Bridging and Fingerprinting: Epistemic Attacks on Route Selection

George Danezis[1] and Paul Syverson[2]

[1] Microsoft Research, Cambridge, UK.
[2] Naval Research Laboratory, USA.

Abstract. Users building routes through an anonymization network must discover the nodes comprising the network. Yet, it is potentially costly, or even infeasible, for everyone to know the entire network. We introduce a novel attack, the *route bridging attack*, which makes use of what route creators do *not* know of the network. We also present new discussion and results concerning route fingerprinting attacks, which make use of what route creators do know of the network. We prove analytic bounds for both route fingerprinting and route bridging and describe the impact of these attacks on published anonymity-network designs. We also discuss implications for network scaling and client-server vs. peer-to-peer systems.

1 Introduction

Anonymous communications were first introduced for electronic mail with the mix network [3] and then extended to internet streams by onion routing [10, 21, 2, 6]. Since then, attempts have been made to totally decentralize the provision of anonymity services. First Tarzan [9, 8], then other systems [15, 26, 12] have applied the peer-to-peer paradigm to ensure that all protocol participants are both clients and routers that anonymize streams.

Besides the differences in the type of traffic carried or division of tasks within the network, all those systems share a common architecture. Initiators of communications relay their messages or streams through third parties to evade identification. The communication contents are encrypted to foil trivial passive linkage, and in some cases countermeasures are applied against traffic analysis, such as making messages uniform size or delaying them or injecting cover traffic.

Despite their common architecture, mix-based, onion-routing, or peer-to-peer anonymizing networks protect against radically different threat models. Mix networks should be secure when under full surveillance, and when a large fraction of routers used are corrupt [1]. Traditional onion routing and stream-based peer-to-peer anonymizers, like Tarzan, are unable to resist even a passive attack and cannot guarantee anonymity if both the initiator and responder of the communication are under surveillance [24]. Similarly, even an entirely passive adversary controlling the first and last node in the path can trace the anonymized stream, unless large amounts of cover traffic are used [23].

A key contribution of this paper is to present a novel class of traffic analysis attacks against relay-based anonymizers, called *bridging* attacks. Bridging uses some a priori information about the route selection of initiators to effectively bridge over honest stages of mixing, making tracing a full path easier for the adversary. As a technique bridging is closely related to *route fingerprinting* [4]. We present an analytic bound on route fingerprinting, using the Tarzan design as an example, and discuss the impact of route fingerprinting against other classes of anonymity systems. In particular, we compare route-fingerprinting resistance for client-server designs vs. peer-to-peer designs and discuss the (encouraging) implications for network partitioning.

2 Fingerprinting

2.1 Young Tarzan leaves telltale fingerprints on the vine.

The early Tarzan design [9] aims to provide strong anonymity against a global eavesdropper using a fully peer-to-peer architecture. The core design is based on ideas from onion routing, with some modifications to distribute services that are otherwise centralized in standard onion routing. The most important distributed service in Tarzan is the directory server providing a list of nodes with their associated keys. Furthermore, Tarzan designers recognized that large scale networks make low-latency traffic more susceptible to tracing, and to alleviate the problem attempt to route multiple streams together by forcing them through restricted routes, called *mimics*—a facet of Tarzan we do not discuss here.

Distributing the directory server functionality over a peer-to-peer network is not straightforward and has deep repercussions on security. Tarzan relies on the use of a Distributed Hash Table (DHT) [20] to store mappings of nodes and keys. A DHT is a peer-to-peer protocol that allows nodes to construct a distributed database mapping keys to values. Nodes are assigned a particular section of the key space for which they store values, and there are efficient $\mathcal{O}(\log N)$ algorithms for finding the node corresponding to a particular key. Tarzan nodes store their directory descriptors as values, and the key of the descriptor is simply its hash.

A Tarzan node joining the network has, as in traditional onion routing, to 'discover' a set of peers along with their directory descriptors containing their cryptographic keys, to be able to construct paths and anonymize streams of traffic. The original Tarzan design required nodes to discover at random only a small subset of other nodes, and used a small subset of those to build anonymous routes. Sampling was performed by selecting a random nonce and finding the closest DHT key and associated directory entry, an operation that is efficient in DHTs.

This approach introduces two problems. First, the sampling procedure is not guaranteed to be uniform in the presence of adversaries prepared to subvert the Distributed Hash Table. For any random key K the client choses, the adversary can simulate a node with a directory descriptor mapping to a close-by key K' that is closer than the closest genuine key K_g. Hence the adversary can easily populate

the client's entries with corrupt nodes, making any routing over them ineffective. This attack is active, and requires the adversary to corrupt the underlying DHT protocols (which is easy, since few DHT designs protect against such attacks effectively). Attacking and defending DHTs is not the focus of this paper and we will not concern ourselves any further with this line of attack.

Second, the client only chooses routes from a small subspace of all nodes, and this subspace is known to the adversary. This in turn can help the adversary identify which routes belong to each client. This family of attacks was briefly introduced in route fingerprinting [4], and in this paper we present a novel attack in this family we call route bridging.

To avoid fingerprinting attacks the final Tarzan design [8] requires each node to know all other nodes—something hardly practical due to the large size and churn of peer-to-peer networks. Our analysis and discussion of these attacks concludes that for weaker threat models this approach may be over-conservative.

