Inferences in the Web

Petteri Kääriäinen
Helsinki University of Technology
petteri.kaariainen@tkk.fi

Abstract

The Internet is a huge information database. With this information it is possible to find inferences which are not obvious to everyone who uses Internet. This paper presents two cases how information can be combined to find inference channels. It also describes salient technologies in a semantic web and how they affect the inference problem.

KEYWORDS: Inference detection, Semantic web, Privacy

1 Introduction

The web contains massive amounts of information. This information is mainly public but some portion of it contains sensitive data which is confidential. There is also data which alone is not sensitive but when combined with other insensitive or sensitive pieces of information it may reveal private things about an individual. This is what we call inference. For example, if someone writes a public blog anonymously using a nick name and publishes his career, first name and city writer's identity may be easily discovered via Google. First query result may be for example, a web-site of the writer's company or an article in a local newspaper. Thus we can say that Google found an inference channel between an anonymous nick name and a real identity.

There are some algorithms for detecting inferences on databases but they are not practical on Internet. This is because of the Internet is so huge[4]. These conventional algorithms propose that association rules are available which describe possible inferences. Another main reason why these algorithms are not applicable to Internet is that no one party controls the whole web so we can't remove inference channels as easily as we could in a database case[7].

Today the traditional Internet is giving way to what is being called the semantic web. The semantic web is an extension of the current Internet which gives attributes to information and so it makes easier to search for accurate information. At the same time when the new web makes it easier to search for information it also becomes easier to make inferences based on information on the Internet. Technologies such Resource Description Framework[1] and Web Ontology Language[14] are two core components of the semantic web.

This paper surveys which kind of inference risks exist in the Internet and how it affects the risks if the semantic web technologies become widely used. First, I give an overview about the semantic web and it's technologies. I also list the main information sources which could be utilized when searching potential inferences. Section 4 presents two papers which have solutions to find possible inference channels before documents are published on the Internet.

2 The Semantic Web

The traditional Internet is designed to be easily readable by humans. The problem is that it is hard to find semantics of information automatically in the traditional web. Information is not categorized thus search engines cannot find for example, which part of an address is the street name and which part is the town. Hence a search engine can query all documents which contains the number 24 but the engine does not know the document tell about the tv-series 24 or for instance some event where were 24 participants. Person who is looking for information about cast of tv-series can do nothing with some event info.

In contrast to the traditional web in the Semantic web, information is presented in XML format thus it is easy to process. XML also allows to add custom tags combined with XHTML tags. The namespace of the tag defines the meaning of every tag. The semantic web requires techniques to process information easily. The World Wide Web Consortium (W3C) has developed two solutions: Resource Description Framework and OWL Web Ontology Language for the semantic web both of which expands XHTML language. The ability to expand XHTML with other languages makes it possible to add necessary features to the web when a new need arises thus the benefit is that we do not have to know all needs beforehand.

2.0.1 The Resource Description Framework

Resource Description Framework (RDF) is an abstract framework to present information[3]. The purpose of RDF is to make it possible to easily process information from the Internet and it also helps interworking of applications. RDF is an extensible framework. That is the reason why RDF does not guarantee that information is logically accurate thus programmers have to ensure it[3]. RDF is defined with an abstract syntax but there is also a XML specification available, which is in use in the semantic web[1].

RDF is presented with triples consisting of a subject, a predicate and a object. For example, in a sentence "*Apple is red*" *apple* is the subject, *is* is the predicate and *red* is the object. To create RDF graph we need a set of these triples. RDF data types are compatible with XML Schema [11] data types but there can also be data types which are not declared in XML schema. For example, Creative Commons Licenses

can be published using RDF based on ccREL[16] language used for example on the home page of White house[20].

