PRIVACY ENHANCED WAP BROWSING WITH MCROWDS - ANONYMITY PROPERTIES AND PERFORMANCE EVALUATION OF THE MCROWDS SYSTEM

Christer Andersson, Reine Lundin, Simone Fischer-Hübner

Karlstad University, Department of Computer Science

christer.andersson@kau.se, +46547002156, Universitetsgatan 2, 665 86 Karlstad, Sweden
reine.lundin@kau.se, +46547001860, Universitetsgatan 2, 665 86 Karlstad, Sweden
simone.fischer-huebner@kau.se, +46547001723, Universitetsgatan 2, 665 86 Karlstad, Sweden

ABSTRACT

While the mobile Internet provides location-based and other useful services, it also introduces new privacy risks. This paper describes mCrowds, an anonymity technology for the mobile Internet developed at Karlstad University, and discusses its theoretical anonymity properties. mCrowds enables anonymous WAP browsing and can further be used to minimize the disclosure of personal information when using location-based services.

Performance is of key importance for mobile Internet technologies, and has for this reason been an important design goal during the development of mCrowds. This paper therefore also studies the theoretical performance properties of mCrowds and the tradeoff between anonymity and performance. Besides, it provides and discusses the results of a practical performance evaluation of mCrowds. These evaluation results are promising as the overhead in performance introduced by mCrowds is relatively small compared to the total response latency when fetching WAP pages via the mobile Internet.

KEYWORDS

Anonymity, privacy, privacy-enhancing technologies, mobile Internet, WAP, location-based services, Crowds, mCrowds, performance evaluation.

PRIVACY ENHANCED WAP BROWSING WITH MCROWDS - ANONYMITY PROPERTIES AND PERFORMANCE EVALUATION OF THE MCROWDS SYSTEM

1 INTRODUCTION

The mobile Internet is a fast growing technology that offers users new, powerful services, such as advanced location-based and context-aware services. While these services can contribute to enriching the user's experience, they also introduce new privacy risks. This is because new kinds of privacy-sensitive information, such as location information or other information about the user's context, are now transmitted to the content providers. This information can possibly be traced and combined with other personal data collected by traditional means, such as cookies, to create extensive user profiles. Further, even information about device capabilities and user preferences in so called user agent profiles can be privacy sensitive.

In most parts of the Western world, data protection laws, as well as international guidelines and directives (such as EU Directive 95/46/EC [1]), require basic privacy principles to be followed. Also, EU Directive 2002/58/EC [2] directly restricts the processing of location data in the mobile Internet. However, due to the global nature of Internet, an international harmonization of legislation is needed for privacy legislation to be effective. In reality, this task is barely achievable, owing to cultural, political and historical differences between countries and continents [3]. Therefore, it is argued in [4] that there is a need to enforce privacy by technology in mobile Internet (by the use of Privacy-Enhancing Technologies, so called PETs [5]), in addition to privacy legislation.

Performance also plays a much more important role in the mobile Internet than it does in the traditional wired Internet. Mobile networks generally have much lower bandwidth capabilities and more transmission errors than wired networks, which results in higher latency. Furthermore, mobile devices have much smaller screens than stationary computers, which place severe constraints on the graphical user interfaces in these small devices. Hence, these constraints on bandwidth and graphical capabilities must be carefully considered when developing PETs for mobile environments so that PETs for mobile Internet gains user acceptance while still providing an adequate level of privacy.

At Karlstad University, we have developed an anonymity technology called mCrowds, which can be used to minimize the dissemination of personal information on the mobile Internet. It does so by enabling anonymous WAP (Wireless Application Protocol) browsing and by minimizing the disclosure of personal information when using location-based services. mCrowds is based on Crowds, a system for anonymous browsing on the traditional Internet developed by Reiter and Rubin at AT&T Labs in 1997 [6]. In short, Crowds works by grouping users into a large anonymity set, a so called crowd. The crowd then issues requests to web servers on behalf of its members. In mCrowds, we combine the concept of a traditional Crowds system applied in a mobile Internet setting with a filtering functionality tailored to mobile requests. This paper compares the level of anonymity offered by mCrowds in a mobile Internet scenario with the level of anonymity offered by the traditional Crowds system in a wired Internet scenario.

Performance has been one of the primary design goals in the development of mCrowds. The traditional Crowds system was chosen to provide a base for mCrowds, since Crowds as a base is supposed to offer better performance properties than the more common anonymity technologies based on Mix-nets [7], such as Web-mixes [8], Freedom System [9] or Onion Routing [10]. This is because Crowds as a base is supposed to provide better scalability properties and further uses

symmetric encryption throughout. This paper elaborates on the trade-off between offered anonymity and performance overhead that the administrator of a crowd must decide upon when configurating mCrowds. Performance can be enhanced by decreasing the expected length of the virtual paths in the crowd along which the traffic are routed. However, an expected path length with a low value also leads to a lower resistance against internal privacy attacks.

