On the Optimal Path Length for Tor

Kevin Bauer[1], Joshua Juen[2], Nikita Borisov[2],
Dirk Grunwald[1], Douglas Sicker[1], and Damon McCoy[3]

[1] University of Colorado at Boulder
{bauerk, grunwald, sicker}@colorado.edu
[2] University of Illinois at Urbana-Champaign
{juen1, nikita}@illinois.edu
[3] University of California at San Diego
dlmccoy@cs.ucsd.edu

Abstract. Choosing a path length for low latency anonymous networks that optimally balances security and performance is an open problem. Tor's design decision to build paths with precisely three routers is thought to strike the correct balance. In this paper, we investigate this design decision by experimentally evaluating several of the key benefits and drawbacks of two-hop and three-hop paths. We find that (1) a three-hop design is slightly more vulnerable to endpoint compromise than a two-hop design in the presence of attackers who employ simple denial-of-service tactics; (2) two-hop paths trivially reveal entry guards to exit routers, but even with three-hop paths the exit can learn entry guards by deploying inexpensive middle-only routers; and (3) three-hop paths incur a performance penalty relative to two-hop paths. Looking forward, we identify and discuss a number of open issues related to path length.

1 Introduction

Design decisions made by low latency anonymizing networks frequently involve achieving a correct balance between security and performance. For example, Tor does not employ cover traffic or add intentional delays in order to ensure performance that is sufficient to support interactive applications such as web browsing. However, this decision has increased Tor's vulnerability to end-to-end traffic correlation. Another key design decision is path length. Tor employs a decentralized architecture of precisely three routers to mitigate any single router's ability to link a source and destination. However, three-hop paths have a performance cost. In this paper, we seek to better understand the security and performance trade-offs related to path length design decisions.

Tor's design — like most low latency anonymizing networks — is vulnerable to end-to-end traffic correlation attacks. If the endpoints are compromised, an adversary can apply any one of many known traffic analysis attacks [1–8] to correlate the source and destination. Conventional wisdom indicates that three-hop paths achieve an appropriate balance between security and performance. However, two-hop paths may be attractive to users seeking improved performance, though it is unclear what security trade-offs two-hop paths may incur.

Through analysis, simulation, and experiments performed on the live Tor network, we critically evaluate the advantages and disadvantages of two-hop and three-hop paths from security and performance perspectives. In addition, we identify and discuss a variety of open issues related to the security and performance of different path length choices.

Path length and security. We consider an adversary who uses selective disruption tactics (as in [4, 9–11]) to force circuits to be rebuilt in the event that a malicious router participates in a circuit that is not compromised. Through simulation of Tor's router selection algorithm fueled by real router data obtained from Tor's trusted directory servers, we show that three-hop paths are up to 7% more vulnerable to path compromise than two-hop paths under the same attack.

One potential disadvantage of a two-hop design is that exit routers can trivially discover clients' entry guards, since they communicate directly. We empirically demonstrate that malicious exit routers can identify clients' entry guards even with three-hop paths by deploying middle-only routers that employ selective disruption. Our results show that an adversary with only ten malicious exit routers and 50 middle-only routers can learn the entry guards for nearly 80% of all circuits constructed. We also analyze the potential to identify clients uniquely through knowledge of their entry guards.

We lastly perform experiments on the real deployed Tor network to show that low cost timing-based traffic analysis techniques that link circuits by their circuit building messages can be highly successful in practice. On the live Tor network with a workload of real user traffic, we show that timing analysis can successfully link 97% of the traffic from clients that we control even before any data traffic is sent.

Path length and performance. In addition to an analysis of path length from a security perspective, we show that shorter paths offer better performance as perceived by end-users in terms of download time. We perform an analysis of typical web browsing behavior and demonstrate that users will see fewer circuit failures with two-hop paths, which results in faster web page loading and an improved user experience.

2 Tor: The Second Generation Onion Routing Design

Tor is the second generation onion routing design providing a low latency anonymizing overlay network for TCP-based applications [12]. One of Tor's primary design goals is to ensure low enough latency to facilitate interactive applications such as web browsing and instant messaging. Tor's system architecture consists of *Tor routers*, which are volunteer-operated servers, *directory servers* that organize information about the Tor routers, and *Tor proxies* (or clients). Tor routers may be configured by their operators to allow connections only to other Tor routers, or to allow exit connections to arbitrary hosts on the Internet. Tor clients query one of the authoritative directory servers to obtain a signed list of the available Tor routers, their public keys, bandwidth advertisements, exit policies, uptime, and other flags indicating their entry guard status and other information.

To establish an anonymous virtual connection through the Tor network to a desired destination, the client must first choose a path (or circuit[1]) of precisely three Tor routers and establish a shared symmetric key with each, using authenticated Diffie-Hellman and a telescoping key agreement procedure. Once the circuit has been created, the client encrypts their data in 512 byte units called *cells* with each key in a layered manner and forwards these cells to the first router in the circuit. Upon receiving a cell, each router removes its layer of encryption using its symmetric key shared with the client and forwards the cell to the next router in the circuit. Finally, after the exit router removes the final layer of encryption, it establishes a TCP connection with the destination and sends the client's data. More details can be found in the Tor Protocol Specification [13].