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/
      02/22-rdf-syntax-ns#"
    xmlns:sem="http://www.niksula.hut.fi/
      pkaariai/seminaari#">
    <rdf:Description
      rdf:about="http://www.niksula.hut.fi/
        ~pkaariai/internetworking seminar">
        <sem:author>Petteri Kääriäinen
          </sem:author>
        <sem:dl>2009-03-19</sem:dl>
        <sem:organization>TKK
          </sem:organization>
    </rdf:Description>
</rdf:RDF>
```

Above is an example of how RDF could be utilized to describe properties of this document (namespace definitions are for layout reasons on two lines). The example uses two namespaces rdf and sem. Sem namespace is a custom namespace used only on this paper. It shows how easy it is to extend RDF. Example contains three RDF triples, for example http://www.niksula.hut.fi/ pkaariai/internetworking seminar, http://www.niksula.hut.fi/pkaariai/seminaari#author, "Petteri Kääriäinen" in subject, predicate, object order.

If some day all documents in the Internet are described with same kind of RDF document as in the example, search engines could easily find all documents which are made by the same author. After collecting all documents written by the same author it would be easy to collect all organizations where the author has worked. This is very short example and in the real world RDF could contain much more meta data and reveal more crucial information about the author than work history.

2.0.2 OWL Web Ontology Language

Web Ontology Language (OWL) adds rules to information which helps to understand relations of data. OWL and ontologies in general can classify things, describe relations of things and add attributes to things[9]. If things are classified consistently to categories which are defined by OWL it is easy to automatically compare things between different web pages and make consistent decisions based on categorized data.

OWL consists of three sublanguages which are from least powerful to the most powerful: OWL Lite, OWL DL and OWL Full[14]. Each language is an extension to the previous language thus a programmer may start with OWL Lite and then moves further if it is too constraint for programmer's purposes. Every OWL document is valid RDF document and every RDF document is valid OWL Full document[14].

With OWL things can be classified for example into people, owners, renters, cars, pick-ups, and vans. OWL could also define that all owners and renters are people and pick-ups and vans are subclasses of cars. If all car rental web sites would use these same categories it would be easy to do a portal which compares rent rates of different rental firms and tells renter which is the cheapest place to rent a van. In the traditional web, it is also possible to do web sites which automatically collects prices from numerous web sites but the search engine has to be modified differently for every site because there is no way of knowing which word means price and which word model of a van.

Another use for OWL is to give semantics to multimedia. That may create new inference channels because without semantics it is harder extract meanings "inside" the pictures than text. OWL could describe pictures common way so search engines could search "inside" pictures and thus we could find inferences also in pictures.

2.0.3 Friend of a Friend and XHTML Friends Network

Friend of a friend (FOAF) is an RDF and OWL based language to describe information about people and relations between them[2]. FOAF describes properties of people (name, email, chatid, address) in machine understandable form and makes it easy to process data which contains personal information.

One interesting characteristic of FOAF is that it contains property foaf:mbox_sha1sum which is SHA1 hash which is calculated from mailto URI of user. Mailbox address can be unique identifier for person but for spam reasons it is not good idea to always publish your mail address. But calculating hash over it and publishing it gives an unique identifier but does not reveal the real email address.

With FOAF it is easy to draw graphs which describe relations of people. There exist automatic tools to create FOAF data[6] thus using FOAF does not need high technical skills.