Finally, to evaluate the performance of mCrowds in practice, a performance evaluation was conducted that measures the performance overhead introduced by mCrowds when browsing anonymously on the mobile Internet. To make the conditions more realistic, an experimental crowd was simulated where the crowd was comprised of peers separated by a relatively large geographical distance. The results of the performance evaluation were encouraging, as the performance overhead was relatively small compared to the total latency. One reason for this is that the communication overhead generated by mCrowds takes place in the traditional wired Internet, and another reason is that a performance-enhanced communication protocol was used between the individual peers in the crowd for performance.

The paper is organized as follows. Section 2 discusses Crowds and mCrowds. Section 3 discusses anonymity properties and discusses theoretical performance properties. Section 4 gives the results of the performance evaluation. Section 5 concludes the paper and presents an outlook on further research.

2 RELATED WORK

This section describes Crowds [6], a system for anonymous web browsing on the traditional Internet developed by Reiter and Rubin at AT&T Labs in 1997, and mCrowds [4], which is a PET that enables anonymous browsing on the mobile Internet. The strategy in developing mCrowds was to utilize an anonymity technology for the traditional Internet and apply it in a mobile Internet setting. As our intention was not simply to "port" an existing anonymity technology, the earlier mentioned special characteristics in mobile Internet were taken into account, such as low bandwidth and many transmission errors. Also for this reason, the choice fell upon the traditional Crowds system, since Crowds is supposed to offer good performance properties.

2.1 Crowds

The main idea in Crowds is that one user's action is hidden within the actions of many other users in a so called *crowd* that issues requests to web servers on the behalf of its members. As a consequence, the web server, as well as other crowd members, cannot determine the original sender since the request is equally likely to have originated from any member of the crowd.

A crowd is built up of a number of *jondos* and one *blender*. A jondo is an application that runs on each user's computer. The traffic in the crowd is routed through *virtual paths*, and each jondo has its own virtual path that passes one or more additional jondos in the crowd before reaching the web server. These static virtual paths are torn down and reconstructed on a regular basis, which allows the possibility to include recently added members in the virtual paths.

The role of a jondo is threefold; first it serves as the user's local proxy server to which the user's web browser forwards HTTP requests. In this case the jondo is the first node in the virtual path. Second, it can serve as an intermediate peer in other jondos' virtual paths and, finally, it can be the last jondo in a virtual path, and in this case acts as a proxy server towards the web server.

The blender is a single server responsible for membership management. Before a user can become a member of a crowd, the user must be registered at the blender. When a user registers, all the other members in the crowd are notified of this event. Further, the blender is also responsible for key distribution. It distributes symmetric keys to the individual jondos, which are used for encryption and decryption, respectively, of packets sent between individual jondos.

The Crowds approach offers very good scalability properties as the capacity of the crowd increases linearly with the size of the crowd. This is because the load is distributed equally among the peers in the crowd. However, one potential problem of the Crowds approach is that users equipped with slow Internet connections degrade the overall performance of the crowd. Another issue that has to be considered is the risk of being associated with other users' requests.

2.2 mCrowds

mCrowds is a research prototype of a PET that enables anonymous browsing on the mobile Internet. As in the traditional Crowds system, a crowd in mCrowds is constituted of many jondos and one blender. In our research prototype, we implemented the jondo and the blender as two separate Java applications. The different entities in the crowd communicate with each other using the *enhanced Crowds protocol* [4], a modified version of the communication protocol used in the traditional Crowds system. One difference compared to the traditional Crowds system is that the role of the blender is diminished, since now the symmetric keys used for encryption and decryption are distributed by the jondos themselves using Diffie-Hellman Key Exchange [11]. This is in contrast to the original protocol, where the blender was responsible for the dissemination of symmetric keys. Another difference that is relevant from a performance perspective is that decryption/encryption is no longer performed by intermediate jondos on the virtual path; only the first and the last jondo on the path encrypt and decrypt the packets, respectively.

Anonymity in mCrowds is gained by combining two approaches. First, the concept of the crowd inherited from the traditional Crowds system is applied in a mobile Internet setting to hide the *initiator* of a request. This is done by hiding the IP address of the user's jondo from the content server. This is important since the jondo could be running on the user's personal computer. Second, the user's local jondo acts as a WAP-tailored filter that filters out or anonymizes possibly privacy-sensitive information *within* the request, such as Capability and Preferences Information (CPI), location data or other personal information. Since the user is able to modify the filter settings herself via the GUI of the jondo, this can be seen as simple step towards identity management. The filter functionality could be adapted easily in the future to anonymize new kinds of information in the request, such as information about the context.

