Membership-Concealing Overlay Networks

Eugene Vasserman Rob Jansen James Tyra Nicholas Hopper Yongdae Kim
{eyv, jansen, tyra, hopper, kyd}@cs.umn.edu
University of Minnesota
Minneapolis, MN 55404

ABSTRACT

We introduce the concept of membership-concealing overlay networks (MCONs), which hide the real-world identities of participants. We argue that while membership concealment is orthogonal to anonymity and censorship resistance, pseudonymous communication and censorship resistance become much easier if done over a membership-concealing network. We formalize the concept of membership concealment, discuss a number of attacks against existing systems and present real-world attack results. We then propose three proof-of-concept MCON designs that resist those attacks: one that is more efficient, another that is more robust to membership churn, and a third that balances efficiency and robustness. We show theoretical and simulation results demonstrating the feasibility and performance of our schemes.

Categories and Subject Descriptors

C.2.4 [**Computer-communication Networks**]: Distributed Systems—*Distributed Applications*; C.2.0 [**Computer-Communication Networks**]: General—*Security and Protection*; K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*

General Terms

Algorithms, Security

Keywords

Security, Privacy, Membership Concealment, Peer-to-Peer Networks

1. INTRODUCTION

One now-widespread threat to the freedom of online speech is the practice of Internet censorship by private and state interests, who use a variety of social and technological means to limit expression or availability of information. The Open Net Initiative (ONI) [1], which catalogs world-wide censorship efforts, categorizes them into four different categories: (i) technical blocking (such as DNS filtering), IP blocking, URL filtering, and content inspection; (ii) search removal, i.e. suppression of web sites or terms from search engines; (iii) take-down, the use of legal or regulatory power to demand the removal of content; and (iv) induced self-censorship, through intimidation including surveillance or the perception of surveillance. In 2006, ONI reported strong evidence of filtering in 26 of 40 countries surveyed [13], with anecdotal evidence suggesting widespread use of social and legal means as well. This list includes Western democracies such as the US and EU member nations. Such prevalence suggests that censorship by governments, ISPs, and corporations represents a valid threat to freedom of speech on the Internet.

As more censorship-*enabling* systems are deployed, we will see increased usage of censorship-*resistance* technologies – tools designed to circumvent the technological filters. However, the use of currently-deployed censorship resistance systems such as Tor bridges [14] and Freenet [10] is problematic if they are explicitly proscribed. If use of censorship circumvention technologies is punishable, then any such system *should also prevent the censor from identifying the system's participants*. We call such systems "membership-concealing networks," and argue that many schemes claiming to provide anonymity or censorship resistance are also trying to achieve membership concealment.

This paper is concerned with the study of membership concealment as an end in itself rather than a means or side effect of another goal. We introduce Membership-Concealing Overlay Networks (MCONs), which are peer-to-peer (P2P) overlays whose membership set is hidden from both insiders and outsiders. Such systems should allow communication while obscuring "real-world" identities of participants. Overlays and membership concealment may sound incompatible, since nodes must always rely on others for communication and connectivity, but it is possible to *minimize the number of other overlay nodes who know the identity of any given node*, to the point where one only needs to disclose one's identity to a small constant number of other nodes. Such systems need pseudonyms to allow for one-to-one communication. Pseudonyms should preserve unlinkability between MCON identities and real-world identities, whether for targeted individuals or for a non-trivial fraction of MCON members. Finally, MCONs must preserve availability by being robust against churn and support scalable and efficient routing and search.

1.1 Relationships between concepts

The concept of membership concealment is not new: organized crime and terrorist networks routinely use compartmentalization to hide the identities of cell members from people outside a given cell (a network is composed of many cells, which mostly act independently). Such networks are not foreign to the computer science community either: overlays with some membership concealment properties have been used for covert activity, such as sharing classified, censored, or copyrighted content. Generally called "darknets,"

CCS'09, November 9–13, 2009, Chicago, Illinois, USA.

these networks are built to be difficult to join or detect, but most do not protect from malicious insiders. One typically becomes a member through social means: an existing member "vouches" for the newcomer [5]. Academically, membership concealment networks have remained less explored than, and frequently confused with, related technologies such as *privacy*, *anonymity*, *unlinkability*, *unobservability*, *pseudonymity*, and *censorship resistance*.[1]

Unobservability. Related to anonymity, unobservability is usually endowed with one of two meanings. Pfitzmann and Hansen define the term to mean that a principal in an anonymity scheme cannot be "observed" to be sending or receiving a message (i.e. other nodes cannot determine whether a node sent or received a message at any particular time) [35]. Some authors have interpreted this to mean that it is difficult to distinguish whether a principal participates in the network or not [25, 19]. The former clearly does not imply membership concealment: a scheme that is unobservable in this sense would remain unobservable if all principals periodically announced their participation. The latter sense is membership concealment in terms of an outsider-only attack, since it is in general necessary for *some* participants to be revealed to others in order for messages to be delivered.

Pseudonymity and anonymity. Pseudonymous credential systems [9, 31, 37] dissociate real-world identities from semi-persistent network identities (pseudonyms). A real-world identity is any information that may reduce the set of candidate identity-pseudonym pairings by a non-trivial amount, such as names, credit card numbers, or IP addresses. MCONs must use pseudonyms to address members, and *for a system to be membership-concealing it must be impossible, with overwhelming probability, to determine the real-world identity of a user with a given pseudonym*.

Anonymity, on the other hand, does not have the persistent identity property, but instead hides any and all identifying information. Consider the relationship between *anonymity* and *membership concealment*. The main goal of an anonymous network is to conceal who is communicating with whom. However, this *unlinkability* or "relationship anonymity" does not require concealment of *who participates in the overlay*, and in fact a scheme with perfect relationship anonymity would not sacrifice this property if the list of participants was broadcast on a regular basis. On the other hand, membership concealment does not guarantee that messages cannot be linked, e.g. each message may contain the pseudonym of both its source and destination, destroying relationship anonymity but preserving membership concealment. MCONs clearly require some type of minimal pseudonymity to prevent a passive insider from simply harvesting identities – for example, messages should not include the real identity of the originator.

While aspects of some anonymity schemes in the literature can be seen as implicit efforts to provide membership concealment, e.g. Bauer's scheme seeks to hide the users of a mix net among a larger set of web users [4], no deployed anonymity scheme explicitly claims to provide membership concealment, and it is largely accepted that sender anonymity (origin obfuscation) can be achieved without it [16, 38]. Some schemes, such as Tarzan [21], explicitly distribute a list of members. However, since this information simplifies certain variants of the intersection attack [48], recent P2P anonymity schemes such as Salsa [33] have mentioned hiding the membership list as a security goal. Unfortunately these schemes do not provide membership concealment under adversarial conditions.

Censorship resistance and availability. Censorship-resistant networks are designed to prevent adversaries from denying users' access to a particular resource or file. This type of system does not require membership concealment: most are designed such that it is difficult to determine what content a given user is accessing or what node is hosting a given file, preventing targeted attacks. Such systems remain censorship-resistant even if the list of participants was public. Membership concealment does not imply censorship resistance: a membership-concealing network may serve unencrypted content, so censorship would only require blocking files that contain selected keywords even if the identities of communicating nodes are hidden. This is similar to the approach used by China's "Great Firewall" [50].

A particularly critical requirement for censorship resistant networks is availability, since an attack against availability is in itself an act of censorship. Some censorship-resistant systems [47] have taken the all-or-nothing approach, assuming that an adversary would want to disable access to selected content, but not to the entire system. We advocate a strictly more powerful adversary model – one that is willing to prevent access to an entire system in order to block some targeted content. Events such as [2, 3] support our position. Even without joining a network or identifying its members, an adversary can block access to "undesirable" content on a large scale by using deep packet inspection or encryption-oblivious protocol fingerprints (and blocking matching packets). Infranet [18] addresses this problem by using steganographic techniques to hide content requests and responses. However, it requires active participation of a number of web servers. Feamster *et al.* extend the Infranet service by adding an extra layer of indirection in the form of untrusted messengers, who pass requests to a forwarder, who then fetches the actual censored content [19]. Tor bridges [14] (discussed in more detail in Section 3) add censorship-resistance functionality to the Tor anonymous overlay [16]. The design is somewhat similar to Infranet with untrusted intermediaries, and both are vulnerable to many similar attacks.