```
<foaf:Person>
  <foaf:name>John Smith</foaf:name>
  <foaf:mbox
    rdf:resource="mailto:john@example.com"/>
  <foaf:knows>
    <foaf:Person>
      <foaf:name>Mary Smith</foaf:name>
      <foaf:mbox
        rdf:resource="mailto:mary@example.com"/>
    </foaf:Person>
  </foaf:knows>
</foaf:Person>
```

The example above shows how FOAF can be used to attach friends of a person. With recursive search it is possible to draw graphs which show relations of a large community.

XHTML Friends Network (XFN)[21] purpose is similar like FOAF. It is little bit easier to attach web page because it use only rel attributes to define friends of person. If user adds a link to his home page with attribute *rel="friend met colleague"* it shows that link targets to person who is co-worker of site owner. For example, Wordpress and Twitter use XFN to show connections between users.

Google provides tools for finding connections which are described with FOAF and XFN[19]. Google indexes all sites which contains FOAF or XFN data and with the tool[19] anyone can easily check own connections. If a user registers to a new service, the service may check user's connections and ask if the user wants to invite his or her friends to use the service.

3 Data Sources

In the Internet, the three most potential data sources which may reveal secret information are: blogs, social web-sites

and home pages. Blogs and home pages are normally without any protection thus everyone can read all information from these sources. The most efficient method to search data on these public sources is to use a search engine for example, Google or Yahoo. Search engines also obtain information from social web-sites but at least some portion of that information is protected by passwords and as a result search engines cannot find it. Some sites also use robots.txt which tells search engines that they are not allowed to index content of that site. This is a soft limit so there are no technical restrictions that search engines could not index that site but normal search engines respect robots.txt.

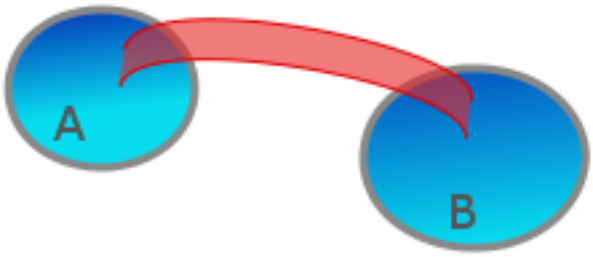

Figure 1: An inference channel

3.1 Blogs

The main idea of blogs is to publish an author's personal ideas to as large number of readers as possible. Some blog writers write with their real names but some bloggers use nick names to ensure anonymity. In section 5.1, we show which problems can still occur if there is inference channels even though writer seems to be anonym and how to lower the inference risk[8].

One of biggest blog-sites is Blogspot which is Google's service. Blogspot gives free blog space for every registered user. Writing a blog is very easy because all the user needs is a network connection and a web browser. Hence a user does not need to install any extra applications on his machine. Counter side of this easiness is that the user finds it easy to add information to the site without thinking is the content good to publish to everyone or not.

3.2 Social Web-sites

Social web-sites had grown very fast on recent years. For example, Facebook has over 175 million active users and average user has 120 friends in his profile[17]. Users are very active and visit daily on the site. Over 18 million users update their status daily[17]. Status is couple words which describe what the user is doing. For example, "Petteri is going to movies" could be my status. Status updates sound very innocent thing but they may reveal lots of sensitive information especially if the user mentions other persons names who are doing same activity with the user.

Other massive social website is linkedin which concentrates on building social networks which combines work carriers together. On linkedin web-site users see what is a person's relation to some other person. Myspace, Flickr and Habbo are other examples of popular social websites.

3.3 Home Pages

Personal and corporate home pages may both give critical information such as phone numbers or addresses which can be the link between private and public data. For spam reasons many companies do not publish direct links to employees e-mails anymore but normally sites give a pattern. For example, firstname.lastname@company.com which gives a hint how to combine employees name and the pattern to get the real email address.

Public home pages should never contain any information which is not intend to be public. That public information may help to create other inference channels which reveal some private data. For instance a politician may publish names of her or his children and schools where they study on her or his campaign page. Based on the names and the schools a nasty reporter may find out children's student party pictures, where they are in some unconventional situation, and create a scandal.

4 Inference Detection Methods

This chapter presents common terms for finding inferences and presents two cases which present solutions to finding inferences.

