Practical Anonymous Communication on the Mobile Internet using Tor

Christer Andersson
Department of Computer Science
Karlstad University
665 86 Karlstad, Sweden
Email: christer.andersson@kau.se

Andriy Panchenko
Department of Computer Science
RWTH Aachen University, Informatik IV
Ahornstr. 55, D-52074 Aachen, Germany
Email: panchenko@i4.informatik.rwth-aachen.de

***Abstract*— This paper proposes and evaluates several architectural designs for enabling anonymous browsing on the mobile Internet. These architectural designs make use of the Tor network in a mobile setting for the provisioning of anonymity to mobile devices. We compare several architectural designs with respect to their anonymity and performance properties. In particular, we are interested in finding a trade-off between anonymity and performance. We also evaluate the architectural designs against other criteria such as practicality, usability, availability, and trust. We show that the most preferable option – given a powerful mobile device and some optimizations in the Tor protocol– is the option where the Tor client is run directly on the mobile device.**

I. Introduction

Privacy is at stake in the mobile Internet[1] (see, e.g., [2]). In this paper, we propose to safeguard privacy in the mobile Internet by enabling anonymous communication; specifically, we suggest to apply the Tor network [7] in a mobile Internet setting. The motivation for using Tor is that we seek a solution that is possible to deploy today, and Tor is the currently most widely used distributed anonymous overlay network.

This paper evaluates the anonymity and performance properties of several architectural designs that employ Tor in a mobile setting, where different design options include where the Tor client is run and which settings are used in Tor for path construction. Here, the degree of anonymity is quantified using the Crowds-based metric [11], while the performance is evaluated by a network measurement involving the real Tor network. In particular, we are interested in finding a trade-off between anonymity and performance. The design options are also evaluated against other criteria, such as practicality, usability, availability, and trust. Finally, we summarize which design options were most successful in meeting most criteria.

The scope of this paper is restricted to anonymous browsing on the mobile Internet. We do not consider, for instance, privacy-enhanced Location Based Services (LBS) or anonymous phone calls, and, thus, approaches such as [4], [8] are outside the paper's scope. Lastly, although this paper's application domain is mobile Internet, many of the results are also valid when applying Tor in the traditional Internet.

[1]In this paper, mobile Internet entails accessing the web via a Public Land Mobile Network (PLMN) – usually via GSM/GPRS/UMTS/EDGE.

The paper is structured as follows: anonymous communication and Tor are introduced in Section II, while Section III describes our proposed system architecture and its several design options. Then, we present the results from the evaluations of the anonymity, performance, and other system properties in Sections IV-VII. Related work is then presented in Section VIII, before the paper is summarized in Section IX.

II. Background

A. Anonymity

Anonymity can be defined as "the state of being not identifiable within a set of subjects, the anonymity set" [10]. Sender anonymity means that a message cannot be linked to the sender, while receiver anonymity implies that a certain message cannot be linked to the receiver of that message [10]. Anonymity both involves preserving the confidentiality of user data in the application layer (application level anonymity) and hiding the network identifiers of the communication partners in the network layer (network level anonymity). The focus of this paper lies mainly on network level anonymity, although we also discuss application level anonymity.

B. Introduction to the Tor Network

Tor [7] is a widely distributed overlay network for anonymizing network traffic. A common usage scenario for Tor is anonymous Internet browsing. Two main components of Tor are the *Tor clients* and the *Tor servers*. The Tor clients constitute the actual user base of Tor. When communicating with other parties (e.g., content providers), the Tor clients set up virtual paths, consisting of several (normally three) Tor servers. These virtual paths constitute anonymous communication channels through which the Tor clients communicate with their respective communication partners.

When setting up the virtual paths, the Tor clients use layered public-key encryption (see Figure 1)[2]. This strategy ensures that both the external communication partner (*D* in Figure 1), as well as the second and third Tor server, cannot determine which Tor client issued which content request. During this process, the sender establishes a shared symmetric key with each Tor server in the path, which are later used during data transfer (avoiding further use of asymmetric encryption).

[2]In Tor, the procedure in Figure 1 is done in several protocol steps.

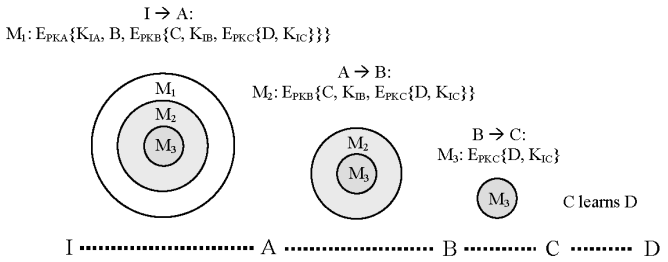

Fig. 1. Illustration of path setup between a sender *I* and an external communication partner *D* (through *A*, *B* and *C*) using layered encryption. PK_A, PK_B, and PK_C are the public keys of *A*, *B*, and *C*. K_{IA}, K_{IB}, and K_{IC} are shared symmetric keys between *I* and the path intermediaries.

III. Proposed System Architecture

The use case assumed throughout this paper is a user browsing the mobile Internet using a mobile client connected to the Tor network. Here, a number of building blocks are needed to enable user anonymity: a mobile device in the wireless domain, a filtering proxy, a Tor client connected to the Tor network, and a content provider on the wired domain. This section describes these subparts and their interplay.

A. Mobile Device

During browsing, the user requests content from a content provider using his mobile device (for instance, a laptop using a GSM/GPRS modem or a mobile phone). This triggers a content request to be sent from the mobile device to the wireless domain. If the Tor client is placed on the user's mobile device, the request is sent via the wireless domain to a randomly selected Tor server. Else, the request is sent via the wireless domain to a Tor client on a stationary computer, which, in turn, sets up the connection with the Tor network.

B. Tor Client

In our application design, the user initiates a path setup in the Tor network between the Tor client and the content provider. While Tor servers normally resides in the wired Internet (e.g., for bandwidth reasons), there are some viable options we consider regarding the placement of the Tor client:

1) The *mobile Tor client* option entails that the Tor client is placed on the user's mobile device;
2) The *home Tor client* option implies that the Tor client is deployed on the user's stationary computer. In this case, the user connects his mobile device to this computer. Now, two options exist: *(i)* either the user only runs a Tor client on on his computer; or, *(ii)* the user both runs a Tor client and a Tor server on his computer;
3) The *third party Tor client* option implies that the Tor client is hosted by a third party, such as the Internet service provider or the mobile operator.