2.1 Tor's Router Selection Algorithm

The manner in which Tor clients select their routers has serious implications for the network's security properties. For example, if a client chooses malicious routers, then they may experience lost anonymity. At Tor's inception, it was composed of only a few high-bandwidth routers and had few users, so it was sufficient to select routers uniformly at random. As the network grew in popularity and router bandwidth diversity, it became necessary to balance the traffic load over the available bandwidth resources, which can be achieved by selecting routers according to their bandwidth capacities. However, Tor routers self-advertise their bandwidth capacities. It has been shown that an adversary can falsely advertise high bandwidth claims to attract traffic and increase their ability to compromise circuits [4, 14].

Recent work has proposed methods to securely verify these self-reported bandwidth claims [15]. Active measurements have been integrated into the Tor network's directory servers to verify routers' bandwidth claims [16]. However, the security of these active measurements has yet to be evaluated.

Tor's router selection algorithm [17] chooses routers with the following constraints:

- A router may only be chosen once per path.
- To prevent an adversary who controls a small network from deploying a large number of routers, each router on a path must be from a distinct /16 subnet (in CIDR notation).[2]
- Each router must be marked as `Valid` and `Running` by the authoritative directory servers.

[1] The terms "path" and "circuit" are used interchangeably throughout this paper.

[2] Tor also allows an operator of many relays to set an advisory `Family` flag that will ensure that their nodes are not chosen twice per path.

– For non-hidden service circuits, each router must be marked as Fast, indicating that the router has at least 100 KB/s of bandwidth or is within the top 7/8 of all routers ranked by bandwidth.
– The first router on the path must be marked as a Guard by the authoritative directory servers. Clients select precisely three entry guards to use on their circuits, and choose new guards periodically.
– The last router on the path must allow connections to the destination host and port.

For general purpose circuits, Tor's path selection algorithm weighs router selection by each router's perceived bandwidth capacity. In order to ensure that there is sufficient exit bandwidth available, the bandwidth of Exit routers is weighted differently depending on the fraction of bandwidth that is available from non-Exit routers. Suppose that the total exit bandwidth is E and the total bandwidth available is T. If $E < T/3$, then Exit routers are not considered for non-exit positions. Otherwise, their bandwidth is weighted by $(E - (T/3))/E$ [17].

Entry guards were introduced to Tor's design to mitigate the threat of profiling and the predecessor attack [14]. Entry guard nodes have special uptime and bandwidth properties. A router is marked as a Guard by the authoritative directory servers only if its mean time between failures is above the median of all "familiar"[3] routers and its bandwidth is greater than or equal to 250 KB/s [18]. By default, clients choose precisely three entry guards to use for their circuits. To ensure that there is sufficient guard bandwidth available, guard node selection is weighted by $(G - (T/3))/G$, where G is the amount of available guard bandwidth. If $G < T/3$, then guard nodes are not considered for non-guard positions [17].

3 Security Analysis

In this section, we study the security implications of Tor's path length. First, we evaluate how an adversary's ability to compromise circuits varies between two-hop and three-hop paths. Second, we explore how two-hop paths reveal circuits' entry guards and discuss the potential for adaptive surveillance attacks. We also describe an attack where an adversary with few exit routers and comparatively many middle-only routers can identify the entry guards on a large fraction of circuits. Third, the amount of information about clients that is revealed by entry guard knowledge is analyzed. Finally, we evaluate a low cost traffic analysis technique that links circuits using only circuit building messages on the live Tor network. This attack's success re-iterates the fact that three-hop paths provide no protection whatsoever against these attacks.

[3] A router is "familiar" if one-eighth of all active routers have appeared more recently than it [18].

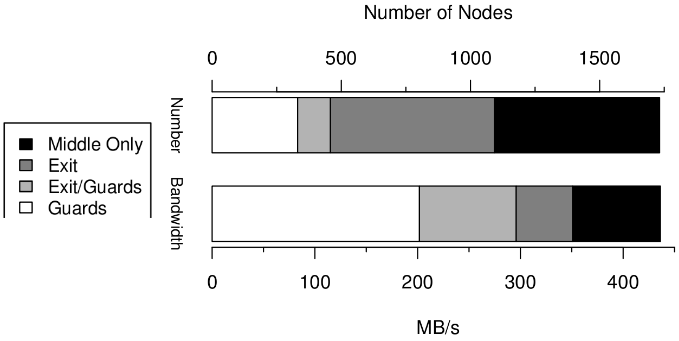

Fig. 1. Distribution of routers used in simulations, as gathered from a directory server

3.1 Selective Disruption and Path Length

To understand the relationship between path length and circuit compromise, we simulate Tor's current router selection algorithm (described in Section 2.1) using router data collected from an authoritative directory server.

Simulation setup. We adopt a simulation methodology similar to Murdoch and Watson [19] in which malicious routers are added to the network and circuit compromise statistics are computed. In particular, we simulate 1 000 clients who choose precisely three entry guards and each construct 100 circuits of length two and three that are suitable for transporting HTTP traffic (port 80).[4] Next, a variable number of malicious routers between 5 and 50 are injected into the network. Each malicious router has 10 MB/s of bandwidth,[5] is marked as a `Guard`,[6] allows port 80 to exit, and is operated on a distinct /16 subnet.

