Chapter 5

Search Privacy Through Personal Control

"I am the master of my fate, I am the captain of my soul"
William Ernest Henley

5.1 Introduction

In chapter 3, we looked at each PET category and asked two questions:

- Does it have the ability to enhance a user's search privacy?
- Is the search engine still a threat to the user's search privacy?

Of the categories analysed, some showed potential but we were unable to provide conclusive answers. In the previous chapter, we investigated the anonymity category in an effort to determine if it can enhance a user's search privacy as well as eliminate the threat of the search engine. Having control over one's identity does contribute to enhancing search privacy, if the search engine does not know who is submitting the query then an accurate search profile cannot be constructed. However, we then showed that the search engine may have a motive for determining who is submitting the query (in the case of thwarting click fraud) in which case anonymity may not be guaranteed. As a result, while PETs in the anonymity category do have the potential to enhance a user's search privacy, they do not eliminate the search engine as a threat to the user's search privacy.

In this chapter, we turn to the category of personal control. Highlighted as "the use of technology to ensure that an individual's personal information is only used in a manner commensurate with the individual's privacy policy", we will determine if PETs in this category can adequately address the issue of search privacy. To evaluate this PET category, we consider a technology that embodies its definition: the Platform for Privacy Preferences (P3P) [32, 31].

P3P allows Web sites to declare their intentions as far as privacy-related matters are concerned. Users of the Web, having explicitly configured their own privacy preferences, are then able to browse through Web sites that have P3P policies with the confidence that their privacy will not be violated.

Although P3P has made an important contribution to the field of privacy, there are two problems:

- Trust: how does one trust what a Web site has published in its P3P policy? Whilst a legitimate Web site has every reason to publish a P3P policy stating its intentions, there is obviously no incentive for an illegitimate Web site to publish its suspicious privacy practices in a P3P policy. Therefore, if an illegitimate Web site publishes a P3P policy at all, why should it not also publish a policy that states its intention of respecting every user's privacy, despite the fact that it will do nothing of the sort?

- Proxies: although the way in which users and their associated P3P agents interact with a P3P compliant Web site has received much attention, the surreptitious role played by the Web proxy server and its impact on P3P has been neglected. We must consider the scenario involving a Web proxy that is situated between a Web user and a P3P compliant Web site.

Without addressing the issue of trust and proxies in P3P, we can go no further in its evaluation as a technology in the personal control category. This chapter is therefore structured as follows: in the next section we briefly look at the design of P3P before dealing with the issue of trust in section 5.3. Section 5.4 then deals with proxies and their place in P3P. This chapter is concluded in section 5.5.

5.2 P3P

P3P has been the focus of much research and criticism [29, 31, 32, 65, 114] and at its most basic level allows Web users (with their associated P3P agents) to automate the protection of their privacy. Web sites publish P3P policies clearly describing their intentions so that Web users can compare these policies to their own set of privacy preferences. Provided that the P3P policy published by the Web site is acceptable, the user may continue to make use of services offered on the Web site. The P3P policy of a Web site typically states what information it may require from a user during a session as well as what it intends to do with the information. The gist of a user's privacy preferences will describe what type of personal information the user is willing to provide, how long it may be stored, for what purpose this information may be used and to whom it may be given.

P3P policies are published in a standardised XML format. Although it is quite possible to peruse them manually, the process of going through a policy is typically automated through the use of an agent that is familiar with the user's preferences. An example of a P3P policy implemented by a Web site is as follows:

```
<POLICIES xmlns="http://www.w3.org/2002/01/P3Pv1">
  <POLICY discuri="http://example.com/privacy.html"
          name="An example of a policy">
    <ENTITY>
      <DATA-GROUP>
        <DATA ref="business.name">
          Example Web Site
        </DATA>
      </DATA-GROUP>
    </ENTITY>
    <ACCESS>
      <nonident/>
    </ACCESS>
    <STATEMENT>
      <CONSEQUENCE>
        We keep standard access logs but everything
        is anonymized.
      </CONSEQUENCE>
      <NON-IDENTIFIABLE/>
    </STATEMENT>
  </POLICY>
</POLICIES>
```

Note the `<NON-IDENTIFIABLE/>` tag, this indicates that information stored by the Web site cannot be traced back to the individual.

P3P policies can be located via Policy Reference Files. These files are responsible for defining which P3P policies apply to certain URIs. Cranor *et al* [31] specify four methods that may be used to find the Policy Reference File that will then be used to look up the appropriate URI:

- It may be located in a *well known* location, for example, http://example.com/w3c/p3p.xml
- A document may indicate a policy reference file through an HTML *link tag.*
- A document may indicate a policy reference file through an XHTML *link tag.*
- The reference file may be indicated through an HTTP header.

5.3 Trust and P3P

Fortunately, the problem of trust is not new to the Web. In the past, environments such as online trading systems needed a mechanism to establish trust between a buyer and a seller that were relatively unknown to one another. This was achieved with the introduction of a Reputation System. Users of the system were granted reputations that could be built up or broken down depending on how they behaved.

In this section, we discuss the problem of associating trust with Web sites within the context of P3P, that is, we identify the need for a Reputation-based System in P3P. We analyse Reputation Systems in detail and discuss the potential problems of extending P3P with a Reputation-based System. As a result we offer an alternative to conventional Reputation Systems in the form of a trusted third party.

5.3.1 Reputation Systems

The trust between a buyer and a seller in a face-to-face sale can be established through several means. Before paying the seller, the buyer must be able to determine whether the goods for sale are, at the very least, genuine. If the goods are acceptable, the buyer must next determine if the seller is trustworthy, i.e., will the goods be delivered after payment has been made? Of course, the transaction is sure to run with fewer problems if both parties have credible reputations and trust each other.

Since the normal elements associated with a face-to-face sale (holding or seeing the goods for example) are not present in an online environment, other mechanisms must be relied upon to establish some degree of trust between the two parties involved before any transaction can occur. This degree of trust depends largely on the reputation of the entities involved. Josang et al [73] point out that although there is a close link between reputation and trustworthiness, it is evident that there is a clear and important difference. In this work, we adopt the definition Josang et al have chosen from the Concise Oxford Dictionary: *reputation is what is generally said or believed about a person or thing's character or standing.*

What is said about an entity, either good or bad, is usually taken from previous dealings and interactions with that entity. Reputation Systems are responsible for gathering these opinions and forming quantifiable reputations for users of a particular system. They serve as a mechanism to establish a degree of trust between participants in the system and do this by collecting, distributing and aggregating feedback regarding participants' past behaviour [102].

As an example, eBay is briefly examined as one of many online trading business-to-consumer and consumer-to-consumer environments. eBay's Reputation System is called the Feedback Forum [102]. Opinions about sellers and buyers come in the form of feedback which is given voluntarily by other buyers and sellers after a sale. Feedback can be either good (+1), bad (-1) or neutral (0). Neutral feedback is also considered part of a feedback system. Feedback points are computed by taking the number of unique users who left good feedback and subtracting the number of unique users who left negative feedback [101]. One seller is said to have a better reputation than another seller if the ratio of positive to negative feedback is better.

Consider sellers Alice and Bob. Alice conducted 500 transactions with only one negative feedback. Because of the ratio of negative to positive feedback Alice has a good reputation. Bob also conducted 500 transactions but received 50 negative feedbacks. Although 90% of Bob's feedback is positive and he is probably a trustworthy seller, he does not appear to be as trustworthy as Alice. In other words, Alice has a better reputation than Bob.