As bandwidth is often a scarce resource in the mobile Internet, there may be a point in saving performance by relaxing the path setup settings in Tor. Thus, we consider several path setup settings in the Tor client:

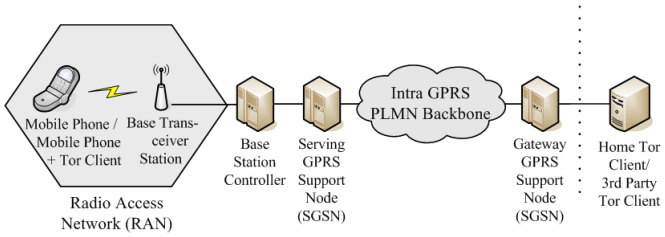

Fig. 2. Wireless domain assuming a WAP 2.0 with a GSM/GPRS PLMN.

- *Standard Tor settings* uses the fixed path length three, and requires that Tor servers in the path should originate from different countries/subnetworks. This is the default settings in the Tor client *OnionCoffee* (see Section VI-C);
- *Performance settings* entails a path length of two and no special requirements regarding the Tor servers;
- For the *proxy settings* the path length is one, and, thus, these settings resemble a one-hop anonymity proxy.

C. The Wireless Domain

The wireless domain includes a Public Land Mobile Network (PLMN) operated by the mobile operator. This paper assumes a GSM/GPRS network (see Figure 2), but there are other options, such as UMTS. Many identifiers are used in the PLMN, including: Mobile Station International ISDN Number (MSISDN), International Mobile Subscription Identity (IMSI), and International Mobile Equipment Identity (IMEI). These identifiers are usually not disclosed outside the PLMN or sent in plain. Instead, temporal identifiers are often used before an encrypted communication line has been established. Yet, in some devices it is possible to enable the automatic disclosure of some of these identifiers to content providers[3].

D. Filtering Proxy

A user may be identified (or, at least, the size of the anonymity set may be reduced) by information in the content request. A filtering proxy, which in this paper is assumed to be placed on the device hosting the Tor client, remedies this situation by allowing the users to specify general rules and patterns for modifying the content request so that personally identifying information can be removed from the content request[4]. Table I depicts an example of a content request for a SonyEricsson W800i phone. This request is always seen in plain by the content provider (and the third Tor server in the path for the cases when end-to-end encryption is not used). In the extreme case, a request may serve as a unique identifier for the content provider. Fields that may help a content provider or Tor server to narrow down the search space includes:

- *User-Agent* – for instance, a user may stand out because he is using a new and uncommon mobile phone. Besides, the mere fact that someone owns an expensive phone constitutes sensitive personal data [9];

[3]E.g., with a Sony Ericsson W800i it is possible to disclose the IMEI.

[4]If the filtering proxy is hosted by a third party, users could be given the possibility to administrate their settings via e.g. a web interface.

- *Accept-Language* – the requested language may enable an attacker to narrow down the search space. For instance, the user may request content in a less common language;
- *CPI fields* – capability and preference information (CPI) in so-called User Agent Profiles allows the content provider to generate content tailored to the characteristics of the requesting mobile device. CPI information have been shown to contain privacy-sensitive information [9].

TABLE I
EXAMPLE OF A WAP REQUEST. THE STARS MARK THE PLACE WHERE THE IMEI NUMBER WOULD BE SITUATED IF THE OPTION TO DISCLOSE THE IMEI IS ENABLED.

```
http://wap.aftonbladet.se/ HTTP/1.1
host:wap.aftonbladet.se
Accept-Language: sv
If-Modified-Since: Mon 06 Oct 2003 14:12:24 GMT
Accept: application/vnd.wap.xhtml + xml,
    application/vnd.wmlc, image/gif,
    application/xhtml + xml, */*, q = 0.9
Accept-Charset: utf-8, utf-16, iso-8859-1,
    iso-10646-ucs-2, Shift_Jis
User-Agent: SonyEricssonW800i/R1L/**********
    Browser/SEMC-Browser/4.2 Profile/MIDP-2.0
    Configuration/CLDC-1.1
x-wap-profile: http://wap.sonyericsson.com/
    UAProf/W800iR101.xml
```

Although specifying the rules for the filtering proxy is outside the scope of this work, we can note that there is a trade-off between customizing device content and anonymity, as the content provider needs to know some characteristics of a device to to generate content tailored for e.g. the screen size of the device. In this context, techniques such as [5] could help the users with determining their application level anonymity.

E. The Wired Domain

In the wired domain, the content request is tunneled through the Tor network (along a virtual path consisting of a number of Tor servers) before reaching the content server. The Tor network (and the filtering proxy) ensures that the content provider and other Tor servers cannot determine which of all Tor clients issued the request (if the first and last Tor server do not cooperate to de-anonymize the user, see Section IV-B).

F. The Content Provider

The user's communication partner is assumed to be a content provider on the fixed Internet (e.g, hosting a mobile Internet web site). Owing to the Tor network, the content provider is not aware of the identity of the user (as the incoming connection is from the last Tor server in the path)

IV. Evaluation Preliminaries

This section introduces our used notation and assumptions.

A. Notation

- We assume that the users running Tor clients constitute the anonymity set $S = \{u_1, u_2, \ldots, u_i, \ldots, u_n\}$;
- $MO = \{mo_1, mo_2, \ldots, mo_n\}$ denotes the set of mobile operators. Each mo_i is a set containing its registered users;
- $C = \{c_1, c_2, \ldots, c_n\}$ denotes the cells of all cellular networks. At any time, a user is receiving service from one cell $\in C$;
- $TTP = \{ttp_1, ttp_2, \ldots, ttp_n\}$ denotes the set of third party operators hosting Tor clients. Each ttp_i contains its registered customers.