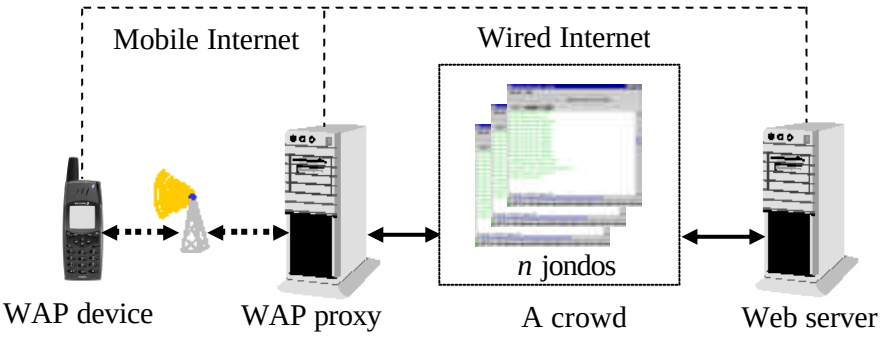

Figure 1: Communication with mCrowds via a WAP proxy.

When a mobile user requests a WAP page from a web server with his/her mobile phone, the request is sent both over wireless and wired Internet, passing a number of different networking entities along the way (see Figure 1 above). Both the WAP proxy and the first jondo on the path are executed in the domain of the user. The WAP proxy[1] is either optional (as in WAP 2.X) or mandatory (as in WAP 1.X). When a WAP proxy is used, the browser in the user's phone is configured to send the request to the WAP proxy; otherwise the request is sent directly to the user's jondo, which in turn is connected to a crowd. More information about the networking entities involved when using mCrowds is given in [4].

[1] Acting as an intermediate between the wireless and wired network.

3 THEORETICAL PROPERITES

Even if the basic anonymity properties in mCrowds are inherited from the theoretical foundations of the traditional Crowds system, some new issues are introduced when moving from a wired Internet scenario to a mobile Internet scenario. This section compares the level of anonymity offered by mCrowds in a mobile Internet scenario with the level of anonymity offered by the traditional Crowds system in a wired Internet scenario. Further, the section elaborates on the trade-off between anonymity and performance that occurs when an administrator of a crowd decides upon the value of the expected path length.

3.1 Anonymity properties in mCrowds

Three dimensions of anonymity are considered in mCrowds, namely the *type of anonymity*, the *potential attackers* and the *degree of anonymity*. These dimensions are described below.

- *Type of anonymity*: the types of anonymity considered are *sender anonymity* and *receiver anonymity*. Sender anonymity means that the identity of a sender is not disclosed to any other communication party in the system. Receiver anonymity means that the identity of the web server is hidden to all other communication parties, except to the sender.
- *Potential attackers*: four types of attackers are considered in mCrowds, namely the *wired domain eavesdropper*, *malicious jondos*, the *web server* and the *wireless domain eavesdropper*. These potential attackers are described below.
    - *Wired domain eavesdropper*: if (a) a WAP proxy is used, the wired domain eavesdropper can observe all messages going to or coming from the "secure domain"[2] shown in Figure 2 below. If (b) no WAP proxy is used, the wired domain eavesdropper can simply observe all messages going to or coming from the user's local jondo (that is, the first jondo on the virtual path). Note that the wired domain eavesdropper replaces the *local eavesdropper* in [6].

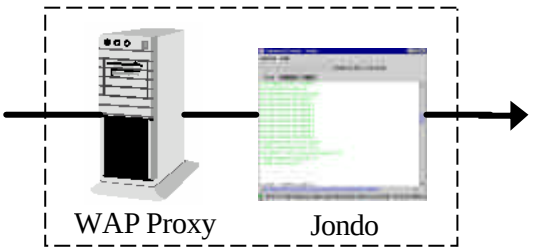

Figure 2: Secure domain when using a WAP proxy.

    - *Malicious jondos*: these are malicious crowd members that work cooperatively to disclose the identity of the sender of a message. Practically, it could also be one attacker controlling many jondos in a crowd.
    - *Web server*: this is the server to which the mobile user connects when fetching a WAP page. Malicious web servers could try to disclose the identity of users in order to misuse information about users.
    - *Wireless domain eavesdropper*: this is an eavesdropper that can eavesdrop on all the information going to and coming from the base station to which the user's requests are sent. The wireless domain eavesdropper does not possess a priori knowledge about the user's exact location and thus cannot eavesdrop directly on the communication going to or coming from the user's mobile phone.

[2] The "secure domain" is not secure by itself. It has to be kept secure, for example by letting the jondo and WAP proxy execute on the same secured computer.

