Don't Clog the Queue! Circuit Clogging and Mitigation in P2P Anonymity Schemes

Jon McLachlan and Nicholas Hopper

Computer Science & Engineering, University of Minnesota, Minneapolis MN 55455 USA
{hopper,jmcla}@cs.umn.edu

Abstract. At Oakland 2005, Murdoch and Danezis described an attack on the Tor anonymity service that recovers the nodes in a Tor circuit, but not the client. We observe that in a peer-to-peer anonymity scheme, the client is part of the circuit and thus the technique can be of greater significance in this setting. We experimentally validate this conclusion by showing that "circuit clogging" can identify client nodes using the MorphMix peer-to-peer anonymity protocol. We also propose and empirically validate the use of the Stochastic Fair Queueing discipline on outgoing connections as an efficient and low-cost mitigation technique.

1 Introduction

Anonymous communication schemes allow their users to communicate with others while concealing who communicates with whom. Deployed anonymity schemes tend to be either high-latency – they provide strong anonymity, but are unsuitably slow for casual Internet browsing – or low-latency, aiming for anonymity against weaker adversaries, but better-suited for casual Internet browsing. The majority of these schemes forward end-user traffic though redirecting relays, using multi-layered encryption between the source and every intermediate relay to ensure both end-to-end privacy and message size unity. This layered encryption creates virtual "anonymous tunnels," or circuits running through N relay nodes. The ultimate node in the anonymous tunnel performs the traditional Internet communication on behalf of the end-user. Returning messages then follow the reverse path through the same N relays, arriving at the source of the anonymous tunnel.

Such low-latency anonymity schemes can be further categorized as either *centralized* systems, in which a relatively small number of nodes act as "servers" that form the tunnels, or *peer-to-peer* systems, in which every end system may act as a relay for any other user. Example centralized systems include AN.ON [1], where tunnels are further constrained to follow one of a few paths through the central relays, and Tor [7], where currently 1000 servers act as relays for an estimated 100,000 users. Example peer-to-peer anonymity schemes include Crowds [22], Tarzan [10], Salsa [19] and Morph-Mix [23]. While peer-to-peer schemes offer the potential to scale more easily to a large user-base, it is interesting to note that to date, only centralized schemes are widely deployed.

Despite having a relatively weak security goal, a variety of attacks against low-latency anonymity schemes have been proposed [2,12,5,21,18,17]. Perhaps the simplest

G. Tsudik (Ed.): FC 2008, LNCS 5143, pp. 31–46, 2008.

is the "clogging" attack proposed by Back *et al.* [2], which we anachronistically call "relay clogging:" an adversary at one end of an anonymous tunnel can determine which nodes participate in the tunnel by sequentially "clogging" each of the possible relays (either within the protocol or at the network level) and looking for a corresponding drop in the throughput across the tunnel. This attack seems unavoidable under the current Internet architecture, but is also quite expensive in terms of the time and bandwidth required.

Murdoch and Danezis [17] proposed and empirically validated a dramatically lower-cost form of this attack, that we call "circuit clogging." In this attack, a malicious server alternately "clogs and unclogs" an anonymous tunnel, while measuring the latency of simple "timing" connections that run across all relays. The relays with latency functions most strongly correlated to the clogging periods are identified as the members of the tunnel. This attack works because in most low-latency schemes, relays divide their outgoing bandwidth among active tunnels, rather than across all tunnels.

In this paper, we present 3 significant contributions. First, we empirically corroborate the effectiveness of the circuit clogging attack against a prototype P2P scheme, MorphMix [23]. Second, building on the observation that in a P2P environment, the client is one of the relays, we empirically demonstrate that circuit clogging in a P2P system can be used to identify the client end of an anonymous tunnel. This makes the attack much more powerful, because it yields direct information about clients, and also because it allows a malicious server to take advantage of prior information about suspected clients: if 10 nodes are suspected as clients, the server can run timing connections across just those 10 nodes to confirm which one is the client. Finally, we propose and empirically validate a strategy for mitigating the circuit clogging attack while maintaining efficiency.

