Chapter 3

Search Privacy

"Ask, and it shall be given you; seek, and ye shall find; knock, and it shall be opened unto you"
Matthew 7:7

3.1 Introduction

It may be argued that many users do not employ the usage of PETs simply because they do not consider their actions online of having the potential to violate their privacy. This may be the case when one considers users that access a number of independent Web sites in a manner where no private information is ever divulged. But what of a handful of Web sites that are integral to the online experience of hundreds of millions of users each and every day? Search engines have become an important part of our experience on the Web. The content of billions of indexed Web pages coupled with billions of user queries looking for information [93] makes search engines a primary point of enquiry [12], a powerful index of what is on the Web and how relevant it may be in so far as what one is searching for.

This dependence on search engines as a source of information and a starting point on the Web today can be viewed upon as a privacy nightmare. Whilst it may not be a violation of privacy when a single day's worth of queries is stored for a single user, the implication of years of queries stored for millions of users worldwide can not be overlooked. Fortunately, there are a number of PETs that serve users in a bid to thwart some of these privacy problems. In this chapter, we will discuss what privacy means in the scope of search. Furthermore, we will focus on the relevant PETs presented in the previous chapter and discuss the strengths and weaknesses of each approach.

This chapter is structured as follows: in section 3.2 we briefly discuss the history of online search and analyse the flow of data amongst all participants concerned. Having analysed exactly what it means to conduct a search, in section 3.3 we examine the search profile stored by a search engine and discuss the privacy problem in search in more detail. Section 3.4 then looks at each of the PETs from the previous chapter and examines their applicability to the search privacy problem.

3.2 Search

The first form of search on the Internet was offered through Archie [37]. This was a database accessible via Telnet [98] that allowed users to query the content of directory listings of publicly accessible FTP [97] servers. In time, the introduction of a distributed document search and retrieval protocol (Gopher [10]) meant that users could search the contents of files and not just the names.

The growing popularity of the Web coupled with the explosive growth of its content meant that it would undergo a similar evolution. In the early days, if a user wanted to find something on the Web he/she either had to know the precise address of the document, or would have to start at a known location (for example, at the list of accessible Web servers maintained by CERN [77]) and navigate through a number of linked documents in the hope of finding something of relevance.

This laborious process of manually wading through content came to an end in 1993, when an MIT student presented the first Web search engine: Wandex [13]. This system consisted of programs on the back end called Web crawlers which navigated the Web and indexed the content found. The content was then searchable in the form of a Web front end which allowed users to query what had been searched. The Wandex search engine was shortly followed by Webcrawler in 1994 [96] and, as a result, full-text search on the Web had finally arrived.

Arasu et al [12] outline the architecture of search on the Web. This can be summarised as the interaction between three entities (illustrated in figure 3.1):

> **The user** - obviously, a user wishes to find something on the Web. Search engines facilitate this process by accepting text based queries which will result in a set of links to Web sites of relevance.
>
> **The search engine** - providing relevant links using text based queries alone is not easy. There are a number of factors that come into play when trying to determine the intent of the user (for example, when searching for the term ‘free’) in addition to knowing which Web sites are better than others. Search engines used to depend on well known information retrieval algorithms [46] to determine which indexed Web sites were of relevance. Unfortunately, these methods do not account for the vast amounts of information that the Web has to offer nor do they leverage off of the intricate structure of the Web. A simplified example is that of the approach which matches text that makes up a user’s query directly to keywords extracted from Web pages indexed earlier. Whilst this results in a fair degree of relevance in some cases, there are a number of obvious pitfalls to this approach. Consider the query ‘free’, this keyword could occur the same amount of times on a simple classifieds page as it could on a page dealing with the intricacies of licensing free software. Calculating a Web site’s worth or relevance with reference to a keyword or query today now employs a number of techniques that exploit the graph-like structure of the Web [7] in addition to the earlier information retrieval approaches. Of note is the Pagerank [25] algorithm, this takes into account what other pages think

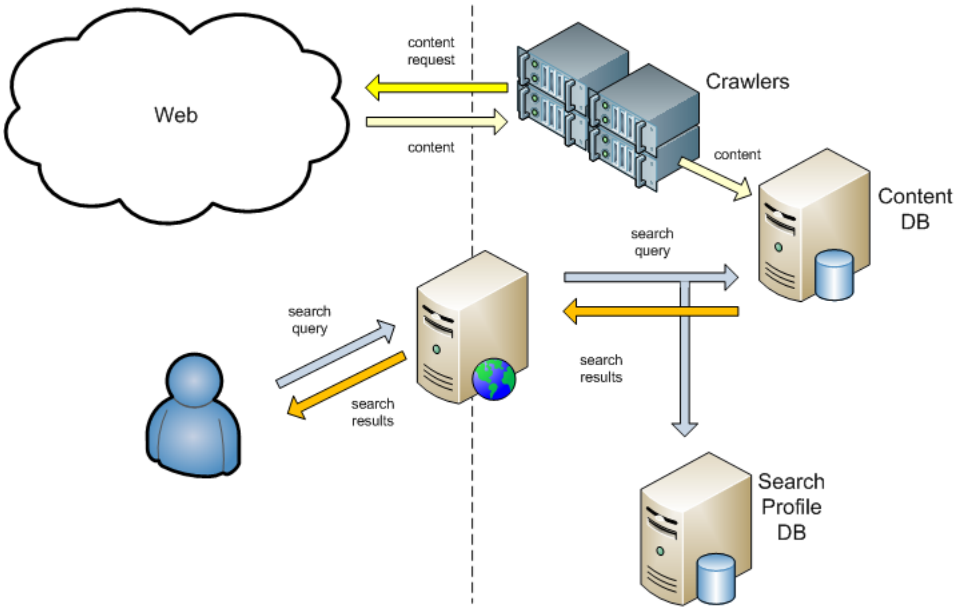

Figure 3.1: A search engine processes a user's query with the goal of delivering a set of links to relevant content on the Web that was crawled, indexed and processed earlier.

about a Web site in addition to the context used when referencing a site.

