user privacy in the semantic Web via tag suppression. In tag suppression, users may wish to tag some resources and refrain from tagging some others in order to hinder privacy attackers in their efforts to profile users' interests. We describe the implementation details of the proposed architecture and provide further insight into the modeling of profiles. In addition, we present a mathematical formulation of the optimal trade-off between privacy and tag suppression rate.

1 Introduction

The World Wide Web constitutes the largest repository of information in the world. Since its invention in the nineties, the form in which information is organized has evolved substantially. At the beginning, web content was classified in directories belonging to different areas of interest, manually maintained by experts. These directories provided users with accurate information, but as the Web grew they rapidly became unmanageable. Although they are still available, they have been progressively dominated by the current search engines based on web crawlers, which explore new or updated content in a methodic, automatic manner. However, even though search engines are able to index a large amount of web content, they may provide irrelevant results or fail when terms are not explicitly included in web pages. A query containing the keyword *accommodation*, for instance, would not retrieve web pages with terms such as *hotel* or *apartment* not including that keyword.

Recently, a new form of conceiving the Web, called the *semantic Web* [1], has emerged to address this problem. The semantic Web, envisioned by Tim Berners-Lee in 2001, is expected to provide the web

[⋆] This work was supported in part by the Spanish Government under Projects CONSOLIDER INGENIO 2010 CSD2007-00004 "ARES" and TSI2007-65393-C02-02 "ITACA", and by the Catalan Government under Grant 2009 SGR 1362.

content with a conceptual structure so that information can be interpreted by machines. The semantic Web requires to explicitly associate meaning with resources on the Web. This process is normally referred to as *semantic tagging*, or simply tagging, and is supposed to play a key role for the semantic Web to become a reality. One of the benefits of associating concepts with web pages is the semantic interoperability in web applications. Furthermore, tagging allows applications to decrease the interaction with users, to obtain some form of semantic distance between web pages and to ultimately process web pages whose content is nowadays only understandable by humans.

Despite the many advantages the semantic Web is bringing to the Web community, the continuous tagging activity prompts serious privacy concerns. More specifically, tags submitted to a web server could be used to derive user's preferences [2] or expertise [3], and thus obtain precise user profiles containing sensitive information such as health, political affiliation, salary or religion. This could be the case of recommendation web sites such as *Last.fm*, *Movielens* or *Jinni*, where user profiles are normally shown by some kind of histogram or tag cloud, as depicted in Fig. 1.

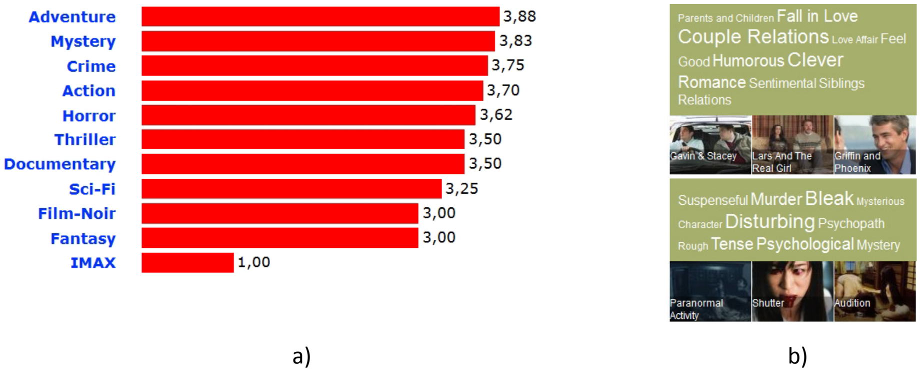

Fig. 1. A histogram (a) and a tag cloud (b) displaying user profiles in *Movielens* and *Jinni*, respectively.

1.1 Contribution and Plan of this Paper

In this paper, we present an architecture that preserves user privacy in the semantic Web via *tag suppression*. More specifically, users may wish to tag some resources and refrain from tagging some others when their privacy is being compromised. The proposed architecture helps users decide which tags should be suppressed in order to hinder privacy attackers in their efforts to profile users' interests. Consequently, this approach guarantees user privacy to a certain extent, at the cost of processing overhead and the semantic loss incurred by suppressing tags, but without having to trust the web server or the network operator.