2.2 Bounding route fingerprinting.

We assume that there are $N+1$ peers in the system, and each of them samples $n < N$ others to create routes. (We assume nodes do not create routes through themselves.) Assume an adversary determines $k < n$ nodes on a particular route. How many peers on average will know all k nodes, and therefore are possible initiators of this route?

Each node can build up to $\binom{n}{k}$ k-tuples out of a maximum of $\binom{N}{k}$ that could exist in the system. Therefore any peer knows those k nodes with probability $p = \binom{n}{k}/\binom{N}{k}$.[1] We define an indicator random variable I_i for each node i that takes the value one when this is the case, and zero otherwise. The expected number of nodes that could be initiators is $A_k = E[\sum_{i=0}^{N} I_i]$ which is at most:

$$A_k = E[\sum_{i=0}^{N} I_i] \leq (N+1)\frac{n^k}{N^k}\left(\frac{N}{N-(k-1)}\right)^k \approx \frac{n^k}{N^{k-1}}, \text{ when } \frac{k-1}{N} \to 0 \quad (1)$$

Proof. We start by the definition and apply linearity of expectations.

$$A_k = E[\sum_{i=0}^{N} I_i] = \sum_{i=0}^{N} E[I_i] \quad (2)$$

$$= (N+1)\cdot p = (N+1)\cdot \frac{\binom{n}{k}}{\binom{N}{k}} = \frac{(N+1)n!(N-k)!}{N!(n-k)!} \quad (3)$$

$$\leq \frac{n^k(N+1)}{(N-k+1)^k} \text{ (keep max. and min. values.)} \quad (4)$$

$$= (N+1)\frac{n^k}{N^k}\left(\frac{N}{N-(k-1)}\right)^k \quad (5)$$

Take the limit $\lim_{\frac{k-1}{N}\to 0}\left(\frac{N}{N-(k-1)}\right)^k = 1$ to conclude the proof.

[1] An equivalent combinatorial formulation is $p = \binom{N-k}{n-k}/\binom{N}{n}$.

2.3 Anonymity loves company, but hates a big crowd.

What does this attack mean in practice? As expected, if the adversary cannot observe any nodes on a path ($k = 0$), anonymity is perfect and $A_0 = N + 1$. This assumption imposes an unacceptably weak threat model.

The first realistic threat model is for the attacker to be the receiver of the communication, and thus to observe just one node in a path, the final one. We expect that given this information ($k = 1$) there are on average $A_1 \approx n$ nodes in the network that could have been the initiators. This is of some interest since it is equal to the number of candidates if the network were split into $\frac{N}{n}$ smaller networks of equal size n, in which all nodes knew all other nodes. In case the adversary controls, and does not merely see a connection originating from the last node, they can associate $A_2 \approx n^2/N$ initiators (the final node and the penultimate node on the path) with each incoming link (since $k = 2$).

Next assuming that the adversary controls the two last nodes on a path (but not the first few). What is the expected number of nodes that could have been the initiator? The two corrupt nodes know that the initiator must have sampled them, as well as the previous node, and therefore $k = 3$. The number of possible initiators is $A_3 \approx \left(\frac{n}{N}\right)^2 n$, and in general for $k > 1$ we have that $A_k < n$, which means that the security of the system will always be worse than if the networks were simply partitioned into smaller cliques.

This attack, and its associated analysis, prove two key intuitions. First it illustrates again, that for some threats the larger the network the less security we get. As N grows the fraction $\frac{n}{N}$ becomes smaller, and in turn the number of candidate nodes that could have created any particular route becomes smaller. Similarly to the predecessor attack against crowds[2] we see that increasing the number of potential senders does not automatically increase anonymity if the system does not ensure that the anonymity sets are constituted using all of them—the route fingerprinting attack illustrates that anonymity may in fact decrease.

Second, one may take a step back and ask "does this attack really matter for onion-routing-based systems?" Onion routing only preserves anonymity against a partial adversary, as long as the first and last node are not compromised [22]. This means that with probability c^2 the system provides no anonymity at all, where c is the fraction of compromised nodes in the network. On the other hand a route fingerprinting attack requires $k \geq 2$ to be truly effective, i.e., to reduce anonymity below the effect of simply splitting the network. The most obvious way for the adversary to achieve this is to compromise at least the final node. If

[2] The predecessor attack was first described and analyzed in the original crowds paper [14], and that design provably prevented predecessor attacks on persistent crowds. However, it was later shown that a predecessor attack was possible when crowds reformed, i.e., every time someone joined a crowd. The same work that uncovered this attack also first observed that anonymity vs. this attack decreases as the crowd size increases [19]. Further analysis of predecessor attacks on crowds and other systems was done by Wright et al. [25].

the final node is corrupt, there is still some anonymity left if $n^2 >> N$, even if the sets of known nodes for each participant are available to the adversary.

To make attacks more effective, more nodes on the path need to be compromised. For short paths ($l = 3$), this attack is no more likely than attack through the normal running of the system. For longer paths, fingerprinting can be used in conjunction with timing analysis, to break the security of paths that start with an honest node. As an example consider an adversary that controls the second and last node on a long path. They are able, using timing analysis, to infer that the two corrupt nodes belong to the same path, and apply fingerprinting to reduce the number of candidate initiators to A_5 (they can identify five nodes known by the initiator: the two dishonest ones, and the three honest nodes surrounding them.) Even for paths of length 3, fingerprinting combined with ordinary correlation attack is slightly more effective than correlation alone.