Crawlers - these programs build up search engine databases by navigating to pages on the Web and locally indexing the content found. Although they do not interact directly with the user, queries from users over time are used when deciding how often a page is revisited by a crawler. Cho et al [68] identify two types of crawlers: *periodic crawlers* which build up brand new content each time a crawl is conducted, i.e. the entire index is renewed, and *incremental crawlers* which update and refresh existing pages in the index over time.

The three entities highlighted essentially deal with the flow of two categories of information: user queries and Web site content. At its core, search on the Web is the process of determining which Web site content best matches the user query provided.

3.3 Search Profiles

It is reasonable to assume that the better a search engine is at matching the intent of the user with relevant content, then the more likely the user will be to return to the same search engine for more information in the future. The incredible efficiency and speed at which modern search engines are able to do exactly this in today's world is astounding. Finding information through the search engines has never been easier. The interesting business model employed by search engines, that of Internet advertising [43], has the search engines not only being easy, but also free. One can truly say that, for the first time, the world's information is literally at one's fingertips.

In this work, a search profile is the history of the queries issued by a user to a specific search engine. In order for a search engine to construct such a profile, it must have the ability to recognise or track users through time. There are a number methods that can be employed by a search engine to achieve this goal, Aljifri et al [8] highlight two of them:

- By recording the IP address associated with the query request. Whether fixed or dynamic, an IP address can be considered Personally Identifiable Information (PII).

- A unique cookie placed on the user's machine through the Web Browser will result in the search engine being able to recognise the user upon his/her return.

Another mechanism that may be used to track a user is that of unique characteristics in the Web request headers, for example, the User Agent string coupled with the browser's dimensions, reported location and number of sessions. Regardless of the technique used, we assume that search engines are aware of and have implemented at least one of the methods available.

The ability to track a user's queries through time means that although it is easy for a user to issue queries to a search engine, unless they are using a PET of some kind then it is just as easy for the search engine to associate the user's query with his/her profile. The nature of the Web is such that every user of a search engine can be mapped to a unique search profile consisting of the following:

- A set of search queries

- The time at which each query was issued

- Where applicable, the link that was eventually of relevance to the user, i.e., the link he/she chose to traverse to in the set of links the search engine matched to the original query. Tracking these links is trivial, hyperlinks to relevant content are simply redirected through the search engine.

3.3.1 Offline Profiles

It can be argued that search profiles are nothing new in the sense that they are the online equivalent of a system that has been around for some time. Consider the library, users of this system are much like users of a search engine since they too are looking for information.

Online users issue a query to a search engine which results in a set of links which redirect to Web pages with more information. The onus is then upon the user to scan the content of the Web pages in order to determine if they are relevant.

Instead of a set of links on a Web page, users in a library have a number of alternatives:

- The library may classify its content using the Dewey Decimal System [38], in which case the user could go directly to the relevant section and begin going through the available sources.
- The user could ask the librarian for assistance or perhaps to make a recommendation.
- The library may have a computer system in place which has already indexed the content of its books. Finding relevant information would consist of using the system to determine which book is relevant, and then finding the book itself.

With this in mind, the search profile built up by the search engine could be compared to the history of a user in a library. In each system, there is a request for information which results in the persistence of a record. The search engine records what the user searched for (a query) and, at the very least, the library records what the user may have checked out (a book, magazine, journal et cetera). If the information stored by the library and the search engine are equivalent, then the applicable privacy regulation of a library could be applied to that of a search engine.

Of course, this is not the case. A search profile is vastly different to the checkout history in a library. When one considers the scenario of a user in a library, a search engine profile would be equivalent to a librarian following the user around and recording not only what may eventually be checked out but, more importantly, how this was found (which books were looked through) and for what purpose (which book provided the most relevance). The reason for this is simple: with the astounding growth of search engine popularity, not only are search engines used as a starting point to almost all queries for information, but search engines are able to track what the user is searching for over time in addition to what was eventually relevant.

3.3.2 A Problem

In 2006, an online search engine (AOL) made one of its own search profile databases available with the intention that it be used exclusively for research [85]. The database contained the profiles of approximately 650,000 of its users over a period of three months. Each query in the profile consisted of the following:

- The username of the user
- The time of the query
- The query itself
- Which link the user followed after the query was submitted

AOL claimed that they had *anonymised* the data by replacing the username for each profile with a random number. What followed was a privacy catastrophe: within a few days of its release the private lives of a number of users in the database were on the Web and open to scrutiny by anyone with basic SQL knowledge and a bit of curiosity. Since then, there are a number of communities and users on the Web[1] that have taken a keen interest on inferring what they can about users in this database. In some cases, users have been identified in so far as where they live, what they do and even what car they drive [19]. On the subject of releasing search profiles for research in a privacy preserving manner, a number of researchers refer to the AOL incident and propose their solutions [4, 76, 128]. Gehrke et al [60] refers to search engines as collecting a “database of intentions”. In referring to the AOL debacle, they conducted a comparative study in which they analyze more sophisticated algorithms for publishing query and click histograms derived from search logs.

