15-744 Project Proposal

Nicolas Feltman and David Naylor

1 Problem Statement

The desire for anonymous communication in the Internet has made it a much-studied issue. Several standard approaches to anonymity have emerged, normally involving some number of explicitly identified proxies due to the dumb nature of current routers. Although none of these methods are perfect, they generally meet most users' needs. Little work has been done, however, exploring ways to make these methods easily utilized by application developers and easily understood by end users.

We propose to consider anonymity in the context of the eXpressive Internet Architecture (XIA) [**?**], a future Internet architecture under development at CMU. We believe that new features in XIA will provide application developers with an easy and consistent way to use existing methods for anonymous communication.

2 Related Work

Existing work related to anonymity can be roughly categorized as either discussion of what it means to communicate anonymously or techniques for doing so.

2.1 Types of Anonymity

The term "anonymity" alone is somewhat vague. When implementing anonymous methods, one must consider three questions: 1) Who wants to remain anonymous? 2) From whom? 3) To what degree? In response to (1), Pfitzmann and Waidner [**?**] consider three scenarios: *sender anonymity*, *recipient anonymity*, and *sender-recipient unlinkability*. As for (2), one typically anticipates either an attacker with limited knowledge (a single host or a small number of colluding hosts) or one with global knowledge (achieved through a large number of colluding hosts). And regarding (3), Reiter and Rubin [**?**] propose three classifications: *beyond suspicion*, *probable innocence*, and *possible innocence*.

2.2 Techniques

Most existing techniques involve routing traffic through a proxy of some kind. In its simplest form, this amounts to forwarding your packet (an HTTP request, perhaps) to the proxy which will in turn forward to its actual destination. There are many ways to improve on this basic idea; in *onion routing*, packets are forwarded through multiple proxies with the use of cryptography to hide all but the next hop address at each step, and *MIXes* [**?**] require the proxy to wait until it has received packets to forward from several users, after which it forwards them on in a different order.

Other techniques include using broadcasts or multicasts to hide the intended recipient or return address spoofing to protect the sender.

3 Status Report

4 Action Plan