The probability of two or more corrupt nodes being on the path, including a last corrupt node is $c(1-(1-c)^{l-1})$. This is always higher than the probability of compromise (c^2) through controlling the first and last node. In such cases the initiator set can be narrowed down to A_3 or fewer nodes, depending on the positions of the corrupt nodes on the path. This demonstrates that fingerprinting does lead to weaker security for onion-routing networks.

2.4 Better to have nothing to do with each other than to stay together in ignorance.

Tor [6] is the current widely-deployed-and-used onion-routing network. Concern about knowledge-based partitioning has deferred any deployment within Tor of a system that gives clients only a partial list of nodes in the network despite the usability, network load, and other issues that have come with maintaining and distributing the increasingly large list to every Tor client. As we have seen, to avoid such knowledge-based attacks the design of Tarzan actually moved in the other direction, towards requiring clients to know the full list.

Our results apply to peer-to-peer versions of onion routing such as Tarzan. In the client-server setting of Tor, the number of clients C is a few orders of magnitude larger than the number of servers N. In that case the number of candidates given k servers on the path is $A_k \approx (n/N)^k \cdot C$. This further increases anonymity when only the last server is compromised (making $k = 2$), hence architectures that allow such systems to scale should not be discarded solely because of the route fingerprinting attack.

To be concrete, at the time of writing, Tor has an estimated 200000-500000 clients and around 2000 routers (server nodes). Suppose we would like to maintain as a security parameter with respect to exit-node route fingerprinting an anonymity set size of 50000. Then, using a conservative number of clients, each one should know about half of the routers. However, note that one could partition both the client set and the network in four such that all clients in a partition know all 500 nodes in one clique and still produce the same resistance to route-fingerprinting by the exit node. This analysis is too simple and overlooks the fact that nodes are not all the same in Tor: they carry widely differing numbers

of circuits (paths) and amounts of traffic; some serve as persistent entry nodes for clients; only about a third are exit nodes, etc. Our analysis illustrates that while scaling such systems can maintain adequate anonymity in the face of route fingerprinting, splitting the network outright may be more desirable.

However, there remain too many concerns for this to be a recommendation in practice: one must securely split the network and clients so that no single authority can take advantage of the splits, and the basic c^2 probability of end-to-end compromise is still affected by network size, etc. To underscore this last limitation let us revisit the analysis of the current Tor network with an anonymity-set security parameter of 50000 clients. Note that the same result as above applies if the client set is partitioned into four even sets of 50000 and, instead of being partitioned evenly, the node set is partitioned into three sets of 10 nodes each and one set of of 1970 nodes. There is no epistemic attack because each client in each set of 50000 knows all the nodes in its assigned partition, but it is much easier for an adversary to monitor all the network connections of ten nodes than the five hundred that would result from an even partition.

Relatedly, onion routing would appear to benefit from a move to a more peer-to-peer design for all of the reasons that make such designs desirable. However, the above shows that a client-server design has some inherent anonymity advantages over a peer-to-peer design, and the assumption that a peer-to-peer architecture would facilitate further scaling up and therefore improve anonymity cannot be justified in general. Specific proposals for P2P designs and deployment strategies thus need to be examined closely to determine if there are indeed anonymity benefits, or at least acceptable anonymity costs.

When fingerprinting is deployed on mix systems instead of onion routing, which are secure with probability $1 - c^l$, it often allows an adversary to deanonymize users much faster than before, and this should be considered a threat. So, in practice, one is not advised to use a Tarzan-like selection strategy for high security mix-based anonymous communications. In case high levels of security are sought, a second attack that leverages the limited knowledge of nodes, *route bridging*, becomes of interest.

3 Route bridging.

> "*We also know there are 'known unknowns'; that is to say we know there are some things we do not know.*"
>
> *Donald Rumsfeld* — U.S. Secretary of Defense

Route bridging assumes that a passive adversary can put some nodes in a mix network under surveillance. It is also relevant to strengthened onion routing schemes that provide protection against correlation attacks, since it provides an alternative method to link incoming and outgoing streams of traffic.

The key intuition behind bridging attacks is that the nodes constructing the routes only know a fraction of all potential routers, as it was the case for the early

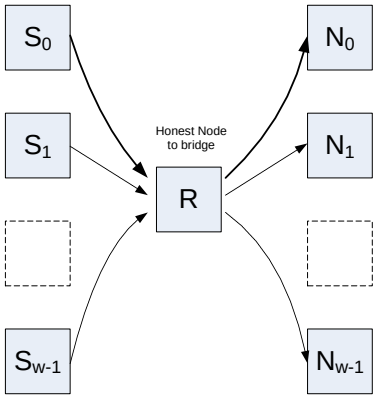

Fig. 1. The setting of the bridging attack when R is the first node.

versions of Tarzan. As a result not all combinations of incoming and outgoing links to/from a router form valid paths—some of those are simply not possible, i.e. no node knows all routers necessary to construct them. In the extreme case some paths do not benefit from any anonymization at all, since for one input link there is a unique possible output link. The key question regarding the route bridging attack is to determine the probability of such a total compromise. It is essentially an epistemic version of the $n-1$ attack [17].