In online trading systems, a buyer will most likely have no idea who the seller is or what the condition of the goods being sold are (other than the condition the seller claims them to be in). With a Reputation System in place, the potential buyer has an accurate idea of how trustworthy the seller is based on their reputation.

Reputation Systems can be divided into two types of architectures: Centralised Reputation Systems and Distributed Reputation Systems [73]. Centralised Reputation Systems have a central authority responsible for the construction of a reputation for a user. A Distributed Reputation System is one in which agents participating in the system are responsible for constructing a user's reputation (e.g. Peer-to-Peer Reputation System) [63, 3, 45].

5.3.2 The Need for Trust in P3P

In section 5.3.1 the critical role performed by Reputation Systems within the context of online trading systems was highlighted. With a Reputation System in place, buyers in the system can easily associate varying degrees of trust with sellers who are generally unknown to them. Without a Reputation System, buyers have no immediate mechanism at their disposal to assist them in deciding who may be trustworthy and who may not. P3P is no exception to the problem of determining who to trust. A potential user of a Web site with a P3P policy may have had no experience with the Web site in the past and as a result has very little from which to judge whether it can be trusted and the Web site's adherence to their published P3P policy.

The intention of P3P is to allow potential users of a Web site to make informed decisions regarding their privacy needs. But how does a user decide whether or not to trust the Web site in question? This Web site may be actively participating with a syndicate that collects personal details (such as full names and email addresses) for the purposes of distributing spam. Despite its unscrupulous approach towards the privacy of its users, the Web site may have a sterling P3P policy that indicates nothing of the sort.

Whilst P3P makes it easier for users to determine if the privacy policy presented by the Web site they are visiting suits their privacy needs, it does not guarantee that the policy presented by the Web site is legitimate. To be precise, there is no guarantee that the actual privacy practices of the Web site are indeed as claimed in its P3P policy.

With the introduction of a Reputation System, trust in a Web site would depend not only on whether or not its P3P policy is acceptable, but also on the reputation associated with the Web site in question. Obviously, a Web site with a bad reputation is less likely to uphold its privacy promises than a Web site with a good reputation.

Implementing a Reputation System within the context of P3P implies several challenges. To better appreciate these challenges, we first examine and discuss the prerequisites of a Reputation System.

5.3.3 Prerequisites of a Reputation System

Various Reputation Systems are available on the Internet: eBay[1], Amazon[2], Yahoo[3] and Bizrate[4]. Common usage includes rating recommendations, articles, buyers and sellers. They are also used by businesses to rate customers and vice versa. Essentially, a Reputation System's main function is the collection and distribution of feedback.

According to Zacharia and Maes [126], one method of building a Reputation System involves the creation of a central agency that records the recent activities of users in the environment. A centralised system monitors users' activities and provides a summary thereof to other users. The centralised system also accepts feedback or comments in order to compute reputation points for each user.

To evaluate the effectiveness of a Reputation System we must examine its properties. Resnick et al [102] list the following three requirements that a Reputation System must meet in order to operate effectively:

- Long-lived entities that inspire an expectation of future interactions
- Capture and distribution of feedback regarding current and past interactions (this information must be visible in the future)
- Use of feedback to guide trust decisions

[1] eBay, http://ebay.com

[2] Amazon Auction, http://auctions.amazon.com

[3] Yahoo! Auction, http://auction.yahoo.com

[4] Bizrate Online Business Ratings, http://www.bizrate.com/ratings_guide/guide.html

In online communities, a low reputation may be the result of a number of reasons (including fraud). Users who receive low reputation points often leave the system and obtain another identity for the purpose of starting afresh with a brand new reputation. This obviously violates the properties of a Reputation System which is built on the main idea that once a user becomes part of the community he or she starts accumulating reputation points. These points are acquired through interactions with other online community users. The users provide feedback to a central reputation system so that other users may have access to such feedback or evaluations to help guide their decisions.

The alternative to a centralised Reputation System is a distributed Reputation system. In this system, users do not submit their feedback to a central authority. Instead, they record their opinion of each experience with other users and provide this information at the request of a "relying party" [73]. When users need reputation information they must find the distributed community of users who have already had direct experience with the specified user. Each user is responsible for computing its reputation points based on interactions with users and private interaction with a target user. Note again that feedback is not submitted to a central authority.

A theme that has received much attention throughout this section is that of a feedback intensive environment. If there is no feedback on a user, reputation points cannot be computed for the user. In other words, constant interactions are required to enable an authority to compute reputation points.

The availability of reputation points also defines a good Reputation System. Users must be able to know that a target user is potentially trustworthy and as a result make a trusted, informed decision. Resnick et al [102] conclude that systems which rely on the participation of large numbers of individuals accumulate trust simply by operating effectively over time. The effectiveness of a Reputation System relies mainly on the honest participation of its community members.

5.3.4 A Reputation System in P3P

We now consider conventional Reputation Systems in a distributed environment like the Web, their properties and why they will fail in P3P. We subsequently propose an alternative to the conventional system as well as a means to visualise (and formalise) reputation.

Conventional Reputation Systems and P3P

Conventional Reputation Systems are built on a centralised platform for the convenience of users who submit and request feedback or reputation points. The Reputation System used in eBay is well studied [73, 101, 102, 126] and several Reputation Systems have been built upon this concept (Yahoo! Auction and Amazon Auction for example). In this section we attempt to model a Reputation System for P3P in an effort to further legitimise the P3P policy presented on a Web site.

A Reputation System within the context of P3P would be responsible for accepting feedback from registered or frequent users of a Web site with respect to the effectiveness of the Web site's privacy policy. For the sake of simplicity feedback provided by users of the Reputation System can be either positive or negative.

Each Web site is obliged to display its reputation points alongside its privacy policy (or provide a link to the central system which contains the score). Thus far, the simple system presented satisfies the three properties of a Reputation System as listed in section 5.3.3.

Unfortunately, the system suffers from two problems. The first problem is the centralised nature of the Reputation System. Since the nature of the Web is distributed, there are a number of difficulties in having a central Reputation System where users can provide their opinion on every Web site's P3P policy.

Even if feedback were only for e-Commerce Web sites, it would still be tremendously cumbersome (the number of e-Commerce sites grows daily). The alternative may be a distributed Reputation System but again the vastness of the Web would make it close to impossible to locate a user to either log or obtain reputation points. The time involved in looking for a user who has a reputation score for a Web site would take up system resources and therefore make it too costly [73].

The second problem with the system presented involves the capturing of feedback. The difference between a trading system and P3P is that there is no tangible product being delivered upon accepting the P3P policy of a Web site. How will a user determine that his/her privacy has been violated by a site that is not adhering to its own P3P policy? We could specify a time period for a user to provide feedback after visiting a Web site, but no definite time period can be specified because the Web site under review may decide to violate its privacy policy (sell a user's details, for example) long after feedback has been provided. Yet another problem is identifying a legitimate user if a definite time period is defined. Registered users can be identified because they have a profile and probably a history. However, unregistered users may visit a Web site once to complete a transaction and are never seen again.

