A Privacy-Aware Architecture for Sharing Web Histories[*]

Alex Tsow and L. Jean Camp
atsow@cs.indiana.edu, ljean@ljean.com

Abstract

Net Trust is a distributed reputation system that identifies fraudulent web sites by aggregating individual opinions and browsing histories over user-selected social networks. This paper examines the security properties intrinsic to any correct implementation of Net Trust's ratings system and the privacy properties arising from the paper's proposed rich-client/thin-server implementation architecture. The ratings system protects against Sybil attacks, corruption by *en masse* bogus ratings distribution, and distinguishes fraudulent web sites from the trustworthy ones by leveraging the browsing behavior in trusted social networks rather than link topology of third-parties. The implementation architecture maintains high data availability while empowering browser-history owners with final control over data access. This paper analyzes Net Trust's participants, attackers, security and privacy goals, and implementation choices.

1 Introduction

Net Trust is a rating system that combines the browsing habits and opinions of friends to identify online fraud, phishing and sleaze. Individual web histories and social networks are sensitive data for many. Much media attention has been shed on people who have been fired for online blog postings, pictures, and other personal content. Perhaps more insidious is the back-door trade of private information via personalization services [4] for consumer profiling [23]. Unlike many social referral systems, browser toolbars, web portals, and other free Internet services, Net Trust does not extract payment from its users through analysis, profiling, or reselling of their private data. For instance, the *Alexa* toolbar—which ranks sites, detects fraud, and makes related link recommendations—transmits and logs web histories, search terms, product purchases and other personally identifiable information [1]; Alexa profits from selling aggregate reports of this data, while its corporate parent, *Amazon.com*, attaches portions of collected data to personally identifiable information. We have designed Net Trust to protect its users from attacks on their sensitive information and ratings integrity by malicious peers, dishonest web sites, and even the Net Trust servers.

Net Trust's security and privacy properties are intended to stop the for-profit compromise of ratings and private information. Economic gain motivates malware's ubiquity and sophistication far more than bragging-rights and malicious character-destruction. The strength of its security and privacy properties is tuned to eliminate systematic abuse by profit-seeking agents. In particular, this falls well short of *computational intractability* as required by cryptographic operations; their security and privacy protections rest on financial calculations rather than algorithmic feasibility.

The remainder of the paper is organized as follows: Section 3 presents the Net Trust rating system and illustrates its use against two fraud scenarios. Section 4 defines Net Trust's potential attackers and the system's security and privacy goals. Section 5 presents the system's implementation. Section 6 discusses how well this architecture achieves the security and privacy goals. Section 7 concludes the paper with a summary of properties achieved and directions for improvement.

2 Related Work

Net Trust's fundamentals—the social foundations [9], economic rationale [3], theoretical fraud simulations [8], and human-computer interface [13]—have been developed over several other studies. This paper focuses on

[*] This work has been funded, in part, by Google and The Institute for Information Infrastructure Protection (The I3P).

its implementation's security and privacy properties.

Combining social networks with peer production to create personalized ratings and recommendations is not a new idea. *StumbleUpon* [25] and *Alexa* [1] are two commercial products that make recommendations based on "friend" activity. Markines, Stoilova, and Menczer's *GiveALink* project [18] discovers relevant web sites (some unknown to Google) with its *social bookmarking algorithms*. Many other social bookmarking systems are summarized by Hammond et al. [11]. Common to these systems is the *inference* of social networks based on browsing, bookmark, or rating similarities. In contrast, Net Trust social networks are formed by explicit invitation. Since end-users control their social scopes, bogus accounts cannot influence ratings without first establishing trust with the peer-targets.

Anti-phishing toolbars help users identify fraudulent web sites. Some—such as SpoofGaurd [5]—use locally evaluated heuristics to determine risk, while others—Google Safe Browsing [10], for instance—depend upon unpersonalized red/green lists. Many current toolbars combine the two approaches, including Net Craft [22] and Microsoft Phishing Filter [24]. Egelman et al. test the effectiveness of ten popular anti-fraud toolbars [26], finding accuracy to be a mixed bag with high false positive rates or high false negative rates.

3 Net Trust Overview

For each web page rendered, the Net Trust toolbar displays an aggregate report over the user's ratings and ratings from the currently selected social network. Additionally it displays (negative/neutral/positive) rankings from up to five self-selected third-party rating sources. See [9] for a full discussion.

3.1 Individual Ratings

One barrier to peer-produced reputation metrics is the effort required rate each subject. Individual ratings in Net Trust may be *explicitly set* through manual intervention or *implicitly generated* according to browsing habits. Normal web browsing activity produces implicit ratings, so users bear no burden to manually evaluate every online resource they encounter.

Implicit ratings count the number of times an individual has visited a web site. This count is subject to a maximum bound (set to 5), an initial time-delay (set to 1 week), a decay penalty for periods of inactivity (beginning at 1 month), and minimum time between visits (set to 1 hour). For a given web site, an implicit rating contains the initial visit date, the most recent visit date, and an integer between 1 and 5 representing the number of visits. The rating value is 0 for the initial time delay, so that phishing sites (whose average lifetime is 3.8 days [2] according to the APWG) cannot gain accidental reputation. Individual visits must be separated by a minimum threshold (1 hour) to ensure that casual exploration of a web site does not artificially boost its rating. After a period of inactivity, an implicit rating looses up to half its value over a decay interval.