3.1 Bridging a first node

We illustrate the attack first in the simplest setting, where an adversary tries to bridge the first, presumably honest, node. In this case we consider w initiators $S_0, \ldots, S_{w-1}$ that concurrently use the honest node R as the very first node in their paths—and the adversary tries to infer the outgoing node $N_0, \ldots, N_{w-1}$ to which each incoming stream corresponds. In the subsequent sections we generalize our results to other settings.

Consider w incoming messages or streams, from $S_0, \ldots, S_{w-1}$ leading to w outgoing messages to $N_0, \ldots, N_{w-1}$, passing through a mix R. (For convenience, we will use 'message' generically below, but observations we make generally carry over to streams as well.) Without loss of generality we assume that the first sender S_0 routes through the mix a message that is destined to node N_0. What is the probability this message is compromised by a route-bridging attack? The link from S_0 to N_0 can be uniquely recovered, if one of two conditions is true (and these are not exhaustive). Either the node S_0 does not know any of the other destination nodes $N_1, \ldots, N_{w-1}$, which we denote as $\Pr[S_0 \not\rightarrow N_1, \ldots, N_{w-1}]$; or none of the other senders $S_1, \ldots, S_{w-1}$ know the destination node N_0, which we denote as $\Pr[S_1, \ldots, S_{w-1} \not\rightarrow N_0]$. We bound the probability of a successful

attack, P_{bridge}, by:

$$\left(1 - \frac{n-2}{N-2}\right)^{w-1} \leq P_{bridge} \leq 2\left(1 - \frac{n-2}{N-2}\right)^{w-1} \tag{6}$$

Proof. First we calculate the two probabilities $\Pr[S_0 \not\rightarrow N_1, \ldots, N_{w-1}]$ and $\Pr[S_1, \ldots, S_{w-1} \not\rightarrow N_0]$. $\Pr[S_0 \not\rightarrow N_1, \ldots, N_{w-1}]$ is the probability each distinct $N_1, \ldots, N_{w-1}$ is not in the set of $n-2$ nodes that S_0 knows and would route through in this way (assuming that the nodes R, N_0, and S_0 itself are excluded). Probability $\Pr[S_1, \ldots, S_{w-1} \not\rightarrow N_0]$ represents how likely it is that no other node from $S_1, \ldots, S_{w-1}$ has N_0 in its set of $n-2$ remaining nodes, after excluding the router node R, as well as their actual outgoing link and the node S_i itself.

$$\Pr[S_0 \not\rightarrow N_1, \ldots, N_{w-1}] = \frac{\binom{(N-2)-(w-1)}{n-2}}{\binom{N-2}{n-2}} = \prod_{i=0}^{w-2}\left(1 - \frac{n-2}{N-i-2}\right) \tag{7}$$

$$\Pr[S_1, \ldots, S_{w-1} \not\rightarrow N_0] = \left(1 - \frac{n-2}{N-2}\right)^{w-1} \tag{8}$$

First we note that if $i > 0$ then $\left(1 - \frac{n-2}{N-i-2}\right) \leq \left(1 - \frac{n-2}{N-2}\right)$ which in turn means that:

$$\Pr[S_0 \not\rightarrow N_1, \ldots, N_{w-1}] \leq \Pr[S_1, \ldots, S_{w-1} \not\rightarrow N_0] \tag{9}$$

The sought probability P_{bridge} is in fact equal to the union of the events described by the probabilities above. Trivially applying the union bound to $P_{bridge} = \Pr[S_0 \not\rightarrow N_1, \ldots, N_{w-1} \cup S_1, \ldots, S_{w-1} \not\rightarrow N_0]$, as well as the fact that one of the probabilities is always larger than the other, we have that:

$$\Pr[S_1, \ldots, S_{w-1} \not\rightarrow N_0] < P_{bridge} < 2\Pr[S_1, \ldots, S_{w-1} \not\rightarrow N_0] \tag{10}$$

The proof can be concluded by substituting for $\Pr[S_1, \ldots, S_{w-1} \not\rightarrow N_0]$.

This attack assumes that the adversary's only information, besides which nodes are known to which system participants, is the router concerned and the nodes providing input and receiving output for a given mix batch. This makes the attack applicable to bridging the first node in the path. The adversary need only know the knowledge set of the target S_0 for the lower bound we have stated to hold; she need not be aware of which nodes are known to the other S_i. Alternatively, she may only be aware of the knowledge sets of the other S_i and not that of S_0. Note that the nodes $N_j \neq N_0$ need not be distinct for these results to hold. In fact they could all be the same node.

Looking at this simple scenario it is clear that as the number of streams or messages crossing a router increases, the probability that any of them is compromised through this route bridging attack decreases. But what order of magnitude should the batch size w be to neutralize the attack? We note that

the probability of security is $1 - P_{bridge} < \frac{(w-1)(n-2)}{N-2}$ (by Bernoulli's inequality), so if the system has to have a chance of providing full security that is close to optimal we should require $1 - \epsilon < \frac{(w-1)(n-2)}{N-2}$, which provides a lower limit on w:

$$w > \frac{(1-\epsilon)(N-2)}{n-2} + 1 \tag{11}$$

So to even start contemplating the possibility of full security the number of mixed messages or streams should be $\mathcal{O}(\frac{N-2}{n-2})$. In a fully peer-to-peer system the number of streams multiplexed is only $\mathcal{O}(l)$, where l is the length of paths in the system. This is usually a small number, way too small to guarantee maximal security.