Additionally, we present an information-theoretic formulation of the trade-off between privacy and tag suppression rate, which arises from our definition of privacy risk. In particular, we measure privacy risk as a divergence between a user's apparent tag distribution and the population's.

Sec. 2 explores the basics of the semantic Web and reviews some relevant approaches related to privacy. Sec. 3 describes our privacy-preserving architecture and focuses on its internal components. Sec. 4 presents our privacy measure and a formulation of the trade-off between privacy and tag suppression rate. Conclusions are drawn in Sec. 5.

2 State of the Art

This section describes the fundamentals of the semantic Web and includes some relevant contributions to privacy within this context.

As mentioned in Sec. 1, the semantic Web requires to explicitly associate meaning with resources on the Web. In order to achieve this meaningful structure, the conceptual description of resources must be described formally. For this purpose, the World Wide Web Consortium (W3C) proposes to use the resource description format (RDF), which is a general-purpose language for representing information on the Web. In RDF, the meaning is encoded by a triple consisting of a *subject*, a *predicate* and an *object*. According to this format, a resource on a web page (subject) is associated with a property (predicate), to which a value (object) is assigned. For instance, in the statement "1984 was directed by George Orwell", "1984" would be the subject, "was directed by" the predicate, and "George Orwell" the object.

Although RDF provides the technology to describe meaning, the semantic Web requires also that concepts and terms share a common definition. Ontologies, which are defined in [4] as "a formal, explicit specification of a shared conceptualization", arise with this aim. In the semantic web context, an ontology is a set of statements where terminology is defined using a specific language. Several languages such as RDF schemas (RDF-S) [5] or ontology web language (OWL) [6] are used to express ontologies.

A number of approaches have been suggested to preserve user privacy in the semantic Web, most of them focused on privacy policies. In the traditional Web, the majority of web sites interact with users to provide them with privacy policies, and allowing them to find out how their private information will be managed. Unfortunately, users do not frequently understand [7] or even read [8] privacy policies. The platform for privacy preferences (P3P) is created to deal with this situation and provides a framework with informed online interactions. More specifically, when a web site supports the P3P, it establishes a set of policies to define how user's private information will be used. Users, in turn, set their own privacy policies to determine what kind of personal information they are willing to disclose to the web sites they browse. Accordingly, when a user browses a web site, P3P compares both the web site's and the user's privacy policies. If they do not match, P3P informs the user about this situation and consequently they decide how to proceed. In the semantic Web, this process is intended to be carried out by autonomous agents. In this context, several policy languages to define privacy and security requirements have been proposed. In [9], the authors suggest a new semantic policy language based on RDF-S to express access control requirements over concepts defined in ontologies. In [10], privacy and authentication policies are incorporated into the descriptions of an ontology called OWL *for services* (OWL-S). Furthermore, the authors implement algorithms for the requester to verify the provider's adherence to policies.

In the context of private information retrieval (PIR), users send general-purpose queries to an information service provider. An example would be a user sending the query:"What was George Orwell's real name?". In this scenario, query forgery, which consists in accompanying genuine with false queries, appears as an approach to guarantee user privacy to a certain extent at the cost of traffic and processing overhead. Building on this principle, several PIR protocols, mainly heuristic, have been proposed and implemented. In [11, 12], a solution is presented, aimed to preserve the privacy of a group of users sharing an access point to the Web while surfing the Internet. The authors propose the generation of fake transactions, i.e., accesses to a web page to hinder eavesdroppers in their efforts to profile the group. Privacy is measured as the similarity between the actual profile of a group of users and that observed by privacy attackers [11]. Specifically, the authors use the cosine measure, as frequently used in information retrieval [13], to capture the similarity between the group genuine profile and the group apparent profile. Based on this model, some experiments are conducted to study the impact of the construction of user profiles on the performance [14]. In line with this, some simple, heuristic implementations in the form of add-ons for popular browsers have recently started to appear [15, 16].

Despite the simplicity of the mechanism described above, an analogous *tag forgery* would clearly not be convenient for the semantic Web, which is the motivating application of our work. Submitting a tag implies the construction of conceptual relations, a much more complex process than just sending a simple query to a service provider. Therefore, users might not be willing to manually tag web content they are not interested in.