Users can explicitly rate a web site with a non-zero integer between -5 and 5. Explicit ratings supplant the implicit ratings for the web site. There are no time delays for validity or decays for inactivity. Moreover, the explicit rating contains no information about initial or visit dates. Explicit ratings may only changed by another explicit user rating interaction. See Figure 1 for the display information.

In addition to numerical ratings, individuals can post textual comments for web pages. This has no effect on the numerical ratings, however Net Trust reports the number of comments in the aggregate display.

3.2 Social networks

Although a level of protection arises from analyzing an individual's browsing habits, utility improves when individual ratings are combined with ratings from friends. Unlike many social referral systems which infer social networks from centralized analysis of individual behavior [18, 25, 1], Net Trust users form their social network by explicit invitation. Only immediate friends have a direct impact Net Trust ratings—although friends of friends have an indirect influence by changing the behavior of immediate friends.

Net Trust reports the following aggregate data over the direct friends in the toolbar: average of negative ratings, average of positive ratings, number of comments, and number of ratings. User interface testing has shown that this information, when suitably displayed, changes trust behavior [13]. Full ratings—the comments and numerical rating for each friend in the network—are available upon request in a separate window.

In general, a Net Trust user should maintain several different ratings profiles, or *personas*. Each persona has a different set of friends. This is important from both a privacy point of view and a utility point of view. One should not mix web ratings between a persona representing one's professional life and a persona representing one's recreational activities. Moreover, these interests are likely to be properly informed by different networks who frequent different web sites.

3.3 Third-party sources

Sometimes an individual's social network is not rich enough to effectively inform trust behavior. *Third-party sources* can help to seed opinion. Third-parties can give one of three ratings—negative, neutral, or positive—to a web site. This rating is indexed entirely by the web site's domain name. The ratings do not mix with the aggregate report from the social network. As with the end-user's social network, individuals get to choose which third-party rating sources they wish to include. For now, the space of third-party sources is limited to those chosen by the Net Trust developers, including the FDIC, BBB, and Site Advisor.

Although third party sources are a theoretical necessity for initializing opinion [8], *this paper limits its security and privacy analysis to Net Trust's peer-generated ratings.*

4 Security and Privacy Goals

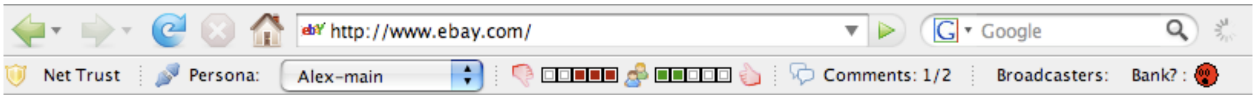
Net Trust requires several kinds of sensitive information to produce its ratings. Among these are partial web histories, personal identifiers, friend identifiers, and authentication credentials. Without appropriate protections, this information is subject to abuse. This section identifies the Net Trust participants, assumptions about their capabilities, and the system's security and privacy objectives.

4.1 The Participants

There are three agents in the Net Trust's peer-generated rating system: *rating-subjects*, *peer-producers*, and *rating-servers*.

The **rating-subject** is the web site which Net Trust evaluates. Their incentives are to promote their own ratings and possibly demote their competitor's ratings. Rating-subjects can host arbitrary content (esp. scripts)—a potential automated-attack vector. Since implicit peer-ratings vary according to visiting patterns (Section 3.1), a web-site may try to automatically influence this with scripted content, e.g. scripted redirects, reloads, and pop-behind windows. Rating-subjects also have control over the structure of their web-site and may manipulate it to confuse the binding of ratings to pages. Moreover, while web sites do not explicitly handle ratings in their role as rating-subjects, they may play strategically as peer-producers.

The **peer-producer** is the essential rating unit in Net Trust. As explained in the Section 3.1, peers alter subject ratings implicitly through their visiting patterns and explicitly through comments or manual evaluation. A social network is defined by one *persona*—a peer account controlled by the end-user—and its collection of chosen friends. In this paper's design proposal, the client maintains persona ratings and writes this data to rating-servers. In principle, a malicious client could write arbitrary ratings to the rating-server for a given persona.


http://www.ebay.com/

Persona: Alex-main

Figure 1: The Net Trust toolbar displays aggregate negative evaluations (red lights on left), aggregate positive evaluations (green lights on right), a comment count, a ratings count, and third-party (broadcaster) evaluations. A full ratings window with friend evaluations is available on demand.

Rating-servers are a shared store for transporting ratings among peers and distributing third-parties ratings to end-users. Their primary actions are processing account creation requests, serving anonymous downloads, and enforcing authenticated uploads.

4.2 Security Goals

Signaling: Net Trust's principle goal is to distinguish fraudulent web sites from legitimate web sites. The approach is more subtle than conventional anti-phishing toolbars which attempt to answer the question "Authentic or phishing?" with a tempered degree of certainty. Instead of directly answering the yes/no question, Net Trust creates a *signal* in the economic sense—a distinguishing property that is costly to forge. This signal in conjunction with the page's semantic content—whose evaluation (is it a login page?) is a human strength but a computer's weakness—produces a trust decision in the user. Preliminary studies of the Net Trust interface suggest that correct signaling produces more effective trust decisions [13].