B. Assumptions

1) Users wanting to be anonymous do not explicitly disclose directly personally identifiable data on the application level. Alternatively, they use an intelligent filtering proxy that assures this assumption on their behalf;
2) The first and last Tor server (and the Tor client, if hosted by a third party) are not operated by the same attacker (or several cooperating attackers). Else, it is trivial for the Tor servers to link the user to the content provider (e.g., by applying various forms of timing analysis);
3) The third party Tor client operators (TTP) are trusted;
4) For each user $u_i \in S$ the following properties hold: *(i)* u_i possesses at least one mobile device capable of connecting to the mobile Internet, and *(ii)* u_i are registered to at least one mobile operator $mo_i \in MO$ for the purposes of using arbitrary mobile services;
5) For each $mo_i \in MO$, the number of registered users from S exceeds two (two is a theoretical minimum, as it is not possible to provide anonymity in a singleton set);
6) For each $ttp_i \in TTP$, the number of registered users from S exceeds two (again, a theoretical minimum).

V. Anonymity Evaluation

This section evaluates the degree of anonymity offered by our system architecture and its several design options. The evaluation is divided into three parts, one for each considered path construction settings in Tor. However, prior to the actual anonymity evaluation we first state our assumed attacker model, and then introduce the anonymity metric used in the anonymity evaluation: the Crowds-based metric.

A. Attacker Model

Before we model the attacker, we state attacker objectives:

1) Expose the identities of all communication partners;
2) Expose the sending user;
3) Expose the receiver.

These objectives are listed in the order of severeness (from the user's perspective). To achieve the first objective, the attacker must meet the other objectives. If an attacker succeeds in doing this, the system is defeated. These objectives are discussed further later in the evaluation. Finally, note that a global eavesdropper is capable of meeting these objectives, as he can observe both communication end-points. Adding mechanisms for thwarting global observers (such as dummy traffic) is very costly performance-wise, and is thus not usable in scenarios like web browsing. Further, as the Tor network is distributed world-wide, the virtual paths can be constructed in such a way that the existence of a global observer would be unlikely. Hence, the global eavesdropper is omitted from

the attacker model. It should be noted, though, that law enforcements of several countries / continents can cooperate and thereby collectively constitute a global eavesdropper.

1) Attackers in the Wireless Domain:

- A *local eavesdropper in the wireless domain* in the current cell of the user. This attacker can monitor all traffic sent between the user and the base station;
- The *mobile operator*: vast amounts of personal information are collected and processed by the mobile operators. In most cases, they can be expected to take measures to protect their internal networks from intruders and avoid the disclosure or leaking of information if it is against current privacy legislation, as they otherwise would risk heavy penalties [9]. Still, we cannot exclude the possibility that, e.g., a rogue employee would commit a privacy attack, or that a mobile operator could be required by law enforcement to disclose customer information. European operators also has to retain personal data according to the EC Data Retention Directive 2006/24/EC [1].

2) Attackers in the Wired Domain:

- A rogue *local Tor client eavesdropper* observing the user's Tor client, monitoring the incoming / outgoing traffic from / to the user's personal computer or mobile device (see Figure 3). This attacker could, e.g., be an employee at the user's workplace (if the Tor client is operated from the user's work computer), or a rogue neighbor (if the Tor client is operated from the user's home);
- A *Tor server operator on the first position in the path* (see Figure 3) recording, e.g., IP addresses;
- A *Tor server operator on the last position in the path* (see Figure 3) recording, e.g., connecting IP addresses;
- Sloppy or rogue *content providers* (see Figure 3) trying to identity the sender to, for instance, conduct extensive user profiling. The content provider may be situated in a country with no stringent privacy legislation, and it may be unclear how far the user's privacy is respected.

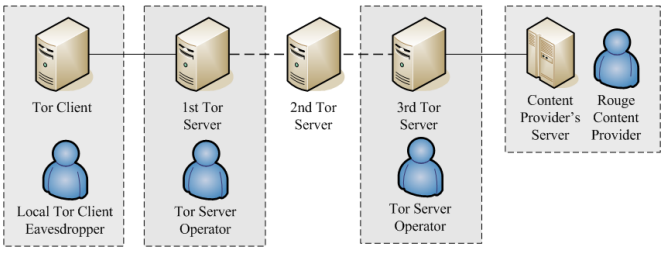

Fig. 3. Attackers in the wired domain.

B. The Crowds-Based Metric

In the Crowds-based metric [11], a user-specific degree of anonymity (A_i) is measured on a continuum between 0 (`provably exposed`) and 1 (`absolute privacy`), were $A_i = 1 - p_i$. The continuum includes the following points:

- `Absolute privacy`: the probability that a given user u_i is linked to a particular message is zero and, hence, $A_i = 1$;
- `Beyond suspicion`: a user u_i in the anonymity set $\mathcal{S} = \{u_1, u_2, ..., u_i, ..., u_n\}$ is `beyond suspicion` if u_i appears no more likely than any other user in $\mathcal{S}$ of being linked to a particular message, that is, $A_i = \max\{A_1, A_2, ..., A_i, ..., A_n\}$;
- `Probable innocence`: p_i that u_i is the sender is less than $1/2$, and hence $A_i \geq 1/2$;
- `Possible innocence`: $\bar{p_i}$ that u_i is *not* the sender is non-negligible, thus $\bar{p_i} \geq 0 + \delta$, where the threshold $\delta > 0$. Hence, we get $A_i = 1 - p_i = \bar{p_i} \geq 0 + \delta$;
- `Exposed`: a given user u_i can be unambiguously linked to a given message, and, hence, $A_i = 0$;
- `Provably exposed`: $A_i = 0$ as above, and it could be proved to a third party that u_i is linked to the message.

The idea to assign 0 and 1 to the continuum and explicitly view A_i as $1 - p_i$ was proposed in [6]. Here, p_i denotes the probability the u_i is the sender / receiver of a message.

C. Anonymity Evaluation: Standard Tor Settings

Below, we present the resulting degree of anonymity assuming standard Tor settings:

1) Local Eavesdropper in the Wireless Domain:

- *Sender anonymity:* assuming an attacker with basic equipment, A_i is `probable innocence` (assuming at least one more user in the same cell $c_i \in C$), as the attacker can only observe (normally encrypted) traffic in the air interface. Yet, with special hardware the attacker may in addition approximate the location of a sending user (by measuring signal strength), and in this case A_i may drop to `possible innocence`. Further, there is a security flaw in the GSM standard that allows an attacker to launch a man-in-the-middle attack by setting up a fake base station, and then silently disable encryption between the user and the fake base station [15]. Equipment to perform this attack can supposedly be found in the black market, but is supposedly expensive. This attack can reduce A_i to `exposed`;
- *Receiver anonymity:* A_i for the receiver is `absolute privacy` for the mobile Tor client option, as end-to-end encryption is used between the Tor client in the wireless domain and the first Tor server. The same goes for the options where the Tor client is residing on the fixed network, provided that the communication between the mobile device and the Tor client is end-to-end encrypted.