1.2 Proposed design

We propose three proof-of-concept designs – one that is more efficient, another that is more robust to membership churn,[2] and yet another which is a hybrid of the first two. All schemes are robust against insider and outsider attack, including targeted attack and network partitioning. Our MCON can be bootstrapped from any social graph of offline face-to-face relationships. (Basing a network on a social network graph allows us to use Sybil attack [17] mitigation systems such as SybilLimit or SybilInfer [49, 23].) Membership is by invitation only, so our network is not "open" in the same sense as other P2P systems, which allow anyone who knows at least one member to become a member themselves. Finally, our designs use distributed hash tables (DHTs) to enable efficient search and ensure that both popular and rare files can be located within a predictable period of time.[3] DHTs are structured overlay networks that allow for very efficient searching [45, 39, 30]. Each DHT node has a random pseudonym, is responsible for responding to queries that are lexicographically close to that pseudonym, and maintains a routing table of $O(\log N)$ peers that enable it to efficiently identify the node responsible for a query.

2. MCON REQUIREMENTS

Informally, we define an MCON to be a *communication system that hides the identities of its members* from both insider and outsider attackers (network members and non-members, respectively), while retaining members' ability to communicate efficiently. The goal is to reveal no information about the network participants that would allow them to be identified in the "real world." (From now

[1] For a thorough treatment of some of these terms, see [35].

[2] Members can go offline without disrupting the network

[3] Files can be arbitrary named data, so "locating files" does not imply a traditional file-sharing system.

on we will refer to the human participants as "users," while denoting their computational presence in the network as "nodes.") Honest users have one fixed network pseudonym, which allows other members to uniquely address them. (We will refer to overlay-level identities as "pseudonyms" and real-world identities as "identities.") For the purposes of this paper, we assume that obtaining a node's network (IP) address is both necessary and sufficient to identify the real-world user of the network.[4]

In addition to hiding member information, this network must be robust to link failure and partitioning: we must maintain *availability* both in the presence of normal network events and attackers. (A related requirement is *node-equity*, i.e. no node is more important to the network than another.) It should also be scalable, allowing for the membership set to grow while maintaining routing efficiency and minimizing communication, computation, and storage overhead. Finally, it should provide efficient search functionality, which can reliably locate any information stored in the network within a predictable time window.

We assume an adversary with the resources of a large ISP or state government. This means that the adversary can monitor or disrupt traffic on some fraction ℓ of links; can communicate with arbitrary nodes on the network; and can selectively "corrupt" or otherwise assume control of some fraction γ of selected nodes. We call this an (ℓ, γ)-adversary. Formally, we say that an overlay network protocol is (Λ, Γ, f)-membership-concealing if no (ℓ, γ)-adversary monitoring $\ell \leq \Lambda$ links and corrupting $\gamma \leq \Gamma$ members can identify more than $f(\gamma, \ell, N)$ members, where N is the total number of MCON participants. When $f(\gamma, \ell, N) = \Theta(\gamma + \ell)$ we call the protocol a membership-concealing network protocol. We note that no overlay protocol that permits communication between peers can be $(\Lambda, \Gamma, o(\Lambda + \Gamma))$-membership-concealing since at least one node must deliver messages to each corrupted or monitored identity, and an adversary can always choose to corrupt or monitor identities with no common neighbors.

3. RELATED WORK

Arguably the first darknet was WASTE [20], designed to facilitate secure collaboration by small groups. Some file sharing applications have recently added darknet features [7], and applications for "friend-to-friend" (F2F) sharing have been developed [27]. The latter scheme is meant to allow sharing through trusted intermediaries, preventing the disclosure of the uploader's identity. It is also fundamentally different from previous darknet designs, since it hides the network member set even from other network members. Unfortunately, all of these networks have similar problems such as forming partitioned groups instead of larger networks, scalability limitations, search efficiency issues, and security vulnerabilities.

3.1 Freenet

The system that is currently most similar to an MCON is Freenet [10]. It is a censorship resistant network which hides the publisher, querier, and storage location of files by obfuscating their names and contents, making it difficult for any party other than the querier to identify the content that is being retrieved. Moreover, Freenet uses recursive routing to reduce the number of nodes who are aware of each other's existence. (Recursive routing proceeds by flooding through intermediate nodes instead of directly between source and destination.) Freenet version 0.7 is designed to allow for two modes of operation: in *opennet* mode, nodes may freely connect to any other opennet node, while *darknet* mode allows connections with other nodes only by prior out-of-band agreement, presumably based on mutual trust [41]. This provides protection from malicious nodes crawling Freenet for membership information. Note that because darknet nodes do not communicate with opennet nodes, there may be many disconnected darknets instead of one large network.

[4]If users voluntarily disclose their real-world attributes then IP addresses become sufficient, but not necessary, to de-anonymize them.

3.2 Tor bridges

Tor [16] is a popular anonymizing network that offers sender anonymity. It consists of a relatively small number of dedicated volunteer "routers," and is thus easily blocked at a national or service provider border by disallowing all connections to those dedicated hosts. Tor designers are actively working to add "bridge" functionality [14] that would make it more difficult to block. Tor bridges are not dedicated routers; they are Tor clients who allow users in censored regions to contact them directly as a first step into the Tor network. Since bridges are client nodes, they are more numerous and experience higher churn than dedicated relays, so blocking them is a more difficult task. This is implicitly a membership concealment feature. Tor currently relies on a publicly-known centralized authority or out-of-band (social) communication for distribution of bridge descriptors, but the authority can itself be blocked. Although the authority takes precautions to avoid providing bridge descriptors *en masse* to anyone who asks, the system is vulnerable to attack: an adversary who controls many IP addresses can query the authority repeatedly, pretending to be different nodes behind different IP addresses.

3.3 Other systems

Other anonymity schemes [25, 19] have also attempted to provide "blocking resistance" by hiding their members among a larger set. However, even if an adversary cannot block access to all members of an overlay, he might be able to block queries for particular types of content. Since most storage networks provide an efficient lookup feature, an adversary knowing the identifying information of the content (hash, ID, etc.) can look up the node(s) storing that content and selectively deny access to those nodes. Censorship resistance requires blocking resistance, but both are orthogonal to membership concealment.

3.4 Using social networks to bootstrap trust

Freenet darknet limits identity disclosure to trusted peers, selected from a network of untrusted members based on past performance and off-line relationships. Turtle [36] is one example of a network that *bootstraps from a social network that expresses mutual trust*. Like Freenet, queries are flooded and do not terminate until either every node in the network has responded or the maximal query depth is reached. Kaleidoscope [44] also uses social networks to distribute proxy information, mitigating Sybil attacks (it is far more likely that Sybil nodes are connected to adversaries than honest nodes). However, the system uses a centralized server to distribute information about proxies to newly-joining nodes, so it is vulnerable to the same attacks as the Tor bridge authority.

Danezis *et al.* also use social networks to bootstrap a Sybil-resistant DHT [11]. Based on the same assumption as above – that adversaries are connected to a social network in few places compared to honest members – the Sybil-resistant DHT builds trust profiles for individual nodes along a query path and favors nodes who usually yield correct results. Since the majority of adversarial (Sybil) nodes will be connected to the DHT through very few honest nodes, those connection points will (with high probability) return Sybil nodes as next hops, eventually producing incorrect results when adversarial nodes misbehave.