There is no doubt that there was a massive privacy violation with the release of the AOL database. With this in mind, we ask at what point was the privacy of these individuals violated:

[1] http://www.aolpsycho.com/, http://www.dontdelete.com/

- When the search engine kept a log of queries?
- When the search engine kept a log of queries associated with usernames and/or IP addresses?
- When the search engine kept a log of queries for more than a day, a week or a month?
- When the search engine released the queries online (after they were supposedly anonymised)?

If we consider the definition of privacy offered by Bellotti and Sellen [116] ("*a subjective notion which may be influenced by culturally determined expectations and perceptions of one's environment*") then the privacy violation could have occurred at any one of these points.

If we refer to the definition from Warren and Brandeis [119] ("*the right to be let alone*") then the violation occurred when the search engine released the queries online. It was this act that resulted in users of the database suddenly coming under the spotlight and receiving attention from the media. Had this not happened, then they would not have been bothered.

We concluded the previous chapter on the note that within the context of this work, we adopt the definition of privacy offered by Westin [120] ("*the ability of the individual to protect information about himself*"). In this definition, emphasis is placed on the ability of the individual to *control* his/her own information. If we consider a search profile as information belonging to an individual, whether personal or not, then neither of the points highlighted apply because the user in question has already lost control over the information, i.e., the search engine has it.

It can be argued that had the search profiles been anonymised more effectively then the privacy problem that followed (regardless of its definition) could have been minimised, or perhaps even avoided entirely. Despite the simple anonymisation process, one can easily look over each user's profile and draw a number of conclusions from the time a query is conducted and what other queries followed/preceded it. The problem with the anonymisation process is that each query still belongs to the original profile in its original order, i.e., each query still has context.

A lone query is not nearly as important as a query which has context. For example, a single query for a wig has a lot more meaning when it is preceded by queries for information on cancer treatment or the location of cancer clinics a few weeks earlier.

One approach to anonymising the data is that of having the queries lose their context. Instead of assigning each profile a random number, each query is assigned a random profile and only then is each profile's username replaced with a random number. This simple process of scrambling the queries is surely far more effective when thwarting the inference of private information for an individual (a profile) based on search queries alone. Of course, the usefulness of the anonymised search profiles for research purposes can be brought into question. On this topic, Götz et al [60] propose the ZEALOUS algorithm. This promises to allow search engines to publish their search logs in a manner that allows for useful research in addition to delivering a strong degree of privacy.

Cooper [28] notes that search profiles "*serve as a unique window into individuals' intentions, desires and behaviors*". In a survey of the techniques search engines could apply to their logs in order to preserve the privacy of their users, Cooper discusses the following approaches:

- Log deletion - not save the queries at all. From the perspective of a search engine taking control over the search privacy of an individual, this is highlighted as the most privacy enhancing technique available.

- Hashing queries - one-way hashes of the queries are stored instead of the queries themselves. Although this is susceptible to sophisticated attacks that exploit the frequency of queries, this technique is still considered valuable in the scenario where government agencies request query logs from search engines.

- Identifier deletion - the removal of PII the likes of an IP address.

- Hashing identifiers - much like hashing a query, this involves the application of a one-way hash function to all identifiers associated with a query.

- Scrubbing content - removing identifiers from the query, this includes Social Security numbers, credit card numbers, addresses et cetera.

- Deleting infrequent queries - proposed by Adar [4], the theory is that removing infrequent queries results in removing identifying queries. Although this approach has the potential to remove the queries of users with unusual interests, the queries are still bound to identifying data the likes of cookies or an IP address.

- Shortening sessions - privacy threats are said to have been mitigated by shortening the length of time that any identifier is associated with an individual.

Given that privacy in this thesis is directly related to the control an individual has over their own information, any approach employed by a search engine to enhance the privacy of an individual is inconsequential. For each of these scenarios, trust has to be extended to the search engine to *do the right thing*. Ultimately, the user has lost control over the search profile because the search engine has it. Since the user has no control over their search profile, the user has no search privacy.

3.4 PETs and Search Privacy

There is no question that search profiles are private. In this section, we highlight each of the PETs discussed in the previous chapter and discuss their potential for enhancing the control a user has over their search privacy. For every PET, we determine if it has the potential to enhance the control a user has over their search profile in addition to whether the search engine still poses as a threat to their privacy. We divide this discussion into the four categories of PETs listed earlier.

3.4.1 Private Communication

Defined as the ability to communicate content only to the recipient specified, PETs in this category have the potential to enhance search privacy in that only the search engine (an intended recipient of the query) will have the search profile. In the absence of this technology, there are a number of entities that may be privy to a user's search profile; for example, the user's ISP or a proxy used in the chain of servers leading to the search engine. SSL [51] is a particularly good example of a PET that fits into this category. If the Web server of a search engine supports this protocol, then users can send queries directly to the search engine with the confidence that no other entity will record the queries. Of course, the search engine still has the queries and trust is then extended to the search engine to record or transform the queries in a manner that does not violate the privacy of the individual. As a result, although PETs in this category have the potential to enhance control in terms of privacy from eavesdroppers, it does not eliminate the search engine as a primary threat to one's privacy.

This does not mean that this category should be discarded. Consider the scenario where the search engine itself is considered the eavesdropper, can there be privacy if we know who is listening and can control exactly what is sent to this entity? With reference to client-side technologies the likes of the TrackMeNot extension [66], could enough noise be generated so as to maintain the search privacy of the individual irrespective of the privacy policy of the search engine? Since users have complete control over what queries are sent to the search engine, it is reasonable to assume that sufficient noise can be generated in the search profile that will eventually be stored by the search engine. When one considers the role that obfuscation can play, this category may have the potential to enhance a user's search privacy. Whether the search engine remains a threat has yet to be discussed.