2) Mobile Operator:

- *Sender anonymity:* A_i for the sender is `exposed` as mobile operators normally require user identification (for accountability and billing purposes). In GSM / GPRS networks, this is done through the IMSI number in the SIM card together with the IMEI number in the mobile device;
- *Receiver anonymity:* A_i for the receiver against the mobile operator is `absolute privacy`, assuming end-to-end encryption between the mobile device and the Tor client.

3) Local Tor Client Eavesdropper:

- *Sender anonymity:* The degrees of sender anonymity differ between the different design options:

  - A_i is `exposed` for the option where the user runs the Tor client on his stationary computer, as the attacker can observe an encrypted content request leaving the computer that does not correspond to any incoming request. When the user also runs a Tor server, we assume that the attacker can differentiate between incoming Tor requests and incoming wireless requests;
  - For the option where the Tor client is hosted by a third party, A_i is `probable innocence`, as the request could equally likely originate from any other Tor user subscribing to ttp_i's services (we assume that only a subset of S subscribe to the same ttp_i);
  - For the design option where the Tor client resides on the mobile device, the attacker constitute the same attacker as the local eavesdropper in the wireless domain, and, thus, A_i is in the range between `probable innocence` and `exposed`, depending on the capabilities of the attacker.
- *Receiver anonymity:*A_i is `absolute privacy`, as encryption is used between u_i and the first Tor server.

4) First Tor Server:

- *Sender anonymity:* The degrees of sender anonymity differ between the different design options:
  - A_i is `exposed` for the option where the user runs the Tor client on his own stationary computer without also running a Tor server, since the first Tor server knows that u_i is the origin sender. If the user also runs a Tor server, A_i is `beyond suspicion`, as from the first Tor server's perspective, the request could equally likely originate from any other user $\in S$;
  - For the option where the Tor client is run on the mobile device, A_1 is `probable innocence` as the first Tor server only learns that the packet originated from a certain mobile operator $mo_i \in MO$. Still, this narrows down the search to some extent, as we assume that only a subset of the Tor users are using the same mobile operator;
  - For the design option where the Tor client is hosted by a third party, A_i is `probable innocence`, as the request could equally likely originate from any other Tor user. Again, we assume that only a subset of the users $\in S$ subscribe to the same $ttp_i \in TTP$;
- *Receiver anonymity:* A_i against the receiver is `absolute privacy` as layered encryption is used between the Tor client and the last Tor server in the virtual path.

5) Third Tor Server:

- For all options, A_i for the sender is `beyond suspicion`, as the content request could originate from any Tor user. When end-to-end encryption is not used between the sender and the content provider, the last Tor server may be able deduce that the request originated from a mobile device by inspecting the communication content. But as mobile devices are ubiquitous nowadays, this does not affect A_i (see assumptions in Section IV-B);
- A_i for the receiver is always `exposed`, as the last Tor server makes a direct connection to the content provider.

6) Content Provider: the sender anonymity is `beyond suspicion` as the last Tor server initiates the connection.

7) Summary for Standard Tor Settings: in Table II, we summarize the degrees of anonymity for standard Tor settings:

TABLE II

SUMMARY OF DEGREES OF ANONYMITY AGAINST SENDERS AND RECEIVERS.

	Mobile Tor	*Home Tor*	*Third Party Tor*
Loc. eavesdropp. in wireless domain	Send. anon. = `prob innocence` / `poss innocence` / `exposed` Rec. anon. = `abs privacy`	Send. anon. = `prob innocence` / `poss innocence` / `exposed` Rec. anon. = `abs privacy`	Send. anon. = `prob innocence` / `poss innocence` / `exposed` Rec. anon. = `abs privacy`
Mobile operator	Send. anon. = `exposed` Rec. anon. = `abs privacy`	Send. anon. = `exposed` Rec. Anonym. = `abs privacy`	Send. anon. = `exposed` Rec. anon. = `abs privacy`
Local Tor Client eavesdropper	Same attacker as *local eavesdropper in wireless domain* (see above)	Send. anon. = `exposed` Rec. anon. = `abs privacy`	Send. anon. = `prob innocence` Rec. anon. = `abs privacy`
First Tor server	Send. anon. = `prob innocence` Rec. anon. = `abs privacy`	Send. anon. = `exposed` (*Tor client*) Send. anon. = `b suspicion` (*Tor client & server*) Rec. anon. = `abs privacy`	Send. anon. = `prob innocence` Rec. anon. = `abs privacy`
Third Tor server	Send. anon. = `b suspicion` Rec. anon. = `exposed`	Send. anon. = `b suspicion` Rec. anon. = `exposed`	Send. anon. = `b suspicion` Rec. anon. = `exposed`
Content provider	Send. anon. = `b suspicion`	Send. anon. = `b suspicion`	Send. anon. = `b suspicion`

D. Anonymity Evaluation: Performance Settings

For these settings, the degrees of anonymity are the same as for standard Tor settings. The path length is however now two, so the third Tor server in Table II is replaced with the *second Tor server*. Event though the (quantitative) degrees of anonymity are not affected, the robustness of anonymity is reduced for this setting: if the first and last Tor server are operated by the same attacker, he will now notice this immediately (in comparison with the standard Tor settings, where the attacker must, e.g., use timing analysis to discover this). If the attacker ends up controlling the full path, this would only de-anonymize the user completely for the option where the user runs the Tor client on his stationary computer without, in addition, running a Tor server. So, for this particular option, the performance settings should preferably be avoided.

E. Anonymity Evaluation: Proxy Settings

Under these settings, the first and last Tor servers become the same attacker: the *single Tor server*. This attacker has the combined knowledge of the first and third Tor server for the standard Tor settings. The degrees of anonymity against this powerful attacker are summarized in Table III. When the user runs the Tor client on his own stationary computer, the single Tor server can link the sender to the content provider.