In low-latency systems like Tarzan or Tor, the threat of route bridging is likely to be dominated by the ability to correlate streams in two locations through simple timing and packet counting for the foreseeable future [13, 18]. In proper mix systems, however, it could prove to be a near-term practical threat. The batch size w provides some guidance on how to set the parameters of each mix to mitigate against the route bridging attack.

3.2 Building bridges further down the road

"*Confusion will be my epitaph, as I crawl a cracked and broken path.*"

King Crimson — Lyrics to "Epitaph"

Bridging could also be applied to the final router in a path. One would, however, need to assume that the adversary knows which ultimate destinations are known to whom. For the anonymity systems we have been considering, these destinations are not assumed to be part of the network; so this information would not be available by the means we described above. Feigenbaum et al. [7] present such an analysis of what a partial network adversary who knows the a priori distribution of ultimate destinations for every client of an onion-routing network can learn by observing the (fully-discovered) network.

What if messages entering router R were from initiators known to the adversary? Note that here the chooser of routes are not the intermediary nodes S_i. Thus it's not the nodes N_{i_j} unknown to S_i that we are considering; it's the nodes unknown to the initiating peer that routed from S_i to R to N_{i_j}. If all paths have been compromised for at least k nodes prior to R, then the bound becomes even tighter in this combination of fingerprinting and bridging.

$$\left(1 - \frac{n-k-1}{N-k-1}\right)^{w-1} \leq P_{bridge} \leq 2\left(1 - \frac{n-k-1}{N-k-1}\right)^{w-1} \tag{12}$$

This situation of so many paths being fully known for more than one hop in their routes is perhaps unlikely; however, we can also determine lower bounds in case just the path of the message entering R from router S_0 and exiting to router N_0 is known to the adversary. Again assuming that the k nodes prior to

R in this path are compromised, we can determine $\Pr[S_0 \not\rightarrow N_1, \ldots, N_{w-1}]$. We cannot say anything about $\Pr[S_1, \ldots, S_{w-1} \not\rightarrow N_0]$ in this case because we do not know about the path nodes chosen prior to the S_i for $i \neq 0$ and cannot trace each back to a unique initiator. For this reason, we cannot give an upper bound. But we can give a lower bound.

$$P_{bridge} = \prod_{i=0}^{w-2} \left(1 - \frac{n-k-2}{N-k-i-2}\right) \geq \left(1 - \frac{n-k-2}{N-k-2}\right)^{w-1} \tag{13}$$

Bridges without compromise anywhere you like. If the path up to R is not compromised the attack becomes less likely to succeed but is still possible in some cases. As before we assume that a node R receives messages from nodes $S_{0\ldots w-1}$ and outputs those messages to nodes $N_{0\ldots w-1}$. Without loss of generality we assume that the message from S_0 is routed through R to N_0 and try to calculate the probability the adversary can infer this without any doubt.

Unlike our assumption so far, the adversary does not a-priori know which set of w initiators are responsible for the w streams going through node R. Our first observation is that the number of potential initiators, N', for each incoming link is much smaller than the total $N+1$ nodes, since they are assumed to know at least nodes S_i and R. According to our results on the fingerprinting attack we expect about $A_2 = E[N'] = (n/N)^2 \cdot N$ potential initiators for each link.

As before we try to calculate the probability an adversary can bridge over node R and uncover the path $S_0 \rightarrow R \rightarrow N_0$. This is possible if *either*:

- there is no initiator that knows nodes S_0 and R and other destinations $N_{1\ldots w-1}$. We denote this as
  $\Pr[(S_0 \rightarrow R) \not\rightarrow N_1, \ldots, N_{w-1} | S_0 \rightarrow R \rightarrow N_0, N']$
  or,
- there are no initiators that know nodes R and N_0 as well as any of the nodes $S_{1\ldots w-1}$. We denote this as
  $\Pr[(N_0 \leftarrow R) \not\leftarrow S_1, \ldots, S_{w-1} | S_0 \rightarrow R \rightarrow N_0, N']$.

The probability of a successful bridging attack in this context is:

$$\left(1 - \frac{n-2}{N-2-(w-2)}\right)^{(w-1)(N'-1)} \leq P_{\text{bridge}} \leq 2\left(1 - \frac{n-2}{N-2}\right)^{(w-1)(N'-1)} \tag{14}$$

Proof. We first calculate the probability
$p_1 = \Pr[S_0 \rightarrow R \not\rightarrow N_1, \ldots, N_{w-1} | S_0 \rightarrow R \rightarrow N_0, N']$ that no other node knows S_0, R and any of the other destinations $N_1, \ldots, N_{w-1}$. This means that the other $N'-1$ nodes have not chosen any of $N_1, \ldots, N_{w-1}$ as part of their remaining $n-2$ nodes. The probability of this happening for any of them is $\binom{N-2-(w-1)}{n-2} / \binom{N-2}{n-2}$ and there are $N'-1$ independent nodes for which this must hold. Hence,

$$p_1 = [\Pr[S_0 \not\rightarrow N_1, \ldots, N_{w-1}]]^{N'-1} = \left[\frac{\binom{N-2-(w-1)}{n-2}}{\binom{N-2}{n-2}}\right]^{N'-1} \tag{15}$$