Unorthodox Reputation Systems and P3P

We have discussed some of the problems concerning the usage of conventional Reputation Systems and their application to P3P. As a solution to these issues, we propose a Reputation System that is based on the dependence of third parties. As mentioned earlier, the distributed nature of the Web causes a centralised Reputation System to be infeasible. A Reputation System built and maintained by the same entity that a user wishes to query would not be feasible either. Extending trust to third parties is a mechanism that is already in use today (through services such as BBBOnLine[5]). We argue that trust in a third party will grow as the number of Web sites supported by the third party increases.

It was pointed out earlier that a Reputation System without feedback (opinions) from entities within the system is essentially useless. In the previous section we discussed the difficulty of providing feedback within the context of P3P. However, if a Reputation System is going to be successful in P3P, it has to cater for feedback of some kind.

We therefore offer an alternative to feedback. We believe that a Reputation System based on the two factors presented in this section will suffice to replace the type of feedback one is generally familiar with in a conventional Reputation System. These two factors are maintained by the third party and are as follows:

- The length of time that the Web site has been registered with the trusted third party.

- The number of confirmed incidents reported against a Web site, i.e., incidents where the Web site has been proven to have violated its own P3P policy.

We believe that a reputation with a 5-year history is more credible than a reputation with a 5-month history (not counting exceptional cases). Building up a reputation as a function of time essentially removes the dependence of the Reputation System on direct feedback alone.

Feedback in the form of confirmed reports cannot be constrained by short time limits, for example, it may be the case that months pass before it is found that a Web site has violated the privacy of a user (perhaps in the

[5] A Better Business Bureau Program - http://www.bbbonline.org

form of unsolicited email). In this example, the user would then report this incident to the trusted third party. After an investigation, the report may be confirmed, in which case the reputation of the Web site in question will be affected negatively. A simple formal solution is discussed in 5.3.4.

Preparing and Presenting Reputation

In our proposed Reputation System, the length of time a Web site has been registered with a third party is an important factor in determining its reputation. Simply presenting a user with graphs depicting time lines, the number of incidents and the confirmed number of incidents (although factual, up to date and accurate) may not necessarily be that helpful. We therefore propose an alternative method to calculate and present or visualise a reputation.

The mechanism used for presenting a reputation can be compared to the cross section of a tree trunk. Each ring is indicative of a period of time. The longer the period of time that a site has been registered, the more rings it will have. For our purposes, a ring can be thought of as a one-month period. Note that rings grow outwards (like in a tree) and that the thickness of the rings (the black outline) decreases as the number of rings increases. We refer to this type of reputation mechanism as the Reputation Ring. Figure 5.1 depicts the Reputation Ring of a user in its early stages.

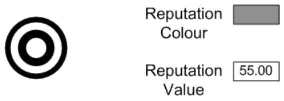

Figure 5.1: Reputation Ring of a new site.

From a presentation point of view, reputation is effectively a measurement on a colour scale ranging from black (completely untrustworthy) to white (trustworthy). The value used to represent the reputation of an individual is a ratio of the number of white pixels to black pixels in the Reputation Ring.

A site's reputation in this system differs from conventional Reputation Systems since new entities in the system do not start off with sterling reputations. Exceptional or flawless reputations have to be earned over time (as depicted in figure 5.2).

A confirmed privacy violation report against a site would darken the applicable ring by a configurable shade of gray. In figure 5.2 the third ring (third month in this case) has been darkened by one shade because of a confirmed report against the Web site. The effects of confirmed reports fade

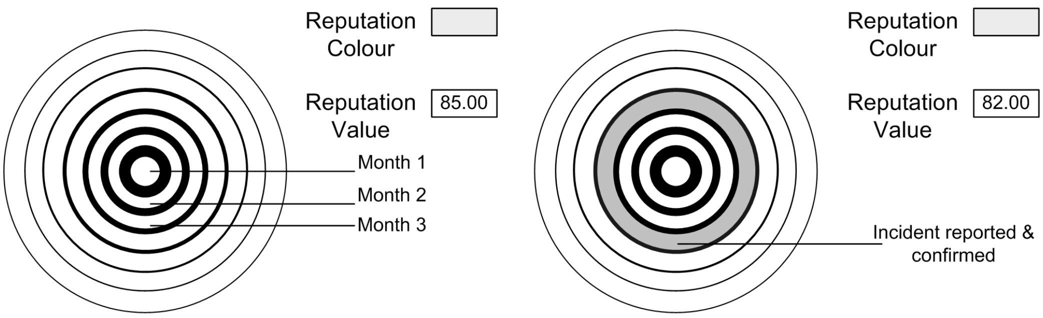

Figure 5.2: A reputation that has improved over time and a similar reputation that has had a confirmed incident reported against it.

as time progresses (with the introduction of more rings there will be more white pixels and so a better ratio).

Formalising a Reputation

A reputation is said to be the ratio of positive feedback to negative feedback. Since feedback in our system is measured in the form of time as well as in the number of confirmed reports against an entity we say reputation is comprised of variables a and b.

Variable a is defined as positive feedback while b is defined as negative feedback. The reputation for a single Web site can be calculated in one of two ways:

1. For a particular period in time. This is similar to calculating the ratio of white to black pixels for a particular portion of the Reputation Ring.

2. The overall reputation of the Web site. This is ultimately what a user would want to see.

For the sake of simplicity we calculate reputation in monthly intervals. And since reputation is the ratio of positive feedback to total feedback, the reputation for month t is defined as follows:

$$R(t) = \frac{a}{a+b} = \frac{a+b-b}{a+b} = 1 - \frac{b}{a+b} \tag{5.1}$$

where a, b are positive real numbers

Since a is a measure of positive feedback and this is proportional to time we denote a at t as $a(t) = t$ where t is a unit of time (equal to the number of months a user has been registered).

We define $b = c + v * r$ where c is a constant that all reputations start off with and can be improved over time; r is the number of confirmed reports against a user and v the desired impact of a confirmed report on a user's negative feedback (this can be compared to introducing a shade of gray to the Reputation Ring). As a result

$$R(t) = 1 - \frac{c + v * r}{t + c + v * r} \tag{5.2}$$

The overall reputation of a Web site is the average of the sum of the reputations over the time it has been registered i.e.

$$\overline{R} = \frac{\sum_{i=1}^{t} R(i)}{t} \tag{5.3}$$

The simple calculation of reputation offered in this section allows for a site's reputation to improve over time. Although reported incidents may have an impact on a reputation because of the way in which we have defined negative feedback (as a weighted constant) the effect of incidents will fade as time passes if the user does not have any more negative feedback.

Future Work

What must be investigated further is the weight of v. How much impact should a confirmed report have on a user's reputation? We gradually increase v from 0.1 to 1.0. Note that although v has more of an impact on the monthly reputation (Figure 5.3) there is not as much of an impact on the overall (Figure 5.4) reputation.

What is of great importance is whether or not v plays any role in the following scenario: user U_1 has been a part of the system for some time. At time T_1 another user U_2 joins the system. A confirmed incident is reported against U_1 at the same time. Once some time has passed, should U_1 and U_2 have the same reputation? Could v play a part in determining the outcome or could it be the case that the nature of a will have more of an impact than v?