TABLE III
ANONYMITY AGAINST *single Tor server* UNDER PROXY SETTINGS.

	Mobile Tor	*Home Tor*	*Third Party Tor*
Single Tor server	Send. anon. = `prob innocence` Rec. anon. = `exposed`	Send. anon. = `exposed` (*Tor client*) Send. anon. = `prob innocence` (*Tor client & server*) Rec. anon. = `exposed`	Send. anon. = `prob innocence` Rec. anon. = `exposed`

F. Observations from Anonymity Evaluation

In this section, we study how the attacker can meet the aforementioned objectives (see Section V-A).

1) *Expose the identities of both communication partners*: the only attacker capable of meeting this goal is the single Tor server under proxy settings (if the user runs only a Tor client on his own computer). For the other combinations of Tor settings and design options, several attackers must cooperate to achieve this goal;
2) *Expose the sender*: this goal may be achieved by the local eavesdropper in the wireless domain, provided that this attacker has powerful enough equipment. Further, the mobile operator can always achieve this goal given current business models. Lastly, both the local Tor client eavesdropper and the first Tor server can achieve this goal (here, the first Tor server can only achieve this goal if the user does not, in addition, host a Tor server);
3) *Expose the receiver*: as the third Tor server under standard Tor settings (second / single Tor server for performance / proxy settings) connects directly to the content server, this attacker can meet this goal.

Above, we can see that the design option where the user runs a Tor client on his own computer without also running a Tor server is problematic. For this reason, the proxy settings should definitely not be used with this design option unless the user in addition runs a Tor server on his stationary computer. However, the relaxed path construction settings may still be acceptable for the design options where the Tor client is placed either on a third party computer or directly on the mobile device. In this case, the combined Tor servers (or single Tor server) cannot reduce the degree of sender anonymity below `probable innocence` (as discussed in Section V-C).

VI. PERFORMANCE EVALUATION

In this section, we describe the results from a performance evaluation in which we measured the performance of our system architecture and its several design options. The real Tor network were used in the performance evaluation, and thus we were conducting a live network measurement. The experiments were conducted during May - July 2007.

A. Experimental Design

We conducted two experiments that we describe in the coming sections. Both experiments were repeated two times:

- The first repetition of the experiments was done solely in the wired domain. It represents the design option were the Tor client is placed either on the user's stationary computer or on a third party company's computer;
- The second repetition of the experiments included both the wireless and wired domains. It represents the design option where the Tor client is placed on the mobile device.

As we always initiate the measurements from the Tor client, all possible combinations of path setup settings and placements of the Tor client are not fully covered in this experiment. Table IV depicts which combinations are fully covered. Although we do not verify this experimentally, placing the Tor client on a high-end mobile device can still be expected to inflict some performance reduction compared to placing it in the wired domain. This is due to the increased amount of Tor protocol traffic that would be sent over the (slower) wireless domain.

TABLE IV
DESIGN COMBINATIONS COVERED IN THE PERFORMANCE EVALUATION.

	Mobile Tor	*Home Tor*	*Third Party Tor*
Standard	2^{nd} repetition (wired & wireless)	1^{st} repetition (only wired part)	1^{st} repetition (only wired part)
Perf.	2^{nd} repetition (wired & wireless)	1^{st} repetition (only wired part)	1^{st} repetition (only wired part)
Proxy	2^{nd} repetition (wired & wireless)	1^{st} repetition (only wired part)	1^{st} repetition (only wired part)

1) Experiment One – Fetching a File from the Content Server: in this application-level use case, we measured the required time for requesting and downloading a file from a content provider to the Tor client. Two file sizes were used – 1 *kB* and 10 *Kb*. The following test algorithm was used:

- For each setting (standard / performance / proxy), repeat:
  - For each file (1 *Kb* / 10 *Kb*), repeat 30 times:
    1) Create a random virtual path and set $i = 1$;
    2) Repeat 30 times: fetch the given file from the content server via the current path. Start the clock when the request is sent from the Tor client and stop the clock when the whole file is downloaded. After the 30 downloads, calculate the average download time x_i (ms). Last, increment i;
    3) Calculate the average y of x_1, x_2,..., x_{30}.

After the test, there will be six combinations of Tor settings and file sizes, presented in the unit milliseconds (ms).

2) Experiment Two – Application Level Throughput: here, the application-level throughput was measured (in Kb/s) when downloading large amounts of uncompressed data from a content server to the Tor client. The actual sending of the requests to the server was not included in the measurements. For each path construction setting (standard / performance / proxy), the average and median throughput was measured 30 times, each time using a new virtual path. Each measurement represents one test case in *MySpeed PC Advanced Edition*[5]. As a comparison, we also measured the throughput – again using MySpeed PC Advanced Edition – without using the Tor network.

B. Variables

The experiments involve the following types of variables:

- *Independent variables* are variables that are explicitly specified by the experimenter. Such variables affect the dependent variables (see below). In the experiments, examples of independent variables are the path construction settings, and the specifications of the computer on which the Tor client and content server are running;
- The *dependent variables* are the explicit outcome of the experiment. The dependent variable in experiment one is the measured time for requesting and downloading a file from the content provider, while the dependent variable in experiment two is the measured application-level throughput between the test server to the Tor client;
- *Uncontrolled variables* have an unknown distribution and they can run the risk of affecting the dependent variables. Explicit measures must be taken to make these variables behave as *controlled variables*. Examples of initially uncontrolled variables in the tests are the bandwidth and processing capabilities of each Tor server in the path. Here, the algorithm for experiment one is an example of a measure that was applied to make them controlled.

C. Test Environment

Below, we state the test environment for experiment one:

- The stationary computer hosting the Tor client was a laptop with the following characteristics: 1.40 GHz Pentium Mobile processor / 512 *MByte* RAM / Windows XP Prof. 2002 Service Pack 2. In repetition one, it was connected to the Internet using a 100 *MBit* wired connection. In repetition two, this computer was connected to the mobile Internet using a mobile phone (SonyEricsson K750i) as a GSM / GPRS modem. The GSM / GPRS modem was connected to the computer using a Bluetooth interface;
- The laptop hosting the Tor client was connected to the real Tor network [7][6]. Virtual paths in Tor were created according to the rules specified by the aforementioned path setup settings (standard / performance / proxy settings);
- The content server hosting the files was a web server at Karlstad University (100 *MBit* bandwidth capabilities). Usually, this server is not under any particularly high load.