3 An Architecture for Privacy Preservation in the Semantic Web

This section presents the main contribution of this work: a privacy-preserving architecture in the semantic Web via tag suppression. More specifically, Sec. 3.1 provides further insight into the construction of user profiles. Sec. 3.2 examines our architecture from a global point of view. Sec. 3.3 focuses on the user-side architecture and goes into the details of its internal functional blocks. The specification of one of these blocks will be given in Sec. 4.

3.1 User Profile Construction

Our architecture contemplates that the profile of a user is directly obtained from specific modules integrated into the user's system. Before giving any details on the construction of user profiles, we will first explore how this information could be represented.

Sec. 1 already mentioned that some recommendation web sites commonly use some kind of histogram to show a user profile, as in the case of *Movielens*, or tag clouds, as in *Jinni*. Bearing in mind these examples, we propose a first-approximation, mathematically-tractable model of user profile as a probability mass function (PMF). Accordingly, we suggest two alternatives to model a user profile. Our first proposal entails certain information loss, as it uses categories into which tags are mapped. On the one hand, this could be difficult to carry out, as the meaning of tags would have to be interpreted in order to classify them into categories, but on the other hand, the description of user profiles could be simplified. Our second alternative represents a user profile by means of tags, which do not necessarily coincide with the semantic tags in the RDF format discussed in Sec. 2. Consequently, this approach could provide a much more accurate description of user profiles, although at the expense of a higher complexity.

Once we have described our proposals to represent a user profile, we will now focus on how to extract this information from a user tag activity. We shall assume that user profiles are modeled by tags, although all considerations also apply to category-based profiles. The naive solution is to locally keep a histogram of all the submitted tags, and to calculate the relative frequency of each tag. Accordingly, this PMF would be updated every time a new tag is generated. However, an improved version would explore contextual information to derive a more accurate profile. A possible approach would be using the vector space model [17], as normally done in information retrieval, to represent web pages as tuples containing their most representative terms. More specifically, the term frequency-inverse document frequency (TF-IDF) would be applied to calculate the weights of each term appearing in a web page. Afterwards, the most weighted terms could be combined with the semantic tag submitted by the user in order to obtain an enriched tag. In the remainder of this section, we shall refer to this enriched tag as *profile tag*, as it will be used by the system to construct the user profile, whereas we shall call *semantic tag*, or simply *tag*, the one created by the user in a format such as RDF. For instance, consider a user browsing a web page and submitting the tag “A conference was held in Copenhagen”. Instead of using this tag to update the user profile, the system would first extract contextual information from the web page as described above, and later, the profile tag “Copenhagen climate conference” would be used to update the user profile, resulting in a more precise description.

Although this section just describes how to construct user profiles, analogous arguments would apply to the modeling of the population profile. Sec. 3.3 gives more details on this.

3.2 Architecture Overview

Our architecture is built on the simple principle of tag suppression. More specifically, a user may wish to tag some resources and refrain from tagging some others when their privacy is being compromised. Our proposal is motivated by the intuitive observation that a privacy attacker will have actually gained some information about a user whenever the user profile differs from the population profile. Accordingly, we now describe an architecture that helps users decide which tags should be suppressed in order to hinder privacy attackers in their efforts to construct a user profile too different from the population profile.

The main component of this architecture is the web and tag server (WTS), a single entity in which web pages and their semantic tags are stored. Users browsing the Web would retrieve those data from the WTSs. The web browser would represent this information so that it could be understood by users. Afterwards, users would generate their own semantic tags and would submit them to the WTSs.

Users would calculate the population profile as the relative frequency of the tags stored in a particular WTS. This could be done by a crawler application collecting the tags submitted to that WTS. Later, this profile would be used to prevent that WTS from deriving accurate user profiles. As the population would be restricted to users tagging in the same WTS, they would have to store a different population profile for each WTS. More details are given in the next section.

3.3 User-Side Architecture

This section examines the internal components of the proposed architecture and goes into the details of a practical implementation.

The user-side architecture is depicted in Fig. 2. As it can be seen there, our proposal is composed by a number of modules, each of them performing a specific task. Next, we provide a functional description of all of their components.

Web Browser. This module is essentially responsible for the communication with the WTS. Specifically, it downloads both the web content and the semantic tags that the user specifies by means of a URL. Afterwards, the web content is delivered to the *context analyzer*, which extracts contextual information from the web page. The web browser is also in charge of submitting the tags proposed by the user to the WTS. Last but not least, this module also retrieves the tags requested by the *tag crawler* component.