A snapshot of all Tor routers was obtained from an authoritative directory server on January 6, 2010. This snapshot (summarized in Figure 1) consists of 1 735 total routers marked as `Valid` and `Running`. Note that the snapshot has sufficient entry guard and exit bandwidth such that both entry guards and exit routers may by used for any position of the circuit, provided that they have the appropriate flags.

Results. Figure 2 shows the fraction of circuits that are compromised as the number malicious routers and amount of adversary-controlled bandwidth increases. First, note that for attackers that do not apply selective disruption, the circuit compromise rate is directly proportional to the adversary's resource in-

[4] We simulate HTTP exit traffic because prior work found it to be the most common type of traffic by connection on the real Tor network [20, 21].

[5] Currently the largest believable bandwidth value.

[6] Obtaining the `Guard` flag only requires that the router demonstrate stability for a relatively short period of time. We anecdotally found that a new router on a high bandwidth link can obtain the `Guard` flag after running for roughly seven days.

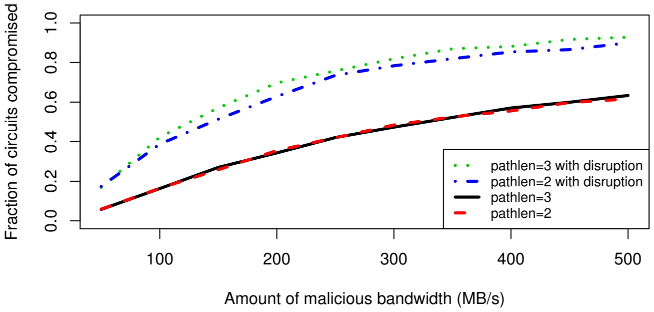

Fig. 2. Fraction of HTTP circuits compromised

vestment: 50 attackers with 10 MB/s of bandwidth each control over half of the network's bandwidth, but are able to compromise just over half of all circuits. Also, the compromise rate is the same regardless of whether three- or two-hop paths are used for each malicious router configuration since the attacker gains no advantage from participating in circuits that are not compromised.

Now consider an adversary whose routers selectively disrupt circuits on which they're chosen that they cannot compromise. Regardless of path length, if the client has the misfortune of choosing three malicious entry guards, then due to selective disruption, their circuits are always compromised. If a client chooses no malicious entry guards, then their circuits are never compromised. Clients that chose one or two malicious entry guards experience circuit compromise with a certain probability. For example, when there are 10 malicious routers and clients use three-hop paths, 38% of clients choose no malicious guards, 47% choose one malicious guard, 13% choose two malicious guards, and 1% choose three malicious guards. Of the clients that choose one or two malicious guards, their circuits are compromised 63% and 85% of the time, respectively. Note that entry guards offer some degree of protection against circuit compromise, since clients that choose no entry guards are safe and the threat increases with the selection of additional malicious entry guards.

As shown in Figure 2, across all malicious router configurations the fraction of circuits compromised is up to 7% higher for three-hop paths relative to two-hop paths. With three-hop paths, when the client selects one or two malicious guards, their circuits are disrupted when they use a non-malicious guard and a malicious middle or exit router (or both). In this case, the circuit is rebuilt and the client may use a malicious guard. With two-hop paths, if the client does not use a malicious guard, then only the exit position can disrupt non-compromised circuits. Since three-hop paths have one additional position from which to disrupt circuits, they exhibit a slightly higher compromise rate relative to two-hops. However, it is unclear if this small increase in the risk of endpoint compromise is a sufficient danger to justify a change in Tor's default path length.

3.2 Adaptive Surveillance Attacks and Path Length

In addition to the threat posed by compromised routers, Tor's three-hop design is ostensibly vulnerable to attacks whereby a powerful ISP or government adversary can monitor a targeted circuit's endpoints' networks to identify the traffic's source and destination. This attack is believed to be difficult with three-hop paths because it relies on a circuit having the misfortune of choosing an entry and exit router that reside within monitored networks. Since Tor achieves network diversity in its route selection in practice [22], this attack would require collusion by many network operators.

However, with two-hop paths exit routers can directly observe the entry guards. Suppose that a client builds a circuit through an adversary-controlled exit router, but uses a non-malicious entry guard. Since the exit router knows the client's entry guard, they could adaptively demand network logs from the entry guard's network through legal channels or other forms of coercion. While this attack requires a powerful adversary and consequently may be unlikely, two-hop paths make the attack technically feasible which may encourage malicious exit routers (or their network operators) to implement it.

While two-hop paths enable adaptive surveillance attacks by leaking entry guards to the exit router, adaptive surveillance is possible even with Tor's current three-hop design. If an adversary deploys both malicious exit routers and malicious middle-only routers, they can collude to identify the entry guards used for every circuit on which they are used for the middle and exit positions. We next show that an adversary who controls few exit routers and comparatively many malicious middle-only routers can identify the entry guard used for a large fraction of circuits.

Simulation setup. Experiments are conducted where an adversary controls only ten exit routers configured to exit HTTP (port 80) traffic and injects 50 and 75 middle-only routers to the Tor network summarized in Figure 1. All malicious routers have 10 MB/s of bandwidth and now disrupt circuits when they do not control both the middle *and* exit positions. We simulate 1 000 clients who each build 100 circuits.