Below, we state the test environment for experiment two:

- Similar computer and Internet settings as in experiment one were used. In addition, the earlier mentioned software MySpeed PC Advanced Edition (version 1.3a build 441 with Java 1.6.0_01) was run on the test computer;
- The content provider was a test server situated in London;

In both experiments, OnionCoffee was used, a Java version of the Tor client developed within the PRIME project[7]. To strictly follow the algorithm in experiment one (see Section VI-A.1), the source code of OnionCoffee was partly modified.

D. Experiment One: Fetching a File from the Content Server

- *First repetition:* here, the Tor client was placed on a stationary computer and the time for requesting and downloading a file from a server was measured. The results are depicted in Figure 4 (error bars indicate the average interquartile range). These results could represent either *(i)* required time for requesting and downloading files when using Tor in a non-mobile scenario, or *(ii)* time spent in the wired domain when requesting and downloading files when using Tor in a mobile scenario, where the Tor client resides on a stationary computer;
- *Second repetition:* here, the time for requesting and downloading the files included both the wireless and wired domain. The results are depicted in Figure 5 (again, error bars indicate the average interquartile range). These results represent the time for requesting and downloading files when using Tor in a mobile scenario where the Tor client resides on the mobile device. We have also marked the results from the first repetition of the same experiment (see the dotted lines in Figure 5). As similar conditions were used in the wired domain in both repetitions, the area above the dotted line can be believed

[5]For more information, see http://www.myspeed.com/pe/advanced.html.
[6]For more information, see http://tor.eff.org/.
[7]For more information, see http://www.prime-project.eu/prototypes/anon.

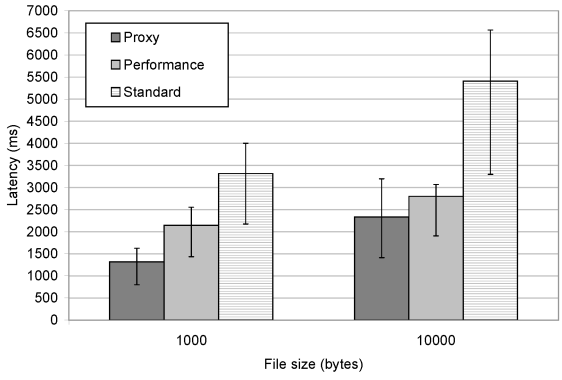

Fig. 4. Experiment one – Tor client placed in the wired domain on a stationary computer. This figure denotes the average time for requesting and downloading a file from the content provider to the Tor client. In a real scenario, the file would still have to traverse the wireless domain to reach the user.

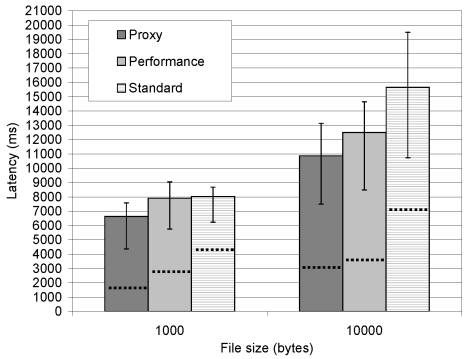

Fig. 5. Experiment one – Tor client in the wireless domain on the mobile device: This figure denotes the average time for requesting and downloading a file from the content provider to the Tor client. As a comparison, this figure also includes the results from the first repetition of the experiment.

to *approximately* correspond to the time spent in the wireless domain, while the part below the dotted line would correspond to the time spent in the wired domain.

E. Experiment Two: Application Level Throughput

- *First repetition:* in this test run, the Tor client was placed on a stationary computer and the application-level throughput in the wired domain when downloading data from a test server was measured. The results are depicted in Figure 6 including the interquartile range. These results can be said to either represent *(i)* the application-level throughput in a non-mobile Tor scenario, or *(ii)* the application-level throughput in the wired domain when using Tor in a mobile scenario, where the Tor client resides on a stationary computer;
- *Second repetition:* the second test was conducted over the mobile Internet. Hence, the subsequent results, which are depicted in Figure 7 (with interquartile range), now includes both the wireless and wired domain. The throughput when not using Tor is denoted in the legend. These results can be said to represent the average application-level throughput when using Tor in a mobile scenario where the Tor client resides on the mobile device.

F. Observations from Performance Evaluation

Below, we point out some observations that can be made when studying the respective results from the two experiments:

- The proxy settings always offered the best average performance and standards Tor settings the worst (except one case in Figure 7, where a few "unlucky" test runs resulted in a worse average for the performance settings than the standard Tor settings). Still, there were great variations between individual test samples for all path setup settings;
- The performance varied greatly depending on the current path. For example, when downloading the 10 *Kb* file using standard Tor settings, the lowest average download time for one test run (i.e., 30 repetitions using a specific path) was 0.6 *s*, while another test run yielded 12 *s*;
- Regarding the second iteration of both experiments, if we would have used a less powerful mobile device than the used laptop regarding processing power (let's say, a regular mobile phone), we may have experienced a higher performance overhead. Currently, there is no available version of Tor that is tweaked for small mobile devices;
- Figure 5 indicates that GSM / GPRS is a bigger bottleneck than Tor. In the future, it would be interesting to repeat repetition two of the experiments using a faster mobile network (e.g., UMTS). This would probably alter the ratio between the time spent in the wireless and wired domain;
- The reduction of the performance overhead between the first and the second repetitions was much greater in experiment two. One probable reason for this is that the propagation delay in the wired domain played a greater role in experiment one, as the sending of the GET request over the Tor network was included in experiment one;
- In both experiments, the average was greater than the median (this is only illustrated for experiment two). For all test cases there were a always couple of test samples with very low performance, and these samples had a bigger impact on the average than the median;
- In the current version of the directory protocol (V2)[8],

[8]The directory protocol version 3 is currently being developed, see http://tor.eff.org/svn/trunk/doc/spec/dir-spec.txt for more information.