4. ATTACKS ON EXISTING SYSTEMS

The primary goal of MCONs is resistance to *member identification attacks*, in which either an insider (MCON member) or an outsider attempts to determine the "real-world" identities of network

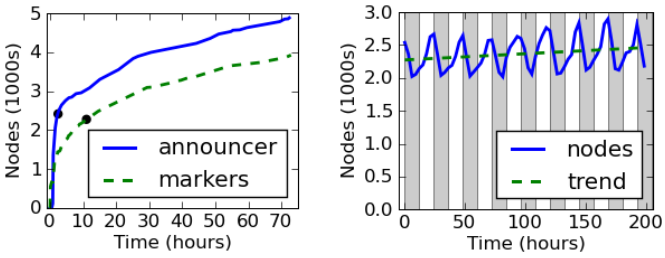
members. This attack may take two forms: existential and targeted. In the former case, which can also be called the *harvesting attack*, an adversary attempts to determine the identities of as many network members as possible. The latter attack allows an adversary to match a network pseudonym with an identity, or to significantly reduce the number of candidate identities for a given pseudonym as a precursor to rubber-hose cryptanalysis.

Most existing systems are vulnerable to at least one type of harvesting attack. The simplest variant exploits systems that do not limit the number of identities that a single member can collect simply by querying the network repeatedly. Since the attack is active, it may be detected and the attacker could be blacklisted, but the adversary can always throttle or otherwise mask his actions to appear benign. A harder-to-detect variation is the *passive harvesting attack*: an adversary runs a network node that logs all direct communication attempts, learning the identities of all other nodes over a long-enough timeline. Both attacks become faster with more adversaries. Another example of a harvesting attack is the *multiple join*, or *bootstrap* attack, in which an adversary either sequentially joins the network multiple times at multiple logical locations (which are the Freenet equivalent of DHT IDs), or creates multiple (Sybil) nodes and simultaneously joins them to the network. Since every joining node must obtain the identity of at least one other network member, multiple joins allow the adversary to learn the IP addresses of a large fraction of network members.

The *celebrity attack* affects systems that use social networks to bootstrap trust [10, 36, 44, 27]. If the social network topology can be discovered then an adversary can choose to corrupt or monitor nodes with many friends, learning disproportionally many other network members. (Mislove *et al.* report node degrees of up to $10,000$ in many popular social networks [32].) Only a few very popular network nodes need to be corrupted or monitored in order to learn the vast majority of network members [12]. This can be generalized to attacks against "tasty targets," applicable when networks that bootstrap from social networks but do not "smooth out" node degree. It also applies to networks with so-called "supernodes" – members who have more power than other members. An MCON should either not contain any targets of compromise that know disproportionally more member information than any other target, or should ensure that such nodes are difficult to identify.

Social networks also expose the constructed MCON to a graph overlap attack – Narayanan and Shmatikov have recently shown that anonymized graphs can be de-anonymized based only on topology knowledge and access to an overlapping non-anonymized graph [34]. This means we cannot anonymize a graph by simply replacing identities with pseudonyms; we must also restrict adversaries from constructing a complete view of the anonymized graph topology and/or perturb the node degree. While some of the above networks would resemble MCONs more closely if they were not vulnerable to the celebrity attack, most of them expose their topology while not enforcing node degree limits.

Another serious attack on a membership-concealing network is the *confirmation attack*. If an MCON requires nodes to respond in a distinctive way to connection attempts, then a non-member adversary can "cast a wide net" and identify a large number of nodes by attempting to connect to them. As an example, consider a network administrator at a large corporation who wishes to identify users on the internal network who are using a file-sharing application. Assume that the most popular application uses a certain port number in the default configuration. Our adversarial network administrator can "probe" each host on the internal network, connecting to that default port, identifying users by the tell-tale replies from the file-sharing client.



(a) **(b)**

Figure 1: Results of Freenet attacks. (a) Total unique Freenet nodes found over time. Dots show the time when all marker nodes were found. (b) Unique running nodes for each 3-hour time period. The cycle is likely the result of day-time versus night-time usage patterns.

Finally, MCONs must resist *protocol identification attacks* when communicating with other network members as well as when joining, leaving, or inviting others to join the MCON. Such an attack would allow passive identification of MCON users by monitoring communication patterns without peeking at content [46]. Consider once again our sneaky system administrator from above. Since there are only a small number of exit points from the internal network to the Internet, our adversary can monitor protocol traffic at those locations, identifying all users of the targeted protocol.

4.1 Attacking Freenet opennet

We implemented a *passive harvesting attack* using well-behaved Freenet clients[5] in "opennet" low-security mode whose only modified behavior is passively logging communication with others. We call these nodes "markers" because we use them to measure the success of our attack – since they have random pseudonyms, the time required to locate all marker nodes will be close to the upper time bound to find all nodes in the Freenet network. We also implemented an *active harvesting attacker*, which announces itself to random logical locations in the network, collecting pseudonyms and IP addresses of responding Freenet nodes (who are located "near" the announcement point in the Freenet logical coordinate space). To eliminate the effects of dynamic IP addresses we only counted node pseudonyms, which are unique and constant.

Figure 1(a) provides a comparison of each of our attacks on Freenet, using 80 marker nodes and a single announcer. Note that the single active attacker outperforms all 80 passive attackers, but collection speed can always be increased by adding more attackers – the bandwidth and processing costs are not a bottleneck. The dots represent the time when each attack discovered all 80 marker nodes, signalling that we have likely enumerated the majority of running nodes. Our passive attackers were able to enumerate all markers in 11 hours, and the active attacker found all of them in 2.5 hours. Figure 1(b) shows the membership of opennet derived from snapshots of 3 hours each – since it took our announcer less than 3 hours to find our markers, we expect this graph to be an accurate measure of the membership of opennet at any given time. (The shaded areas are 8pm to 8am GMT.) We observed between 2,000 and 3,000 *running* opennet nodes at any given time. While the total number of *existing* opennet nodes cannot be counted accurately since a large number are likely to remain offline for the duration of the experiment, we discovered a total of 11,100 unique node pseudonyms.

4.2 Attacking Tor bridges

We also launched a *passive harvesting attack* against Tor bridges using an unmodified Tor router, configured as a middleman (non-exit) node and offering 100MB/sec of bandwidth (attracting disproportionally many client connections). Our router should receive connections from clients, bridges, and other Tor routers. We weed

[5]Based on Freenet 0.7 Build #1204 r25665 (2-17-2009)

out routers using several tests, including TLS handshake fingerprinting, querying all running directory authorities to see if they know the router, as well as by connecting back to the router and examining its descriptor [15]. Once we eliminate the routers we are left with clients and bridges. To differentiate between them we attempt a connection using common bridge ports. If our connection succeeds and we get a bridge descriptor [14], then it is not a client and we launch a *confirmation attack* by extracting the fingerprint from the descriptor and querying each directory authority. A router will appear in at least one, while bridges will not. (Note that these tests are all performed in real time.) Since we expect that all bridges will eventually connect to our router due to Tor's selection rules, we can eventually build a complete list of bridges. We collected 61 unique Tor bridge identities in 4 days, a clear vulnerability in the membership concealment capability of Tor bridges.

5. DESIGN

This section outlines three proof-of-concept MCON designs.[6] Using a social network based on offline relationships as a starting point, we bootstrap our MCON from a small fully-connected "seed" network of "social neighbors" (nodes connected by an edge in the social network). The MCON grows by having existing members invite new nodes based on social relationships. When joining, nodes are assigned persistent pseudonyms and DHT IDs. After the MCON is built and after a period of DHT routing table discovery, we use a VRR-like protocol [8] to allow nodes to communicate with the DHT over a small set of "physical neighbors." We define physical neighbors as those nodes who are allowed to directly communicate over IP. (This system may be considered a double overlay – we use DHT communication for efficient search, over a source-routing overlay, over IP.) To avoid celebrity attacks, every MCON node can only communicate directly with a constant k other nodes, since direct IP communication is sufficient to break membership-concealment. Most communication takes place through the DHT overlay, which connects any two nodes by $\log N$ logical hops, where N is the total number of members. This contributes significantly to the scalability and efficiency of our system.