- *Degree of anonymity*: this means that anonymity is measured on a continuous scale ranging between the two extremes *absolute privacy* and *provably exposed* (for more information, see [1]). Three intermediate points are of special interest, namely *possible innocence* (from the attacker's point of view, there is a non-trivial possibility that the real sender is someone else), *probable innocence* (from the attacker's point of view, the sender appears no more likely to be the originator of the request than to not be the originator) and *beyond suspicion* (the initiating sender appears no more likely to have initiated the request than any other potential sender in the system). Of these intermediate levels of anonymity, possible innocence is the weakest and beyond suspicion the strongest.

These three dimensions are combined in the table below to describe the degree of anonymity offered against the potential attackers mentioned above. In Table 1 below, 'P' stands for probability, and the notation P() means that, for large crowds, the anonymity level converges towards the level enclosed within the parentheses.

Table 1: The different levels of anonymity in mCrowds.

	Sender Anonymity	Receiver Anonymity
Wired Domain Eavesdropper	beyond suspicion	P(beyond suspicion) for $n \to \infty$
Malicious Jondos	probable innocence	P(absolute privacy) for $n \to \infty$
Web server	beyond suspicion	N/A
Wireless Domain Eavesdropper	beyond suspicion	beyond suspicion

A similar table was presented for the traditional Crowds system in [6], where some of the anonymity properties differ. First, the wireless domain eavesdropper is naturally not present in [6]. Second, in mCrowds, the sender anonymity against a wired domain eavesdropper is "beyond suspicion", in contrast to the traditional Crowds system, where the level of sender anonymity against the local eavesdropper is "exposed". This is because, in a mobile Internet scenario, all outgoing requests from the aforementioned secure domain have a corresponding incoming request, and hence no new request is created.

If a secure communication link is available between the user's mobile device and the WAP proxy, both receiver and sender anonymity against a wireless eavesdropper is "beyond suspicion". The Wireless Transport Layer Security (WTLS) protocol [12] could be used to create the secure link. However, it should be noted that even if no encryption is used between the user's mobile device and the WAP proxy, advanced equipment and expert knowledge are needed to eavesdrop on the wireless communication.

3.2 Performance properties in mCrowds

A crowd C can be described by the set of parameters $C = \{n, c, p_f, l, r_c\}$ where

- n is the number of members in the crowd
- c is the number of malicious jondos (see section 3.1). This parameter is normally known only to the attacker.
- p_f is a system-wide probability constant chosen by the administrator of the crowd. During the creation of the virtual paths, it denotes the forward probability that a jondo will extend the path to another jondo, instead of ending the path.
- l is the expected length of the virtual paths, that is, the expected average length of the virtual paths
- r_c is the resistance against malicious jondos. $r_c = c_{max}/n$, where c_{max} is the maximum number of malicious jondos that the system can handle while still guaranteeing anonymity.

From a performance perspective, l is obviously an important parameter since each jondo in the virtual path causes an additional overhead in performance. In [6], l is expressed in terms of p_f as

$$l = (2 - p_f)/(1 - p_f) \tag{1}$$

When studying (1), it is obvious that a low value of p_f results in a lower value of l, and thus lower performance overhead. On the other hand, as will be shown below, a low value of l also leads to a low value of r_c. This in turn leads to weak resistance against malicious jondos. Thus, the administrator of a crowd has to make a trade-off between performance (minimizing l) and privacy (maximizing r_c) when deciding upon the value of p_f.

From [6] it follows that, for the anonymity properties to hold in Crowds, and hence in mCrowds, the following inequality must be satisfied.

$$n = \frac{p_f(c+1)}{p_f - 0.5} \tag{2}$$

The inequality in (2) can be rewritten as.

$$p_f = \frac{0.5}{1-(c+1)/n} \tag{3}$$

By exchanging variable c with the constant, c_{max}, the inequality in (3) can be expressed as an equality as shown below. For a given n and p_f, c_{max} is the maximum value of c that still satisfies (3).

$$p_f = \frac{0.5}{1-(c_{max}+1)/n} \sim \frac{0.5}{1-r_c} \quad (\text{if } c_{max} >> 1) \tag{4}$$

The forward probability, p_f, is a probability value that by definition is always bounded in the interval [0, 1]. For the cases when $c_{max} >> 1$, the value of r_c in equation (4) is bounded by the interval [0, 0.5), since the value of p_f would otherwise be outside the scope of a value for a probability. From this, it follows that the p_f in (4) can only be varied in the interval [0.5, 1). For the cases when $c_{max} >> 1$ does not hold, r_c is bounded by the interval [0, 0.5 – 1/n).

From the discussion above, it can be noted that r_c cannot exceed 0.5 (0.5 – 1/n, for the cases when $c_{max} >> 1$ does not hold), regardless of how the value of p_f is chosen. Because of this, the number of malicious jondos must never be equal to or more than n/2. This means that the number of malicious jondos in a crowd must always be less than half of the number of members in the crowd; otherwise, the property of "possible innocence" cannot be guaranteed for a sender.