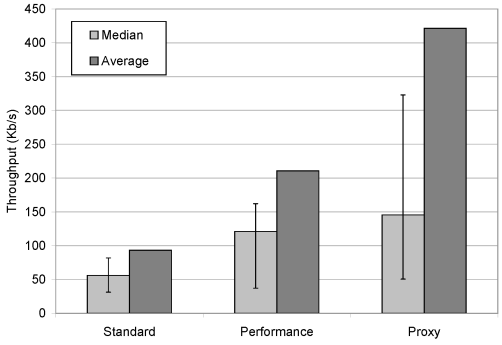

Fig. 6. Experiment two – Tor client is placed in the wired domain on a stationary computer. The application-level throughput when downloading files from a test server is denoted here. As a comparison, the corresponding throughput when not using Tor was measured to be $\approx 13\ Mb/s$.

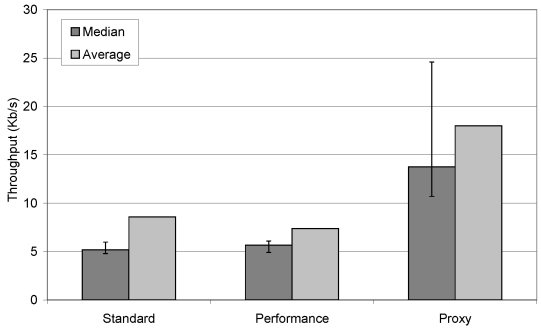

Fig. 7. Experiment two – Tor client is placed in the wireless domain on the mobile device. The application-level throughput when downloading files from a test server is denoted here. As a comparison, the corresponding throughput when not using Tor was measured to be $\approx 40\ Kb/s$.

the directory servers are divided into two categories: first and second level servers. The Tor client first contacts one of the first level servers and requests a so-called network status document that includes the list of active Tor servers and the addresses of the secondary servers where the descriptors of single Tor servers can be download. One problem is that the primary directory servers are usually overloaded. Another problem is that the amount of information that must be downloaded before the Tor client starts building circuits currently goes beyond several megabytes. For instance, we observed that the size of each update from the directory servers was roughly 2.5 MB. Further, the Tor client refreshes its directory information in regular time intervals. If the Tor client runs on a mobile device this may hamper performance, and, it may further be very costly unless the user pays for the connection using a flat rate. There is a clearly a need for a mobile Tor friendly directory protocol, which would allow to minimize the amount of data that must be downloaded. The latter could be realized as a separate information service that would assist mobile Tor users.

To summarize, although Tor in general was shown to impose a significant performance overhead, we can see from the test results that it is clearly possible to improve performance significantly by relaxing the path setup settings. Also, there are other performance bottlenecks, such as the mobile network.

VII. Evaluation of Other System Properties

There are other important properties besides anonymity and performance. This section evaluates a selection of other important system parameters. Specifically, we evaluate whether a given placement of the Tor client: *(i)* offers a high degree of availability; *(ii)* is possible to deploy today with current business models (practicability); *(iii)* avoids single points of trust; and *(iv)* is usable without vast experience on anonymity technologies (usability). Each combination of design option and criterion is graded between one to three, depending on the degree of fulfillment. The grade '$*$' implies fulfillment to a low degree, '$**$' entails fulfillment to a medium degree, while '$***$' means fulfillment to a high degree. In the evaluation, we do not apply any internal weighting between the criteria.

A. Mobile Tor Client Design Option

- *Availability*: Tor can easily be restarted after a crash ($***$);
- *Practicality*: if the communication protocol between the directory servers and the Tor client is optimized (see Section VI-F) this option is practical for mobile high-end devices such as laptops ($***$). It is not yet practical for low-end devices, as the cryptographic operations involved in setting paths are performance demanding [13]. There is currently no version of the Tor client for low-end mobile devices ($*$) – yet, this may change in the near future;
- *Trust*: the user runs the Tor client on his own device ($***$);
- *Usability*: Some user experience is needed, as the user must configure the Tor client on his mobile device ($**$).

B. Tor Client on User's Computer Design Option

- *Availability*: problems may arise when the Tor client running at the user's home for some reason crashes ($*$);
- *Practicality*: as powerful computers with fast broadband access are common nowadays, there are no major issues besides the requirement to set up a fixed Tor client ($**$);
- *Trust*: the Tor client is run on the user's computer ($***$);
- *Usability*: a fairly high degree of user experience is needed, as the user himself must configure the Tor client ($**$). If the user additionally sets up a Tor server, even more knowledge is needed ($*$). Also, the user must be in the vicinity of the computer to alter the Tor client settings (although there are ways around the latter requirement);

C. Third Party Tor Client Design Option

- *Availability*: the third party company would be responsible for keeping the Tor client up and running ($***$);
- *Practicality*: some issues needs to be resolved to make this option practical, for instance, there must be a viable business model for such services (i.e., users must be willing to pay a third party for such a service) ($*$);
- *Trust*: there is a major trust dependency between the user and the third party. Also, the third party may be legally forced to both retain data (for instance, to comply with [1]), and later release it to law enforcement ($*$);
- *Usability*: given the availability of intuitive Tor client interfaces, there are no major issues for usability ($***$);

D. Discussion on Evaluation of Other System Properties

We can see that each design option has its pros and cons. Given a powerful enough mobile device and an optimized communication protocol with the directory servers, however, the design option where the Tor client is run on the mobile device would be the most preferable option.

VIII. Related Work

A. Anonymous Overlay Networks for Mobile Internet

Although anonymous overlay networks specifically targeted for the mobile Internet are relatively uncommon, a handful approaches have been proposed in recent years, including:

- *mCrowds* [3] is a variant of Crowds [11] for enabling anonymity against malicious service providers and rogue users by forming a peer-to-peer network on the wired Internet. All users operate an anonymity proxy on a computer under their control. Content request from mobile clients are randomly routed through a subset of these proxies (starting with the user's own computer) before reaching the content server. The user's proxy also acts as a filter. Compared to our proposal, mCrowds requires the usage of a non-deployed protocol in the fixed network;
- In [13], a framework for providing anonymity in mobile Internet is proposed. The users connect their mobile phones via a Security Provider (SP) to a deployed anonymous overlay network, such as Jap or Tor. The SP acts as a TTP providing an interface between the user and the anonymous overlay network. The SP also helps users

by performing cryptographic operations on their behalf when setting up a virtual paths. A potential problem is that the SP constitutes a single point of failure and trust. Further, the framework in [13] neither presents an anonymity analysis nor a performance evaluation;