Sybil attacks: Net Trust resists *Sybil attacks* [7], where a single entity who controls large numbers of peer-producers (*Sybil accounts*) contaminates ratings in honest peer-producer social networks. Preventing this from happening to *any* honest account is more than we can guarantee, however a weaker—but arguably as useful—goal is to prevent Sybil accounts from contaminating large numbers of honest peer-producers. Gauging the relationship of Sybil accounts to vulnerable accounts is a matter of judgment and attacker resources. Measuring this asymptotically, one might demand a linear relationship exceeding a specified constant. In general, the cost function, f—which maps a target number of vulnerable honest accounts n to the number of Sybil accounts necessary for corruption—can be given a specific asymptotic bound, g; i.e. there exists an N such that for all $n > N$, $f(n) \leq g(n)$.

For example, search engine manipulation by adding links from bogus web pages into the subject page is a possible Sybil attack because a constant number of bogus web-pages compromises ratings for *all* search engine users. The attacker does not have to invest more resources to reach a larger audience. In this sense, the cost function f is constant and would be asymptotically bounded by any increasing function. For practical purposes, we take a linear bound to preclude effective Sybil attacks.

Owner-enforced access control: Peer-producers must be able to explicitly grant and revoke read access to their browsing histories. Granting read access may require assistance from third parties (e.g. rating-servers), however revocation should proceed without third-party cooperation. We do not require resiliency to peer betrayal: if Alice trusts Bob with her browsing history, then Bob can betray her by forwarding the data to unauthorized parties. Practically speaking, limiting access across peer accounts controlled by the same user is not possible.

Third-party scripting: Web site visits and social networking changes must be filtered through explicit user actions. Scripted redirects, reloads, and pop-under frames should not influence the peer history database. Moreover, remote scripting of social network changes should be impossible: the *Samy* worm is one notorious example of a server exploit resulting in the automatic non-consensual addition of over 1 million friends to a MySpace profile [14].

Rating-server authentication: Rating-servers in Net Trust must be unspoofable. Subverting the data-distribution mechanism destroys the system.

Writer authentication: Writes into the rating-servers must be limited to account owners.

Read integrity: Reads from the rating-servers must match the values asserted by the authors.

4.3 Privacy Goals

Linking resistance: Unique identifiers are necessary to reference peer ratings, however we want to prevent *linking* these references with personally identifying information (PII) about their account holders. Trusted peers—those inside a subject's social network—may link a subject's Net Trust account with its offline identity. A weaker linking problem is the discovery that an individual controls a given set of peer accounts without uniquely identifying the individual. Third-parties (e.g., rating-servers) should not be able to link account identifiers with users or each other by examining IP addresses, access patterns, or raw account data.

Confidential social networks: Social networks in Net Trust are collections of immediate friends. Thus, the social network for identifier A is the set of peer-identifiers who have granted history access to A. The only party with full knowledge of the social network is the peer account owner: Alice may know that she is in one of Bob's social networks, but she can not discover Bob's other friends. This is contrary to online many social networking web sites, where discovering friends-of-friends is one of the key interactions. Third parties must be unable to discover friend relationships among peers.

Account deniability: Anyone may participate in Net Trust as a peer producer. Net Trust does not attempt to verify the claims of any account holder. This gives rise to a deniability property: the plausible claim that someone else could have generated a particular account with arbitrary ratings without the knowledge or authorization of anyone else. In particular, an account appearing to belong to Jane Doe could have be generated by Don Jones. There's no strong binding between accounts and end-user identities.

Request granularity: In general the page rendered by a browser is function of the URI (domain, directory, filename, CGI variables), HTTP post data, server-state, client-state (e.g. cookies, cache, browser history, browser plug-ins) and protocol details (e.g. IP address). The complete download details reveal far too much information: account names, numbers, and search terms are commonplace in CGI variables and cookies. Net Trust limits request history data to the URI's hostname and top-level directory.

Temporal granularity: Net Trust evaluates web sites based on the visit times and places from peer histories. Too much data in the ratings file and too frequent an update to ratings servers could betray timely knowledge of browsing habits to both the ratings servers and the peers. Clearly, time-stamping every visit to a particular web page discloses too much about personal browsing behavior. Similarly updating rating-servers too often can betray near real-time browsing behavior to a malicious peer who polls the data for changes. Similarly, if there are central record accesses and updates for each visit (an option in the Google Safe Browsing toolbar [10]), the server and malicious peer could deduce that a target peer is looking at a specific page at a specific time after a short latency.

Correct social context: A peer-persona that represents the end-user's recreational interests may carry history that should be kept separate from a persona which represents the end-user's professional interests. The emergence of the abbreviation NSFW (Not Suitable For Work) in email attests to the recognition of this phenomenon. While largely a social problem, the user interface and interaction policies can have profound effects on isolating browsing history among same-owner personas. It is our long-term goal to prevent inappropriate or embarrassing history from appearing in socially targeted personas.

5 Design

5.1 The Data Aspect

Anonymous but consistent peer identification: Net Trust uniquely identifies peers with *reference-keys* generated by the collision resistant hash of a user-asserted nym (a nickname which may not be unique), email address, and random number. These assertions are not subject to verification, however the hashing regimen serves as a *commitment* to the specified information and prevents someone who only knows the reference-key from recovering the nym and email address.