Many popular DHT designs use iterative routing, where a node will communicate with its DHT neighbor to ask for the IP address of the next DHT hop, with which it will then communicate directly. The process repeats until the desired node is found, at which time the origin and the destination can communicate directly over IP. We cannot achieve membership concealment in the iterative scheme, since an adversary can learn the IP addresses of all intermediate nodes as well as the final destination by repeatedly searching the network. In recursive routing, nodes communicate only with their DHT neighbors, who forward requests to the next DHT hop on behalf of the originator. In this scheme, all communication between source and destination happens through multiple intermediaries. Sometimes recursive routing carries the benefit of plausible deniability of query origin – a node receiving a message from a physical neighbor cannot distinguish whether that neighbor originated or forwarded the message.

Our design relies on a trusted central authority (the Membership and Invitation Authority, or MIA) to invite new nodes into the MCON and act as a key issuer. The MIA is also responsible for keeping track of node degree, ensuring that nodes do not exceed the global constraint. To prevent Sybil attacks, the authority can use existing systems such as SybilLimit or SybilInfer [49, 23], which use a social network to bound the number of Sybil identities accepted into the network. Note that *membership concealing properties of our scheme do not depend on the number of Sybil nodes in the network*, provided they are not connected to honest nodes. Since honest nodes will not directly communicate with anyone other than their neighbors, Sybil nodes without edges to honest nodes will only affect the robustness of routing in our network, not its security. Since future designs will distribute the functionality of the MIA throughout the network, we want to minimize our current dependence on it. To that end, the MIA is only needed when a new node joins. Moreover, it does not have to respond in real time, and so can be offline and does not constitute a central point of failure *for denial of availability attacks*. Unlike the Tor authority, nodes need never contact the MIA directly, so it can remain hidden.

Below we present three MCON designs: the first is more efficient, the second is more robust in high-churn situations, and the third is a hybrid. They can all be split into three major components: **invitation and join**, **route discovery**, and **overlay routing**.

Invitation and join. The network is built by starting from a small "seed" and adding nodes one by one, expanding it to form the full MCON. While the seed can be an arbitrary group of social network nodes matching certain mutual connectivity parameters, growing that network is challenging. In our system, nodes who are already part of the MCON invite other nodes with whom they share connections in the social network. Nodes must receive multiple invitations in order to join the MCON, and the entire process must be somehow mediated to ensure admission control and key distribution for the MCON. While this is currently handled by a centralized entity, future designs will incorporate distributed computation of this information by MCON members.

Route discovery. Once node A has been admitted to the MCON, it must construct a DHT routing table for efficient communication. The routing table consists of sourse routes to other DHT nodes that share different prefix lengths with A. Routes are discovered by flooding requests over the "physical" network. Since nodes only communicate with their neighbors, route responses must conceal information about intermediate nodes. We accomplish this by using private routing tokens. A node building a routing table obtains information about the next hop (one of his direct neighbors), the destination (one of his DHT neighbors), and no information about the intermediate nodes.

Overlay routing. Finally, once a node builds his DHT routing table, he can route to arbitrary DHT keys. As in route discovery, communication happens strictly through the node's "physical" neighbors, and DHT communication is recursively routed. MCON communication consists of two layers: *routing to a DHT hop*, and *routing between DHT hops*. In the first step, the node uses the collected private routing tokens to deliver a message to the first DHT hop. That DHT node will then use her routing table to transport the message to the next DHT hop, and so on until the destination is reached. Nodes never communicate directly with anyone other than their physical neighbors, and layers of encryption prevent the exposure of the DHT message as well as the source and destination. We ensure resistance to confirmation and brute-force scanning attacks by using *strong binding* – an MCON node will only communicate directly with her physical neighbors, ignoring all messages from other nodes (enforced by cryptographic signatures).

5.1 Efficient design

Network construction. We start network construction with a clique of $\lceil k/2 \rceil$ social network neighbors, where k is the maximum number of allowed MCON physical neighbors. The MIA iteratively "grows" the network by finding nodes to invite. Node A can be invited if: a) A has at least $\lceil k/2 \rceil$ social friends in the current MCON, b) those friends have at most $k-1$ "physical neighbors," and c) A is not in the MCON, and has not been previously invited. Call A's friends satisfying (a) and (b) her potential physical neighbors.

[6] We omit full algorithms due to space constraints.

Once A has been identified, the MIA randomly chooses $\lceil k/2 \rceil$ of A's potential physical neighbors, tells them A's new pseudonym, and instructs them to (1) add A to their list of physical neighbors and (2) send an invitation to A with their IP addresses, MCON pseudonyms, public keys, and DHT ID. Once A receives the invitation and joins, the MIA assigns her a private key and a set of identity-based private keys [6].

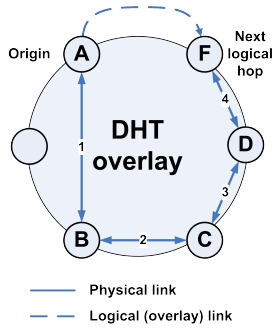


Figure 2: Routing to a logical hop over 4 physical hops

Route discovery request. After joining, A can discover her logical neighbors and build a routing table. The routing table is composed of source routes to DHT nodes whose pseudonyms share a common prefix with A's pseudonym. (For instance, in the Kademlia DHT protocol [30], A acquires K routing table entries for every i-bit prefix of A's pseudonym.) These source routes are discovered by scoped flooding over the physical links of the MCON, analogously to a wireless ad-hoc routing protocol, such as Virtual Ring Routing [8]. A continues sending discovery requests, increasing the scope of each by one, until her entire routing table is filled. (This constitutes a depth-limited breadth-first search of the network.) While expensive, floods are only needed during initial route discovery. A uses these source routes to establish onion routes to each of its routing table entries, similar to Tor tunnels [16]. Onion-wrapped source routes ensure that for most routes neither the source nor the destination learn anything about each other except that they share a common pseudonym prefix. Moreover, most intermediate nodes in a source route know neither the source nor the destination, and cannot determine if any message is addressed to the same node as any other message, ensuring unlinkability.

Route discovery messages are in the form of $(\text{ID}, \text{scope}, IBE_{\text{prefix}}(z, g^x, R))$, where IBE is identity-based encryption [6], scope is the flood depth, g^x is a Diffie-Hellman half-key,[7] R is a route descriptor, z is a random number, and ID is $h(h(z))$, with h being a cryptographically secure hash function. IBE_{prefix} is an identity-based encryption to an i-bit prefix of A's pseudonym [6], meaning that only a node matching the search parameters can open the message. The route descriptor is a random bit-string of some fixed size. A stores z, the prefix, the route, and the DH half-key for later reference. When relaying a route discovery message, each node will decrement the scope by one, dropping messages whose scope is 0. Relaying nodes will also record the request ID and physical neighbor from whom it came.[8] These records are kept either until a reply is received or a timer expires.

Route discovery reply. When node F receives a route discovery request which he can decrypt (meaning F's DHT ID contains the prefix to which the message is encrypted), he generates a DH half-key and composes a response in the form of $(\text{ID}', g^y, R, E_{k_1}(z), \text{MAC}_{k_2}(E_{k_1}(z)))$, where k_1 and k_2 are keys derived from the full DH key, i.e. $k_1 = h(0, g^{xy})$ and $k_2 = h(1, g^{xy})$. ID' is $h(z)$, the pre-image of the request ID. The response also includes R from the route discovery message (unchanged), and a message authentication code (MAC) of z. F keeps a record of z, D, and the DH key for later use. (Every message from the same source must be tagged with z to allow F to look up the shared key.) F sends this response to the physical neighbor D from whom he received the request, and also floods the original request, decrementing the scope. The resulting source route is shown in Figure 2.