This attack strategy has a low cost for the adversary, since they do not need to demonstrate router stability (as is necessary to obtain the guard flag). In addition, all malicious middle-only nodes could be deployed on the same /16 network and all malicious exit routers could be deployed on a second /16 network. Thus, the resources required to launch this attack are modest.

Results. For an attacker with 10 malicious exit routers and 50 middle-only routers, the adversary can identify the entry guard for 79% of all circuits constructed. When the attacker deploys 75 middle-only routers, they discover the client's entry guard for 85% of all circuits. For these circuits, the adversary could apply pressure and potentially coerce the entry guard (or its network operator) to reveal the identity of the client.

Table 1. Daily statistics for clients per entry guard and entropy estimates

No. of Samples	**Minimum**	**Maximum**	**Median**	**95% Confidence Interval**
$n = 737$	680	164 000	8416	(24 104, 27 176)
Entropy	8.20	0.29	4.57	(3.05, 2.88)

Perhaps the most compelling argument in favor of three-hop paths for Tor is that the middle router hides the entry guards from exit routers. By using a middle router, a malicious exit typically knows only information about the client that is leaked by their applications. However, if malicious exits collude with middle routers who can observe the entire circuit, it becomes feasible for the exit to learn a large fraction of the total client population's entry guards.

To make matters worse, deploying a relatively large number of middle-only routers causes a global change in Tor's router selection process. In these experiments, when 50 middle-only routers are introduced, the aggregate entry guard bandwidth G and aggregate exit router bandwidth E no longer satisfies $G \geq T/3$ and $E \geq T/3$, respectively, where T is the total bandwidth. In this network configuration, exit routers may only be used for the exit position and entry guards may only be used for the guard position. This enables the adversary to focus their few exit routers toward occupying the exit position and maximize their ability to conduct adaptive surveillance.

3.3 Entry Guard Linkability

With a two-hop design, we know that malicious exit routers can discover clients' entry guards. It is possible that clients' entry guards may be uniquely identifying or place clients into small anonymity sets. To understand the extent to which knowledge of clients' entry guards may be identifying, we next analyze publicly available data on entry guard usage from the Tor Metrics Project [23]. From this data, eleven entry guards provide information about the number of clients that they observe over time.[7] Table 1 presents a statistical summary of the number of clients observed by each entry guard on a daily basis. With this data, we can estimate how much identifying information is leaked through knowledge of a client's entry guard.

We apply the standard entropy metric from information theory [24] to measure how much information is revealed about a user by their entry guard selections. The total number of unique Tor users per day is currently estimated to be between 100 000 and 300 000 [25]. Thus, without any additional knowledge, 17.61 bits of information are necessary to uniquely identify a Tor user.[8] Now suppose that a malicious exit router knows a particular client's entry guard. On average, roughly 25 000 clients use the same entry guard, so this knowledge leaks only

[7] To preserve users' privacy, this data is aggregated by country of origin, quantized by multiples of eight, and compiled daily.

[8] This analysis assumes that 200 000 unique clients use Tor each day.

2.96 bits of information about a user's identity. Even in the worst case when a client shares a guard with as few as 680 other clients, only 8.20 bits are revealed (the full entropy results are shown in Table 1).

If an attacker knows all three of a particular client's entry guards, the client may be more identifiable since a choice of three guards may be significantly more unique than a single guard. While it is usually difficult to link a client across multiple entry guards, if a client inadvertently identifies herself — perhaps by logging-in to a website or using an application that does not support SSL/TLS — over time her full set of entry guards could be leaked to a malicious exit router. Tor clients do, however, expire their entry guard selections periodically, which may help to protect users from this type of profiling.

We should also point out that, even with three-hop paths, linkability pitfalls still exist in Tor. First, a Tor circuit can be used by several connections, which can be trivially linked by the exit router. Second, the predecessor attack shows that the entry guards used by a client can be learned after $O(1/(f_m f_e))$ circuit constructions on average, where f_m and f_e are the probabilities that a malicious router will be chosen as the middle and exit router, respectively [26]. Selective disruption and other techniques [27] can be used to increase the speed of such attacks.

3.4 Low Resource Traffic Analysis on the Live Tor Network

Prior work has shown that end-to-end traffic correlation attacks launched against low latency anonymous networks can achieve near perfect accuracy [2]. To support interactive or delay-sensitive applications, Tor does not explicitly delay or batch messages to help defend against end-to-end traffic correlation attacks. Consequently, Tor's design assumes that these attacks can achieve high accuracy in practice. In fact, such an attack has been proven effective against the live Tor network in 2006 [14]. Since then, a low resource traffic analysis technique has been proposed that uses only circuit construction messages to link a source and destination before any data is sent [4]. This approach allows low bandwidth attackers to maximize the number of circuits compromised, but this low cost attack has yet to be validated on the live Tor network. We next evaluate this traffic analysis approach on the live Tor network.