Peer-ratings: (Figure 2) Net Trust generates peer-ratings locally but shares them via the rating-servers. Figure 2 summarizes the file's hierarchical structure. The top-level contains the peer's reference-key—but not its generators—and a date indicating its last modification. The children of the top-level structure hold truncated URIs which store the web site's domain (minus the leading `www.`, if present) and the top-level directory, if present. The record contains no CGI variables, individual pages or script-references. Truncated URIs have either a *visit-node* or an *explicit-rating-node* and an optional *comment-node*. The visit-node stores the dates of initial and last visits as well as the visit count from one to five. When an explicit-rating-node is set, it stores only positive or negative value rating from one to five; it stores no dates or visit-count. The comment-node stores text for a one-line comment.

Invitation and acceptance format:(Figure 3) Invitation and acceptance messages between peers have the same format. It is simply a list of the following data: local nym (non-unique screen name), communicating email address, random blinding string, Net Trust identifier (the hashed concatenation of the first three fields),

```
<peer-ratings> := <reference-key> <mod-date>
                  <location>*
<location>     := <domain>/<top-dir>
                  (<visit> | <exp-rating>)
                  <comment>?
<visit>        := <first-date> <last-date>
                  (1|2|3|4|5)
<exp-rating>   := (-5|-4|-3|-2|-1|
                    1| 2| 3| 4| 5)
<comment>      := <text>
```

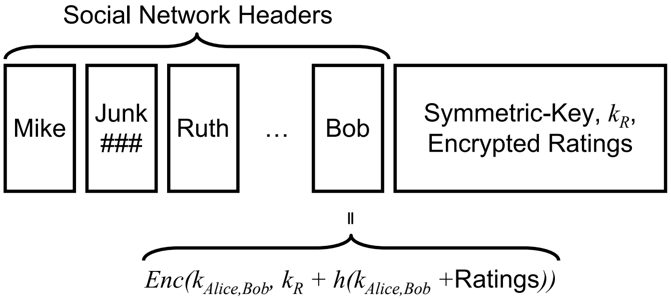

Social Network Headers
Mike
Junk ###
Ruth
…
Bob
Symmetric-Key, k_R, Encrypted Ratings
=
$Enc(k_{Alice,Bob}, k_R + h(k_{Alice,Bob}$ +Ratings$))$

Figure 2: The left-hand side shows the hierarchical structure and contents peer-ratings data. The right-hand side shows the export format which encrypts the ratings with a randomly chosen symmetric key, k_R. Access is granted to authorized peers (e.g., Bob) by encrypting k_R and a ratings checksum (a keyed-hash) with a shared secret-key ($k_{Alice,Bob}$). The header blocks perform the dual role of access control and concealing the social network (since blocks are indistinguishable from random data).

a header position for access keys (Figure 2), and a seed for generating a shared secret-key. The shared secret key is the hashed concatenation of the two seeds according (smaller seed appears first); thus, the order these messages does not change the shared secret—there is no distinction between invitation and acceptance messages.

Local social-network management: Each Net Trust client manages its own social network; no other party has direct access to this data. Clients do not share it with ratings servers or other peers. A locally stored file contains reference-keys, email addresses, nyms, and shared secrets (for write authentication with ratings-servers) for each of the end-user's peer-personas. Net Trust defines a persona's social network with a list of immediate friends. The list contains the reference-key, generating information (the nym, email address, and random number), header position, and shared secret-key for each friend. The same peer may appear in multiple local social networks.

Encrypted export format: (Figure 2) When exporting ratings of persona A, the client encrypts the ratings data with a randomly chosen secret-key, k_R. For each friend, B, in the social network, the client concatenates k_R with a hashed checksum of the ratings, and then encrypts it with a shared secret key $k_{A,B}$. The client stores the block in a header position reserved for B. There are n blocks in the header, unallocated positions are padded with random-looking junk.

5.2 Architectural and Behavioral Aspects

The Net Trust system combines two kinds nodes: lightweight rating-servers and full-featured clients.

Server: The rating-servers collect and distribute peer-ratings and third-party recommendations. Net Trust limits their role as much as possible to the transport of these two data sets. Because of the data storage choices explained in Section 5.1, the servers have no direct knowledge of end-user identities nor do they have access to social relationships.

Rating-servers run a CGI-script supporting four kinds of peer operations: *new*, *read*, *write*, and *status*. The *new* operation specifies a reference-key and authorization-secret. If the reference-key is unused, the server creates a new account by recording the key and authorization secret. The *read* operation specifies a reference-key. If valid, the server returns the corresponding peer-rating set. *Write* operations specify a reference-key, peer-rating file, and *hash map authentication code (HMAC)* [15] using their shared secret. The server performs syntactic input validation prior to updating its database of peer-ratings. Given a reference-key, the *status* operation returns the current peer-rating set's modification date.

Client-Server communication: Rating-servers communicate with clients over the SSL-encrypted protocol, HTTPS. To prevent server-side spoofing, clients only communicate with servers that present certificates signed by the Net Trust signing authority. The client may optionally contact rating-servers through an anonymous proxy service—Tor [6], for example. This prevents servers from using the client IPs in privacy-compromising attacks (full discussion in Section 6).