The problem of mitigating the circuit clogging attack represents an interesting optimization problem involving tradeoffs between efficient resource usage and vulnerability to several related attacks. The attack works because of interference between tunnels running through a relay, suggesting several possible mitigations. For example, one solution is for each relay to strictly divide its resources into N time slots or buckets, and map each circuit to a unique bucket; unused buckets represent wasted resources, while a relay with all buckets full would turn away additional circuits. With the possible exception of minor effects due to the scheduling code, this solution eliminates any interference between circuits, but uses system resources very inefficiently, and opens the way for new attacks. An adversary can essentially perform the $N-1$ [24] attack by keeping all but one of the buckets in each relay full, then intermittently testing whether an additional circuit succeeds; or the adversary can simply deny service by filling all buckets in all relays.

Another seemingly obvious alternative is to let relays accept any number of circuits but to equally allocate resources between all circuits, busy-waiting when a given circuit is inactive. This solution is less resource-wasteful in a friendly environment than the previous, but makes the relay-clogging attack very simple: to test whether a relay is involved in a tunnel, we can build a few additional circuits through the relay and watch for a corresponding drop in circuit throughput.

Our proposed mitigation mechanism is based on an existing queueing discipline, *Stochastic Fair Queueing*, which assigns circuits randomly to one of a small number of queues, which each receive equal service. Service within a queue is divided equally

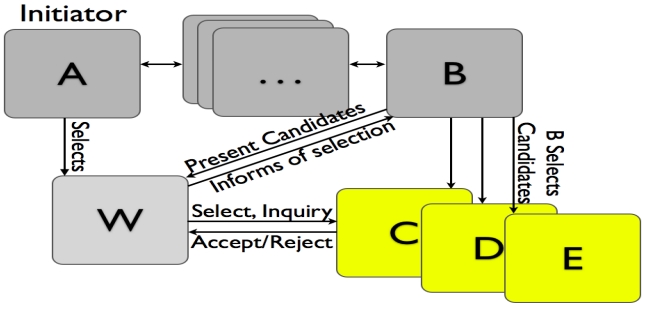


Fig. 1. Simplified MorphMix Next-Hop Selection

among just the active circuits. When the number of queues is set correctly, this scheme does not waste system resources and does not facilitate the $N - 1$ or relay clogging attacks; and since any given circuit will intersect with a target circuit with small probability, it limits the amount of interference an attacker can cause with a single circuit. Our experiments show that while an attacker can still mount a circuit clogging attack using what we call an "N-probe" variant, the costs of mounting this attack are significantly increased, and can be controlled as a function of the number of queues in a relay.

In the remainder of this paper, we discuss the MorphMix P2P anonymity scheme and related work in section 2 and the results of our circuit clogging experiments on MorphMix in section 3. We then present our SFQ mitigation mechanism and experiments analyzing its effectiveness and efficiency in section 4, along with experimental analysis of a possible counterattack. Finally we discuss implications of these findings in section 5.

2 Background

Morphmix. Our experiments were conducted against the Java implementation of the MorphMix P2P layered encryption anonymity scheme [23], selected mainly for the existence of a working prototype. Abstracting away the details of encryption algorithms and packet formats, the main distinguishing characteristics of such schemes are the mechanisms for peer discovery and circuit construction.

In MorphMix, these mechanisms are closely related. A morphmix node joins the network by "bootstrapping" from any existing node by sending a request to initiate an encrypted link. The node responds with a list of the other nodes it is connected to. This process can be repeated to discover a small set of nodes. Thereafter an honest node learns about new nodes in the process of circuit construction, shown in Figure 1. When extending a circuit, the node currently at the end of the circuit provides a list of several of its neighbors to a "witness" node chosen by the initiator. The witness node chooses the next hop and also returns the list of neighbors through the tunnel. Thus, a node never needs to know about more than a few other peers, but through repeated connection attempts and tunnel building, can quickly learn many nodes.