Once D receives a route reply, he looks up $h(h(z))$ in a table of previous request IDs to verify that the request was correctly opened and to find the next hop where the response must be sent. He constructs a "route token," encrypts it with his public key, and prepends the resulting ciphertext to the route contained in the response. He also removes an equal-length token from the end of the route string. The new token identifies the next and previous hops along that route, and can only be decrypted by D. The encryption should include a random component to prevent every token that points to the same node from being identical, since the route is visible to intermediate nodes. D then sends the response to the appropriate physical neighbor. In this way, the originator of the request (A) will get back a series of tokens that are meaningless to her, but that comprise a source route to her logical neighbor. Additionally, since A generated the original random route string, she can ensure that it passes some rudimentary sanity checks: she knows the exact length of the route since she iteratively increased the scope of the flood, allowing her to check that only the correct number of routing tokens have been changed. An incorrect count would indicate that someone is not following the protocol, and the route should be discarded.

Since A is likely to get multiple replies to a route request, she must arbitrarily select a reply message from which to extract routing information. (More than one message can be used to build redundancy into the routing table.) We must be careful to defend against multiple adversarial replies, although any such countermeasure would ultimately prove futile since a single adversary "close" the A may respond on behalf of any and all adversaries in the network. To reduce the number of adversaries in A's routing table, she should first select a random physical neighbor from whom she has received at least one response, and then select one of those responses at random. The intuition behind this strategy is that the distribution of DHT IDs should be similar independent of which "direction" in the network the request is routed, so the number of responses from each physical neighbor should be comparable. A large response set may indicate an adversarial node. We avoid wormhole attacks [26] since A already knows the cost of each route – the scope of the flood – and thus knows that each route returned for a given scope has the same cost.

After route discovery, A sets up shared keys with each node along the physical route in a process similar to constructing a Tor tunnel [16]. While A does not know the identities or pseudonyms of nodes along the route and cannot authenticate their DH keys, the final DH key (shared with F) is authenticated, since only a node with a given pseudonym prefix could have decoded the half-key. A can use this information to detect man-in-the-middle attacks.

DHT routing. Once the logical routing table has been built, MCON routing is identical to DHT routing, with the exception that messages to every logical hop must traverse a number of physical hops.

When searching, A hashes the search term to determine the destination (X) and the closest logical hop in A's routing table (F). She then retrieves the DH key and index z shared with F, along with the source route (B, C, D) and the associated keys shared with each physical hop. A uses identity-based encryption to encipher the hash of the search term so only a node logically close to X can open the message. (Authenticating the final logical hop prevents arbitrary DHT nodes falsely claiming responsibility for a given key.) She composes a message containing z and the resulting ciphertext, encrypts it to F using the shared DH key, attaches a MAC, and onion-wraps it such that each physical hop must remove a layer of encryption to forward the message. Route tokens

[7] If the shared DH key is g^{xy}, where x and y are private keys, then one DH half-key is g^x and the other is g^y.

[8] If identical requests are received from multiple neighbors, all their identities are stored.

are included in the onion-wrapped portions so that each hop only receives its own token.

DHT (overlay) messages are in the form of $(\mathrm{ID}, E_{\mathrm{P}_1}(R_1, E_{\mathrm{P}_2}(R_2, E_{...}(R_{...}, z, E_{k_1}(M)), \mathrm{MAC}_{k_2}(ID, z, E_{k_2}(M)))))$, P_i is the ith physical hop in the route and ID is the message identifier, followed by repeated layers of onion encryption containing route tokens. The inner-most onion layer is composed using the verified DH key shared with the logical hop and contains a message M for the final DHT hop. M is in the format of $IBE_{\mathrm{L}}(h(\mathrm{search\ term}))$, where L is the DHT (logical) destination.

Onion wrapping. Onion-wrapping messages and randomizing routing tokens prevents A, B, and C from linking messages or learning the MCON topology. However, any one of them may monitor the amount of time between query and response along a given source route, and can deduce the magnitude of the ID prefix match between the source and destination nodes. (Messages sent to an early logical hop along a route will take a long time to return a result, while messages sent from the last logical hop to the query destination would see an almost immediate response.) We can prevent this attack by asking each logical hop to delay query responses for a fixed amount of time. Since the number of logical hops required to complete a query is uniform for a given network size, and since each logical hop knows its logical distance from the destination, the required delay can be estimated within a factor of 2. This increases the total time required to receive a response to every query, but also ensures that the time is constant (so query failure is easy to detect).

If the query originator or a logical hop fails to receive a response (within a timeout period) from the next logical hop (which can be the result of failure at the logical hop or any physical hop along the way), it picks the next best logical hop and repeats the attempt, until it either succeeds (receives a response) or runs out of logical hops to try. In the latter case, it would admit failure, and not return a response. While this is not a complete solution for churn, it does provide a certain level of robustness against offline nodes and packet loss. A more robust scheme is discussed in the next section.

5.2 Robust design

Robustness is somewhat tricky to achieve in the efficient design, since a single offline physical hop along a source route renders the entire source route unusable. In this section, we discuss a design that trades increased robustness for decreased efficiency and larger number of disclosed IP addresses. We employ what we call a "skipping stones" approach:[9] in addition to sending a message to a single hop along a physical route, the message is sent to *each neighbor of that hop*. Each of those nodes sends to each of the neighbors of the next physical hop, and so on. This reduces the probability of failure because only one node per "neighborhood" needs to be honest and online in order for a message to get through. To that end, all MCON nodes must know not only the IP addresses and cryptographic keys of each neighbor, but also addresses and keys of each neighbor's neighbor. The MIA will reveal that information during the bootstrap phase. Furthermore, the entire neighborhood needs a shared key for use with route tokens. This key can either be given out by the MIA or agreed-upon by neighborhood members using a key agreement scheme such as in [28].

Although DHT routing does not change in the robust scheme, we must alter our physical hop routing and discovery to accommodate neighborhood-wide routing decisions. When a neighborhood receives a route discovery reply, a majority of neighbors must come to a consensus regarding the contents of their routing token (and thus the previous and next hops for the reply). They sign the agreed-upon token using a threshold signature scheme [43],[10] which requires m out of n nodes to partially sign a message before a full signature can be derived. For a simple majority, m would be $\lceil \frac{n+1}{2} \rceil$. Each node can then independently encrypt the signed token with the shared neighborhood key, prepend it to the route reply, and forward it. Note that majority agreement is required only during route discovery and during shared key exchange with the originator. Multiple route replies are handled the same way as in the efficient scheme, but now they are far less likely to be malicious since multiple nodes must agree on the route – a malicious majority at one of the intermediate neighborhoods would be required to produce a compromised route.

In order to send a message in the robust scheme, the originator sends messages to each neighbor of the next physical hop along the route. When a message arrives, each neighbor can independently decrypt the enclosed routing token (using the neighborhood key) and verify the signature to ensure it is correctly formed. Since only the final destination can determine message validity by verifying the enclosed MAC, intermediate nodes will not know if a message is legitimate. If an intermediate node gets different messages with identical IDs, it must forward one of each copy, potentially increasing the number of messages proportionally to the number of adversaries encountered en route.

While offering superior robustness under heavy churn, this scheme has higher overhead than our efficient scheme. We face a constant-factor increase in the number of real-world identities every MCON member knows, since every node must now keep track of the IP addresses not only of its physical neighbors, but also of their neighbors. However, since the number of identities each node knows is still constant, this does not compromise membership-concealment. We also lose plausible deniability: any one of a node's neighbors can perform packet counting [42] and timing attacks [16] to determine if a message is being forwarded or originated. However, we can recover plausible deniability by using cover traffic.

5.3 Hybrid design

The hybrid scheme maintains most of the robustness properties of the previous scheme while significantly reducing communication costs. We take a similar approach to Saia and Young [40] and modify our robust scheme such that nodes discover the identities of their neighbors' neighbors only if $h_1(ID_1) \mod m = h_2(ID_2) \mod m$ for some small constant m, where ID_1 and ID_2 are the DHT IDs of the two nodes. If the equality does not hold, nodes simply do not learn about each other. Since introductions are handled by the MIA, this invariant is trivial to enforce. Intuitively, this design probabilistically guarantees that every node of the next neighborhood receives at least one copy of each message. As we increase the modulus m, fewer messages are sent and robustness decreases. However, this reduction is acceptable when we consider that message overhead (combining communication time, bandwidth, and cryptographic overhead) is reduced by a factor of m.