Now note that $p_2 = \Pr[N_0 \leftarrow R \not\leftarrow S_1, \ldots, S_{w-1} | S_0 \rightarrow R \rightarrow N_0, N']$ is in fact equal by symmetry to p_1. Since bridging is successful if either of those holds, by the union bound we get:

$$\left[\frac{\binom{N-2-(w-1)}{n-2}}{\binom{N-2}{n-2}}\right]^{N'-1} \leq P_{\text{bridge}} \leq 2\left[\frac{\binom{N-2-(w-1)}{n-2}}{\binom{N-2}{n-2}}\right]^{N'-1} \tag{16}$$

The lower bound is simply derived by assuming that only one of the two events takes place.

We can loosen a bit the bounds in order to get some intuitions about how the different quantities influence the probability of successful bridging. We note that:

$$\frac{\binom{N-2-(w-1)}{n-2}}{\binom{N-2}{n-2}} = \prod_{j=0}^{w-2}\left(1 - \frac{n-2}{(N-2)-j}\right) = \alpha \tag{17}$$

By assigning to the fraction in α the maximum and the minimum values j assumes we get:

$$\left(1 - \frac{n-2}{N-2-(w-2)}\right)^{w-1} \leq \alpha \leq \left(1 - \frac{n-2}{N-2}\right)^{w-1} \tag{18}$$

We substitute the derived inequalities for α into eq. 16 to derive our final bound on the probability of successful bridging in eq. 14. Intuitions about its behaviour are present in the next section.

3.3 But can the army walk across it? Building bridges in the real world

In the previous sections we described bridging and derived analytic bounds when the first node is compromised, when paths from all or from just a specific source to an honest mix are compromised, and even for the more general case where an adversary tries to bridge an arbitrary honest node in the network. Let us now examine the relevance of this attack to real world systems.

The first difficulty in applying the attack relates to the threat model it assumes. A local passive adversary is required to observe all incoming and outgoing messages or streams around the node to be bridged. Mix systems usually try to protect against such adversaries, but stream-based anonymization systems, which are already susceptible to timing attacks, do not. Yet even in the case of stream-based systems, such as onion routing (including Tor), an adversary might find it advantageous to use bridging if possible: it only requires connection information, rather than the exact timing of packets traveling in the network. If applicable, bridging requires several orders of magnitude less information about each link and node than timing attacks—and this information can be inferred through sampling network packets [11] or observing short windows of traffic.

A global passive adversary may be required to discover the sets of nodes known by each initiator in the system, depending on the exact network discovery protocol employed. Tarzan proposed the use of a DHT that can easily be infiltrated by a few nodes to observe all other nodes' activity. Current widely-used distributed anonymizing systems (Tor, mixmaster, mixminion) use a distributed but more centralized directory architecture to provide routing information. If any of these were to move away from assuming that every client in the network knows all servers, it could be subject to epistemic attack if just one of the directory servers is dishonest. It may be possible to bootstrap off using a core anonymizing network known to all clients that could be used to obtain node information from directories or to use private retrieval or other techniques to counter these. However, more research is needed to determine if there are scalable, efficient, and secure techniques for partial network discovery in any directory system from centralized to diffusely distributed. In case node discovery is unobservable by the adversary, the attacker would have to resort to monitoring the network to infer the sets of nodes know by each initiator. Distributing such unobservable sets for each client is an open research problem.

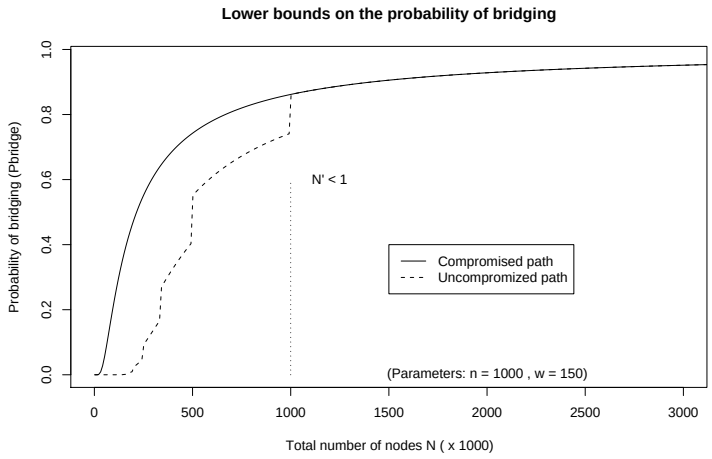

Fig. 2. The effectiveness of bridging given a compromised or uncompromised path.

The analytic bounds provided can be of great help to attackers or to system designers wishing to evaluate security, but they offer little intuition into the effectiveness of the attack in a realistic setting. To illustrate we assess the probability of success of bridging in a network where each node knows $n = 1000$ others and the batch size of relays is $w = 150$. Figure 2 plots this probability as the total number of nodes in the network N grows. As expected, the lower bound for the probability of success if the initiator of the connection is known (the compromised path case) is always greater or equal to the case where the

initiator is not known. If the path has not been linked unambiguously to an initiator (uncompromised path case), then the probability is lower according to the expected number of nodes that could be initiators of an observed link. Since $N > n^2$, we expect the number of such initiators to be at most one, and the probabilities of success for the two cases are equal for average N'.