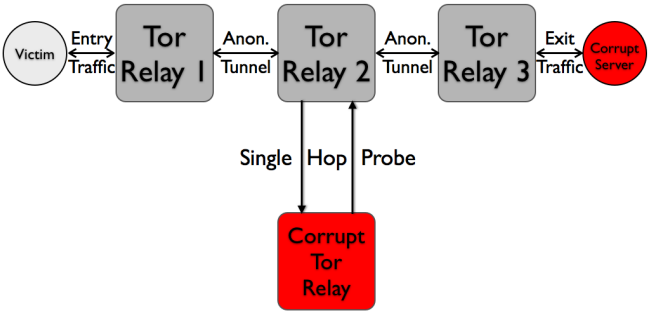


Fig. 2. Murdoch and Danezis Experiment

Since there is, by design, no admission control in MorphMix, a sybil attack [8] or other large-scale collusions are an obvious concern, especially given that random nodes assist in the construction of a circuit. Thus MorphMix includes a Collusion Detection System which attempts to identify nodes that seem to prefer selecting each other as next hops in anonymous tunnel creation. This scheme, which is essentially irrelevant to our paper, is based on locally collected statistics about the behavior of other nodes.

As with most low-latency anonymity schemes, MorphMix is mainly concerned with local active adversaries. In other words, they aim to provide anonymity against localized eavesdroppers, packet injections, packet replays, and most importantly against prying recipients of anonymous traffic. These adversaries are assumed to control only a few nodes within the MorphMix network. The circuit clogging attack (on Tor, and on MorphMix) falls within this threat model, motivating the need to mitigate this vulnerability in not only MorphMix, but in all low-latency schemes.

Related Work. As discussed above, Murdoch and Danezis [17] introduced the circuit clogging attack and measured its effectiveness against Tor. The basic setup of their experiment is illustrated in Figure 2: a "probe" machine, running a corrupt Tor node, initializes a circuit of length one – looping back to the probe machine – through each Tor node. The victim connects to the corrupt server through a circuit, and the server modulates its response in 30-60 second "on" periods of high data rate, followed by 30 second "off" periods with no outgoing data. The latency of packets sent through the "probe" tunnels was then correlated against the on/off period of the corrupt server. In their experiments, they correctly identified 11/13 true positives, with no false positives. Once the nodes in a Tor circuit are identified, they can be used to link two connections from the same exit node. Recently, another paper [12] discussed a way to extend this attack, using circuit latency measurements, to learn some information about the network location of the client.

Tabriz and Borisov [27] derived a way to model the statistical correlations used by victim nodes to detect collusion and simulated the same statistical tracking within the colluding nodes themselves. This way, the colluding nodes were able to intelligently select other colluding nodes without overstepping their statistically inferred node selection "canaries," avoiding the CDM on victimized MorphMix nodes while still

undermining a substantial portion of the MorphMix network. This attack, however, requires several colluding morphmix nodes.

Several other timing-based attacks on low-latency anonymity schemes have been proposed. *Packet-counting* attacks [2,25,29,14,3] work by examining either the packet flux over several time periods or the inter-packet arrival times of a flow to find correlations between flows. This information can be used to identify flows where the adversary controls the first and last node in a circuit, or for a global observer to identify all flows. Several countermeasures have been performed, such as link padding and defensive dropping [14]; the extent to which these defeat an active adversary engaged in counting attacks is uncertain [5]. A similar attack can allow a local observer to identify anonymously downloaded websites by their packet-count fingerprints [11].

An explicitly stated goal of some P2P anonymity systems, such as Salsa [19], is to hide the complete list of participants in the scheme. This is to prevent *intersection attacks* [6,15], that work by observing the differences in network traffic based on a node's presence or absence in the anonymity protocol. Contrary to the suggestion of [28], we demonstrate that MorphMix does not effectively hide the list of participating nodes.

3 Attacking Morphmix

To evaluate the effectiveness of circuit-clogging against the MorphMix prototype, we ran a series of experiments against small MorphMix deployments. The first set, run against a small deployment on the PlanetLab [4] wide-area testbed, replicated the Murdoch and Danezis experiment and were used to measure the experiment's ability to distinguish between circuit relays, circuit initiators, and non-circuit nodes. The second set, conducted against a local lab deployment, measured the difficulty of finding the set of nodes in a morphmix network.

