Crowds: Anonymity for Web Transactions

MICHAEL K. REITER
Bell Laboratories, Lucent Technologies
and
AVIEL D. RUBIN
AT&T Labs—Research

In this paper we introduce a system called Crowds for protecting users' anonymity on the world-wide-web. Crowds, named for the notion of "blending into a crowd," operates by grouping users into a large and geographically diverse group (crowd) that collectively issues requests on behalf of its members. Web servers are unable to learn the true source of a request because it is equally likely to have originated from any member of the crowd, and even collaborating crowd members cannot distinguish the originator of a request from a member who is merely forwarding the request on behalf of another. We describe the design, implementation, security, performance, and scalability of our system. Our security analysis introduces *degrees of anonymity* as an important tool for describing and proving anonymity properties.

Categories and Subject Descriptors: C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Applications*; K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*; K.4.4 [**Computers and Society**]: Electronic Commerce—*Security*

General Terms: Security

Additional Key Words and Phrases: Anonymous communication, world-wide-web

1. INTRODUCTION

Every man should know that his conversations, his correspondence, and his personal life are private.—Lyndon B. Johnson, president of the United States, 1963–69

The lack of privacy for transactions on the world-wide-web, or the Internet in general, is a well-documented fact [Brier 1997; Miller 1997]. While encrypting communication to and from web servers (e.g., using SSL [Garfinkel and Spafford 1997, Ch. 12]) can hide the content of the transaction from an eavesdropper (e.g., an Internet service provider, or a local

Authors' addresses: M. K. Reiter, Bell Laboratories, Lucent Technologies, 700 Mountain Avenue, Room 2A-342, Murray Hill, NJ 07974; email: reiter@research.bell-labs.com; A. D. Rubin, AT&T Labs—Research, 180 Park Avenue, Room A239, Florham Park, NJ 07932–0972; email: rubin@research.att.com.

system administrator), the eavesdropper can still learn the IP addresses of the client and server computers, the length of the data being exchanged, and the time and frequency of exchanges. Encryption also does little to protect the privacy of the client from the server. A web server can record the Internet addresses at which its clients reside, the servers that referred the clients to it, and the times and frequencies of accesses by its clients. With additional effort, this information can be combined with other data to invade the privacy of clients even further. For example, by automatically `finger`ing the client computer shortly after an access and comparing the idle time for each user of the client computer with the server access time, the server administrator can often deduce the exact user with high likelihood. Some consequences of such privacy abuses are described in Miller [1997].

In this paper we introduce a new approach for increasing the privacy of web transactions and a system, called *Crowds*, that implements it. Our approach is based on the idea of "blending into a crowd," i.e., hiding one's actions within the actions of many others. To execute web transactions in our model, a user first joins a "crowd" of other users. The user's request to a web server is first passed to a random member of the crowd. That member can either *submit* the request directly to the end server or *forward* it to another randomly chosen member, and in the latter case the next member chooses to submit or forward independently. When the request is eventually submitted, it is submitted by a random member, thus preventing the end server from identifying its true initiator. Even crowd members cannot identify the initiator of the request, since the initiator is indistinguishable from a member that simply forwards a request from another.

In studying the anonymity properties provided by this simple mechanism, we introduce the notion of *degrees* of anonymity. We argue that the degree of anonymity provided against an attacker can be viewed as a continuum, ranging from no anonymity to complete anonymity and having several interesting points in between. We informally define these intermediate points and, for our Crowds mechanism described above, we refine these definitions and prove anonymity properties for our system. We expect these definitions and proofs to yield insights into proving anonymity properties for other approaches as well.

An intriguing property of Crowds is that a member of a crowd may submit requests initiated by other users. This has both negative and positive consequences. On the negative side, the user may be incorrectly suspected of originating that request. On the positive side, this property suggests that the mere availability of Crowds offers the user some degree of deniability for her observed browsing behavior, if it is possible that she was using Crowds. Moreover, if Crowds becomes widely adopted, then the presumption that the computer from which a request is received is the computer that originated the request will become decreasingly valid (and thus decreasingly utilized).

The anonymity provided by Crowds is subject to some caveats. For example, Crowds obviously cannot protect a user's anonymity if the content

of her web transactions reveals her identity to the web server (e.g., if the user submits her name and credit card number in a web form). More subtly, Crowds can be undermined by executable web content that, if downloaded into the user's browser, can open network connections directly from the browser to web servers, thus bypassing Crowds altogether and exposing the user to the end server. In today's browsers, such executable content takes the form of Java applets and ActiveX controls. Therefore, when using Crowds, it is recommended that Java and ActiveX be disabled in the browser, which can typically be done via a simple preferences menu in the browser.

The rest of this paper is structured as follows: In Section 2, we state the anonymity goals of our system more precisely and introduce the notion of *degrees* of anonymity. This gives us sufficient groundwork to compare our approach to other approaches to anonymity in Section 3. We describe the basic Crowds mechanism in Section 4 and analyze its security in Section 5. We describe the performance and scalability of our system in Sections 6 and 7, respectively. We discuss crowd membership in Section 8, the system's user interface in Section 9, and the obstacles that firewalls present to wide scale adoption of Crowds in Section 10. We conclude in Section 11.

2. GOALS

2.1 Anonymity

As discussed in Pfitzmann and Waidner [1987], there are three types of anonymous communication properties that can be provided: sender anonymity, receiver anonymity, and unlinkability of sender and receiver. *Sender anonymity* means that the identity of the party who sent a message is hidden, while its receiver (and the message itself) might not be. *Receiver anonymity* similarly means that the identity of the receiver is hidden. *Unlinkability of sender and receiver* means that though the sender and receiver can each be identified as participating in some communication, they cannot be identified as communicating *with each other*.