Context Analyzer. This module is aimed to process the web content that is either requested by the user or explored by the tag crawler. Particularly, it performs this task by using the vector space model and the TF-IDF weights commented on in Sec. 3.1. As a result, a tuple of weighted terms is internally generated for each web page. Later, the context analyzer takes a number of the most weighted terms of each tuple, and sends them to the *profile tag generator* module. The selection of these terms could be done according to these two possible alternatives: a user could choose either a fixed number of terms n, or those terms with weights above a threshold t. This selection poses an inherent compromise between accuracy and complexity, regardless the alternative chosen. The higher the resulting number of terms, the higher the accuracy in the description of the profile tag, but the higher the difficulty to handle that user profile.

Tag Crawler. This module retrieves the tags stored in a WTS. Namely, the web browser gives the tag crawler the URL specified by the user. The tag crawler browses then the web pages stored in the corresponding WTS and retrieves the other users' tags. These retrieved tags are submitted to the profile tag generator module linked to the *population profile constructor* block.

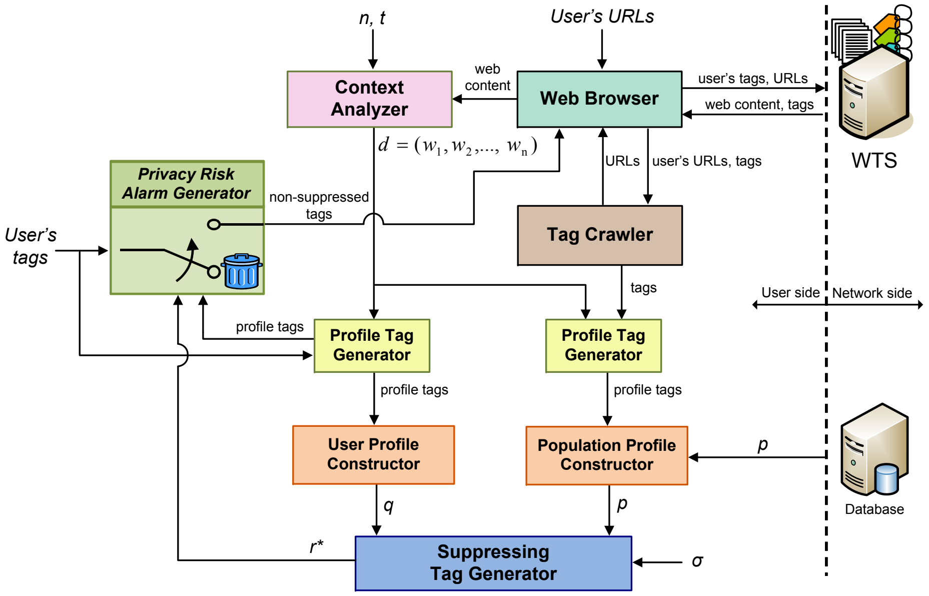

Fig. 2. Internal components of the user-side architecture.

Profile Tag Generator. This component generates profile tags from both the semantic tags and the contextual information given by the context analyzer. The architecture is composed of two profile tag generator modules. One of these modules derives profile tags from the tags proposed by the user, and the other generates them from the population's tags retrieved by the tag crawler. The resulting profile tags are delivered respectively to the modules *user profile constructor* and population profile constructor. In addition, the user's profile tags are sent to the *privacy risk alarm generator* block.

Population Profile Constructor. It is responsible for the estimation of the population's tag profile. As the concept of population is limited to users tagging in a common WTS, this module requires to store a population profile for each WTS. Specifically, this module obtains profile tags from one of the profile tag generators. Based on these profile tags, the population profile constructor proceeds as follows: if the profile tag is not included in the population profile, a new entry for it will be automatically created. However, if the profile tag already exists the population profile will be just updated. Alternatively, this block could query databases containing this kind of information. This would be the case, for example, of a future application similar to *Google Insight*.

User Profile Constructor. Analogously to the population profile constructor, this component generates the user's tag profile. Specifically, this module receives profile tags from the profile tag generator dealing with the tags proposed by the user. These profile tags update the user profile like the population profile constructor module does.