4.1 Common Terms

Inference detection algorithms are based on finding **association rules** $A \Rightarrow B$ [4]. In this formula, A is a collection of key words and B is a collection of sensitive words. Collections A and B are two distinct sets like in figure 1. If we want to protect sensitive data B, be we have to somehow break the association rule. If both A and B are already publicly available on the Internet it is almost impossible to break the association rule but before A or B are published it can be possible.

Figure 1 shows how an association rule from A to B makes an inference channel(red bridge in the figure) between document group A and B. In some situations, it may be enough to remove one word from group A to break the inference channel.

Different association rules can be compared with **support** and **confidence** values[5]. Confidence can be calculated by formula $\frac{|A \cup B|}{|A|}$ where $|A \cup B|$ is amount of documents which contains word from both groups A and B and $|A|$ is amount of documents which contain a word from group A. Confidence tells how confident association rule $A \Rightarrow B$ is. Support can be calculated by formula $\frac{|A \cup B|}{|N|}$. $|N|$ is total amount of documents. Support describes frequency of association rule.

One way to compare relevance of inference results is term frequency-inverse document frequency value (**tf-idf**). In this value tf describes a local relevance of a given term in one document. It is calculated by dividing the term count by total count of all terms in a document. Idf describes how often the term is presented in whole document set.[15] Idf

can be calculated by formula $log(N/n_j) + 1$ where N is the total amount of documents and n_j is the amount of documents where the term is found. In web-based inference detection the document set is the whole Internet so we cannot know the accurate document total count so idf value has to be estimated.[10]

4.2 Web-based inference detection

Staddon et. all [13] have invented tools which help to search unintended inferences in documents before they are published. These tools use Google search to search possible inferences. The tools first extract the main keywords from documents to be published. To select the most important keywords they use tf-idf value to compare a relevance of every word in the documents. Then the tools make queries with these keywords from a reference corpus (for instance Internet) to find more relevant keywords. The purpose of reference corpus is to give as much information about the topic as possible. After collecting all keywords, tools return potentially dangerous inferences.

To show how hard it is to avoid unintended inferences Staddon et al. [13] present an example about FBI document[18] which was published on the Internet after redacting all trivial identification terms such as names and social security numbers. The intention was to keep person's identity secret. The problem was that even if many words were redacted some remained and they formulated association rules that made it possible to know that the FBI document was a redacted profile about Osama bin Laden.

To make redaction better Staddon et al.[13] removed all references to places, dates near September 11, 2001, and citations. After this manual redaction they calculated tf-idf values for all unredacted words and selected words which have top tf-idf value for next step. They made Google queries with these selected words and reviewed the search results. After the review, they removed from the document the keywords which made association rules. They did this process iteratively as long as any association rules exists. Result was that they redacted almost thirty words more than FBI had redacted.

4.3 Detecting Privacy Leaks Using Corpus-based Association Rules

Chow et al.[4] has further developed the algorithms of Staddon et al. [13]. They use simpler algorithms to get massively better performance than Staddon et al. algorithms. For example, evaluating single inference is 100 times slower with Staddon et al.[13] algorithms than with Chow et al.[4] algorithms.

The main portion of association rule mining focuses on finding association rules with high support and high confidence. In inference detection we are interested in every association rule which might reveal secret information. That is why the algorithm of Chow et al.[4] collects also association rules which have only large support but confidence does not matter so much. It is smaller problem if we redact some document which does not contain secret information than if we do not redact document with confidential data.