3.4.2 Anonymity

Anonymity is control over one's identity. The identity of an individual in this space may include information the likes of the IP address of the user's machine.

If a PET in this category allows a user to send queries to a search engine in a manner that does not reveal information about his or her identity, then the search engine does not know the true source of the query. Without this information, an accurate search profile cannot be constructed.

Anonymity PETs have the potential to greatly enhance the search privacy of a user. Whether or not the search engine still remains a threat though remains to be investigated. An issue that warrants further analysis is that although identifying information may not be explicitly provided, there may be the opportunity for it to be inferred. Since this is a topic of a future chapter, in this section we conclude the issue of search engines being a threat to search privacy as questionable.

3.4.3 Personal Control

This category of PET aims to ensure that any information shared by the user will be used in a manner that adheres to his/her privacy policy. For example, if the user prefers only to use search engines that do not keep any logs (store any of his queries), then PETs in this category would warn the user when this preference is in jeopardy.

Although these PETs have the potential to greatly enhance search privacy, it is not obvious how. In the case of the example above, how does one know when a search engine is not going to keep logs? Much like the previous category, we acknowledge that there is potential to enhance control over one's search privacy with this PET, but how this is done and whether or not the search engine remains a threat has yet to be investigated.

3.4.4 Organisational Safeguards

PETs in this category are aimed at the organisation, not the individual. Essentially, the organisation implements mechanisms that limit internal misuse so as to safeguard the interests of their users.

Consider the Internal Revenue Service (IRS) as an example. It may be the case that tax assessors need not be granted access to all tax returns all of the time. The IRS could implement mechanisms that only allow tax returns to be assessed by assigned individuals and only for the period for which an assessment is valid. Of course, there are exceptions (in the event of an audit, for example). In so far as issuing tax refunds to tax payers, the department responsible may only require access to their bank details and not the tax return in question. As a result, the IRS could store the bank details in one database and the tax returns in another, with separate DBAs assigned to each.

In the context of this research, the organisation responsible for implementing safeguards is the search engine. As noted in section 3.3.2, there are a number of options available to the search engine in so far as safeguarding their user's privacy. Ultimately, since this is a category of PETs that deals only with the organisation and its own internal controls, there can be no enhancement of a user's control if it has already been lost. Whether or not the user has willingly lost control over his/her search profile, the fact remains that in this category, total control over a profile lies in the hands of the organisation.

3.4.5 Summary

Table 3.1 summarises our evaluation of the PETs discussed thus far.

3.5 Conclusion

We began this chapter with an overview of search on the Web. This included a brief discussion of the history of search and an analysis of the three key entities involved: the user, the search engine and the crawlers. With a high level understanding of what search entails, we introduced search profiles. Essentially, these are logs of a search user's queries over time. In addition to storing queries, a search profile contains numerous other pieces of information; this includes the external link that was eventually followed and the IP address of the user in question.

We then introduced the privacy problem inherent with storing a search profile and referred to the database released by AOL in 2005 as an example

PET	Enhances control	SE Threat
Private Communication	Y	?
PGP	Y	Y
SSL	Y	Y
OTR	Y	Y
Steganography	?	?
Obfuscation/TrackMeNot	Y	?
Anonymity	Y	?
Anonymous proxies	Y	?
Crowds	Y	?
Onion routing (Tor)	Y	?
Personal Control	Y	?
P3P	Y	?
IE8 InPrivate blocking	N	Y
Cookie Managers	Y	Y
Organisational Safeguards	N	Y
E-P3P	N	Y
Hippocratic Databases	N	Y

Table 3.1: Each PET discussed in the previous chapter is assessed according to whether or not it has the potential to enhance search privacy and if so, whether the search engine still remains a privacy threat.

of the nature of privacy violation that a search engine is capable of. This lead us to the question of what search privacy means. Having looked at some of the definitions of privacy offered in the previous chapter, we define search privacy as *the control one has over one's search profile*. The absence of control implies the absence of privacy.

This lead to an examination of each of the PET categories highlighted earlier. We wanted to know if any of these could provide control over a user's search profile and in doing so, provide the user with search privacy. The results are unclear. The PETs that are not applicable have been easily identified but the categories that are applicable leave us asking if the search engine is still a threat. Whilst they have the potential to enhance control over a search profile, we are left unsure as to whether or not they provide enough control so as to no longer consider the search engine a threat to one's privacy.

Over the next three chapters, we examine each PET category in detail. If a category does not have the potential to provide search privacy, we investigate why this is the case and if it can be remedied.

Chapter 4

Search Privacy Through Anonymity

"Anonymity is a shield from the tyranny of the majority. It thus exemplifies the purpose behind the Bill of Rights, and of the First Amendment in particular: to protect unpopular individuals from retaliation – and their ideas from suppression – at the hand of an intolerant society."
McIntyre v. Ohio Elections Commission, 514 U.S. 334 (1995)

4.1 Introduction

In the previous chapter, we emphasised the notion that search privacy is essentially control over one's search profile. Through an anonymising network, a user could indirectly exercise control over their search profile by concealing their identity from the search engine. The premise is that if the search engine does not know the true identity associated with a search query, then it cannot construct an accurate search profile.

In this chapter, we explore the feasibility of attacking an anonymising network from a search engine's perspective (an external attack). If this is successful, PETs in the anonymity category may not necessarily be the best option for search privacy.