Experimental setup. We deploy two Tor routers[9] hosted on a 100 Mb/s network link onto the live Tor network. Each router has a distinct configuration: (1) One Tor router is configured as a non-exit and after roughly ten days of uninterrupted operation, it obtained the `Guard` flag from the authoritative directory servers. (2) A second Tor router is configured with the default exit policy.[10] During their operation, both routers sustained roughly 3 MB/s of traffic.

To evaluate the expected success of traffic analysis, we operate our own Tor clients and attempt to link their circuits to their destinations. Upon building

[9] These routers ran software version `Tor 0.2.1.20`.

[10] Ports often associated with outgoing e-mail, peer-to-peer file sharing applications, and high security risk services are blocked.

a circuit, each client downloads `www.google.com`, tears down the circuit, and repeats this procedure. To preserve users' privacy, we ignore traffic at the entry guard that is not produced by one of our clients.[11] Note that we do not retain any linkable data nor do we attempt to deanonymize any other clients but our own.

Traffic analysis methodology. We apply a traffic analysis technique in which circuits are linked by their circuit building messages before the clients send any data cells. This approach leverages the fact that Tor's circuit establishment procedure sends a fixed number of circuit building messages in an identifiable pattern.

Briefly, circuit linking via circuit building messages works as follows. First, our entry guard ensures that the circuit building request is from a client and not a Tor router. Next, it is necessary to ensure that the next router for our entry guard is the same as the previous router for our exit router (with a tight time difference). Finally, the circuit building messages for the entry, middle, and exit routers should occur in increasing chronological order. More details about our linking procedure can be found in [4].

Results. On the live Tor network, our clients build a total of 1 696 circuits that always use our entry guard. Of these 821 circuits use our exit router and 875 circuits use a different exit router.[12] The middle routers are chosen according to Tor's default selection algorithm. Through traffic analysis, we link their circuits with 97% accuracy, 0.6% false negatives (6 false negatives in total), and 6% false positives (52 false positives in total). We regard these results as a lower bound on attainable traffic analysis success, as it should be possible to increase the accuracy by also using data cells to link circuits. Also, we observe that circuits that use a popular (*i.e.,* high bandwidth) middle router tend to be more prone to false positives. Thus, an attacker who sees a positive result with a low bandwidth middle router can be more confident in the result. Given the high accuracy and the relatively easy manner in which the traffic analysis was conducted, we confirm that three-hop paths offer no protection against low cost timing attacks.

4 Performance Analysis

We have already studied Tor's path length from a security perspective. We next examine its performance implications. Since the vast majority of Tor traffic is interactive web browsing [20, 21], we investigate the performance benefits of a two-hop design from a web browsing end-user's perspective.

Experimental setup. In order to understand Tor's performance in a manner that reflects the quality of a user's experience, we simulate real clients accessing the 15 most popular websites[13] over Tor version 0.2.1.24 with Polipo version 1.0.4 and measure the download times. Experiments are conducted in February

[11] This data collection procedure was approved by the University of Colorado's Institutional Review Board.

[12] This setup allows us to count the number of false positives that occur during linking.

[13] We consider the 15 most popular websites according to `http://www.alexa.com`.

2010 over the course of four days.[14] Circuits are constructed according to Tor's default router selection algorithm and the Firefox browser downloads one of the web pages.

In the event of a circuit failure, Firefox's default behavior is to time-out after two minutes. However, real users may be impatient and explicitly force the browser to reload the page by pressing the "refresh" button. Prior work has found that users of low latency anonymous networks tend to tolerate no more than four seconds of latency [28]. Thus, in the event of a circuit failure, we assume that users wait not the full two minutes for their browsers to time-out, but precisely four seconds before explicitly reloading the page.

Results. A CDF of download times for two- and three-hop paths is shown in Figure 3. For three-hop paths, half of all web page downloads take longer than 12 seconds, while for two-hop paths, half complete in over 8 seconds. The mean download time for three-hop circuits is over 28 seconds, which is twice the expected download time for traffic over two-hop circuits (14 seconds).

We observe that circuit failures tend to be a significant cause of the additional expected download times with three-hop circuits.[15] 21% of circuits fail with three-hop paths, but only 15% of circuits fail with two-hop paths. The observed unreliability of three-hop circuits may contribute to high download times, as some users may wait unnecessarily for their browser to time-out. In these experiments, we assume that the user can quickly identify that their session has stalled (*i.e.,* by observing that no web content has loaded) and refresh the page after waiting four seconds for content to appear. However, some users may take significantly longer to launch another web request.

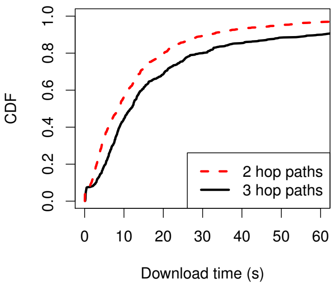

Fig. 3. Time to download popular web pages

5 Discussion

Having discussed Tor's path length from security and performance perspectives, we next discuss a variety of open issues related to path length.

[14] While prior studies have found that Tor's performance varies by time of day [28, 29], a more recent study did not identify diurnal patterns in Tor's traffic load [20]. Thus, we do not believe the time of day to be a significant factor that effects performance.

[15] A circuit failure occurs when a circuit fails after the circuit has been established and at least one data stream has been attached. This is different than a circuit building failure, where a chosen circuit cannot be built.