6. THEORETICAL ANALYSIS

Our MCON designs do not share the flaws of existing schemes such as Freenet [10], Tor bridges [14], Turtle [36], or Kaleidescope [44]. The latter two, being based on social network, are susceptible to targeted corruption and celebrity attacks since nodes are not degree-constrained, and therefore some are "tasty targets" for compromise.[11] Freenet opennet is vulnerable to the same attacks,

[9] A stone skipped over water makes contact with the surface repeatedly, creating ripples at each contact point.

[10] Threshold signatures allow some nodes to disagree or be offline during route discovery.

[11] We note a celebrity could split split her contact lists into many nodes with a small number of neighbors each, and remain a logically tasty target while maintaining a low target profile at the

and also to both passive and active harvesting. Bridges are vulnerable to confirmation and passive harvesting attacks. Moreover, our designs provide unlinkability, and our efficient design provides plausible deniability. Unlike Freenet and OneSwarm, our search completes within a guaranteed time bound while making rare files as easy to find as popular files.

6.1 Membership concealment intuition

To verify that our designs do not fall victim to identity disclosure, we check that 1) only physical neighbors communicate directly over IP (preventing *harvesting*), 2) no adversary can query arbitrary Internet hosts or otherwise elicit an IP-level response identifiable as an MCON message (preventing *confirmation*), and 3) no adversary learns the identity of a node who does not directly connect to corrupted or monitored nodes (preventing information *leakage*). In our system, (1) and (2) are handled by the strong binding property – nodes will only respond to messages that are signed by their physical neighbors, and neither initiate nor respond to IP-level contact with any other nodes using the MCON network protocol. (3) presents a greater challenge: a powerful network-monitoring adversary may monitor not only individual nodes but entire networks, and use some encryption-oblivious fingerprinting technique to identify MCON members [46]. The defense is protocol-level obfuscation (steganography) such as used in [22], which, while not explicitly implemented in our current system, is a natural extension.

While we impede graph de-anonymization attacks by perturbing the maximal MCON node degree, making it independent of nodes' social degree, our main defense is to prevent topology exploration by both insider and outsider adversaries. The success of the latter mechanism depends on the quality of traffic obfuscation. Remaining attacks are discussed below.

Recall our (ℓ, γ)-adversary, who can monitor ℓ links and can corrupt γ network members. Since he can only learn k additional members from every member he corrupts or monitors, he is limited to learning at most $k\ell + k\gamma$ correctly-functioning members ($k^2\ell + k^2\gamma$ in the robust scheme). Without protocol obfuscation, we say that our network is $(\frac{N}{2k}, \frac{N}{2k}, f)$-membership-concealing for $f(\gamma, \ell, N) = k\gamma + k\ell$, since $f(\gamma, \ell, N) = \Theta(\gamma + \ell)$, where N is the total number of MCON participants. In the robust scheme, the network is $(\frac{N}{2k^2}, \frac{N}{2k^2}, f)$-membership-concealing for $f(\gamma, \ell, N) = k^2\gamma + k^2\ell$. If we use protocol obfuscation then membership hiding properties will depend on the details of the steganographic system, but with perfect obfuscation our efficient network would be $(N, \frac{N}{k}, f)$-membership-concealing and our robust network would be $(N, \frac{N}{k^2}, f)$-membership-concealing for $f(\gamma, \ell, N) = k\ell$ and $f(\gamma, \ell, N) = k^2\ell$, respectively.

6.2 Churn

Churn, or the constant leaving and re-joining of nodes, causes problems in peer-to-peer networks – nodes in such networks are not expected to be long-lived, and if all of a peer's contacts go offline, the peer will be disconnected from the network and must re-join, discovering new (online) network contacts in the process. Churn is particularly problematic in MCONs because disconnected nodes are not allowed to acquire new MCON contacts and *any* level of churn reduces the efficiency of our routing scheme, by invalidating some optimal routes. Node degrees in the MCONs must be large enough to handle churn, and yet small enough to minimize identity exposure. We use a very strong churn model in our analysis: we do not assume any relationship between the online status of a node from one moment to the next, i.e. any node has the same probability of being offline at any time, independent of its previous online status. While we do not currently consider nodes who permanently leave the MCON, we can add an MIA-mediated revocation system – nodes who have been offline for a long time can have their keys revoked. Neighbors of those nodes can then be allowed to acquire more neighbors, since they will still not know more than k MCON members – the revoked node no longer counts among the member set. Neither do we consider social network churn because we do not use social network edges as trust relationships. Therefore, the loss of a social network edge need not affect the topology of the MCON. As for new edges, we support issuing invitations as long as all other conditions, such as node degree, continue to hold. The MIA can discover such edges as they are created.

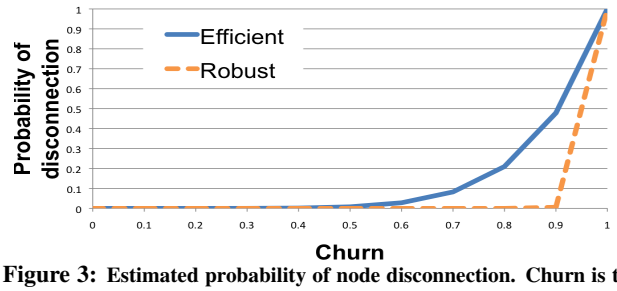


Figure 3: Estimated probability of node disconnection. Churn is the fraction of MCON members who are offline.

Connectedness. A node becomes disconnected when all of his physical neighbors are offline. Assuming nodes come online and go offline independently of each other, the probability of disconnection is c^k, where c is the churn rate and k is the MCON node degree limit. Roughly, this means that at $k = 7$, around 90% of MCON nodes have to be offline for a node to be disconnected half the time in our efficient scheme. The chance of disconnection in the robust scheme is very small when churn is less than 90%. This is shown in Figure 3. Analysis of Freenet data shows a churn rate in the vicinity of 70%[12], if we meaning that we can be almost certain that nodes are always connected in our scheme. However, a problem can occur that causes nodes to permanently lose track of each other: if node A goes offline, and B, A's physical neighbor, goes offline sometime later, and they both change IP addresses before returning, they will have no way to communicate with each other when re-joining. The solution is to have nodes periodically publish their signed IP addresses to a known DHT location, combined with a random value and encrypted with their physical neighbors' keys. This ensures that physical neighbors can always keep track of each other while providing no information to unauthorized parties.

Reachability. Nodes cannot reach a network destination (even if both nodes are technically connected) if there is no DHT route between them. This may happen if all required DHT hops are down themselves, or if they are not reachable through source routes. In our efficient scheme, the probability that all nodes along a given source route are up and forwarding packets is $(1-c)^d$, with offline probability c and route length d, which is at worst the network diameter. The robust scheme is more forgiving since it uses more resilient source routes – only *one neighborhood node* in every source route needs to be forwarding packets. In this scheme, our failure probability becomes $(1-(1-c)^k)^d$. Note that since k is constant, we cannot *guarantee* resilient routing, but this is unlikely to be a problem in practice – while we need $O(\log \log N)$ nodes per group for provable resilience, we can set k to 11 and support a network of

network layer. This attack is unlikely at the social network layer, since a celebrity must maintain her celebrity status to get contacts, and any system that enforces a maximal node degree at the membership concealing layer will not create multiple pseudonyms from a single social network-level identity.

[12]discounting nodes that we see only once throughout the experiment

100 billion nodes. In the next section we present simulation results measuring reachability when re-routing is taken into account.

Denial of service attacks. An unfortunate sideeffect of plausible deniability in the efficient scheme is the inability to prevent nodes from flooding the network, since it is impossible to determine if a node legitimately initiated such a flood or is forwarding the message for another node. This leads to the problem of denial of service (DoS) attacks through network floods. We can counter this using data-oblivious throttling, where neighbors of a node sending packets faster than a certain threshold will refuse to forward some of those packets, independent of their ultimate origin or destination. This prevents undue usage of network bandwidth but degrades the maximum possible performance of the network.