Figure 2 illustrates clearly that the probability of bridging is not negligible: if nodes know only 1-in-500 other nodes, it is already higher than $1/2$ even if the initiator is unknown. When the initiator is known the probability of compromise rises above $3/4$. Furthermore those are lower bounds, and the adversary is very likely to be able to do better in practice.

The analysis of bridging we present is centered around the probability of successful attack P_{bridge}. This represents the probability that an adversary using the techniques describe is able to infer *with absolute certainty* the link between an incoming and outgoing message or stream. Even when this is not possible, bridging will lead to a severe reduction in anonymity. Despite the theoretical number of output streams being w, an adversary is very likely able to reduce the candidate output streams, even if they never manage to isolate a single one. This can be used to reduce anonymity and to skew the probability distributions describing who might be the sender or receiver of a message.

Similarly, an adversary with some incomplete information about which nodes are known to which users might still perform some variant of bridging to reduce anonymity. The adversary could also perform more sophisticated variants on bridging. For example there may be relations between the sets of nodes known to the other originators of streams that affect what patterns are possible amongst the observed streams that are not attacked and what can thus be inferred about the attacked stream. In that sense, our P_{bridge} actually represents only the simplest form of bridging attack. Bridging can also be performed alongside other attacks, integrating different constraints of anonymous paths like length, or the lack of cycles. This will increase the probability of successful bridging. Measures of anonymity [16, 5] taking into account those effects could be used to quantify any reduction in anonymity, but deriving analytic results in this setting might be hard. This is a promising avenue for future research.

4 Conclusion

In this paper we have examined effects of partial network knowledge on anonymity, based on both what is known and what is not known by those building routes through an anonymity network.

We presented a simple analytic bound on route fingerprinting, which is based on what route builders know about the network, and introduced a new attack, route bridging, which adds consideration of what clients do not know about the network. We also proved analytic bounds for different cases of route bridging. We illustrated our results on the initially published Tarzan design, which we found to be vulnerable to our attacks.

Successful attacks on Enigma in WWII were based on a property of the device that it would never produce an output letter that was the same as its input. Using this "nonoccurrence" statistical analysis made it possible to break encrypted messages. With the introduction of route-bridging attacks we show again that in security one must pay attention not only to what can happen but also to what cannot happen.

Our results also suggest that any attempt at scaling anonymity networks by limiting node discovery to a level below full network discovery should be carefully compared to simple partitioning as a first test. While it may be possible to maintain anonymity by such limitation, one may obtain better results, at least in this regard, simply by partitioning. On the other hand, our results also showed that the threat of epistemic attack is substantially mitigated in a client-server architecture such as that of Tor, and there is reason for cautious optimism that this threat will not preclude scaling of the design.

Wright et al. [24] suggested that to protect against passive logging attacks one might be better off choosing both entry guards and exit guards. For Tor and other three-hop anonymity systems, however, random middle nodes could do route fingerprinting with $k = 3$ and in fact a very small n as well. That is, the middle node will see both ends; so $k = 3$. And, because guards are used at both ends, any client will be choosing entry and exit nodes from a small persistent set. Current entry guards for Tor start with a default of three nodes. Which entry and exit guards are chosen by a client would not be directly apparent to the adversary from node discovery or from observation of a single route selection but would instead have to be discovered by observing repeated connections. An adversary owning a single node was able to quickly uncover entry guards by watching repeated connections (at least for circuits used by hidden services) on the live Tor network of several hundred nodes that existed in early 2006 [13]. Clearly this requires further examination. As we have noted, having orders of magnitude more clients than servers substantially diminishes such threats for Tor itself. But, vulnerability would grow if the ratio of clients to servers were to drop and the size of the network to persist or grow. This seems to be a basic difficulty for pure peer-to-peer anonymity designs unless we can anonymize network discovery without having the anonymization that the network can provide once discovered.

Acknowledgments. We would like to thank Richard Clayton, with whom the first versions or route fingerprinting were developed, and Roger Dingledine of the Tor project, for his keen interest in what happens, security-wise, when nodes only have a partial view of the network. Aaron Johnson and Emilia Käsper each read a draft of this paper and offered many helpful suggestions, for which we are grateful to them. Work by Paul Syverson supported by ONR.

References

1. Nikita Borisov, George Danezis, Prateek Mittal, and Parisa Tabriz. Denial of service or denial of security? How attacks on reliability can compromise anonymity.

In Sabrina De Capitani di Vimercati, Paul Syverson, and David Evans, editors, *CCS'07: Proceedings of the 14th ACM Conference on Computer and Communications Security*, pages 92–102. ACM Press, October 2007.