Suppressing Tag Generator. This module is the core of the proposed architecture as it is directly responsible for the user privacy. First, this component is provided with both the user and the population profile. In addition, the user specifies a *tag suppression* rate σ, which is a parameter reflecting the proportion of tags that the user is willing to suppress. Next, this module computes the optimum tuple of suppressing tags r^*, which contains information about the profile tags that should be suppressed. Finally, this tuple is given to the *privacy risk alarm generator* module. The suppressing tag generator block is specified in Sec. 4 by means of a mathematical formulation of the trade-off between privacy and tag suppression rate.

Privacy Risk Alarm Generator. The functionality of this module is to warn the user when their privacy is being compromised. When the user submits a tag to the system, this module waits for the profile tag generator to send the profile tag corresponding to the semantic tag. Additionally, this module receives the tuple r^* and proceeds as follows: if the probability of that profile tag in r^* is positive, a privacy risk alarm is generated to warn the user, and it is then for the user to decide whether to eliminate the tag or not. However, if that probability is zero, the system is not aware of any privacy risk and then sends the tag to the web browser.

Having examined each individual component, we will next describe how this system would work. Initially, the user would browse a web page and would submit tags to a WTS. The contextual information derived by the context analyzer would be used to transform these tags into profile tags, and then construct the user profile. At the same time, the tag crawler would retrieve semantic tags from that WTS, and analogously the population profile would be constructed. Both the user profile and the population profile would be used to calculate the tuple r^* every time these profiles were updated. At a certain point, the user could receive a privacy risk alarm when trying to submit a new tag. If this was the case, the user would have to decide whether to eliminate the tag or not.

4 Formulation of the Trade-Off between Privacy and Tag Suppression Rate

This section presents our privacy criterion and a formulation of the trade-off between privacy and tag suppression rate in the semantic Web, which is used to specify one of the functional blocks in Sec. 3.3.

Sec. 3.1 explained how certain recommendation web sites show user profiles. In particular, we mentioned that this information is normally displayed using histograms or tag clouds. Now, we provide a more formal approach to describe user profiles. Specifically, we model user *tags* as random variables (r.v.'s) on a common finite alphabet of n categories or topics, or more specific tags. This model allows us to describe user profiles by means of a PMF, leading to a similar representation than that shown in Fig. 1a. Accordingly, we define q as the probability distribution of the tags of a particular *user* and p as the distribution of the *population*'s tags. In line with Sec. 3.3, we introduce a *tag suppression* rate $\sigma \in [0,1)$, which is the ratio of suppressed tags to total tags. Thus, we define the user's *apparent* tag distribution s as $\frac{q-r}{1-\sigma}$ for some suppression policy $r = (r_1, \ldots, r_n)$ satisfying $0 \leqslant r_i \leqslant q_i$ and $\sum r_i = \sigma$ for $i = 1, \ldots, n$.

Inspired by the privacy criteria proposed in [18], we use an information-theoretic quantity to reflect the intuition that an attacker will be able to compromise user privacy as long as the user's apparent tag distribution diverges from the population's. Specifically, we consider the Kullback-Leibler (KL) divergence [19], which may be interpreted as a measure of discrepancy between probability distributions. Accordingly, we define *privacy risk* as the KL divergence between the apparent distribution and the population's, that is,

$$\mathrm{D}(s \,\|\, p) = \mathrm{D}\left(\frac{q-r}{1-\sigma} \,\middle\|\, p \right).$$

Supposing that the population is large enough to neglect the impact of the choice of r on p, we define now the *privacy-tag suppression rate* function

$$\mathcal{R}(\sigma) = \min_{\substack{0 \leqslant r_i \leqslant q_i \\ \sum r_i = \sigma}} \mathrm{D}\left(\frac{q-r}{1-\sigma} \middle\| p\right), \tag{1}$$

which characterizes the optimal trade-off between privacy (risk) and tag suppression rate, and formally expresses the intuitive reasoning behind tag suppression: the higher the tag suppression rate σ, the lower the discrepancy in terms of the KL divergence between the apparent distribution and the population's, and the lower the privacy risk. In addition, this formulation allows us to describe the functional block *suppressing tag generator* in Sec. 3.3. Namely, this module will be responsible for solving the optimization problem in (1).

Our privacy criterion in the formulation of the privacy-tag suppression rate function is justified, on the one hand, by the arguments in the literature advocating entropy maximization [20], as our privacy measure may be regarded as an extension of Shannon's entropy [19], and on the other hand, by the rationale behind divergence minimization and information gain minimization [18].