Invitation and Acceptance Data Fields

```
<target-email>
<nym> <sender-email> <blinder> <id>
<header-position>
<shared-secret-seed>
```

$$id = h(nym + sender\_email + blinder) \qquad (1)$$

$$k_{Alice,Bob} = h(seed_{Alice} + seed_{Bob}) \qquad (2)$$

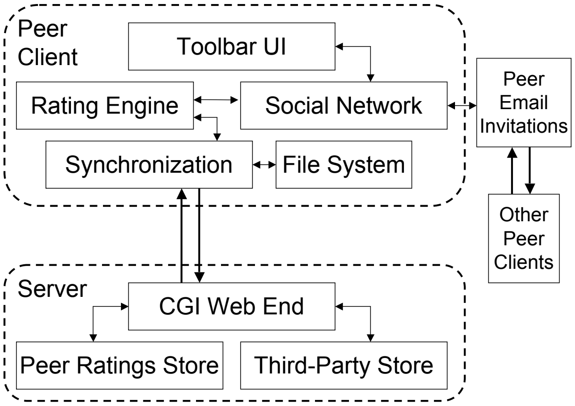


Figure 3: The left-hand side shows the data fields for invitation and acceptance messages. A shared secret-key $k_{Alice,Bob}$ is generated from the hash of two concatenated secret seeds. This key encrypts the ratings-access key and checksum in the specified header position (Figure 2). The right-hand side illustrates the component view of Net Trust.

Peer-to-peer communication: Net Trust does not enforce a method for transferring invitation (and acceptance) messages between peers. The client saves an invitation file to a user-specified location. Users must then transfer this data to the intended recipients by a confidential method of their choice. We envision email as a practical solution. Users can encrypt email for additional assurance. Alternatives include chat sessions over SSL and person-to-person transfers of physical media (disk, flash, CD).

Invitation exchange commences manually, however Net Trust provides additional support for transactions over email. The client maintains a list of incoming invitations and outgoing invitations. When generating an invitation, the client prompts its user for a target email address. The client first checks to see if the address matches any address on incoming invitation list. If so, it generates the outgoing invitation, computes the shared secret key, adds the peer to the address book, and removes the incoming invitation. If not, the client adds the invitation to the outgoing list. When importing an invitation file, the roles are reversed: the client checks the outgoing invitation list and completes the transaction when an incoming invitation matches the target email address. Otherwise, the invitation is added to the incoming list.

Client: The Net Trust client has four main components: ratings-engine, social network manager, record synchronizer, and display.

The ratings-engine evaluates web sites as specified in Section 3 using locally stored data from peers and trusted authorities. Upon visiting a web site, the ratings-engine computes its evaluation, then records the visit in the currently selected peer-persona. When viewing a web-site, the end-user may optionally set an explicit rating or post a comment to the current peer-persona. There is no rating-server communication during any of these ratings operations.

The social networking module stores the identifier, identifier generators, authentication credentials, and shared secrets keys for all user personas and friends. The module maintains an "address book" that stores access tokens for peers in all local social networks. Users may add peers from the address book into the social network of any persona under their control. When a user wishes to stop participating in a social network, the client software revokes access by discontinuing inclusion of the randomly selected ratings key, k_R, in the specified header. The target's synchronization module (below) detects this ouster since it can no longer decrypt the downloads.

The synchronization module is the only part of Net Trust that interacts with rating-servers. It abstracts read/decrypt and write/encrypt processes, accessing the local filesystem and/or the rating-servers as necessary. The client stores all ratings files in cleartext locally. It automatically applies the symmetric-key (de/en)cryption to incoming and outgoing peer-ratings according to the specification above. This module further schedules remote accesses to conceal social networks. A parameter t_{min} defines minimum time separation between remote accesses. A second parameter, t_{fresh}, defines the freshness requirement; i.e., an age past which the file is scheduled for update. As the ratings engine requests access to peer-ratings, the synchronization module will make local reads and writes. A JavaScript timer, set to t_{min}, transfers control to the synchronization module at regular intervals to check for synchronization opportunities. It first adds any expired ratings to a refresh list, then randomly selects an element from the list to synchronize with the remote server.

Net Trust is a web browser toolbar which displays the aggregate peer ratings (average negative ratings, average positive ratings, number of comments) and up to five third-party classifications. An on-demand external window shows full evaluations by enumerating ratings and comments for each peer in the network. While we reserve full discussion of the user interface for another paper [13], Figure 1 illustrates its salient features.

6 Discussion

6.1 Security Properties

Belief in Net Trust's capacity to generate effective signals is a prerequisite for designing a secure implementation. A previous theoretical simulation demonstrates that the rating system has economic utility [8]. The simulation's core assumption is that peer-selected social networks exhibit similar browsing habits—an instance of a more general result on human behavior [20]. Data from a recent study on browsing behavior, which collects 10 months of aggregate data over roughly 100,000 users throughout the Indiana University campuses [21], suggests that users in the same social network have similar browsing habits. The common thread among this population is their affiliation with Indiana University (IU) and it is no surprise that among the top 10,000 most consistently visited domains (i.e., those visited at least once in more than 95% of the tested days), IU-related domains account for 30.5% of human-initiated web traffic. Top rankings for non-IU `.edu` domains, webmail, news, and banking reflect substantial local bias (see Appendix for details). We find this encouraging since geography is one of the most reliable predictors of small social networks [16].

One of the core properties of Net Trust's social-network-based ratings system is its intrinsic *resistance to Sybil attacks*. Malicious peers must first be accepted into the victim's social network before influencing ratings. Moreover, there must be sufficient penetration to override honestly-generated ratings. Committing mass ratings-fraud requires controlling lots of peer accounts *and* large-scale infiltration of social networks. Thus the Sybil attack must target victims individually. Contrast this with open-access search-engine manipulation, where rating-up (or down) a page with bogus links produces a global effect.