4.2 Motive

Popular Internet protocols the likes of TCP/IP were not designed with anonymity in mind. A TCP/IP message is likely to be routed through a collection of different servers employing logging policies that effectively expose the sender (source IP) and receiver (destination IP) of the message. As a result, entities on the Internet that are not making an effort to be anonymous are leaving trails of their activities behind regardless of what they access on the Internet or where they gain access to the Internet. Fortunately, as noted in the preceding chapter, there are a number of anonymising PETs which leverage off of current Internet protocols to provide varying degrees of anonymity [87, 27, 50, 53, 80, 90, 99, 110].

Though anonymising PETs are making it easier for legitimate users to remain anonymous, they are also making it easier for those wishing to exploit the benefits of anonymity. Examples of this exploitation include using anonymising networks to send spam and commit click fraud.

Click fraud is the intentional clicking on Pay Per Click advertising [17, 44, 121] and falls into two categories. In the first category, an attacker (possibly an unscrupulous competitor of an advertiser) generates false clicks on an advertiser's adverts in order to deplete their budget. In the second category, a fraudulent Web site owner configures a fake HTTP traffic generator [1] to generate traffic to his site (as well as click on the ads – indirectly generating revenue). Daswani and Stoppelman [34] present a case dealing with this type of fraud on a larger scale (the fraudsters used a botnet of approximately 100,000 machines).

[1] Clicking Agent, http://www.clickingagent.com

Usage of an anonymising network to commit click fraud makes it extremely difficult for the advertising company to detect whether or not the clicks being paid for are genuine. Exposing the fraudster involves exposing the users behind the anonymity network or, simply put, attacking the anonymity network.

Exposing a click fraud scheme is important in this work because it is a valid motive for a search engine. Though there may be other reasons for a search engine to attack an anonymity network, we focus on this scenario because it is feasible in the world of search as we know it today. Search engines are inextricably intertwined with advertising [70] and, as click fraud remains a threat [88], it has the potential to influence their bottom line.

Several types of attacks on anonymity have been proposed and discussed in detail within the field of anonymity networks [123, 81, 125]. Attackers have been viewed as entities that are able to monitor all traffic to and from machines on the network as well as collaborating participants of the anonymising network which pool their resources and knowledge of the network in order to expose legitimate members. If we are to be in a position so as to expose schemes the likes of click fraud, we must investigate the implications of an attack which has received little attention, i.e., an attack from outside of the anonymity network (this is the same perspective of a search engine).

The intention of this chapter is as follows: we present and discuss two principles that, when combined, may allow for a successful external attack against an anonymising network. The first principle depends on a characteristic that is common amongst anonymising networks: the unlikelihood of the sender of a message being the machine that actually delivers the message. The second principle is the ability of an attacker to recognise related requests to an end server through a shared identifier of some kind. We apply these principles in a simulated external attack against an existing anonymity model (crowds) and present the results.

This rest of this chapter is structured as follows: in section 4.3 we briefly discuss the field of anonymising technologies and provide an overview of the crowds anonymity model. Section 4.4 makes a case for attacking crowds, i.e., why it is that crowds is an eligible candidate for our attack. In section 4.5 we discuss the assumptions made for the attack against crowds. Section 4.6 then moves on to discuss the attack itself. The simulated attack is then presented, the results of which are discussed in section 4.7. This chapter is then concluded in section 4.8.

4.3 Background

Pfitzmann et al [94] define anonymity as the "*state of being not identifiable within a set of subjects*". We view the subjects in question as machines that participate on a network. An attacker plays the role of a Web server and is said to have identified a machine on the network when he can make educated guesses as to which requests on the network originated from that machine.

Pfitzmann et al [95] discuss three forms of anonymity that are possible on a network:

> Receiver anonymity - referring to the receiver of a message. The sender may be known, the message itself may be observed but the receiver of the message is anonymous (the degree of anonymity associated with the entity is discussed later in this section).
>
> Sender anonymity - referring to the entity from which the message originated. This is similar to receiver anonymity except it applies to the sender only.
>
> Unlinkability of Receiver and Sender - this form of anonymity hides the relation between the sender and the receiver of a message.

There are a number of anonymity models that employ numerous techniques to provide anonymity in one form or another. Chaum [26] employs the use of *mix* machines to delay the delivery of encrypted messages so as to hide the source of the message (typically used for email). Wright et al [122] provide an overview on a host of different mix technologies as well as on the current state of research within the field of anonymity.

Crowds [100] offers various degrees of anonymity through the notion of blending into a crowd. The degree of anonymity, with respect to the sender of a message, is defined as an informal continuum ranging from *absolute privacy* to *probable innocence* and eventually *provably exposed.* Refer to [39] for more detail on quantifying anonymity.

Each member in a crowd (a jondo) receives and forwards messages on behalf of any other member in the crowd. When a jondo decides to send its first message via the crowd, a random path must be configured to serve as the route taken for all messages sent by that jondo from that point onwards. The route is established as follows:

1. The sender picks a random jondo and forwards the message to him.
2. This jondo then flips a coin which determines whether to forward the message to yet another jondo (in which case this step is repeated) or to send the message to the end server.

Upon receiving a response from the end server, the message is routed back along the path to the initial sender of the message.

The authors of crowds define three types of attackers:

- A local eavesdropper is defined as someone that can monitor all communication from a single computer (jondo). Though no sender anonymity is offered from this type of attack (since the attacker can see all messages initiated from the computer), Rieter and Ruben describe how receiver anonymity (the end server) tends towards *beyond suspicion* as the size of the crowd tends towards infinity.