5.3.5 Reputations Systems Bring Trust to P3P

In this section, we identified the problem of trust within the context of P3P. Although P3P allows Web sites to describe their privacy practices in a structured manner (in the form of a policy) there is no mechanism to guarantee

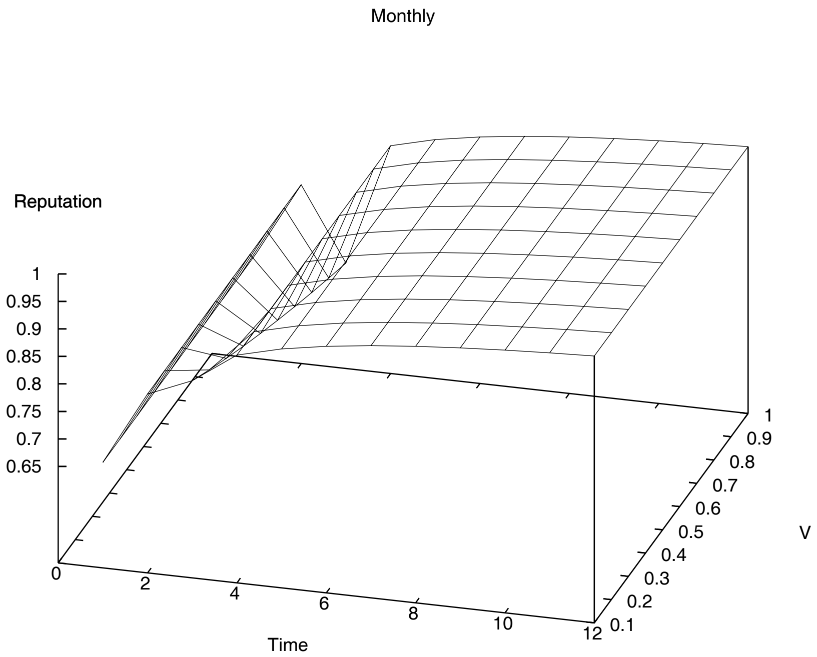

Figure 5.3: The effect of v ranging from 0.1 to 1.0 using a 12 month reputation with a single confirmed report against it.

users that a Web site will adhere to the P3P policy specified. It was suggested that a solution to this problem may lie in the form of a Reputation-based System. Although a Reputation System does not provide a user with any guarantees, it does give a user enough information so as to better decide how much trust to associate with a Web site.

Having discussed Reputation Systems, we have pointed out that a conventional Reputation System is unlikely to work in a P3P environment mainly because of the lack of feedback. As a result we have proposed an alternative to the conventional approach in the form of a trusted third party. Feedback, although not immediate, is implemented as a function of time while confirmed reports from users of a Web site are dealt with in the same way. Furthermore, reputations of Web sites in the system that we have proposed do not start off with exceptional reputations. Instead a new reputation is given an average rating and then encouraged to improve over time.

We discussed an alternative method to visualising reputation by compar-

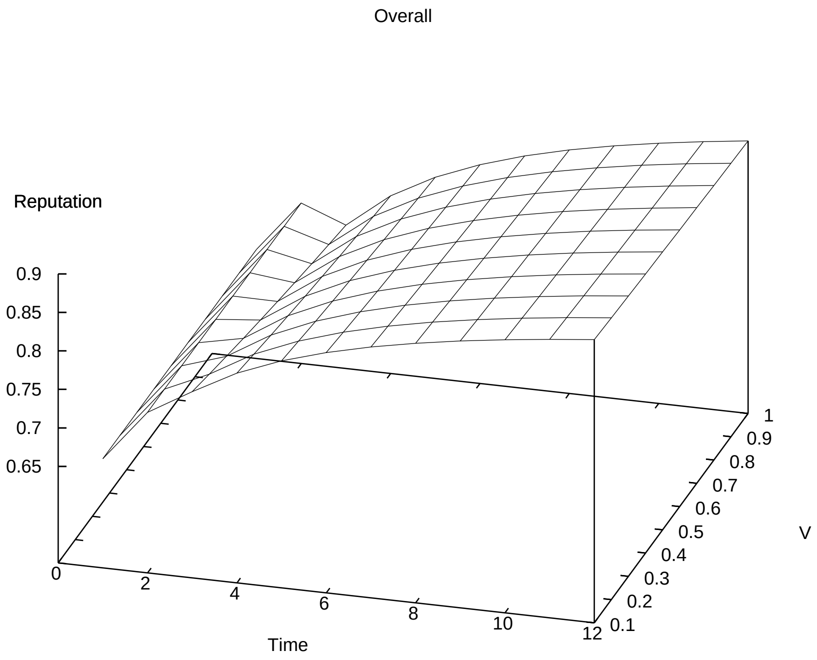

Figure 5.4: The effect of v ranging from 0.1 to 1.0 using a 12 month reputation with a single confirmed report against it.

ing a reputation to the cross section of a tree. We suggested a simple formula that may serve as a guideline towards building up a reputation as a function of time. Possible future work may involve investigating the significance of v (the desired impact of a confirmed report on a user's negative feedback).

From the previous sections, it is clear that without a Reputation System of some kind, trust between a user and a Web site within the context of P3P will have to have been earned through some other means. In this work, the Reputation System we have proposed allows users to gauge the trustworthyness of a site. For search engines that have exposed a P3P policy and earned a good reputation, their users have all they need to make informed decisions as to whether or not they adhere to their own privacy requirements.

5.4 Proxies and P3P

Although the way in which users and their associated P3P agents interact with a P3P compliant Web site has received much attention, the surreptitious role played by the Web proxy server and its impact on P3P has been neglected. In this section we consider the scenario of a Web proxy that is situated between a Web user and a P3P compliant Web site.

In focusing on the role played by the Web proxy from a P3P perspective we will discuss why it is imperative that the Web proxy is identified as a possible privacy threat. In doing so, it will be made clear that the Web proxy must not be excluded in so far as providing a P3P policy to the Web user. The P3P policy provided by a proxy however, brings with it a new set of problems. We analyse these problems and discuss potential solutions.

Transparent proxies are included in our analysis of proxies as potential privacy threats primarily because their nature is somewhat different to that of regular proxies and therefore deserve special attention. A transparent proxy functions like any other proxy with the exception that it is not explicitly configured by a user. Usage of a transparent proxy is in most cases configured by a network administrator on behalf of a user. As a result, although a user may have configured his Agent to make use of a trusted proxy, usage of the trusted proxy may be through an unknown, untrusted and undetected proxy, i.e., the transparent proxy.

The problem presented by chained proxies presents as much of a threat to privacy as transparent proxies and is therefore also included in this thesis. The chained proxy scenario arises from one proxy acting as a client to another proxy. A user may trust proxy P_1 and is content to access P3P services through proxy P_1. As was the case with transparent proxies though, the user may be implicitly going through yet another proxy which he does not know or trust (in the case of proxy P_1 being configured to use proxy P_2 in order to access the Web).

In this section, we focus on why the Web proxy must be considered a threat to privacy. We then discuss the problems introduced by a proxy in more detail as well as possible solutions. This includes a discussion of chained proxies, transparent proxies and the semantics of P3P from a proxy perspective.

5.4.1 Dealing with Web proxies in P3P

The P3P 1.1 Specification [31] only recognises proxies as a cache that may be holding P3P policies belonging to the Web site that a user wishes to access. There is no discussion as to how Web proxies should implement P3P or the way in which user agents should work with proxies that implement P3P.