B. Approaches for Enhancing/Measuring Performance of Tor

Below we introduce some related work on performance:

- There are some projects aiming to reduce the performance overhead in Tor by optimizing the virtual path construction. One example is ongoing work at RWTH Aachen University on improving the performance in Tor by, e.g., downloading the bandwidth status from Tor servers, measuring the round trip time on the created paths, and optimizing path construction based on certain heuristics. Another example the work presented in [12] that seeks to optimize the selection of the middle node in the path with respect to the overall latency of the path;
- In a performance comparison between Tor and AN.ON, Wendolsky *et al.* observed that the performance in Tor varied depending on the time of the day (European mornings offered slightly better performance than afternoons) [14]. We did not observe a similar behavior in our data set. The reason for this can be twofold. First, their tests was conducted about one year earlier than ours (early 2006); the number of Tor servers have increased significantly during this time. Second, we did not apply the same tests on our data set as in [14], and thus we may have missed to spot minor time-related variations.

IX. Conclusion & Outlook

This paper proposed architectural designs for enabling anonymity on the mobile Internet by applying Tor in a mobile setting. Several options were discussed regarding the placement of the Tor client, and the rules regarding path setup. From these evaluations, we could make the following observations:

- In the anonymity evaluation, we showed that the home Tor client option offered the least degree of anonymity (unless the user additionally hosts a Tor server). We also showed that the single Tor server in the path represents a potentially powerful attacker for the proxy settings;
- In the performance evaluation, mainly the settings regarding path setup were studied. We concluded that the performance and proxy settings reduced the performance overhead significantly in most of the assessed scenarios;
- Concerning the evaluation of practicality, usability, availability, and trust, all options had their pros and cons. Yet, given a powerful mobile device and an optimized directory server protocol, the most viable option would be to place the Tor client on the mobile device.

Lastly, as the performance varied greatly between different test cases in the performance evaluation, there is obviously room for other performance enhancements besides relaxed path setup settings. Possible future work would be to combine such performance-enhancing path setup settings with strategies for intelligent path construction (see Section VIII-B). In this case, it is important to study how the degree of anonymity would be affected. Another topic for future work is to study how the Tor client can be tweaked for low-end mobile devices (including the optimization of the directory server protocol).

Acknowledgements

Thanks to Dogan Kesdogan, Lexi Pimenidis, and Tobias Kölsch (RWTH Aachen), as well as Simone Fischer-Hübner (Karlstad University), for contributing with ideas during the paper writing process. Parts of this work have been funded by the EU FP6 project PRIME (Contract No. 507591) and the IST FIDIS (Future of Identity in the Information Society) Network of Excellence project (Contract No. 507512).

References

[1] Directive 2006/24/EC of the European Parliament and of the Council of 15 March 2006 on the retention of data generated or processed in connection with the provision of publicly available electronic communications services or of public communications networks and amending directive 2002/58/ec. Official Journal L No.105, 13 Apr 2006.

[2] Christer Andersson, Simone Fischer-Hübner, and Reine Lundin. Enabling Anonymity in the Mobile Internet using the mCrowds Approach. In Penny Duquenoy, Simone Fischer-Hübner, Jan Holvast, and Albin Zuccato, editors, *Proceedings of the IFIP WG 9.2, 9.6/11.7 Summer School on Risks and Challenges of the Network Society*, volume 2004:35, pages 178–189. Karlstad University Studies, 4–8 Aug 2003.

[3] Christer Andersson, Reine Lundin, and Simone Fischer-Hübner. Privacy Enhanced WAP Browsing with mCrowds: Anonymity Properties and Performance Evaluation of the mCrowds System. In *Proceedings of the ISSA 2004 Enabling Tomorrow Conference*, Gallagher Estate, Midrand, South Africa, 30 Jun – 2 Jul 2004.

[4] Alastair R. Beresford and Frank J. Stajano. Location Privacy in Pervasive Computing. *IEEE Pervasive Computing*, 2(1):46–55, January 2003.

[5] Sebastian Clauss. A framework for quantification of linkability within a privacy-enhancing identity management system. In Gnter Mller, editor, *Proceedings of Emerging Trends in Information and Communication Security (ETRICS)*, volume LNCS 3995, pages 191–205, Berlin Heidelberg, 2006. Springer Verlag.

[6] Claudia Díaz. Anonymity Metrics Revisited. Daghstuhl Seminar on Anonymous Communication and its Applications, Oct 2005.

[7] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th USENIX Security Symposium*, Aug 2004.

[8] Leonardo A. Martucci, Christer Andersson, Wim Schreurs, and Simone Fischer-Hübner. Trusted Server Model for Privacy-Enhanced Location Based Services. In *Proceedings of the 11th Nordic Workshop on Secure IT-systems (Nordsec 2006)*, 19–20 Oct 2006.

[9] Mikael Nilsson, Helena Lindskog, and Simone Fischer-Hübner. Privacy Enhancements in the Mobile Internet. In *Proceedings of the IFIP WG 9.6/11.7 2nd Working Conference on Security and Control of IT in Society (SCITS-II)*, 15–16 Jun 2001.

[10] Andreas Pfitzmann and Marit Hansen. Anonymity, Unlinkability, Unobservability, Pseudonymity, and Identity Management - A Consolidated Proposal for Terminology v0.27, 20 Feb 2006. See `http://dud.inf.tu-dresden.de/literatur/`.

[11] Michael Reiter and Avi Rubin. Crowds: Anonymity for Web Transactions. In *DIMACS Technical report*, pages 97–115, 1997.

[12] Stephen Rollyson. Improving Tor Onion Routing Client Latency. Technical report.

[13] Emin Islam Tatlý, Dirk Stegemann, and Stefan Lucks. Dynamic Mobile Anonymity with Mixing. Technical Report TR-2006-007, Department for Mathematics and Computer Science, University of Mannheim, 27 Mar 2006.

[14] Rolf Wendolsky, Dominik Herrmann, and Hannes Federrath. Performance Comparison of low-latency Anonymisation Services from a User Perspective. In *Proceedings of the 7th Workshop on Privacy Enhancing Technologies (PET 2007)*, 20–22 Jun 2007.

[15] Svein Yngvar Willassen. Forensics and the GSM Mobile Telephone System. *International Journal of Digital Evidence*, 2(1), 2003.