By itself, the per-victim-cost property deters mass manipulation with bogus peer accounts. However Net Trust further increases the per-victim-cost with its *identity commitment* scheme. When registering an account, the peer must commit to an email address and a nym. If a Sybil attack were to attempt large scale social engineering by self-inviting into victim social networks, the messages must offer socially meaningful nyms and email addresses. For instance, if Jane Doe and Don Jones are friends, an attacker who wants to join Don Jones's network would maximize the chances of success by spoofing Jane Doe's identity. Perhaps the attacker would register a Net Trust account with the email address, `janedoe69@free-email.org`, and nym, `JaneDoe`. This account will help the attacker compromise Jane Doe's friends, but work against compromising others. Since bogus peer accounts must be tailored to their victims, the social-engineering Sybil attack requires large numbers of Net Trust accounts.

Assuming (incorrectly) that it only takes one friend betrayal to subvert rating, and that no person has more than n social contacts, the cost function mapping number of vulnerable accounts to number of Sybil accounts is bounded below by the linear function, $\lambda(x).x/n$ since no Net Trust account can represent more than one email address. Thus, even in idealized circumstances Sybil attacks are infeasible.

The rating-servers, which grant account resources, can throttle this kind of excessive account creation with proof-of-work [17] schema. The problem could be made even more expensive by automated client-side validation of email invites. At the most rudimentary level, validation compares the email address commitment with the sender address. A more sophisticated version adds a header check for valid domain keys [12].

Net Trust's encrypted export format (Figure 2) ensures *read integrity* with symmetric-key encryption and with keyed collision-resistant hashing of the cleartext ratings. Since streaming operation of block ciphers encrypts by XORing plaintext with pseudorandom strings of the necessary length, ciphertexts are vulnerable to predictable modification: negating a ciphertext bit also negates the corresponding plaintext bit. If an active attacker modifies the encrypted ratings with this method, he must also update the friend header blocks to include the correct checksum. This is not possible because the attacker does not know the shared secret key (e.g., $k_{Alice,Bob}$) necessary to compute the collision-resistant hash. The same barrier prevents an active attacker *who knows the ratings access key, k_R,* from rewriting the ratings and encrypting again with k_R. If the checksum were unkeyed hash, the attacker could compute the hash of the original and the bogus record, then flip the appropriate bits in the encrypted header blocks. Thus ratings-servers may be granted read access—as form of compensation— without compromising read integrity.

A related—but much weaker—active attack is the *replay attack*, where a malicious adversary intercepts honestly constructed messages and re-transmits them (possibly out of order and possibly dropping some) to the intended recipients. Since ratings bear a time stamp, undetected out-of-order replay of messages is not possible, although an adversarial ratings-server may selectively delay or deny service.

Write authentication is achieved through secret sharing at the time of account registration. The Net Trust client follows the HMAC protocol for authenticated server writes [15]. The client leverages SSL to authenticate rating-servers; valid rating-servers present certificates signed by the Net Trust certificate authority.

In general, it is very difficult to tell when web site visits are the result of user actions or scripted actions. Net Trust mitigates ratings manipulation through *third party scripting* with a inter-visit threshold. Visits which increase ratings must be separated by a minimum time, currently set to one hour. This prevents a web-page's javascript from reloading the page multiple times to "rate-up" the URI. Web-scripting can still load other pages—each one incurring a one hour re-visit clock—in pop-up/under windows or by sequentially visiting a set of colluding pages. Anti-pop-up/under technology embedded in Firefox mitigates the first attack, however we implement no countermeasures for scripted sequential visits inside the same window. Emergence of such an attack would be evidence of Net Trust's widespread adoption—we defer a solution until this point.

Attacks to manipulate an individual's social network must be directed at that individual's client rather than any centralized server. Right now, remote manipulation of social networks by third-party scripting presents only minimal risk since there is no interface for remove management. However, such attacks (and far more) would be possible if third-party scripts can compromise the *chrome* [19] level of access in Mozilla—a privileged mode that includes full user-level access to the filesystem, bookmarks, browser history, and program execution.

To our knowledge, Net Trust's purely local social-network administration is novel. In this paradigm, a critical security property is *owner-enforced access control*, essentially imposing membership symmetry. Peers grant read access to friends by generating encrypted header blocks (with shared secret-key, $k_{Alice,Bob}$) that contain the access key, k_R, to the ratings file. Peers revoke access by replacing the header block with pseudorandom data. Since peer-producers randomly select k_R for every ratings update, prior knowledge of k_R will not decrypt future posts.

6.2 Privacy Properties

Net Trust's separation of personal identity and social network data from the centralized rating-servers form the foundation for many of its privacy properties.

Extracting the personal identity (i.e., the email and nym commitment) from the reference-key is computationally intractable because of the randomly chosen blinding constant and the hash function; without the blinding constant, it might be possible to extract the email and nym with a dictionary attack over likely pairs.

If ratings-servers have access to peer-rating sets, then the partial web-history could potentially identify the user. Net Trust stores only the URI's domain and top-level directory in the history, reducing the danger that identifying information—such as user-IDs that are common in CGI variables—is directly committed to the rating-set. Net Trust's *limited URI record* prevents the contents of searches, specific online profiles, and many other dynamic web details from propagating to rating-servers and peers. Still, this privacy mechanism depends on web site organization; visits to personal web pages, blogs, and employer/employee resources can bind an offline identity to a Net Trust reference key. Rating-servers may further attempt classify or identify users based on their IP addresses. Such attacks are mitigated by using anonymizing tools, such as proxies or Tor [6], to mask the client machine's IP address.