- Collaborating crowd members are seen as jondos that pool their information regarding the crowd together in an effort to expose crowd members. An attack is described where collaborators on a path of jondos used to send a message try to determine if their immediate predecessor on the path is the sender. Reiter and Rubin conclude that *probable innocence* is guaranteed for sender anonymity as long as $n >= 3(c+1)$ where n is the size of the crowd and c the number of collaborating jondos. Wright et al [124, 123] formally define this type of attack (the *predecessor attack*) and examine it within a broader scope.

- The end server: Rieter and Rubin make a strong case for sender anonymity against an attack from the end server. They suggest that the anonymity of the initial sender of the message is *beyond suspicion*.

Since this chapter concerns itself with an attack launched from the end server (an external attack), we discuss details of this attack and assumptions made for the attack in sections 4.5 and 4.6.

4.4 The Case for an External Attack

In this section we discuss the elements that contribute to making a case for attacking crowds. The two principles that must apply to make crowds an eligible candidate for an attack are as follows:

1. The unlikelihood of the sender being the machine that ultimately delivers the message.

2. Recognition of related requests through the usage of a shared identifier.

In satisfying these two principles, we move on to discuss the attack itself.

4.4.1 Principle 1 - an unlikely sender

Reiter and Rubin [100] state, that from an end server's perspective, each member of a crowd is equally likely to have been the sender of a particular request. If I denotes the event where a member of the crowd initiates a message and S the event where a member of the crowd sends the message then $P(I \cap S|k) = 1/n$ where k is the path length (dependent on the probability of forwarding a message in the crowd) and n the number of members in the crowd.

Note that the value of k actually has no impact on the probability of the sender delivering his own message. The reason is that at step $k - 1$ the currently chosen member of a crowd must decide where to forward the message to and since he is as eligible a member as anyone else in the crowd, including the sender, it is reasonable to say that the probability of the sender being chosen is $1/n$. Since the decision taken at step k - 1 is independent of the size of k and will occur for all values of k greater than 1, we say $P(I \cap S|k) = P(I \cap S)$.

This suggests that as n tends towards a significantly large number it is unlikely that a participant in a crowd will issue requests to an end server when it is also the sender of the request. The first principle of our attack has therefore been satisfied.

4.4.2 Principle 2 - recognising related requests

The first principle allows an attacker to start making assumptions about the machines that are part of the anonymising network (the crowd in our case). Being in a position to make these kinds of assumptions is not entirely sufficient when attempting to compromise the anonymity of the sender though; a mechanism to recognise requests must exist, i.e., a way in which to relate requests to one another (and ultimately the sender of the requests).

The mechanism adopted in this chapter is generally accepted by the Web community and is in widespread use on the Web as we know it today: that of Web servers issuing cookies to Web browsers [2].

Cookies allow Web servers to maintain state and effectively recognise repeat visits from users to a given URL. When a new user requests a page from a Web site, the response headers may include a unique identifier that the user's browser would then store in its cookies file on the hard drive. Further requests to the Web site will include this identifier. The usage of cookies will ensure that all related requests share a unique identifier. Although the

[2] Using Cookies with CGI, W3C Journal Volume 1 (available online at http://www.w3j.com/4/s3.shishir.html)

authors of crowds suggest disabling cookies on the jondo agent, support is available for it.

By supporting cookies in a crowds environment we have satisfied our second principle for an external attack against an anonymising network.

4.5 Assumptions

In our attack against crowds we make four assumptions:

1. All participants in the crowd have cookies enabled.

2. All participants forwarding requests in the crowd will not strip out cookies.

3. There is a large enough group of colluding servers to ensure that a large number of requests from the crowd are forwarded to at least one of these servers. The large group of colluding servers allows us to define a cookie farm: a depot from which colluding servers lookup and assign cookies to participants in the crowd.

4. In issuing cookies from the cookie farm to members of the crowd, the colluding servers are able to learn which machines (jondos) are part of the crowd. This assumption is not entirely unreasonable. If a machine that has already been handed a cookie starts making requests to the colluding servers using a different cookie (or without any cookie at all), the colluding servers can mark the machine as a potential member of a crowd (due to it acting as a proxy for other machines).

Since it is unlikely for a participant p_1 to represent himself when making a request to an end server, we can say that it is equally unlikely that a cookie issued to participant p_1 actually belongs to p_1.

When the colluding servers receive a request from a participant in a crowd where no cookie has been issued, they are now in the position to make the following assumption: in issuing cookie C to participant P it is highly unlikely that C actually belongs to P. The postulate drawn is therefore as follows:

For a significantly large crowd, P_i is not the owner of C_k

where i denotes any member of the crowd that has made a request and k denotes the cookies that were issued to P_i.

4.6 The Attack

This section describes the attack through the aid of an example. The example presented uses a small crowd consisting of three participants (p_1, p_2, p_3). Although the first principle presented in this chapter relies on a significantly large crowd, we use a small crowd so as make it easier to appreciate the gist of the attack.

When participant p_1 initiates a request for the first time in the crowd he will typically select a random member in the crowd (let this be p_2) to whom the message will be forwarded. This member will then decide whether or not to send the request to the server in question or forward it to another random member of the crowd.

The act of forwarding the message to a random member versus sending it to the server results in the configuration of a random path used to route messages through for the sender. Upon establishment of the path, it will serve the sender until the central authority of the crowd (the *blender* [100]) issues a command to all participants to reset their paths, for example, in the event of a participant joining the crowd. Table 4.1 depicts a mapping of senders (S) to their respective participants that will ultimately be responsible for delivering the message (S').