Consider the P3P policy of section 5.2. This policy essentially states that users of the site will not be logged, i.e., all identifiable data (including the IP address) will be anonymised. If user Bob has specified that he does not want his IP address logged when accessing a Web site, this policy will meet his requirements.

In order to consider the issues that arise when adding a proxy to P3P we include a P3P proxy in the scenario above. For the moment, a P3P proxy is merely a Web proxy with a P3P policy defined for itself, i.e., a P3P compliant proxy.

To minimise the impact of introducing a P3P proxy, the associated P3P policy is accessed via a Policy Reference File which would in turn be accessed via any of the methods indicated in a previous section. The P3P proxy we will use in this example is configured to log all Web access attempts (including the time and IP Address) indefinitely.

In accessing the Web site in question, Bob's P3P agent realizes that Bob's browser is configured to make use of a P3P Web proxy (note that this is not a transparent proxy) which will then access the Web site on Bob's behalf. Bob's P3P agent must therefore scrutinize the policy of the P3P Web proxy to ensure that Bob's privacy is not being violated before making use of any other service on the Web.

Since the Web proxy's configuration stores identifiable information indefinitely, further usage of the proxy will result in a direct conflict with Bob's privacy preferences. If Bob is serious about not wanting his IP Address logged then he will not be able to access any Web sites at all (regardless of whether or not the P3P policies on the Web sites he will be accessing are acceptable).

The consequence of introducing a third party into the P3P framework may have dire implications since P3P was designed with only two parties in mind:

1. A user on the Web with privacy preferences employing the services of

2. a Web site with a P3P policy on a Web server.

The addition of a P3P Web proxy cannot be regarded as just an entity that needs to be scrutinized before accessing the Web. The nature of the proxy is such that there are many opportunities to violate the privacy preferences of an individual who uses it. For example, aside from the logging implications of a proxy, simply caching an object may result in a violation of privacy. A cached object means an attacker using the proxy will be able to determine whether or not a particular object on the Web has been accessed by another user of the proxy, perhaps opening the door to an inference attack of some kind.

Note that Web proxies issue requests to the Web for more than just one user (consider work or academic environments). Though many users may have similar privacy preferences, it is unlikely that their preferences will be exactly the same. If it is the will of the Web proxy administrator to respect the privacy of each of the users accessing the proxy then the administrator will either have to configure a policy that satisfies all users (whether this is feasible remains to be seen) or configure separate policies for each user (adding significant workload to the proxy).

Alternatively, a user could choose to make use of multiple proxies depending on his privacy expectations. It may be the case that proxy P_1 has a different privacy policy to proxy P_2. A user could then decide to employ the services of proxy P_1 for some sites and proxy P_2 for others. Though this seems like a viable solution, this does not apply to users that have only one proxy as a point of contact with the Web (in the case of administrators not agreeing to configure multiple proxies).

5.4.2 P3P Web Proxy Problems

By including the Web proxy in P3P, several problems arise that demand immediate attention. We discuss these problems and examine ways in which they may be resolved. Briefly, the problems are as follows:

> *The semantics of P3P policies.* As mentioned previously, the P3P framework was designed to address the privacy needs of a user accessing a Web site. The addition of a third party into the framework (the Web proxy) will result in subtle changes to the semantics of P3P policies.
>
> *Complicated proxy policies.* Satisfying the privacy needs of each user using a proxy may result in incredibly complex proxy policies. Since configuring these policies will be an arduous task, we must investigate a simpler alternative.
>
> *Transparent and chained proxies.* Transparent proxies are not explicitly defined to be used by the Web user. They may be the result of a sophisticated networking architecture and are therefore unseen by the Web user. Chained proxies occur when a single proxy acts as a client to a second proxy, this proxy will then issue a request to yet another proxy (or the Web) on behalf of the client proxy. What must be investigated is

how these proxies will identify themselves to the Web user and whether or not this identification is necessary.

In considering and discussing these problems it will be evident that Web proxies cannot be ignored within the P3P architecture.

P3P Semantics

The central theme of P3P centers around a client and a Web site. The Web site provides a policy, after perusal of the policy the client may decide whether or not to continue usage of the Web site.

In communicating with a proxy, a user is indirectly communicating with a Web site. As a result, the privacy policy of the proxy as well as the Web site must be acceptable to the user before any indirect communication between the user and the Web site can begin.

A simple solution to this problem is to have the proxy agree to implement the privacy policy of the site that is being accessed through it. If a user is content with the P3P policy of the site then it stands to reason that he should be content with the same policy being applied on the proxy that he is using to access the site.

Whilst this solution is at first attractive, it has the disadvantage of the proxy not having a say in the policy that is to be implemented. It may be the case that the proxy does not employ the same privacy practices at all and even if it was willing to change its practices for the duration of the session, the overhead in doing so for multiple sessions and many users may be far too much to deal with.

The solution proposed in this work allows for a proxy to specify separate policies for individual sites. Because of the indirect means of communication between different Web sites and multiple users, a proxy must be able to provide more than one single policy which describes how it deals with all data collected. It must be able to accommodate for detailing how it intends to handle information with regards to separate entities (Web sites) on the Web.

This is best explained with another example: although Bob has no problems in having his details logged (both at the proxy and at the Web site) when he accesses an online weather service, he is not willing to compromise any privacy at all in so far as electronic payments or online banking is concerned. On the one hand he does not mind the proxy keeping logs for *generic* Web access, but on the other hand he does not want any details logged at all for what he deems as *private* and *confidential*.

In order to support multiple policies we propose changes to the Policy Reference File (the `<POLICY-REF>` element in particular). The Policy Reference File, as mentioned earlier, refers to a P3P policy (or policies) and describes various attributes regarding the policy. We propose the addition of the **siteURI** parameter (as detailed in Table 5.1) which will typically denote the entity (Web site) for which the policy being referred to will be applied. The addition of this parameter allows a proxy P3P policy to specify which URI the P3P policy in question refers to.

```
policy-ref =
  <POLICY-REF
    about="URI-reference"
    [siteURI="URI-reference"]>
      ...
  </POLICY-REF>
```

Table 5.1: An optional parameter extension to the POLICY-REF element.

Absence of this parameter in the `<POLICY-REF>` tag of a P3P proxy policy denotes a *generic* policy, i.e., the policy that applies to all sites accessed via the proxy. Only sites that have defined policies (via the **siteURI** parameter) on the proxy will be excluded from the *generic* policy. Essentially, the **siteURI** parameter provides a mechanism for the proxy Policy Reference Files to indicate which P3P policies apply to certain URIs.

Figure 5.5 illustrates the process of a proxy looking up the appropriate policy to apply for the user in question. This is based on the URI of the site being accessed by the user. The process is as follows:

Step 1 - Client sends a request to the proxy to access a Web site on its behalf.

Step 2 - The proxy looks up the policy associated with the Web site being accessed. The proxy may choose to categorise P3P policies per user in addition to the siteURI tag.

Step 3 - If there is a P3P policy that applies to this user for the Web site in question, it is loaded by the proxy. If there is no applicable P3P policy then the generic policy (depicted as * in the figure) is loaded for the duration of the transaction with the user.

Step 4 - The response from the Web server is forwarded to the client.