5.1 User-Configurable Path Lengths

Since two-hop paths offer better performance, it may be tempting to allow users who value performance over security to use two-hop paths while users who need stronger security may use three-hop paths. Suppose that most users value performance and consequently, Tor chose a default path length of two hops. Security-conscious users could optionally use three hops to take advantage of the additional security that three-hop paths offer against adaptive surveillance. However, clients who choose to use longer paths may be identified as desiring additional security, which alone could draw an adversary's attention. Furthermore, it has been argued that most users tend to keep default options, even when the defaults may not be optimally suited to their needs [30]. Allowing users to configure their own path lengths assumes that users understand the full security implications of their choice, which is unlikely, particularly for novice users. Thus, all users should be encouraged to use the same path length.

5.2 Potential Liabilities for Exit Routers

Beyond the potential risks of identifying users who desire stronger security by their path length choice, two-hop paths could be a liability for exit router operators. With three-hop paths, exit routers know nothing about clients other than what may be revealed by their traffic. However, with two-hop paths, exit routers are exposed to clients' entry guards; thus, they are no longer agnostic with regard to the clients whose traffic they transport. Exit routers could be presented with subpoenas to reveal entry guard information to governments or law enforcement agents, which increases the risks associated with operating an exit router. Since Tor's exit bandwidth is relatively scarce yet essential to the network's ability to function properly, liabilities for exit router operators should be minimized to attract additional exit routers.

5.3 Secure Bandwidth Estimation

The attacks that we describe in Sections 3.1 and 3.2 are particularly dangerous in the absence of secure bandwidth verification, since malicious routers could otherwise inflate their perceived bandwidth to attract traffic. With secure bandwidth estimates in place, it will no longer be possible to carry out these attacks with few resources. However, it is important to remember that such attacks are still within reach of medium-to-large organizations, or even determined individuals: at current hosting rates, running a 10 MB/s node for one week (long enough for a node to be declared a guard) can cost less than $1 000;[16] thus, the financial resources required to attack the network successfully are moderate at best. Additionally, attackers may be able to insert their own high-bandwidth nodes into the Tor network by compromising computers at well-provisioned institutions.

[16] See, for example, `http://aws.amazon.com/s3/`.

5.4 Does a Two-Hop Design Discard Many Routers?

Many Tor routers are not configured to allow exit traffic and are not fast and/or stable enough to be an entry guard. These routers are only used for the middle position. We next consider whether a two-hop design would discard a significant number of middle-only routers and their collective bandwidth.

From the directory server snapshot analyzed in Section 3, we find that 639 routers may only be used for the middle position. These routers collectively contribute about 85 MB/s of bandwidth. To understand how bandwidth is distributed among non-exit and non-guard routers, Figure 4 shows a CDF of these routers' bandwidth contributions. Half contribute less than 50.3 KB/s each and only 11% offer the 250 KB/s necessary to meet the bandwidth criterion for the guard flag. These higher bandwidth routers collectively contribute 54.3 of the 85 MB/s of middle-only bandwidth. If stable enough, they could eventually obtain the guard flag and be used for the entry position.

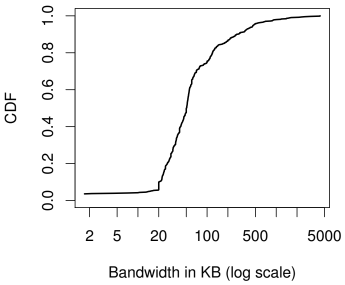

Fig. 4. Bandwidth contributions from middle-only routers

6 Related Work

Security in low latency anonymous networks. An early security analysis of low-latency anonymous networks suggests that an anonymous path is compromised if its endpoints are controlled by an adversary; the expected success of such an attack is roughly $(c/n)^2$, where there are c malicious routers, n total routers, and clients choose routers uniformly at random [31]. As networks such as Tor evolved, it became necessary to balance the traffic load over a diverse set of volunteer routers, making the task of analytically modeling path compromise more challenging. To meet this challenge, Murdoch and Watson propose that path compromise be analyzed empirically through faithful simulation of the underlying routing mechanism in the presence of different threat models [19]. We adopt a similar empirical approach to reasoning about Tor's security properties.

Selective disruption attacks. Selective disruption attacks are a form of denial-of-service (DoS) that allow an adversary to increase the number of circuits compromised. These attacks work as follows: a malicious router who uses selective disruption should refuse to forward traffic in the event that they participate in a circuit that is not compromised. This causes the circuit to fail and be rebuilt, providing an opportunity for malicious routers to compromise another circuit.

Bauer *et al.* show that an attacker with only six malicious Tor routers who utilizes the selective disruption strategy can compromise over 46% of all clients' circuits in an experimental Tor network with 66 total routers [4]. Similarly, Borisov *et al.* demonstrate that an adversary who uses selective disruption experiences a significantly greater path compromise rate. Without selective disruption, an attacker who controls 50% of the network's bandwidth can compromise 25% of all circuits, but with selective disruption they can compromise up to 66% of circuits [9]. However, their analysis was performed using a highly simplified model of Tor path selection. They also found that, for mix networks, increased path length results in greater susceptibility to selective disruption attacks, but did not analyze the effects of path length in Tor. We examine how selective disruption attacks are less effective with two-hop paths than three hops.