The relation between the expected path length, l, and the resistance against malicious jondos, r_c, can be expressed by combining equations (1) and (3) as shown below.

$$l = \frac{1.5 - 2r_c}{0.5 - r_c} \tag{5}$$

To further illustrate the relationship between p_f, l and r_c, equations (4) and (5) are plotted in Figure 3 below[3]. These figures can be used by the administrator of a crowd to study how the value of p_f affects the values of r_c and l.

One important factor in the choice of p_f is the expected size of the crowd. Consider a crowd C_1 = {100, c, 0.56, 3.27, 0.1}, which is a relatively small crowd optimized for performance. In order to break the property of "possible innocence" for a sender, an attacker needs to be in control of 10 or more of the jondos in the crowd (that is, c = 10), which cannot be considered an infeasible task for an attacker. If we instead consider a crowd C_2 = {10000, c, 0.56, 3.27, 0.1}, the attacker would

[3] $c_{max} >> 1$ is assumed in both figures.

now need to be in control of 1000 jondos in the crowd to break the anonymity properties (that is, c = 1000). This would make the attacker's task much more difficult. As a rule of thumb it could be argued that the larger the crowd, the lower the necessary p_f and, accordingly, the shorter the expected path length.

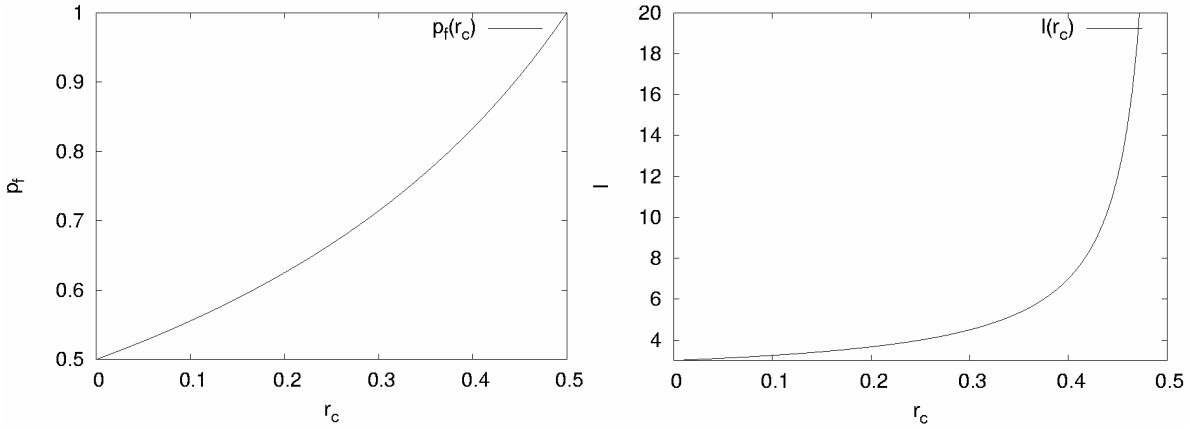

Figure 3: (a) equation (4) plotted (b) equation (5) plotted.

Finally, p_f should preferably not be too close to 1, as this would lead to very long virtual paths. By studying Figure 3 above, it can be noted that l in equation (4) starts to increase very rapidly around $r_c \tilde{} 0.4$, which in turn corresponds to $p_f \tilde{} 0.83$. For this reason, striving for $r_c > 0.4$ runs the risk of resulting in very long virtual paths and hence high performance overhead. Therefore, ideally, the size of the crowd should be large enough to make it infeasible for an attacker to control more than approximately 40% of the jondos in the crowd.

4 PERFORMANCE EVALUATION

This section describes a performance evaluation that was made to measure and analyze the overhead in performance introduced by mCrowds. The hypothesis is that the performance overhead is relatively small compared to the total round trip time when browsing via the wireless network. In order to validate this hypothesis, a performance analysis was done in which the performance overhead introduced by mCrowds was thoroughly isolated and measured. This was done by measuring the time required to fetch WAP pages of different sizes via an experimental crowd in which the length of the virtual paths was varied.

The goal of the performance analysis was to measure the impact on the performance overhead caused by a) *the size of the WAP page to be fetched* and b) *the length of the virtual path.* This was done in two experiments. The first experiment measured the total time spent in the crowd when fetching different WAP pages. As earlier mentioned, the crowd itself is situated on the traditional wired Internet, and thus the measurements in this experiment do not include the mobile Internet. The second experiment measured the total latency for fetching the same WAP pages from a web server to a mobile phone; now via the both the wired and the mobile Internet (see Figure 1 for an illustration of the environment of mCrowds). The second experiment was conducted both in a situation in which mCrowds was disabled and with a typical mCrowds configuration. The purpose of this experiment was to examine how significant the delays caused by mCrowds were in comparison to the total communication time.