Personal identities are already disclosed to friends in the form of self-selected nyms and email addresses.

However, Net Trust is a method which leveraging one's existing social network, not a method for discovering new ones; friends should know each other's email addresses independently of the Net Trust system.

The ability to open accounts in Net Trust without verification of email addresses or nyms creates a *deniability* property. If someone wants to publicly expose the browsing habits of an individual by disclosing the reference key and generating information, there is little assurance that the account (and hence the history) is controlled by the same person who controls the email address. Our system specifically avoids public-key digital signatures on the ratings files to preserve deniability. Jane Doe could have opened a Net Trust account using a reference-key generated with Don Jones's email and nym. Rating-servers create accounts based only on reference-keys; they are oblivious to the generating information. One way deniability could be compromised is by sending a phishing email to the victim which urges them to visit an unusual web site. The victim is identified if the web site subsequently appears in the rating-set corresponding to the suspected reference-key.

While identities of immediate friends are shared among peers, the reference-keys (and personal identities) to friends-of-friends remain inaccessible. It is very difficult for peers to discover other members of a friend's social network are since there is no remote interface for accessing this information. Moreover, the encrypted header blocks leak no social networking information since they are indistinguishable from random data; header size is fixed due to padding.

Ratings-servers have an advantage for inferring network-membership among peers since they have a central view of ratings access. Their basic strategies for discovering social networks are client IP address logging and timing analysis on read requests. IP logging would be the most expedient method to track social networks; however, it is easily defeated through IP anonymizers. Timing analysis of read events can also link friends. The naive synchronization protocol, which updates all ratings at the time of persona load, trivially discloses the network to a server which logs access times—it simply looks for clustered access. The current synchronization mechanism counters this by randomly scheduling updates over a parameterized time window. These parameters are adjustable to accommodate a changing anonymity set—a value that will grow as Net Trust gains more users.

The Net Trust synchronization policy and history format limits the tracking of user activity. The most invasive level of monitoring would record the time of every visit and send high-frequency low-latency updates to rating-servers. Net Trust only records the initial visit date, last visit date and visit count (a value bounded by 5). Like read updates, server write updates are scheduled with parameterized frequency and delay. Thus, even malicious parties with read access who poll ratings-servers for updates are throttled by the client's choice of update granularity.

7 Conclusion

In summary we have designed a privacy-enhancing distributed reputation system for identifying malicious web sites. We have argued that this design enables both peer-production of web-site ratings *and* individual control of personal data. *Sybil attack resilience*, *identity commitment*, *server authentication*, *write authentication*, *server-to-client encryption*, *timing thresholds* and *read integrity* protect the system from malicious ratings manipulation. While based on individual web browsing habits and opinion, Net Trust ensures privacy with *owner-enforced access control*, *linking resistance*, *confidential social networks*, *account deniability*, and limited disclosure of browsing details. Net Trust exists today as an operational prototype which constructs online reputation by sharing web histories among self-selected social networks. All ratings and social network behavior is completely functional, while implementation of the encrypted exchange format is an ongoing process. A daily build is available at `nettrust0.ucs.indiana.edu`.

References

[1] Alexa web search — privacy policy. Accessed 15 June 2007, `http://www.alexa.com/site/help/privacy?p=TBStartpage_W_t_40_B1`, 17 March 2003.

[2] APWG. Phishing activity trends: Report for the month of April, 2007. Accessed 15 June 2007, `http://www.antiphishing.org/reports/apwg_report_april_2007.pdf`.

[3] L. J. Camp. Reliable, usable signaling to defeat masquerade attacks. The 2006 Workshop on the Economics of Information Security *(WEIS 2006)*, 26-28 June 2006.

[4] R. K. Chellappa and S. Shivendu. Incentive design for free but no free disposal services: The case of personalization under privacy concerns. The 2007 Workshop on the Economics of Information Security *(WEIS 2007)*, 7-8 June 2007.

[5] N. Chou, R. Ledesma, Y. Teraguchi, and J. C. Mitchell. Client-side defense against web-based identity theft. In *NDSS*, 2004.

[6] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[7] J. Douceur. The Sybil Attack. In *Proceedings of the 1st International Peer To Peer Systems Workshop (IPTPS 2002)*, March 2002.

[8] A. Friedman. Information networks and social trust. Social Science Research Network, February 2006. Accessed 15 June 2007, `http://papers.ssrn.com/sol3/papers.cfm?abstract_id=882370`.

[9] A. Genkina and L. J. Camp. *Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft*, chapter Case Study: “Net Trust”. John Wiley & Sons, 2007.

[10] Google safe browsing for Firefox. Accessed 15 June 2007, `http://www.google.com/tools/firefox/safebrowsing/`, 2007.

[11] T. Hammond, T. Hannay, B. Lund, and J. Scott. Social bookmarking tools (I): A general review. D-Lib Magazine, April 2005. Accessed 15 June 2007, `http://www.dlib.org/dlib/april05/hammond/04hammond.html`.