S	S'
p_1	p_2
p_2	p_3
p_3	p_1

Table 4.1: A map of senders and participants responsible for delivering the message.

The assumptions made in section 4.4 allow the colluding servers to determine if the sender represented by S' has been issued a cookie from the server. Since the cookie farm is a central repository of cookies and all colluding servers are using the cookie farm, the colluding servers are able to begin mapping out which cookies have been issued to which participants in the crowd. Under the postulate that it is highly unlikely for a sender to represent himself, the colluding servers can say with a fair degree of certainty, that since cookie c_1 has been assigned to participant p_2 it is unlikely that cookie c_1 actually belongs to p_2, i.e., the colluding servers can start building sets of cookies that have been assigned to participants and moreover, sets of cookies that are likely to belong to participants. Let A_i denote the set of cookies handed to p_i. In our example the servers know the following:

$A_1 = \{c_3\}$
$A_2 = \{c_1\}$
$A_3 = \{c_2\}$

Let B_i denote the set of cookies that p_i could simply not be (for example, the case where new cookies are generated and handed out to new members joining the crowd). If we define C_i as the set of cookies that p_i is likely to be then, using our small sample and concentrating on p_1, we can say

$B_1 = \emptyset$
$C_1 = \bar{A}_1 \bigcap \bar{B}_1 = \{c_1, c_2\} \bigcap \{c_1, c_2, c_3\} = \{c_1, c_2\}$

With the crowd we have described thus far, the colluding servers are in a position to make the following assumptions about each member:

p_1 does not have the cookie c_3 it probably has cookie c_1 or c_2.

p_2 does not have the cookie c_1 it probably has cookie c_2 or c_3.

p_3 does not have the cookie c_2 it probably has cookie c_1 or c_3.

Now consider the scenario where another participant joins the crowd. Because the paths used by each of the senders has to be “forgotten”, it is not likely that the same participant will be mapped to the same sender when making future requests to the colluding servers. If the cookie issued to each participant S has not been deleted upon the crowd resetting then let table 4.2 depict the routes that have been configured as a result of another jondo joining the crowd.

S	S’
p_1	p_3
p_2	p_1
p_3	p_4
p_4	p_3

Table 4.2: A map of senders and participants upon a new jondo joining the crowd.

Since new participants are now representing the same cookies, the servers are in a better position to make assumptions as to which cookies belong to which machine(s) from the crowd. With what the servers know about the cookies assigned to the crowd, and with the postulate that a cookie assigned to a participant probably does not belong to the participant, we know:

$A_1 = \{c_3, c_2\}$
$A_2 = \{c_1\}$
$A_3 = \{c_2, c_1, c_4\}$
$A_4 = \{c_3\}$

$B_1 = \{c_4\}$
$B_2 = \{c_4\}$
$B_3 = \{c_4\}$
$B_4 = \{c_1, c_2, c_3\}$

$C_1 = \bar{A}_1 \bigcap \bar{B}_1 = \{c_1\}$
$C_2 = \bar{A}_2 \bigcap \bar{B}_2 = \{c_2, c_3\}$
$C_3 = \bar{A}_3 \bigcap \bar{B}_3 = \{c_3\}$
$C_4 = \bar{A}_4 \bigcap \bar{B}_4 = \{c_4\}$

Note that B_1, B_2 and $B_3 = \{c_4\}$. We know each of these participants could not possibly have c_4 since this cookie did not exist when they were part of the crowd. Evidently, this attack is much harsher on new members to the crowd. In the case of p_4, the colluding servers know that it cannot be holding cookies c_1, c_2 or c_3 since the machine was not part of the crowd when those cookies were handed out. Therefore, they can with a fair degree of certainty assume that $B_4 = \{c_1, c_2, c_3\}$ and as a result, p_4 is holding the c_4 cookie.

4.7 A Simulation

To test the effectiveness of this attack a simulator was written. The simulator implements the behaviour of a crowd on the one hand as well as the colluding servers on the other. Each test using the simulator is carried out as follows:

1. Instantiate a crowd of size n.

2. Initiate a single request from each member of the crowd to the colluding servers (assuming all members of the crowd send approximately the same amount of traffic [48]). In making an initial request, the random path for each member is configured.

3. If a colluding server receives a request from a member of the crowd that does not have a cookie then a cookie from the cookie farm is issued

to it. The colluding servers keep track of which cookies were handed out to which members of the crowd.

4. The colluding servers try to make assumptions about the crowd, i.e., which cookies belong to which members.

5. A new member is added to the crowd.

6. If the crowd is below a predefined limit the test proceeds to step 2, otherwise the test is complete.

As discussed in section 4.4, the number of jondos that participate in the routing of a message (the path length) has no impact on the probability of delivering one's own message. Since the probability of forwarding a message from one jondo to another directly influences the path length, this value was constant for each of the tests and set to 0.5.

Figure 4.1 depicts the results of the tests conducted. The percentage axis indicates the ratio of assumptions made by the colluding servers that were correct. The α axis depicts the degree of certainty required by the colluding servers before making an assumption.