5 Concluding Remarks

There exists a large number of proposals for privacy preservation in the semantic Web. Within these approaches, tag suppression arises as a simple strategy in terms of infrastructure requirements, as users need not trust an external entity. However, this strategy comes at the cost of processing overhead and the semantic loss incurred by suppressing tags.

Our main contribution is an architecture that implements tag suppression in the semantic Web. The proposed architecture helps users refrain from proposing certain tags in order to hinder attackers in their efforts to profile users' interests.

We describe the implementation details of our architecture. Specifically, the core of the system is a module responsible for calculating a tag suppression policy. The system uses this information to warn the user when their privacy is being compromised and it is then for the user to decide whether to eliminate the tag or not.

We present a mathematical formulation of the optimal trade-off between privacy and tag suppression rate in the semantic Web, which arises from the definition of our privacy criterion.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scient. Amer. (May 2001)
2. Michlmayr, E., Cazer, S.: Learning user profiles from tagging data and leveraging them for personal(ized) information access. In: Proc. Workshop Tagging and Metadata for Social Inform. Org. Workshop in Int. WWW Conf. (2007)
3. John, A., Seligmann, D.: Collaborative tagging and expertise in the enterprise. In: Proc. Col. Web Tagging Workshop WWW. (2006)
4. Gruber, T.R.: A translation approach to portable ontology specifications. Knowl. Acquisition **5**(2) (1993) 199–220
5. Brickley, D., Guha, R.V.: RDF vocabulary description language 1.0: RDF schema. W3c recommendation, W3C (February 2004) http://www.w3.org/TR/2004/REC-rdf-schema-20040210/.
6. OWL Working Group, W.: OWL 2 Web Ontology Language: Document Overview. W3C Recommendation (27 October 2009) Available at `http://www.w3.org/TR/owl2-overview/`.
7. Mcdonald, A.M., Reeder, R.W., Kelley, P.G., Cranor, L.F.: A comparative study of online privacy policies and formats. In: Proc. Workshop Privacy Enhanc. Technol. (PET), Berlin, Heidelberg, Springer-Verlag (2009) 37–55
8. Jensen, C., Potts, C., Jensen, C.: Privacy practices of internet users: Self-reports versus observed behavior. Int. J. Human-Comput. Stud. **63**(1-2) (2005) 203–227
9. Kagal, L., Finin, T., Joshi, A.: A policy based approach to security for the semantic web. In: Proc. Int. Semantic Web Conf. (2003) 402–418
10. Kagal, L., Paolucci, M., Srinivasan, N., Denker, G., Finin, T., Sycara, K.: Authorization and privacy for semantic web services. IEEE J. Intelligent Syst. **19**(4) (2004) 50–56
11. Elovici, Y., Shapira, B., Maschiach, A.: A new privacy model for hiding group interests while accessing the web. In: Proc. ACM Workshop on Privacy in the Electron. Society, ACM (2002) 63–70
12. Shapira, B., Elovici, Y., Meshiach, A., Kuflik, T.: PRAW – The model for PRivAte Web. J. Amer. Soc. Inform. Sci., Technol. **56**(2) (2005) 159–172
13. Frakes, W.B., Baeza-Yates, R.A., eds.: Information Retrieval: Data Structures & Algorithms. Prentice-Hall (1992)
14. Kuflik, T., Shapira, B., Elovici, Y., Maschiach, A.: Privacy preservation improvement by learning optimal profile generation rate. In: User Modeling. Volume 2702/2003 of Lecture Notes Comput. Sci. (LNCS)., Springer-Verlag (2003) 168–177
15. Howe, D.C., Nissenbaum, H.: TrackMeNot (2006)
16. Toubiana, V.: SquiggleSR (2007)
17. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM **18**(11) (1975) 613–620
18. Rebollo-Monedero, D., Forné, J., Domingo-Ferrer, J.: From *t*-closeness-like privacy to postrandomization via information theory. IEEE Trans. Knowl. Data Eng. (October 2009)
19. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Second edn. Wiley, New York (2006)
20. Jaynes, E.T.: On the rationale of maximum-entropy methods. Proc. IEEE **70**(9) (September 1982) 939–952