[12] T. Hansen, D. Crocker, and P. Hallam-Baker. DomainKeys Identified Mail (DKIM) Overview: draft-ietf-dkim-overview-05. IETF Internet Draft, 11 June 2007. Accessed 15 June 2007, `http://www.ietf.org/internet-drafts/draft-ietf-dkim-overview-05.txt`.

[13] P. Hariharan, F. Asgharpour, and L. J. Camp. Nettrust — recommendation system for embedding trust in a virtual realm. Accessed 15 June 2007, `http://www.ljean.com/files/Recommend2007.pdf`.

[14] M. Kotadia. Samy opens new front in worm war. cnet News.com, 15 October 2005. Accessed 15 June 2007, `http://news.com.com/Samy+opens+new+front+in+worm+war/2100-7349_3-5897099.html`.

[15] H. Krawczyk, M. Bellare, and R. Canetti. Hmac: Keyed-hashing for message authentication. RFC 2104, February 1997.

[16] D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins. Geographic routing in social networks. *Proceedings of the National Academy of Sciences*, 102(33):11623–11628, 2005.

[17] D. Liu and L. J. Camp. Proof of work can work. The 2006 Workshop on the Economics of Information Security *(WEIS 2006)*, 26-28 June 2006.

[18] B. Markines, L. Stoilova, and F. Menczer. Bookmark hierarchies and collaborative recommendation. In *Proceedings of The Twenty-first National Conference on Artificial Intelligence (AAAI-06)*, 16-20 July 2006.

[19] N. McFarlane. *Rapid Application Development with Mozilla*. Prentice Hall PTR, 2003.

[20] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, pages 415–444, 2001.

[21] M. Meiss, F. Menczer, S. Fortunato, A. Flammini, and A. Vespignani. Ranking web sites with real user traffic. In preparation, 2007.

[22] Netcraft anti-phishing toolbar. Accessed 15 June 2007, `http://toolbar.netcraft.com/`.

[23] A. Odlyzko. Privacy, economics, and price discrimination on the internet. In *ICEC '03: Proceedings of the 5th international conference on Electronic commerce*, pages 355–366, New York, NY, USA, 2003. ACM Press.

[24] M. Security. Phishing filter: Help protect yourself from online scams. Accessed 15 June 2007, `http://toolbar.netcraft.com/http://www.microsoft.com/protect/products/yourself/phishingfilter.mspx`, 2007.

[25] Stumbleupon ≫ privacy policy. Accessed 15 June 2007, `http://www.stumbleupon.com/privacy.html`, 9 May 2007.

[26] Y. Zhang, S. Egelman, L. Cranor, and J. Hong. Phinding phish: Evaluating anti-phishing tools. In *Proceedings of the 14th Annual Network & Distributed System Security Symposium (NDSS 2007)*, San Diego, CA, March 2007.

A Browsing data at Indiana University

Top 10 Non-IU `.edu` domains

Domain	% of total traffic	note
`www.npac.syr.edu`	0.08422	
`cc.ivytech.edu`	0.03330	local community college
`www.bsu.edu`	0.02227	in-state college
`www.mccsc.edu`	0.01955	county-public school system
`www.ivytech.edu`	0.01296	local community college
`www.south.mccsc.edu`	0.01274	local high school
`www.purdue.edu`	0.01238	in-state
`www.columbia.edu`	0.01015	
`muse.jhu.edu`	0.00825	
`owl.english.purdue.edu`	0.00807	in-state

Table 1: There is a high instance of local institutional traffic. It is surprising that a local high-school garners more traffic than IU's in-state sibling, Purdue University.

Top 10 News domains

Domain	% of total traffic	note
`www.idsnews.com`	0.99893	university student paper
`www.msnbc.msn.com`	0.94356	
`www.cnn.com`	0.52084	
`espn.go.com`	0.23755	
`news.chinatimes.com`	0.19678	
`sportsillustrated.cnn.com`	0.18627	
`www.foxnews.com`	0.18548	
`newsinfo.iu.edu`	0.17503	university press office
`iuhoosiers.cstv.com`	0.16467	university athletics news
`www.nytimes.com`	0.16320	

Table 2: The university student paper tops the list. Note the high interest in sports reporting. It is surprising that the China Times outranks the New York Times.

Top 10 Webmail Domains		
Domain	% of total traffic	note
`mail.google.com`	0.76486	
`webmail.iu.edu`	0.46231	university mail
`www.exchange.iu.edu`	0.23175	university mail
`www.hotmail.com`	0.22801	
`mail.yahoo.com`	0.19018	
`webmail.aol.com`	0.13843	
`webmail.indiana.edu`	0.10354	university mail
`mail.daum.net`	0.07011	a Korean language site
`www.gmail.com`	0.06197	
`webmail.iupui.edu`	0.05639	university mail

Table 3: Though Indiana University servers rank highly in this list, Gmail exceeds their capacity. It is interesting that a Korean language webmail service outranks the IUPUI webmail.

Top 10 Financial Institutions		
Institution	% of total traffic	note
Chase	0.05860	
IUCU	0.03500	university credit union
TIAA-CREF	0.01795	university retirement manager
National City	0.01631	
IMCU	0.01057	state credit union
Fidelity	0.00825	university retirement manager
Fifth Third	0.00745	regional bank
Regions Bank	0.00548	regional bank
Bank of America	0.00522	
Monroe Bank	0.00467	local bank

Table 4: Note the high instance of regional traffic.