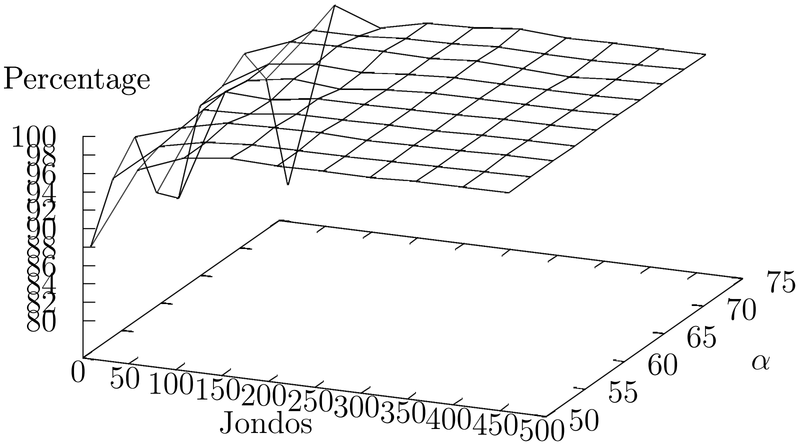

Figure 4.1: Correct assumptions made by the colluding servers

In the case where 200 cookies have been issued to the crowd, with $\alpha = 70\%$ the colluding servers will only try to make an assumption for any member of the crowd (predict which cookies match to which machines) when the number of cookies in the prediction is less than 70% of the cookies handed out. In this case, less than 60 cookies.

Figure 4.2 depicts (as a percentage) the number of crowd members that were exposed for each attack. This suggests that most of the members being exposed in the crowd are new members to the crowd (in doubling the crowd, 50% of the members in the crowd were exposed). Closer analysis of the assumptions made by the colluding servers has revealed that this is indeed the case. With the crowd size only doubling in size there are very few original crowd members that were exposed by the colluding servers.

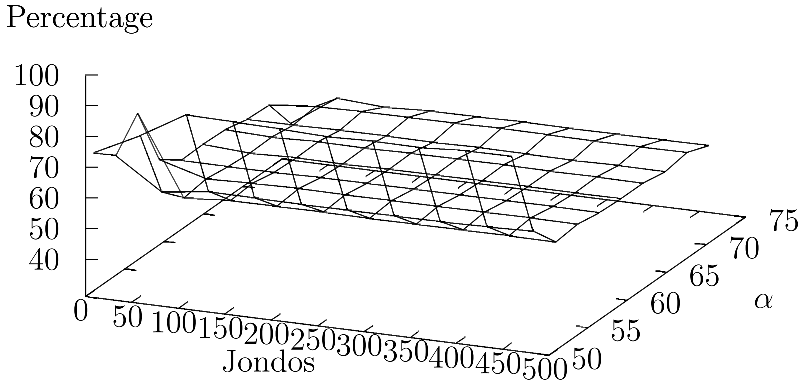

Figure 4.2: Percentage of crowd members that were exposed through an attack.

With this in mind, we conducted another test where we grew the crowd past the double original size limit. Figure 4.3 depicts the results of this test and clearly shows that the number of original members exposed by the colluding servers tend to be far greater as the crowd keeps growing.

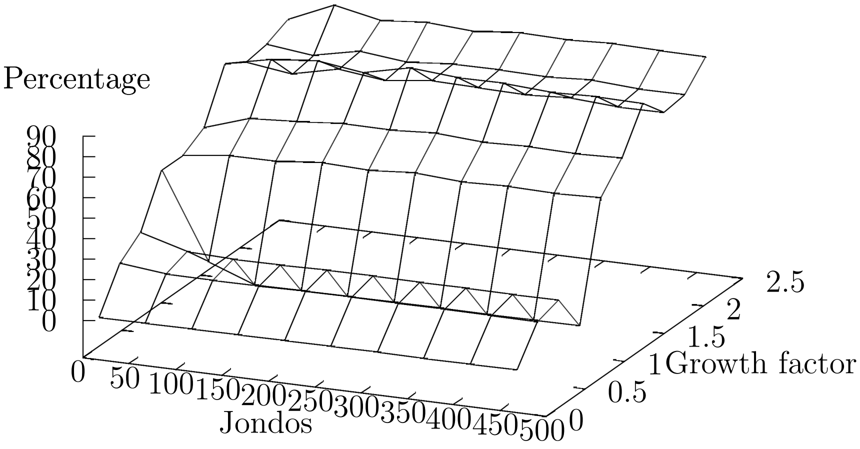

Figure 4.3: Percentage of the original crowd that has been exposed through an attack. A growth factor of 1.0 indicates that the crowd grew by 100% of its original size.

4.8 Conclusion

We began this chapter by discussing the motive for externally attacking an anonymising network. We then presented two principles which, when combined, may result in successfully exposing members of an anonymous network by means of an attack which has received little attention.

The first principle of the attack relies on the sender of a message not representing himself when delivering the message to an end server (this is common in anonymity networks). The second principle depends on a mechanism for recognising related requests; the mechanism used in this chapter was that of cookies. In issuing cookies to members of the crowd under the postulate that a cookie issued to a participant probably does not belong to that participant, the external attacker is able to start creating sets of cookies to which a candidate is likely to belong.

The attack discussed in this chapter was tested by means of a simulation. The results showed that with sufficient growth of the crowd, a significant number of original crowd members may be exposed. The impact of these results may serve so as to shed some light on the often overlooked external attack. This is especially true for crowds. When considering an end server (external attacker) the authors of crowds rate the anonymity of a participant sending a message as *beyond suspicion*. The results of the attack proposed in this chapter suggest that the anonymity of the participant tends more towards *possible innocence*, i.e., "there is a nontrivial probability that the real sender is someone else".

In this chapter, we have shown that under certain conditions the usage of an anonymising network does not guarantee anonymity from an external party (the likes of a search engine). Since search engines have a motive for attacking an anonymity network, the result of which may be successful, it is reasonable to assume that anonymity PETs are not the ideal environment for search privacy.