A second aspect of anonymous communication is the attackers against which these properties are achieved. The attacker might be an eavesdropper that can observe some or all messages sent and received; collaborations consisting of some senders, receivers, and other parties; or variations of these [Pfitzmann and Waidner 1987].

To these two aspects of anonymous communication, we add a third: the *degree* of anonymity. As shown in Figure 1, the degree of anonymity can be viewed as an informal continuum. For simplicity, we describe this continuum with respect to sender anonymity, but it can naturally be extended to receiver anonymity and unlinkability as well. On one end of the spectrum is *absolute privacy*: absolute sender privacy against an attacker means that the attacker can in no way distinguish the situations in which a potential sender actually sent communication and those in which it did not. That is,

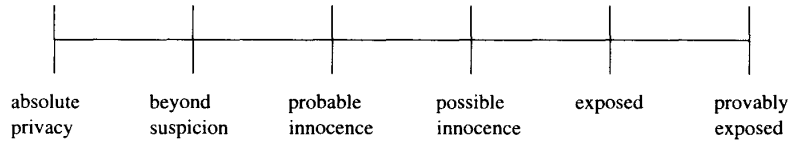

Fig. 1. **Degrees of anonymity:** Degrees range from *absolute privacy*, where the attacker cannot perceive the presence of communication, to *provably exposed*, where the attacker can prove the sender, receiver, or their relationship to others.

sending a message results in no observable effects for the attacker. On the other end of the spectrum is *provably exposed*: the identity of a sender is provably exposed if the attacker can identify the sender of a message, and can also prove the identity of the sender to others.

For the purposes of this paper, the following three intermediate points of this spectrum are of interest, listed from strongest to weakest.

—**Beyond suspicion**: A sender's anonymity is beyond suspicion if though the attacker can see evidence of a sent message, the sender appears no more likely to be the originator of that message than any other potential sender in the system.

—**Probable innocence**: A sender is probably innocent if, from the attacker's point of view, the sender appears no more likely to be the originator than to not be the originator. This is weaker than beyond suspicion in that the attacker may have reason to suspect that the sender is more likely to be responsible than any other potential sender, but it still appears at least as likely that the sender is not responsible.

—**Possible innocence**: A sender is possibly innocent if, from the attacker's point of view, there is a nontrivial probability that the real sender is someone else.

It is possible to describe these intermediate points for receiver anonymity and sender/receiver unlinkability as well. When necessary, we define these intermediate points more precisely.

Which degree of anonymity suffices for a user obviously depends on the user and her circumstances. Probable innocence sender anonymity should prevent many types of attackers from *acting* on their suspicions (therefore avoiding many abuses, e.g., cited in Miller [1997]) due to the high probability that those suspicions are incorrect. However, if the user wishes to avoid any suspicion whatsoever—including even suspicions not sufficiently certain for the attacker to act upon—then she should insist on beyond suspicion sender anonymity.

The default degree of anonymity on the web for most information and attackers is *exposed*, as described in Section 1. All recent versions of Netscape Navigator and Internet Explorer are configured to automatically identify the client computer to web servers, by passing information including the IP address and the host platform in request headers.

Table I. Anonymity Properties In Crowds

Attacker	Sender anonymity	Receiver anonymity
local eavesdropper	exposed	$P(\text{beyond suspicion}) \xrightarrow[n\to\infty]{} 1$
c collaborating members,	**probable innocence**	$P(\text{absolute privacy}) \xrightarrow[n\to\infty]{} 1$
$n \geq (p_f/(p_f - 1/2))(c + 1)$	$P(\text{absolute privacy}) \xrightarrow[n\to\infty]{} 1$	
end server	**beyond suspicion**	N/A

2.2 What Crowds Achieves

As described in Section 1, our system consists of a dynamic collection of users, called a *crowd*. These users initiate web requests to various web servers (and receive replies from them), and thus the users are the "senders" and the servers are the "receivers". We consider the anonymity properties provided to an individual user against three distinct types of attackers:

—A **local eavesdropper** is an attacker who can observe all (and only) communication to and from the user's computer.

—**Collaborating crowd members** are other crowd members that can pool their information and even deviate from the prescribed protocol.

—The **end server** is the web server to which the web transaction is directed.

The above descriptions are intended to capture the full capabilities of each attacker. For example, collaborating members and the end server cannot eavesdrop on communication between other members. Similarly, a local eavesdropper cannot eavesdrop on messages other than those sent or received by the user's computer. A local eavesdropper is intended to model, e.g., an eavesdropper on the local area network of the user, such as an administrator monitoring web usage at a local firewall. However, if the same LAN also serves the end server, then the eavesdropper is effectively global, and we provide no protections against it.

The security offered against each of these types of attackers is summarized in Table 1 and justified in the remainder of the paper. As indicated by the omission of an "unlinkability of sender and receiver" column from this table, our system serves primarily to hide the sender or receiver from the attacker. In this table, n denotes the number of members in the crowd (for the moment we treat this as static) and $p_f > 1/2$ denotes the probability of forwarding, i.e., when a crowd member receives a request, the probability that it forwards the request to another member, rather than submitting it to the end server. (p_f is explained more fully in Section 4.) The boldface claims in the table—i.e., probable innocence sender anonymity against collaborating members and beyond suspicion sender anonymity against the end server—are guarantees. The probability of beyond suspicion receiver

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.