Given the danger of selective disruption, Danner *et al.* describe an algorithm for detecting selective disruption attacks that requires a number of probes that scales linearly with the network size [10]. However, such an active probing approach may introduce high load into the network. Also, active probing could be gamed by an intelligent attacker who can recognize the probes, or the adversary could disrupt circuits probabilistically to blend in with expected background circuit failures. Ultimately, the DoS strategy allows attackers to perform traffic analysis on a far greater number of circuits than would otherwise be possible.

End-to-end traffic correlation attacks. Prior work has shown that end-to-end traffic correlation attacks are highly effective against low-latency anonymizing networks. Levine *et al.* demonstrate through simulation that the performance of timing-based traffic correlation attacks is dependent on network conditions, but they show that an adversary can correlate traffic with perfect accuracy when the packet drop rate is very low [2]. For circuit linking experiments carried out on a small experimental Tor network, Bauer *et al.* report only 12 false positives out of over ten thousand successful correlations [4]. However, their traffic load was light and uniform, which may have contributed to the extremely low false positive rate. Also, Syverson and Øverlier report a negligible false positive rate for a traffic correlation attack on a Tor hidden service [14]. In this paper, we verify that similarly high traffic correlation accuracies can be expected for low-resource traffic analysis attacks launched on the real Tor network.

Alternate router selection strategies. Given the threat of malicious routers positioning themselves at circuit endpoints for a large number of circuits, Snader and Borisov propose that clients have the ability to "tune" the router selection process between security and performance [32]. Choosing routers more uniformly at random reduces the end-user's risk of choosing malicious routers who inflate their bandwidth claims to attract traffic, however, at the potential cost of choosing low bandwidth routers and experiencing poor performance. In addition, Sherr *et al.* propose that link-based attributes (such as latency or jitter) be used to select routers rather than node-based attributes (like bandwidth) [33]. However, these proposed routing techniques have yet to be adopted in practice. Consequently, we only consider Tor's current router selection algorithm in our subsequent analysis.

Prior performance analyses. Beyond the security properties of low latency anonymizing networks, recent work has investigated their performance characteristics. It has been shown that users are more likely to use anonymous communication services that offer better performance [34]. In a performance study of Tor and AN.ON [35] (a mix cascade) from the end-user's perspective, Wendolsky *et al.* find that Tor is subject to unpredictable performance and observe that users exhibit a four second tolerance to delay [28]. However, Tor users often experience significant delays beyond this user tolerance threshold [29].

To help explain Tor's poor observed performance, Reardon and Goldberg identify that because Tor multiplexes many streams over the same TCP connections, congestion control interference among different circuits is produced [36]. These unintended interactions often cause very high delays for end-users. TCP-over-DTLS, an alternate transport design, is proposed to improve performance. Our work is complementary to these prior studies. Since end-users are sensitive to excessive delays, we quantify the performance improvement that can be expected with a two-hop design and argue that such a design may offer even more improvement in combination with TCP-over-DTLS.

7 Conclusion

We critically evaluate Tor's path length and consider the advantages and disadvantages of a two-hop and three-hop design. We show that two-hop paths are slightly less vulnerable to circuit compromise attacks than three-hop paths, but two-hop paths are trivially vulnerable to adaptive surveillance and introduce potential liabilities for exit node operators. While performance is improved with shorter paths, we conclude that there is no strong argument for reducing Tor's path length. However, we identify a number of open issues that could effect this decision. Our hope is that this paper encourages further investigation into the security and performance trades-offs of various path lengths.

Acknowledgments

We thank Prateek Mittal, Mike Perry, and the anonymous reviewers for their helpful comments and suggestions.

References

1. Serjantov, A., Sewell, P.: Passive attack analysis for connection-based anonymity systems. In: Proceedings of ESORICS 2003. (October 2003)
2. Levine, B.N., Reiter, M.K., Wang, C., Wright, M.K.: Timing attacks in low-latency mix-based systems. In: Proceedings of Financial Cryptography (FC '04). (February 2004)
3. Shmatikov, V., Wang, M.H.: Timing analysis in low-latency mix networks: Attacks and defenses. In: Proceedings of ESORICS 2006. (September 2006)