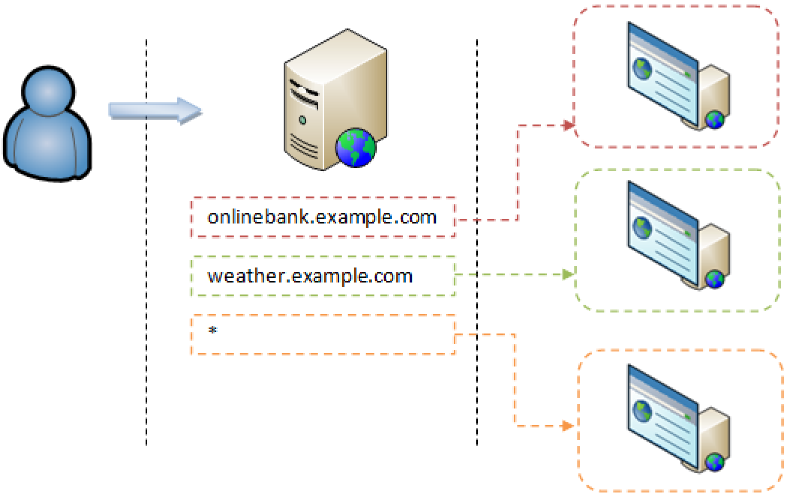

Figure 5.5: User accesses a Web site through a P3P compliant Proxy. If applicable, the proxy will load and adhere to a policy specific to the site being accessed.

The practical implications for proxies that choose to make use of the **siteURI** parameter may be catastrophic. Three problems are immediately apparent:

- The proxy may be subjected to additional stress as P3P policies are requested by users for sites that they may wish to access through the proxy.
- There is additional overhead on the entire process of requesting an object from a Web site as the P3P policy of the proxy for a site is looked up by the proxy upon each initial request to a Web site.
- If an administrator can be convinced that it is imperative for the proxy to strive towards meeting the privacy expectations of all its users then there may be considerable complexity in implementing generic proxy policies for the users, or alternatively, implementing different policies (possibly for different users) for different sites that are accessed via the proxy.

The first problem has already been anticipated in the P3P framework and is circumvented via a simple leasing scheme through the `EXPIRY` element in the Policy Reference File. The first time a user requests a policy from the proxy for a site then the user, in looking up the EXPIRY element, may gauge how long the policy will be valid for. This approach reduces the overhead of always requesting the policy from the proxy for a site each and every single time the user wishes to access the site.

A caching solution may help to alleviate stress on the proxy in so far as addressing the second problem i.e. a proxy can cache policies for quick retrieval at a later stage. Unfortunately, there may still be significant overhead in the initial lookup of a policy.

The third problem is not as easily tackled. We discuss this problem in detail in the next section.

Complicated vs Customised Proxy Policies

A silver bullet policy that addresses the privacy needs of all individuals making use of one proxy will be a rare feat since privacy preferences of users across the Web obviously differ tremendously. Attempting to create a single privacy policy for a P3P proxy that is a cross section of most users' privacy preferences will be an arduous task that, in most cases, will surely fail. The introduction of the **siteURI** tag into the <POLICY-REF> element alleviates the problem only slightly since it allows for proxies to simplify their policies by using multiple policies for different sites.

Needless to say, there is room for improvement in addressing complex privacy policies. Large proxies may have to implement a number of privacy policies to cater for the growing demands of their users. Unfortunately, this solution may not be feasible at all for larger proxies when one considers the sheer number of new sites that may be accessed by hundreds of thousands of users every day.

Simpler schemes may involve generic policies that are served to the general population of a proxy and customised policies that are served to a handful of users either because they pay for the customised policy or maybe because they simply don't fit the generic mould.

5.4.3 Transparent and Chained Proxies

In order to make an informed decision in the P3P architecture, a Web user needs to be aware of all elements that may be a threat to his privacy. We have identified the proxy as a potential threat to the Web user's privacy. A proxy must implement a P3P privacy policy detailing its intentions to the Web user. Transparent and chained proxies introduce another problem to the P3P architecture. In the event of a transparent proxy being present, the user may not be aware of its existence therefore any decision made by the user with regards to his privacy is not an entirely informed decision. It must therefore be the responsibility of a transparent proxy (of any Web proxy) to identify itself as a proxy to the P3P Agent of a Web user.

One mechanism that could be used for identification is that of the proxy detecting when a P3P policy of a site is accessed through it and injecting its own policy into the policy of the site that is sent back to the user. This approach has two benefits. The first is that of the actual identification and the second is that of having saved the user a trip to the proxy to lookup the appropriate proxy policy. A disadvantage of this approach is the overhead incurred in having to monitor all traffic accessed through the proxy.

The identification process we propose may take place through any of the means described in the P3P Specification [31]: HTTP Response Headers, embedded link tags, et cetera. In the case of HTTP Response Headers, we propose the addition of an optional field to the P3P header, the `proxy-policy-ref-field`:

`[proxypolicyref= URI-reference]`

Table 5.2: Optional addition to the P3P Header.

The addition of the `proxy-policy-ref-field` in the header (inserted by the proxy) will point to the URI of the P3P Policy implemented by the proxy. Not only does the addition of this field serve as a means that can be used by the proxy to identify itself as a part of a Web-based transaction, but it also points to the P3P policy published by the proxy.

In addition to the optional field, we propose the addition of a new element to the P3P vocabulary to enhance privacy policies for P3P proxies. The addition of the `<PROXY>` element as one of the root elements (an element in the `<META>` namespace - the same namespace used by `<POLICY-REF>` namespace) will allow P3P agents to immediately identify the P3P policy as one belonging to a P3P proxy.

In identifying the policy as a policy of a P3P proxy, the `<PROXY>` element will also give details as to any other proxies (or sources) that may be used in retrieving Web objects. These may be chained or transparent proxies. Table 5.3 describes the `<PROXY>` element.

The `<PROXY>` element contains an optional parameter denoting whether or not the proxy itself is a transparent proxy. Elements within the `<PROXY>` element may refer to other proxies that are used by the current proxy. It may be the case that in accessing a site, the current proxy will make use of

```
proxy =
 <PROXY about="URI-reference" [transparent=true/false]>
   [<PROXYSOURCE siteURI="URI-reference" [transparent=true/false]/>]
 </PROXY>
```

Table 5.3: The PROXY element.

another proxy. The user will now be made aware of this process and will be able to query the privacy policy of the other proxy used.

We allow the proxy to specify which proxy will be used when accessing a site by adding another extension to the `<POLICY-REF>` element. The optional proxySource parameter in the `<POLICY-REF>` element will typically refer to one of the `<PROXYSOURCE>` elements defined in the `<PROXY>` namespace.

```
policy-ref =
 <POLICY-REF
  about="URI-reference"
  [siteURI="URI-reference"]
  [proxySource="URI-reference"]>
    ...
 </POLICY-REF>
```

Table 5.4: An optional parameter extension to the POLICY-REF element.

These simple additions to P3P allow proxies to achieve two important objectives that must be realised in an effort to minimise the impact of proxies as a threat to privacy:

1. Proxies can identify themselves to a Web user, even in the case of the proxy being transparent.

2. Proxies can list any additional proxies that may be used (via the `<PROXYSOURCE>` element) in addition to when each of these proxies will be used i.e. for which sites they will be used (from within the `<POLICY-REF>` element).