As mentioned above, the mobile Internet was included in the test environment of the second part of the experiment. Mobile Internet is known to be "lossy" and unstable by nature. Hence, even though repeated measurements were made, the results of the second experiment do have a somewhat lower quality than those of the first. However, since the emphasis of the performance analysis was on the first experiment, while the results of the second part are more used as an illustrative comparison to those of the first, this does not constitute a great problem.

4.1 Variables

This section introduces the most important variables of the experiment. The terminology describing the different variables is as follows: the *dependent* variables are the explicit outcome of the experiment. *Independent* variables are variables that are explicitly specified by the experimenter. These variables affect the dependent variables. *Uncontrolled* variables have an unknown distribution and they can run the risk of blurring the outcome of the experiment. Hence, explicit measures must be taken to make uncontrolled variables *controlled*.

- **Controlled variables**. The time spent on the mobile Internet when fetching the different WAP pages in the second part of the experiment is an uncontrolled variable due the aforementioned reasons regarding the unstable nature of the mobile Internet. To make this variable controlled, each measurement was repeated 50 times.
- **Independent variables**. The configurations of the workstations on which the jondos are running, including the version of the Java Virtual Machine, constitute an independent variable. Another independent variable is the simulated network between the jondos in a virtual path, which will be described in the next section.
- **Dependent variables**. Both experiments measured the performance overhead introduced by mCrowds in different ways. The first part of the experiment measured the total time spent *inside* the crowd. This measurement does not include the mobile Internet. In the second experiment the total *response latency* needed to fetch the aforementioned WAP pages was measured. This measurement includes the mobile Internet.

4.2 Test environment

The performance evaluation was executed by fetching WAP pages from a web server to a mobile phone emulator, which was running on a Pentium IV 1.4 GHz with Windows XP. In the second experiment the phone emulator was connected to the mobile Internet via a GPRS modem (an Ericsson T68i mobile phone). The GPRS connection was a "sharp" GPRS network operated by Swedish telecom operator Telia [13]. As an intermediate between the GPRS network and the wired Internet, the WAP proxy Kannel [14] was used (running on a PIII 450 MHz with Linux Debian). Kannel was configured to use a local jondo as an HTTP proxy. This jondo was in turn a member of the experimental crowd. In the first experiment the phone emulator was connected directly to Kannel, skipping the mobile Internet. Otherwise, the test environment was the same in both experiments.

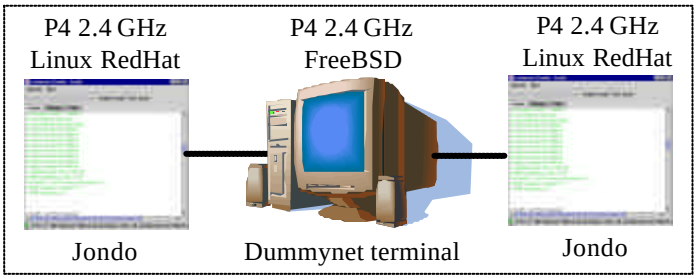

Figure 4: The simulated crowd.

The experimental crowd was comprised of two jondos interconnected via another computer running the Dummynet application (see Figure 4). Since [6] assumes that a crowd spans a large graphical region, Dummynet was used to introduce a constant propagation delay of 10 milliseconds between the jondos in the simulated crowd. Furthermore, Dummynet was also used to limit the bandwidth capacity between the jondos to 10 MBit/s, which is a common capacity of a broadband connection in the country of the authors. The fact that the crowd is only comprised of two jondos does not constitute a problem, since the request will simply be passed back and forth between the two available jondos for the cases when $l > 2$.

4.3 Experimental design

As mentioned earlier, two experiments were conducted. These are described below.

- **First experiment**. Here, the performance overhead generated *inside* the crowd was isolated and studied. This was done by fetching WAP pages of different sizes (1kB, 5kB, 10kB, 15kB, 20kB and 25kB), where the data sizes around 1 – 5kB could represent WAP pages, while the larger data sizes could represent content that could be downloaded with the phone, e.g. games or pictures. Furthermore, since the experimental crowd was assumed to be a crowd optimized for performance (see section 3.2 for a discussion of this) that spans a large geographic region, the length of the virtual path was varied between 2 and 6 jondos[4].
- **Second experiment**. This experiment measured the total round trip time for fetching the earlier mentioned WAP pages of different sizes (1kB, 3kB and 5kB) from the web server to the mobile phone. Thus, the request travels both over the wired and the mobile Internet. In the second experiment a typical length of a virtual path in a performance enhanced crowd was used (4 jondos).