4. Bauer, K., McCoy, D., Grunwald, D., Kohno, T., Sicker, D.: Low-resource routing attacks against Tor. In: Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2007), Washington, DC, USA (October 2007)
5. Houmansadr, A., Kiyavash, N., Borisov, N.: Rainbow: A robust and invisible non-blind watermark for network flows. In: NDSS, The Internet Society (2009)
6. Ling, Z., Luo, J., Yu, W., Fu, X., Xuan, D., Jia, W.: A new cell counter based attack against Tor. In: CCS '09: Proceedings of the 16th ACM conference on Computer and communications security, ACM (2009) 578–589
7. Murdoch, S.J., Zielinski, P.: Sampled traffic analysis by Internet-exchange-level adversaries. In Borisov, N., Golle, P., eds.: Privacy Enhancing Technologies. Volume 4776 of Lecture Notes in Computer Science., Springer (2007) 167–183
8. Back, A., Möller, U., Stiglic, A.: Traffic analysis attacks and trade-offs in anonymity providing systems. In Moskowitz, I.S., ed.: Proceedings of Information Hiding Workshop (IH 2001), Springer-Verlag, LNCS 2137 (April 2001) 245–257
9. Borisov, N., Danezis, G., Mittal, P., Tabriz, P.: Denial of service or denial of security? How attacks on reliability can compromise anonymity. In: Proceedings of CCS 2007. (October 2007)
10. Danner, N., Krizanc, D., Liberatore, M.: Detecting denial of service attacks in Tor. In: Financial Cryptography and Data Security, Berlin, Heidelberg, Springer-Verlag (2009) 273–284
11. Tran, A., Hopper, N., Kim, Y.: Hashing it out in public: Common failure modes of DHT-based anonymity schemes. In: ACM Workshop on Privacy in Electronic Society. (2009)
12. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. In: Proceedings of the 13th USENIX Security Symposium. (August 2004)
13. Dingledine, R., Mathewson, N.: Tor protocol specification. https://www.torproject.org/svn/trunk/doc/spec/tor-spec.txt
14. Øverlier, L., Syverson, P.: Locating hidden servers. In: Proceedings of the 2006 IEEE Symposium on Security and Privacy, IEEE CS (May 2006)
15. Snader, R., Borisov, N.: EigenSpeed: Secure peer-to-peer bandwidth evaluation. In: Proceedings of the 8th International Workshop on Peer-to-Peer Systems (IPTPS). (2009)
16. Perry, M.: TorFlow: Tor network analysis. http://fscked.org/talks/TorFlow-HotPETS-final.pdf
17. Dingledine, R., Mathewson, N.: Tor path specification. https://git.torproject.org/checkout/tor/master/doc/spec/path-spec.txt
18. Dingledine, R., Mathewson, N.: Tor directory protocol, version 3. https://www.torproject.org/svn/trunk/doc/spec/dir-spec.txt
19. Murdoch, S.J., Watson, R.N.M.: Metrics for security and performance in low-latency anonymity systems. In: Proceedings of the Eighth International Symposium on Privacy Enhancing Technologies (PETS 2008), Leuven, Belgium (July 2008)
20. McCoy, D., Bauer, K., Grunwald, D., Kohno, T., Sicker, D.: Shining light in dark places: Understanding the Tor network. In: Proceedings of the 8th Privacy Enhancing Technologies Symposium. (July 2008)
21. Loesing, K., Murdoch, S., Dingledine, R.: A case study on measuring statistical data in the Tor anonymity network. In: Workshop on Ethics in Computer Security Research. (January 2010)
22. Edman, M., Syverson, P.F.: AS-awareness in Tor path selection. In: Proceedings of the 2009 ACM Conference on Computer and Communications Security (CCS). (2009) 380–389

23. Tor metrics portal: Data. http://metrics.torproject.org/data.html#stats
24. Shannon, C.: A Mathematical Theory of Communication. In: Bell System Technical Journal. Volume 27. (1948) 379–656
25. Loesing, K.: Measuring the Tor network: Evaluation of client requests to the directories. Tor Project Technical Report (June 2009)
26. Wright, M.K., Adler, M., Levine, B.N., Shields, C.: The predecessor attack: An analysis of a threat to anonymous communications systems. ACM Trans. Inf. Syst. Secur. **7**(4) (2004) 489–522
27. Abbott, T., Lai, K., Lieberman, M., Price, E.: Browser-based attacks on Tor. In: Privacy Enhancing Technologies Symposium. Volume 4776 of Lecture Notes in Computer Science., Springer (2007) 184
28. Wendolsky, R., Herrmann, D., Federrath, H.: Performance comparison of low-latency anonymisation services from a user perspective. In Borisov, N., Golle, P., eds.: Proceedings of the Seventh Workshop on Privacy Enhancing Technologies (PET 2007), Ottawa, Canada, Springer (June 2007)
29. McCoy, D., Bauer, K., Grunwald, D., Tabriz, P., Sicker, D.: Shining light in dark places: A study of anonymous network usage. University of Colorado Technical Report CU-CS-1032-07 (August 2007)
30. Dingledine, R., Mathewson, N.: Anonymity loves company: Usability and the network effect. In: Workshop on the Economics of Information Security. (June 2006)
31. Syverson, P., Tsudik, G., Reed, M., Landwehr, C.: Towards an analysis of onion routing security. In Federrath, H., ed.: Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability, Springer-Verlag, LNCS 2009 (July 2000) 96–114
32. Snader, R., Borisov, N.: A tune-up for Tor: Improving security and performance in the Tor network. In: Proceedings of the Network and Distributed Security Symposium (NDSS), IEEE (February 2008)
33. Sherr, M., Blaze, M., Loo, B.T.: Scalable link-based relay selection for anonymous routing. In: 9th Privacy Enhancing Technologies Symposium (PETS '09). (August 2009)
34. Köpsell, S.: Low latency anonymous communication - How long are users willing to wait? In: ETRICS. (2006) 221–237
35. JAP. http://anon.inf.tu-dresden.de
36. Reardon, J., Goldberg, I.: Improving Tor using a TCP-over-DTLS tunnel. In: Proceedings of the 18th USENIX Security Symposium. (August 2009)