While Staddon et al.[13] algorithm has to process all search engine results this algorithm lets search engine do the job. Confidence of two terms is estimated by two search engine queries. If we are calculating confidence for $A \Rightarrow B$ thus we first query A and then A and B. Then we divide query result count for A by result count for query result A and B. By this way search engine makes biggest work and we have to do only basic dividing calculations. This is called PMI-IR estimate for confidence.

Chow et al.[4] did test where they tried to find inferences between some medical terms and HIV. They collected 70 top rated terms and asked some medical expert to check if inferences are correct. The expert said that 53 of 70 were correct implications. When investigating results the authors noticed that also some other terms than which were in group of 53 were also highly inferencing but medical expert did not notice inference between term and HIV. This shows clearly how hard it is to find inferences manually.

5 Solutions

5.1 How to Hide Identity on Blogs

Frankowski et al. [8] show how user's privacy is not assured even thought a user seems to be anonym. The paper contains test case where writer's identity is solved based on writer's movie ratings. Purpose of the paper is to examine how easily writer can be identified and which kind of methods there exist to make identifying harder.

The authors defined a k-identification concept. K-identification means that if the user who has made movie ratings is k:th first in result list of the identification algorithm, user is k-identified. K-identification is easier if the user has rated rare rated movies because in that case result count is already small and algorithm has only few choices to pick.

The authors got the best result if blog posts contained numerical ratings for movies. If there was an algorithm which could invent ratings from context of blog text it would also be useful. With ratings they get almost 50% of users 1-identified (i.e. actual writer was first hit).

The authors also examined some tactics to mislead detection algorithms and that way to hide writer's identity. The authors tested to suppress rarely rated movies from review dataset to hide writer's identity. Test showed that dataset should be suppressed by 88% of all data thus suppressing is not very efficient method because it hides also so much useful information.

After testing suppressing results the authors tested how efficient it could be to mislead detection algorithms with false ratings. Earlier the Authors find out that rarely rated movies are the most identifying items but in misleading case user should add ratings about most popular movies which expand the set of possible matches on the user most. This shows that user can quite easily lower identification risk by adding few extra ratings about the most popular movies.

5.2 Platform for Privacy Preferences

The Platform for Privacy Preferences Project (P3P) provides user agents information about privacy policies of web site in

standard format. It does not assure that a site fulfills policies but P3P makes it easier to compare what sites promise to do. P3P specifies XML Schema for privacy requirements of site so users web browser may easily automatically check if sites policy match to browsers predefined policy.[12]

P3P specification defines two information categories: "identified" data and "non-identifiable" Data. Identified data contains all data which can easily be attached to some individual. Non-identifiable data is data which is anonymizied. For example, a list may contain only first names of users so actual identification is impossible with this list.

Server may return P3P policy of site in HTTP-headers or with link tag. In both cases policy is defined by linking to XML document which describes the policies of site. Each policy may describe for instance which kind information site collects and how user can get it.

P3P does not prevent anyone using private data for making inferences. If user's predefined privacy policies are strict enough and sites honour P3P policy the user hopefully does not give private data which may allow making inferences. Thus P3P is not the most important tool to prevent inferences but it gives little help to a user who wants to protect his or her privacy.

6 Discussion

The combination of traditional technologies and new semantic web technologies makes it easier to create inferences. Aim of the semantic web is to give semantics to information and that way make automatic data queries more accurate. Thus algorithms in chapter 4.3 give more accurate inferences if search engines give more accurate answers to search queries. At the same time if inference detection is easier it is also easier to be protected from unintended inferences because users may find them easier before publishing documents.

Wider use of RDF and OWL makes function of search engines easier. Search engines can easily find relation between different web sites and can so independently make inference detection. Thus search engines can someday answer questions like "Who is the son of the president of Finland?". In that day Google is a big inference detection site.