What must receive further attention in future research is a mechanism for handling policies that are inaccessible to the Web user. Consider the following scenario: proxy P_1 uses proxy P_2 to access a Web site. proxy P_1 has a P3P policy and proxy P_2 has a P3P policy. In accessing proxy P_1, a

Web user is made aware of the chained proxy framework that proxy P_1 is a part of. The user knows that proxy P_1 uses proxy P_2 to access Web site, it is therefore essential that he scrutinizes proxy P_2's privacy policy before making use of any Web sites.

But what will the outcome be if the user is unable to access proxy P_2's privacy policy (possibly due to firewall restrictions)? There may be a solution to this problem by including a cached copy of proxy P_2's policy on proxy P_1. This approach however may have serious performance implications when several proxies are used in succession, which proxy policies should be stored where, and for how long?

5.4.4 Identification and Separation

In discussing the P3P proxy, this section has centered itself around two themes: Identification and Separation. It must be the responsibility of the Web proxy to identify itself to the Web user as a Web proxy. Identification is possible through the proposed optional field in the HTTP Response Header.

Having addressed identification of a proxy, one can begin to address the issues that arise with transparent and chained proxies. We have discussed these types of proxies as well as the problem they present in a P3P environment. A potential solution has been proposed in the form of extending policy files to specify the nature of the proxy being accessed as well as the nature of any additional proxies that it may make use of.

Having identified itself, the proxy enables users to identify policies on the proxy that will be applied when accessing specific sites through it. This is achieved via the **siteURI** parameter of the `<POLICY-REF>` element. Absence of this parameter denotes a generic policy, i.e., a policy that will be applied to all sites without policies.

In recognising the proxy as a privacy threat we have discussed several issues that are of importance when the P3P Web proxy is introduced; in particular, we have discussed issues relating to semantics and complex policies as well as transparent and chained proxies.

This is by no means a complete solution to the P3P proxy problem. Issues that require further attention in future research are as follows:

- Examining whether the burden of complex policies can be addressed as discussed in section 5.4.2. Could a feasible solution lie in the copying of policies or perhaps a simple kind of categorisation process?

- An investigation into the implications of a proxy caching Web objects. It may be in the best interest of the user not to have any Web objects that he has requested cached on the proxy at all since caching copies of a Web object opens the door several types of attacks, including timing attacks [49].

- Firewalls and Gateways pose just as much of a threat to privacy within the context of P3P as Web proxies. Can the strategies proposed in this thesis also be applied to them?

- Since there are several intermediaries involved in a typical Web transaction, the potential for violation of privacy preferences extends beyond the proxy and onto different layers of the Open Systems Interconnection Reference Model (OSI [35]). Though the approach adopted in this thesis may apply to some of the intermediaries (a firewall is also on the application layer) it is not as easily adopted by intermediaries that are further down in the OSI model. Since a router operates on the network layer of the OSI model it is not a candidate for the type of approach presented in this thesis. Future research would do well to investigate if it is acceptable to simply accept several types of intermediaries as static privacy threats.

5.5 Search Privacy and Personal Control

We have looked at two problems with a technology that embodies the personal control category: P3P. The first issue we dealt with was that of trust: how can a user trust the privacy policy of a Web site? The solution proposed in this work was in the form of Reputation Systems. Positive feedback of a reputation is delivered into the system as a function of time and negative feedback in the form of confirmed privacy violations. With a Reputation System in place, future users can make informed decisions when deciding whether or not to trust the policy of a Web site.

The second issue with P3P is that of proxies. Making decisions without taking into account the practices of the underlying proxy means that a user may be making an uninformed decision. We proposed that proxies should themselves implement P3P policies. We highlighted the themes of Identification and Separation. Our solution to dealing with transparent proxies implies that they are willing to surrender themselves (in the form of identifying that they exist). As much as proxies that identify themselves can fall into the Reputation System presented and in time become trusted, a transparent proxy with suspicious privacy practices has no incentive to identify itself (and start earning a bad reputation).

P3P does enhance the control that users have over their privacy when employing the services of legitimate Web sites. With a Reputation System in place and proxies that implement policies this control is greater because decisions made by the users are more informed. If one accepts that there are entities on the Web that are simply beyond one's control (routers, firewalls, transparent proxies), then the solutions offered in this chapter may be sufficient.

In the event that this naive view of privacy is acceptable, we ask the original questions related to search privacy:

- *Does this PET category have the ability to enhance a user's search privacy?*
  The answer to this is simple: absolutely. This category allows for more informed decision making. Not only does a user know more about what will be done with information collected by the Web site in question (the search engine in this case), but the user can use the reputation of the site to determine whether or not its promises are genuine.

- *Is the search engine still a threat to the user's search privacy?*
  Regardless of whether we are accepting a naive view of privacy, there is still a problem that remains in each of the categories we have looked at thus far: the user's queries are being sent to the search engine. Once the search engine has constructed a search profile from these queries, what happens to it is beyond the control of the user (consider for example if the search engine is compromised or if information is accidently sold). With this in mind, the search engine is still a threat to the user's search privacy.

Given that search engines are still a threat (even with a naive view of privacy), we continue the search for a category that can satisfy our definition of search privacy.

Chapter 6

Search Privacy Through Private Communication

"You have no privacy anyway, get over it"
Scott McNealy, CEO Sun Microsystems, 1995

6.1 Introduction

Private communication between two parties can be achieved through encryption and/or obfuscation. The former, if supported by the search engine, is useful since it thwarts eavesdroppers from recording an individual's search profile. Whilst this does enhance control over one's search profile, it does not eliminate the search engine as a threat to one's privacy (since the search engine still receives the queries and therefore has ultimate control over the search profile stored). The latter is the subject of discussion in this chapter. If we consider the search engine as an eavesdropper (an entity we know will intercept all communication) then, since encryption is not an option, we want to determine if we can obfuscate the data in a manner that will eliminate the search engine as a threat to search privacy.

We assume that the search engine will not adhere to the commitments made in its privacy policy (whether this is intended or not is inconsequential). The TrackMeNot (TMN) extension to the Firefox browser makes a similar assumption. The premise of this technology is to generate noise from the client (by sending false search queries to the search engine), eventually resulting in the real search queries being "*hidden*".

We begin this chapter by evaluating this approach as a means of enhancing control over a search profile. Having discussed this briefly in a previous chapter, it is evident that it does enhance a user's search privacy; what remains to be answered is whether or not the search engine is still a threat.

6.2 TrackMeNot

Howe et al [66] refer to TrackMeNot as means of achieving privacy through obfuscation. They describe the technology as a "lightweight Firefox browser extension designed to ensure privacy in Web search by obfuscating a users actual searches amidst a stream of programmatically generated decoy searches". They realise that there may be an effort from the search engines to thwart this technology and as a result have implemented several features that make it difficult to do so.

The first feature is that of Dynamic Query Lists (DQL). This is a list initially seeded by random terms retrieved from a number of RSS feeds upon being installed on a user's machine for the first time. It is then used by TMN as a source for queries sent to the search engine. In time, TMN will mark terms in the DQL for substitution. Results from queries generated from these terms will be scanned for eligible replacements. Essentially, the DQL evolves through time to be unique to the TrackMeNot user in question. As a result, the queries generated by the tool and the search profile eventually stored by the search engine will also be unique to the user.