Even without plausible deniability, the robust (and hybrid) scheme falls victim to DoS due to the amplification factor of messages – for every message sent by a node, multiple message must be sent by recipients. While nodes can refuse to forward duplicate messages, adversarial intermediaries modifying messages will cause both the original and the modified messages to be propagated. With enough adversaries, the final destination could be overwhelmed with messages, all of which require decryption and verification.

7. SIMULATION RESULTS

We simulated MCON construction and routing using the Orkut dataset from Mislove *et al.* [32]. The data contains 3,072,606 nodes,[13] with an average node degree of 74. To test the robust routing scheme we generated a smaller synthetic social network using a modified version of the algorithm in [29]. Our network contained 1,324,134 nodes and had a degree distribution, diameter, and clustering coefficient comparable to the Orkut dataset.

7.1 MCON construction

From the social network dataset, we constructed an MCON with a node degree limit of 7.[14] The bootstrap protocol randomly selects an initial seed clique of four nodes ($\lceil \frac{7}{2} \rceil$) from the social network and iteratively adds nodes to the seed based on social network relationships. The final MCON contained just over 85% of the nodes in the social network. Slightly more than 35% of MCON nodes had under-full routing tables, resulting in average node degree of 5.997, with an average pairwise physical distance of 10.

7.2 Routing and search

Our DHT uses the Kad routing protocol (a variation of Kademlia [30]), using routing table of 16 buckets of 8 entries each. The average number of DHT hops between any two MCON nodes is 2.5, which translates to an average of 13 physical hops in the efficient case, and 26 hops in the robust case. (The probability distribution of physical hops per query with no churn is shown in Figure 4.) Note that due to the greedy nature of routing table construction, which preferentially incorporates the nearest node with a given prefix match, the average number of physical hops per logical hop is *lower* than the average number of physical hops between any two random nodes in the MCON. Assuming average round trip times of 180ms (computed from the "King" dataset [24]), a search should complete in fewer than 2.5 seconds without time padding.

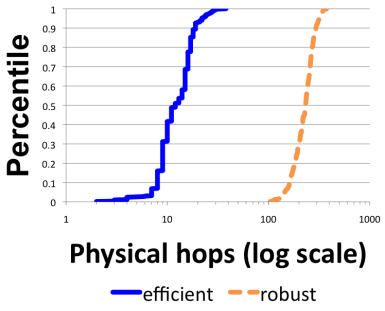


Figure 4: Cumulative distribution of physical hops per DHT query

[13]Less than 12% of Orkut's network at the time of collection

[14]We also simulated $k = 5$ and $k = 9$. The results conform to expectations – smaller k reduces connectivity and efficiency.

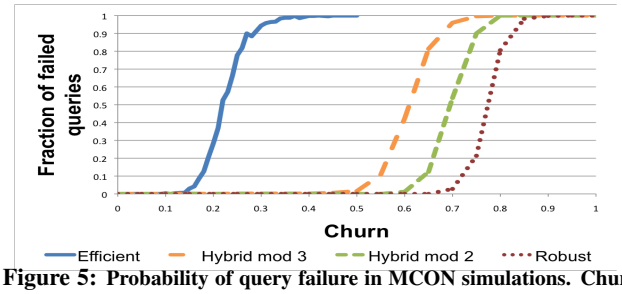


Figure 5: Probability of query failure in MCON simulations. Churn is the fraction of MCON members who are offline.

When a route fails, we select the next best route and continue trying until we reach 25 failed routes per node or the query succeeds. The rates of DHT query failure with churn for all three schemes are shown in Figure 5. Data was collected using 500 independent trials, routing between two randomly-selected nodes. In the efficient scheme, the worst-case number of logical hops is 18 and the worst case for physical hops is 178, which translates to a query time of just under 33 seconds. In the worst case for the robust scheme, average performance is 127 logical hops and 395 physical hops, which would require 71 seconds on average. (Note that many of these physical hops are contacted in parallel, making the time estimate strictly pessimistic.) The efficient scheme reached its performance limit at 21% churn, and the robust scheme at 75% churn. Hybrid scheme performance depends on the modulus.

Finally, the robust scheme may provide worse-than-expected resilience in certain topologies, such as when adversarial nodes form clusters in the MCON due to dense social network relationships among them. Clusters reduce the adversaries' knowledge of honest MCON nodes (since most of her neighbors are malicious), but impede routing – adversarial clusters have a high chance of forming neighborhoods with malicious majorities. However, assuming route discovery proceeds correctly, we only require *one honest node* per neighborhood for message forwarding to succeed.

8. CONCLUSION

In this paper we initiate a systematic study of membership concealment as a security goal. While the idea has been implicitly described in other work, it was not rigorously defined, and therefore only implemented in an ad-hoc fashion, usually resulting in vulnerabilities. We presented attacks against two well-known censorship resistance tools (Freenet and Tor bridges), and described three designs for membership-concealing overlay networks (MCONs). One design is efficient, one is more robust to churn, and one is a hybrid, balancing robustness and efficiency. In simulation, churn significantly degrades the performance of all schemes, but the robust scheme performs well under churn up to 75%. From a combination of theoretical analysis and simulation, we conclude that both schemes are practical, offering bounded-time search that locates both popular and rare files equally well. *In the worst case*, our search time is less than 90 seconds in the robust scheme, and less than 35 seconds in the efficient scheme.

Some open problems remain with our designs. First, our "infection" approach to constructing the MCON, while mitigating bootstrap attacks, is nonetheless cumbersome. A better approach would be to somehow allow people to ask for membership in the MCON while preserving security. Second, the current route discovery mechanism requires a flood of the network at node join time. This imposes significant message overhead, so we need a mechanism that is more efficient and still maintains membership concealment properties and sender-receiver unlinkability. Finally, although our Membership and Invitation Authority can remain offline and hidden, it still represents a central point of failure[15] – if it were com-

[15]with the exception of availability attacks

promised then the complete membership of the network would be discovered. In principle, the functions of the MIA can be carried out using secure multi-party computation. We leave finding efficient algorithms for this computation for future work.

ACKNOWLEDGEMENTS

The authors would like to thank Jon McLachlan and Eric Chan-Tin for invaluable feedback on early version of the design, Hal Peterson for feedback on the latter versions of the paper, and our anonymous reviewers for their very helpful suggestions. This work was supported by NSF grants CNS-0546162 and CNS-0709048 and CNS-0917154.

9. REFERENCES

[1] Opennet initiative. http://opennet.net/.

[2] China 'blocks' itunes music store. BBC News, August 2008.

[3] Pakistan blocks youtube website. BBC News, February 2008.

[4] BAUER, M. New covert channels in HTTP: Adding unwitting web browsers to anonymity sets. In *Proceedings of the Workshop on Privacy in the Electronic Society (WPES)* (2003).

[5] BIDDLE, P., ENGLAND, P., PEINADO, M., AND WILLMAN, B. The darknet and the future of content distribution. Tech. rep., Microsoft Corporation, 2002.

[6] BONEH, D., AND FRANKLIN, M. Identity-based encryption from the Weil pairing. *SIAM Journal on Computing 32*, 3 (2003), 586–615.

[7] BUSKIRK, E. V. LimeWire adds private file sharing. http://blog.wired.com/business/2008/12/lime-wire-adds.html, 2008.

[8] CAESAR, M., CASTRO, M., NIGHTINGALE, E., O'SHEA, G., AND ROWSTRON, A. Virtual ring routing: network routing inspired by DHTs. In *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)* (New York, NY, USA, 2006), ACM Press, pp. 351–362.

[9] CHAUM, D. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM 24*, 2 (February 1981).

[10] CLARKE, I., SANDBERG, O., WILEY, B., AND HONG, T. W. Freenet: A distributed anonymous information storage and retrieval system. In *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability* (July 2000), pp. 46–66.