As long as the main part of documents in the Internet are without RDF and OWL advantage of using these technologies is not very clear. If there are not services which exploit new technologies it does not give any value to use the technologies. So it is long and slow process until every site in the Internet is made with RDF and OWL. For other hand using RDF and OWL does not harm for traditional services thus there are no reasons why not use the technologies.

7 Conclusion

This paper describes the main concepts of inference detection in the traditional and the semantic web. It presents the main concepts of OWL and RDF, which are the two main technologies in the semantic web. When the semantic web comes to wider use these technologies allow easier automatic data processing. Thus it may be even harder to avoid unintended inferences as nowadays.

Section 4 describes two cases of how people can use a web to find inferences. The cases show how powerful tool a search engine is and how hard it is to break inference channels. If that kind tools can be modified so that every web user can easily use them everybody could try to check unintended inferences before they publish documents to the Internet.

The paper also covers the case about blogs and shows how easily a user can be detected based on his movie comments. Case shows that making fake movie ratings is the most powerful way to mislead inference detection tools.

Like this paper shows there exist some tools which help to protect user's privacy from inferences but there is no one silver bullet which could solve the whole problem. Especially when the semantic web technologies allow very fast automatic information processing in the whole Internet to make inferences, the protection may be almost impossible.

References

[1] D. Beckett. RDF/xml syntax specification (revised). W3C recommendation, W3C, Feb. 2004. http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/.

[2] D. M. L. Brickley. Foaf vocabulary specification 0.91. Technical report, Foaf, Nov. 2007. http://xmlns.com/foaf/spec/.

[3] J. J. Carroll and G. Klyne. Resource description framework (RDF): Concepts and abstract syntax. W3C recommendation, W3C, Feb. 2004. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.

[4] R. Chow, P. Golle, and J. Staddon. Detecting privacy leaks using corpus-based association rules. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 893–901, New York, NY, USA, 2008. ACM.

[5] E. Dasseni, V. S. Verykios, A. K. Elmagarmid, and E. Bertino. Hiding association rules by using confidence and support. In *IHW '01: Proceedings of the 4th International Workshop on Information Hiding*, pages 369–383, London, UK, 2001. Springer-Verlag.

[6] L. Dodds. Foaf-a-matic, 2009. http://www.ldodds.com/foaf/foaf-a-matic.html.

[7] C. Farkas. *Data Confidentiality on The Semantic Web: Is There an Inference Problem?*, pages 73–90. Idea Group Inc., 2005.

[8] D. Frankowski, D. Cosley, S. Sen, L. Terveen, and J. Riedl. You are what you say: privacy risks of public mentions. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 565–572, New York, NY, USA, 2006. ACM.

[9] J. Heflin. OWL web ontology language use cases and requirements. W3C recommendation, W3C, Feb. 2004. http://www.w3.org/TR/2004/REC-webont-req-20040210/.

[10] M. Klein and M. L. Nelson. A comparison of techniques for estimating idf values to generate lexical signatures for the web. In *WIDM '08: Proceeding of the 10th ACM workshop on Web information and data management*, pages 39–46, New York, NY, USA, 2008. ACM.

[11] A. Malhotra and P. V. Biron. XML schema part 2: Datatypes. first edition of a recommendation, W3C, May 2001. http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/.

[12] M. Schunter and R. Wenning. The platform for privacy preferences 1.1 (P3P1.1) specification. W3C note, W3C, Nov. 2006. http://www.w3.org/TR/2006/NOTE-P3P11-20061113/.

[13] J. Staddon, P. Golle, and B. Zimny. Web-based inference detection. In *SS'07: Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, pages 1–16, Berkeley, CA, USA, 2007. USENIX Association.

[14] F. van Harmelen and D. L. McGuinness. OWL web ontology language overview. W3C recommendation, W3C, Feb. 2004. http://www.w3.org/TR/2004/REC-owl-features-20040210/.

[15] H. C. Wu, R. W. P. Luk, K. F. Wong, and K. L. Kwok. Interpreting tf-idf term weights as making relevance decisions. *ACM Trans. Inf. Syst.*, 26(3):1–37, 2008.

[16] *ccREL*, 2009. http://wiki.creativecommons.org/CcREL.

[17] Facebook - statistics, 2009. http://www.facebook.com/press/info.php?statistics.

[18] Redicted osama document. http://www.judicialwatch.org/archive/2005/osama.pdf.

[19] Social graph api. http://code.google.com/intl/en/apis/socialgraph/.

[20] White house - copyright notice, 2009. http://www.whitehouse.gov/copyright/.

[21] Xhtml friends network, 2009. http://gmpg.org/xfn/.