The second feature is that of Selective Click-Through. In an effort to mimic a normal user, TMN will click-through on (navigate to) some of the links returned from a query (the "more results" link, for example).

The third feature is Real-Time Search Awareness (RTSA). This monitors the user's browser and makes TMN capable of detecting when search queries have been initiated by the user. This serves as the foundation for two additional features: (1) the exact headers used by the actual user can be used in queries sent by TMN and (2) instead of randomly spaced search queries, TMN can issue a number of queries within close proximity to the user's actual search queries.

6.3 Recognising TMN

The authors of TMN have created a tool which mimics search queries submitted by humans. With such a rich feature set, one can easily argue (and the authors do) that it will be frustrating for the search engines to distinguish between TMN generated queries and actual user's queries.

From a user's perspective, TMN provides an enhancement of control over one's search profile in the form of obfuscation. In addition to legitimate search queries submitted by the user, TMN generates a number of false queries and submits these on the user's behalf. The search profile stored by the search engine is therefore a result of a number of legitimate queries hidden in and amongst a sea of false queries.

From a search engine's perspective, not only are their resources being consumed by false queries, but the profiles stored no longer serve as extremely rich sources for research and data mining. We believe these two reasons serve as a strong motive for a search engine to want to be in a position to distinguish between queries that are legitimate and queries that have been falsely generated.

Distinguishing between these types of queries is essentially a matter of determining whether or not the entity behind the query is a machine. Search engines already have incredible financial interest in solving this problem, specifically for the purpose of thwarting click fraud (as discussed in chapter 4). Since dealing with click fraud is related to solving the problem of distinguishing machine generated activity from that of a human, it seems plausible that this technology can be applied to thwarting TMN.

The following two scenarios are simple examples of the technology that can be used to detect queries generated by TMN:

> *Invisible content* - the search engine provides results containing links that cannot be seen by a human, i.e., they are not visible on the screen and can only be *seen* by the machine. For example, the colour of the link could be the same colour of the background. As legitimate invisible links, the search engine knows that when they are navigated to then the associated search query from which it was served was probably generated by a machine. Of course, this example does not take screen reading software into account.
>
> *Tracking users* - Javascript is a common technology on the Web which can be used to serve as a foundation for frameworks that track and monitor user behaviour when visiting a Web site [16, 15]. Previous research has focused on tracking the mouse movement of a user in order to determine their search query intent [62]. We believe a similar approach can be used to determine if a user on the Web site is human by concentrating on two simple characteristics: (1) when clicking on a link, is the position of the click always in the same place and (2) when moving the mouse from one position to another, is it always in straight lines?

This is by no means a comprehensive list of the techniques that can be used to determine if a search query has been generated by a human. We highlight these techniques only to emphasize that search, or the act of searching on the Web of today, has become an integral part of our lives that is far more complex than just the process of sending text queries through to a search engine. Whilst it is not impossible to mimic the behaviour of a human in the scenarios above, perfecting this type of technology (simulating the activities of a human) is far beyond the scope of what the authors who proposed the search privacy through obfuscation technique had originally intended.

6.4 Obfuscation and Search

As we highlighted earlier, private communication between a sender and a recipient can be achieved through encryption and/or obfuscation. Within the context of search, encryption would protect queries sent to the search engine from eavesdropping but it would not protect the query from the search engine itself. Since the search engine receives and processes the query it is able to construct a search profile and is therefore still considered a threat to search privacy. One could encrypt the query so that the search engine would be unable to see it but, for obvious reasons, this would be counterproductive.

The second option is that of obfuscation. The authors of TMN chose to use an obfuscation technique that hides real queries to the search engine amongst a number of falsely generated queries. Unlike steganography, where one tries to conceal the very existence of the message exchanged between a sender and recipient [72], this technique generates noise around the legitimate queries sent to a search engine (the intended recipient) resulting in an inaccurate search profile stored. This method of obfuscation extends more control into the hands of the searcher since he/she is able to indirectly influence what it is eventually stored by the search engine. The increase of control can be said to enhance privacy since, as discussed earlier, privacy is directly related to the control one has over one's search profile.

Unfortunately, the search engine remains a threat to one's privacy if it is able to differentiate between queries that have been falsely generated (by a machine) and queries that are legitimate (sent by a human). In the previous section, we discussed the motive a search engine would have for being in a position to do this and also pointed to previous work that could be applied to solving this problem. Essentially, detecting this form of obfuscation can be achieved by detecting queries that have been generated by a machine. What remains to be discussed is whether or not this obfuscation can be detected if it is not generated by a machine.

In chapter 4 we discussed search privacy through anonymity, i.e., using crowds to conceal one's identity from a search engine. Upon joining a crowd the participant (a jondo) is assigned a random path through the crowd that will relay messages to one another and eventually the intended recipient (the search engine) on the jondo's behalf. We now consider a crowd that has been modified as follows: instead of assigning a jondo a random path upon joining the crowd, a path is assigned each time a legitimate query is issued. The jondo chosen to be the exit point from the crowd will exhibit similar behaviour to the user that initiated the search (similar to the TMN features discussed earlier).

For example, user A joins the crowd and starts using search engine X. Queries from user A to X may be routed through the crowd and eventually through user B, who was also using search engine X at the time. In this scenario, the false queries are still routed through to a user's search profile. The difference is that the entity behind the false queries is not a machine.

In this scenario, the problem of determining a user that is obfuscating his/her search profile is no longer related to detecting queries generated by a machine. A jondo in the crowd will essentially have its legitimate queries hidden amongst a sea of legitimate queries that have been generated by other jondos (humans in the crowds).

Unfortunately, this does not mean that we have finally found search privacy. The problem of detecting queries generated by a machine has been shifted to detecting queries generated by other humans, surely a much more complex problem. Whilst this does offer promise in so far as obfuscation techniques it has also introduced another problem: that of trying to achieve privacy through anonymity. We concentrated on this problem in chapter 4 and solutions in this category were shown to be unsatisfactory in so far as eliminating the search engine as a threat to one's search privacy.

6.5 Conclusion

In this chapter, we have investigated the category of private communication as a means to achieving search privacy. Having identified encryption as an unsuitable approach within the context of search, we turned to obfuscation. Essentially, we want to hide information from the very entity we are sending it to.

The TrackMeNot extension set out to achieve this by hiding legitimate queries amongst a number of falsely generated queries. The authors of TMN implemented an array of rich features which make detection of the false queries extremely difficult.

Instead of focusing on the queries themselves, we believe that the search engine is still a threat if it chooses to focus on the activities behind the queries and not just the queries themselves. A number of advances on the Web make this a feasible initiative. Furthermore, we discussed a motive for the search engines wanting to be in a position to detect these activities and pointed to current initiatives that could be applied to this effort (thwarting click fraud).

With a motive and the ability to differentiate the artificially generated queries from the genuine ones, the search engine can "deobfuscate" the search profile. We then discussed a modification to crowds that would have users with similar search behaviours submitting queries on behalf of each other. The difference in this scenario is that a human being would be behind each *false* query (essentially thwarting the deobfuscation technique proposed – machine vs. human activity). Unfortunately, this technique depends on search privacy through anonymity (something we have already shown is not viable).

Since an accurate search profile can be maintained regardless of a user's attempts to obfuscate it (be it via machines using tools like TMN or through human activity in the crowds scenario), the search engine remains a threat to search privacy and our search for search privacy continues.