4.4 Test results

The results from the first experiment are plotted in Figure 5(a), where each curve represents the response latency for increasing data sizes for a specific *l*. In this figure, the error bars indicate the standard deviation. Figure 5(b) is a theoretical plot of the sum of the transmission and propagation delay[5], where each curve represents a specific *l*. The values in the curves in Figure 5(b) have been calculated theoretically using equation (6) below. In this equation, *d* is the propagation delay while *bw* is the bandwidth. Figure 5(b) can be said to represent a theoretical lower bound on the response latency for a crowd configured as the one used in the experiments.

$$\textit{transmission + propagation delay} = (l - 1)[(\textit{data size} / \textit{bw}) + 2d] \quad (6)$$

It can be noted that both Figure 5(a) and Figure 5(b) follow the same behavior, as the increase rate of the curves grows with increasing *l*. This is due to the transmission delay, which is positively dependent on both *l* and the data sizes. The propagation delay on the other hand is positively dependent on *l* but independent of the data size.

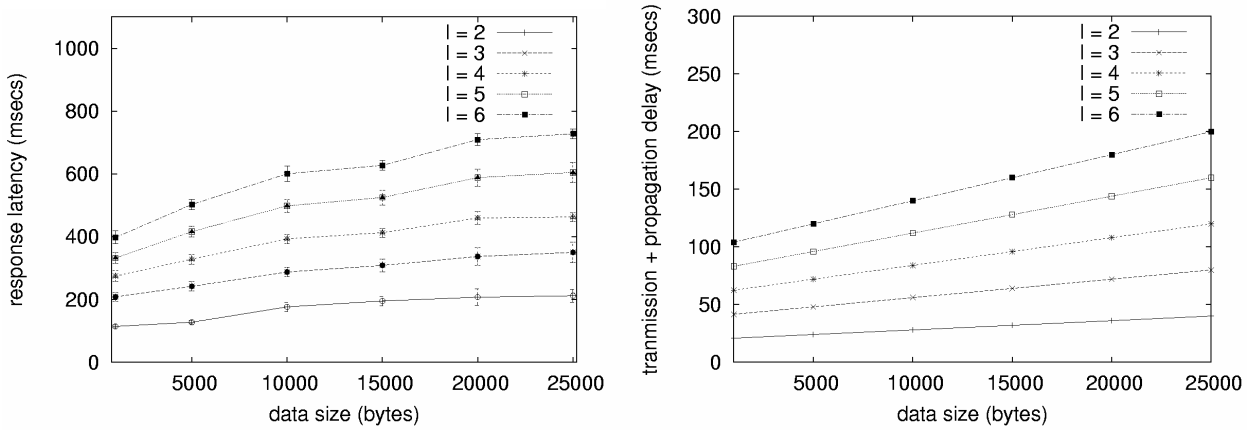

Figure 5: (a) results from first experiment (b) transmission + propagation delay.

[4] The enhanced Crowds protocol [4] was temporarily modified so that the number of jondos could be explicitly specified by the first jondo. This alteration of the protocol did not produce any side effects that affected the performance.

[5] The transmission delay is due to the limitation on the bandwidth capacity of 10MBit/s, while the propagation delay is due to the 10msec delay between the individual peers in the crowd. Further, the transmission delay due to the GET request sent to the web server when requesting a WAP page is neglected.

Studying the results in Figure 5(a), one can see that the response latency never exceeded 800 milliseconds. Thus, the price in performance when browsing anonymously is always less than a second for our experimental setting. Further, comparing Figure 5(a) and 5(b), it can be noted that the transmission and propagation delay only constitute a relatively small portion of the total response latency (in the region of 20 ~ 30 %). The rest of the response latency is due primarily to performance overhead in the jondo software, including encryption/decryption time and the transmission time when sending or receiving data from the WAP proxy, the web server or other jondos. Since mCrowds is a research prototype written in Java, it is probably possible to further decrease the performance overhead, for example by optimizing the code or implementing the jondo software in a programming language optimized for performance.

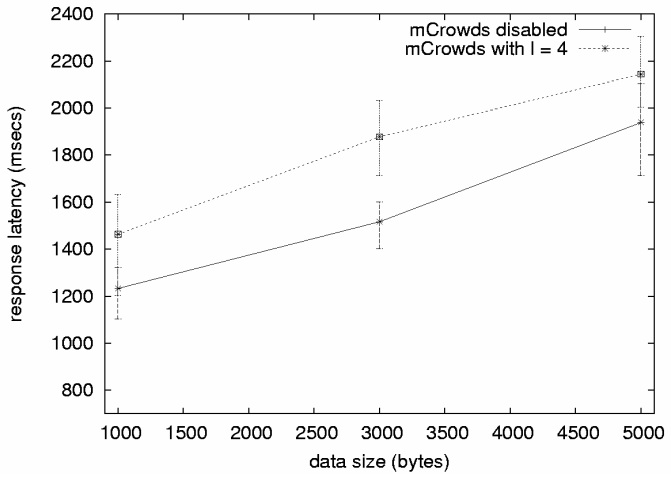

Figure 6: Results of the second experiment.