[11] DANEZIS, G., LESNIEWSKI-LAAS, C., KAASHOEK, M. F., AND ANDERSON, R. Sybil-resistant DHT routing. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS)* (2005).

[12] DANEZIS, G., AND WITTNEBEN, B. The economics of mass surveillance and the questionable value of anonymous communications. In *Proceedings of the Workshop on The Economics of Information Security (WEIS)* (June 2006).

[13] DEIBERT, R. J., PALFREY, J. G., ROHOZINSKI, R., AND ZITTRAIN, J. *Access Denied: The Practice and Policy of Global Internet Filtering (Information Revolution and Global Politics)*. MIT Press, 2006.

[14] DINGLEDINE, R., AND MATHEWSON, N. Design of a blocking-resistant anonymity system. https://www.torproject.org/svn/trunk/doc/design-paper/blocking.html, 2007.

[15] DINGLEDINE, R., AND MATHEWSON, N. Tor protocol specification. http://www.torproject.org/svn/trunk/doc/spec/tor-spec.txt, 2008.

[16] DINGLEDINE, R., MATHEWSON, N., AND SYVERSON, P. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium* (August 2004).

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

[18] FEAMSTER, N., BALAZINSKA, M., HARFST, G., BALAKRISHNAN, H., AND KARGER, D. Infranet: Circumventing web censorship and surveillance. In *Proceedings of the 11th USENIX Security Symposium* (August 2002).

[19] FEAMSTER, N., BALAZINSKA, M., WANG, W., BALAKRISHNAN, H., AND KARGER, D. Thwarting web censorship with untrusted messenger delivery. In *Proceedings of the Workshop on Privacy Enhancing Technologies (PET)* (2003).

[20] FRANKEL, J. Archived at: http://slackerbitch.free.fr/waste/, 2003.

[21] FREEDMAN, M. J., AND MORRIS, R. Tarzan: A peer-to-peer anonymizing network layer. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002)* (Washington, DC, November 2002).

[22] FU, X., GRAHAM, B., BETTATI, R., AND ZHAO, W. On countermeasures to traffic analysis attacks. In *Proceedings of the Information Assurance Workshop* (2003), pp. 188–195.

[23] G. DANEZIS, G., AND MITTAL, P. SybilInfer: Detecting Sybil nodes using social networks. In *Proceedings of the Annual Network and Distributed System Security Symposium (NDSS)* (2009).

[24] GIL, T. M., KAASHOEK, F., LI, J., MORRIS, R., AND STRIBLING, J. The "King" data set. http://pdos.csail.mit.edu/p2psim/kingdata/, 2005.

[25] HEYDT-BENJAMIN, T. S., SERJANTOV, A., AND DEFEND, B. Nonesuch: a mix network with sender unobservability. In *Proceedings of the Workshop on Privacy in the Electronic Society (WPES)* (2006).

[26] HU, Y., PERRIG, A., AND JOHNSON, D. Packet leashes: a defense against wormhole attacks in wireless networks. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)* (2003), vol. 3, pp. 1976–1986.

[27] ISDAL, T., PIATEK, M., KRISHNAMURTHY, A., AND ANDERSON, T. Friend-to-friend data sharing with OneSwarm. Tech. rep., University of Washington, 2009. http://oneswarm.cs.washington.edu/f2f_tr.pdf.

[28] KIM, Y., PERRIG, A., AND TSUDIK, G. Communication-efficient group key agreement. In *Proceedings of the IFIP TC11 Annual Working Conference on Information Security: Trusted Information: The New Decade Challenge* (2001), Kluwer Academic Pub, pp. 229–244.

[29] KLEINBERG, J. The small-world phenomenon: An algorithmic perspective. In *Proceedings of the ACM Symposium on Theory of Computing* (2000).

[30] MAYMOUNKOV, P., AND MAZIERES, D. Kademlia: A peer-to-peer information system based on the XOR metric. In *Proceedings of the International Workshop on Peer-to-Peer Systems (IPTPS)* (2002), vol. 258, Springer, p. 263.

[31] MAZIÈRES, D., AND KAASHOEK, M. F. The design, implementation and operation of an email pseudonym server. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)* (November 1998), ACM Press.

[32] MISLOVE, A., MARCON, M., GUMMADI, K., DRUSCHEL, P., AND BHATTACHARJEE, B. Measurement and analysis of online social networks. In *Proceedings of the ACM SIGCOMM conference on Internet Measurement* (New York, NY, USA, 2007), ACM, pp. 29–42.

[33] NAMBIAR, A., AND WRIGHT, M. Salsa: A structured approach to large-scale anonymity. In *Proceedings of the ACM conference on Computer and Communications Security (CCS)* (October 2006).

[34] NARAYANAN, A., AND SHMATIKOV, V. De-anonymizing social networks. In *IEEE Symposium on Security and Privacy* (May 2009).

[35] PFITZMANN, A., AND HANSEN, M. Anonymity, unobservability, and pseudonymity: A consolidated proposal for terminology. Draft (v.0.31), July 2000.

[36] POPESCU, B. C. Safe and private data sharing with turtle: Friends team-up and beat the system. In *Proceedings of the Cambridge International Workshop on Security Protocols* (2004).

[37] RAO, J. R., AND ROHATGI, P. Can pseudonymity really guarantee privacy? In *Proceedings of the 9th USENIX Security Symposium* (August 2000), USENIX, pp. 85–96.

[38] REITER, M., AND RUBIN, A. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security 1*, 1 (June 1998), 66–92.

[39] ROWSTRON, A., AND DRUSCHEL, P. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Proceedings of IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)* (2001), Springer, pp. 329–350.

[40] SAIA, J., AND YOUNG, M. Reducing communication costs in robust peer-to-peer networks. *Information Processing Letters 106*, 4 (2008), 152–158.

[41] SANDBERG, O. Distributed routing in small-world networks. In *Proceedings of the Workshop on Algorithm Engineering and Experiments (ALENEX)* (2006).

[42] SERJANTOV, A., AND SEWELL, P. Passive attack analysis for connection-based anonymity systems. In *Proceedings of ESORICS 2003* (October 2003).

[43] SHOUP, V., AND GENNARO, R. Securing threshold cryptosystems against chosen ciphertext attack. *Journal of Cryptology 15*, 2 (January 2002), 75–96.

[44] SOVRAN, Y., LIBONATI, A., AND LI, J. Pass it on: Social networks stymie censors. In *International Workshop on Peer-to-Peer Systems (IPTPS)* (February 2008).

[45] STOICA, I., MORRIS, R., LIBEN-NOWELL, D., KARGER, D., KAASHOEK, M. F., DABEK, F., AND BALAKRISHNAN, H. Chord: A scalable peer-to-peer lookup service for Internet applications. In *Proceedings of the Conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)* (New York, NY, USA, 2001), ACM Press, pp. 149–160.

[46] SUN, Q., SIMON, D. R., WANG, Y.-M., RUSSELL, W., PADMANABHAN, V. N., AND QIU, L. Statistical identification of encrypted web browsing traffic. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy* (Berkeley, California, May 2002).

[47] WALDMAN, M., AND MAZIÈRES, D. Tangler: a censorship-resistant publishing system based on document entanglements. In *Proceedings of the 8th ACM Conference on Computer and Communications Security (CCS 2001)* (November 2001), pp. 126–135.

[48] WRIGHT, M., ADLER, M., LEVINE, B. N., AND SHIELDS, C. An analysis of the degradation of anonymous protocols. In *Proceedings of the Network and Distributed Security Symposium (NDSS)* (February 2002), IEEE.

[49] YU, H., GIBBONS, P. B., KAMINSKY, M., AND XIAO, F. SybilLimit: A near-optimal social network defense against Sybil attacks. In *Proceedings of the IEEE Symposium on Security and Privacy* (2008), pp. 3–17.

[50] ZITTRAIN, J., AND EDELMAN, B. Internet filtering in China. *IEEE Internet Computing 7*, 2 (2003), 70–77.