2. Philippe Boucher, Adam Shostack, and Ian Goldberg. Freedom systems 2.0 architecture. White paper, Zero Knowledge Systems, Inc., December 2000.
3. David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 4(2):84–88, February 1981.
4. George Danezis and Richard Clayton. Route fingerprinting in anonymous communications. In *Sixth IEEE International Conference on Peer-to-Peer Computing, P2P 2006*, pages 69–72. IEEE Computer Society Press, 2006.
5. Claudia Díaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity. In Roger Dingledine and Paul Syverson, editors, *Privacy Enhancing Technologies: Second International Workshop, PET 2002*, pages 54–68. Springer-Verlag, LNCS 2482, 2003.
6. Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, pages 303–319. USENIX Association, August 2004.
7. Joan Feigenbaum, Aaron Johnson, and Paul Syverson. A probabilistic analysis of onion routing in a black-box model. In Ting Yu, editor, *WPES'07: Proceedings of the 2007 ACM Workshop on Privacy in the Electronic Society*, pages 1–10. ACM Press, October 2007.
8. Michael J. Freedman and Robert Morris. Tarzan: A peer-to-peer anonymizing network layer. In Vijay Atluri, editor, *Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002*, pages 193–206. ACM Press, November 2002.
9. Michael J. Freedman, Emil Sit, Josh Cates, and Robert Morris. Introducing Tarzan, a peer-to-peer anonymizing network layer. In Peter Druschel, M. Frans Kaashoek, and Antony I. T. Rowstron, editors, *Peer-to-Peer Systems: First International Workshop, IPTPS 2002*, pages 121–129. Springer-Verlag, LNCS 2429, 2002.
10. David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. Hiding routing information. In Ross Anderson, editor, *Information Hiding: First International Workshop*, pages 137–150. Springer-Verlag, LNCS 1174, 1996.
11. Steven J. Murdoch and Piotr Zieliński. Sampled traffic analysis by internet-exchange-level adversaries. In Nikita Borisov and Philippe Golle, editors, *Privacy Enhancing Technologies: 7th International Workshop, PET 2007*, pages 167–183. Springer-Verlag, LNCS 4776, 2007.
12. Arjun Nambiar and Matthew Wright. Salsa: A structured approach to large-scale anonymity. In Rebecca N. Wright, Sabrina De Capitani di Vimercati, and Vitaly Shmatikov, editors, *CCS 2006: Proceedings of the 13th ACM Conference on Computer and Communications Security*, pages 17–26. ACM Press, October 2006.
13. Lasse Øverlier and Paul Syverson. Locating hidden servers. In *2006 IEEE Symposium on Security and Privacy (S&P 2006)*, pages 100–114. IEEE CS Press, May 2006.
14. Michael Reiter and Aviel Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, June 1998.
15. Marc Rennhard and Bernhard Plattner. Practical anonymity for the masses with morphmix. In Ari Juels, editor, *Financial Cryptography: 8th International Conference FC04*, pages 233–250. Springer-Verlag, LNCS 3110, 2004.

16. Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In Roger Dingledine and Paul Syverson, editors, *Privacy Enhancing Technologies: Second International Workshop, PET 2002*, pages 41–53. Springer-Verlag, LNCS 2482, 2003.
17. Andrei Serjantov, Roger Dingledine, and Paul Syverson. From a trickle to a flood: Active attacks on several mix types. In Fabien Petitcolas, editor, *Information Hiding: 5th International Workshop, IH 2002*, pages 36–52. Springer-Verlag, LNCS 2578, October 2002.
18. Andrei Serjantov and Peter Sewell. Passive attack analysis for connection-based anonymity systems. In Einar Snekkenes and Dieter Gollmann, editors, *Computer Security – ESORICS 2003, 8th European Symposium on Research in Computer Security*, pages 116–131, October 2003.
19. Vitaly Shmatikov. Probabilistic model checking of an anonymity system. *Journal of Computer Security*, 12(3-4):355–377, 2004.
20. Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek, and Hari Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Trans. Netw.*, 11(1):17–32, 2003.
21. Paul Syverson, Michael Reed, and David Goldschlag. Onion Routing access configurations. In *Proceedings of the DARPA Information Survivability Conference & Exposition, DISCEX'00*, volume 1, pages 34–40. IEEE CS Press, 1999.
22. Paul Syverson, Gene Tsudik, Michael Reed, and Carl Landwehr. Towards an analysis of onion routing security. In Hannes Federrath, editor, *Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability*, pages 96–114. Springer-Verlag, LNCS 2009, July 2000.
23. Parvathinathan Venkitasubramaniam, Ting He, and Lang Tong. Anonymous networking amidst eavesdroppers. Pre-print available as arXiv:0710.4903v1 at arxiv.org, October 2007.
24. Matthew Wright, Micah Adler, Brian Neil Levine, and Clay Shields. Defending anonymous communication against passive logging attacks. In *2003 IEEE Symposium on Security and Privacy*, pages 28–43. IEEE CS Press, May 2003.
25. Matthew Wright, Micah Adler, Brian Neil Levine, and Clay Shields. The Predecessor Attack: An Analysis of a Threat to Anonymous Communications Systems. *ACM Transactions on Information and System Security (TISSEC)*, 4(7):489–522, November 2004.
26. Li Zhuang, Feng Zhou, Ben Y. Zhao, and Antony I. T. Rowstron. Cashmere: Resilient anonymous routing. In *2nd USENIX Symposium on Networked Systems Design and Implementation (NSDI 2005)*, pages 301–314. USENIX Association, 2005.