The results of the second experiment are illustrated in Figure 6 above. The unstable nature of the mobile Internet can be seen in this figure. Here, the error bars indicate the interquartil distance, as normal distribution was not assumed. Comparing the results from the first and the second experiment, it can be noted that the performance overhead is relatively small. Depending on the length of the virtual path, the overhead in performance is approximately between 10 and 30 %, which is a promising result considering that mCrowds is a research prototype.

5 CONCLUSIONS AND OUTLOOK

This paper argued that mobile Internet introduces new privacy risks and that privacy legislation alone is not sufficient to secure informational privacy for users. There is thus a need to develop privacy-enhancing technologies in addition to privacy legislation. Our contribution is mCrowds, which is a privacy-enhancing technology that enables anonymous WAP browsing on the mobile Internet. It was further discussed that performance plays an important role in the mobile Internet, since mobile Internet often suffers from low bandwidth capabilities and high rates of transmission errors. For that reason, performance was one of the primary design goals in mCrowds.

A number of experiments were made to evaluate the performance of mCrowds in practice, in which the performance overhead generated by mCrowds was measured. To make the conditions more realistic, the experiments used an experimental crowd that simulated a large geographical distance between peers in the crowd. The subsequent results of this performance evaluation were encouraging as the overhead in performance introduced by mCrowds was relatively small compared to the total response latency when fetching WAP pages via the mobile Internet. One reason for this is that the communication overhead generated by mCrowds takes place in the traditional wired Internet; another reason is that we enhanced the communication protocol used between the individual peers in the crowd for performance. Furthermore, since mCrowds is only an initial research prototype, there is room for further optimizations that may be able to reduce the performance overhead.

The results of our performance analysis can serve as a comparison to other approaches for anonymity on the mobile Internet. The area of anonymity and identity management (IDM) on the mobile Internet is growing fast, and such technologies will become more common in the coming years. Our contribution in the form of mCrowds can be seen as one of the initial steps.

In the future, we will separate the WAP-tailored filtering functionality from the rest of the jondo functionalities. When separated from the underlying anonymous communication protocol, the filtering functionality will be easier to study and extend. As a first step towards mobile IDM, the filtering functionality could be part of a "personal privacy proxy" for mobile devices offering users basic IDM functionalities. This privacy proxy could in turn be connected to an underlying communication network, e.g. mCrowds. This approach is more generic and would allow for the possibility to compare the mCrowds approach with other kinds of anonymous networks.

6 REFERENCES

[1] European Union, Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data, http://www.cdt.org/privacy/eudirective/EU_Directive_.html, accessed 11 June 2004.

[2] European Union, Directive 2002/58/EC of the European Parliament and of the Council concerning the processing of personal data and the protection of privacy in the electronic communications sector, Brussels, 12 July 2002, http://www.etsi.org/public-interest/Documents/Directives/Standardization/Data_Privacy_Directive.pdf, accessed 20 April 2004.

[3] Simone Fischer-Hübner. Privacy and Security at Risk in the Global Information Society. In: D. Thomas, B. Loader (Eds.): *Cybercrime.* Routledge. London and New York, 2000.

[4] C. Andersson, S. Fischer-Hübner, R. Lundin: Enabling Anonymity in the Mobile Internet Using the mCrowds Approach, Proceedings of IFIP WG 9.2, 9.6/11.7 Summer School on Risks and Challenges of the Network Society, Karlstad/Sweden, 4-8 August 2003.

[5] Registratiekamer, the Netherlands and Information and Privacy Commissioner: Privacy-Enhancing Technologies: The Path to Anonymity, Volume II, Achtergrondstudies en Verkenningen 5B, Rijswijk, 1995.

[6] M. Reiter, A. Rubin: Crowds: Anonymity for Web Transactions, 1997.

[7] D. Chaum, Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms, Communication of the ACM, 24 (2), pp. 84-88, 1981.

[8] JAP, http://anon.inf.tu-dresden.de/, accessed 11 June 2004.

[9] I. Goldberg, A. Shostack, Freedom Network 1.0 Architecture and Protocols, Zero-Knowledge Systems, Inc, Freedom Papers, 1999.

[10] D. Goldschlag, M. Reed, P. Syverson, Hiding Routing Information, in R. Anderson (Ed): *Information Hiding*, LNCS 1174, Springer Verlag, Berlin, 1996.

[11] W. Diffie, M. Hellmann, New Directions in Cryptography, IEEE Transactions on Information Theory, November 1976.

[12] WAP Forum, The Wireless Application Protocol, http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html, accessed 11 June 2004.

[13] Telia, http://www.telia.se/, accessed 11 June 2004.

[14] Kannel: Open Source WAP gateway. http://www.kannel.org, accessed 11 June 2004.