3.1 Wide-Area Experiments

Setup. In our first set of wide-area experiments, we performed two sets of twenty runs each using the setup shown in Figure 3. In this experiment, the *victim* node initiates

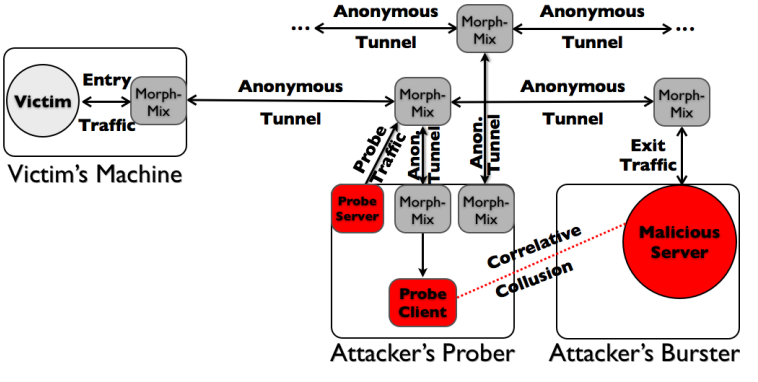
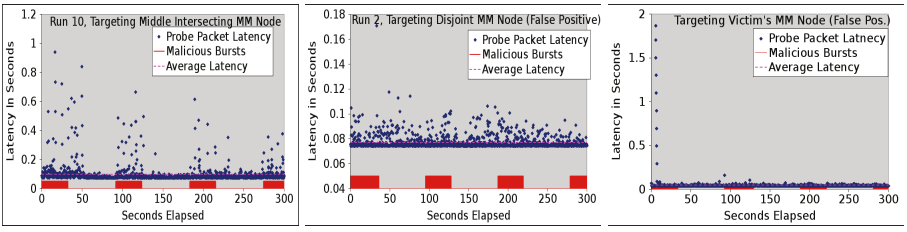


Fig. 3. Circuit clogging attack on MorphMix, probe targeting a middleman relay

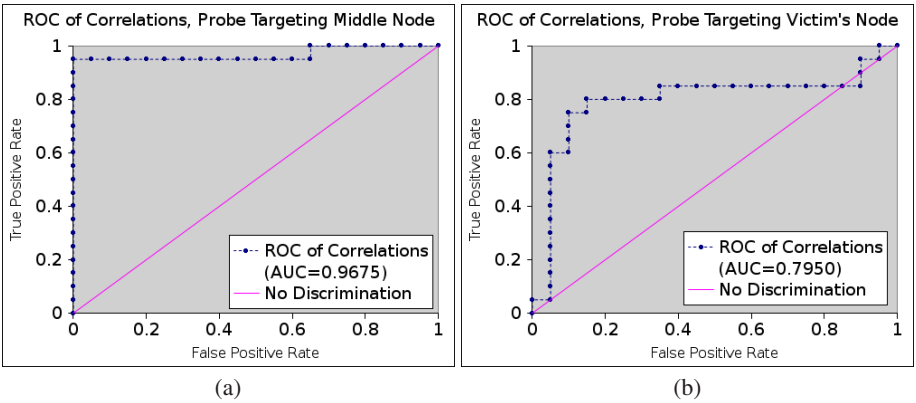


(a) (b) (c)

Fig. 4. Representative experiment results: (a) and (b) show probe latencies targeting a relay node, while (c) shows probe latencies targeting a disjoint node

a circuit of length three and connects to the (malicious) *burst server*. Meanwhile, the *probe server* builds two *probe circuits* that each run in a separate MorphMix instance and connect to one external MorphMix instance. The first circuit passes through an *intersecting* node in the victim's circuit, while the second passes through a *disjoint* node outside of the victim's circuit. These probe circuits then connect back to the probe server, which sends a short (64 byte) packet every 0.2 seconds to measure the latency of the probe circuits. In *middleman* runs, the intersecting morhpmix instance is the third relay in the circuit, while in *victim* runs, the intersecting instance runs on the victim node. Finally, the burst server alternates between 30 second "on periods," during which it sends a 10KB payload every 25ms, and 60 second "off periods," where it sends no data. Each run, testing two nodes, ran for 5 minutes.

Results and Analysis. Figure 4 shows the results of three runs. Fig. 4(a) shows the expected results — when targeting a relay in the circuit, latency increases dramatically during the burst server's "on" periods compared to the "off" periods. Fig. 4(b) shows a "false negative" run — latency remains quite high during the off periods. This may well be due to the fact that we selected the burst server load by measuring the average



(a) (b)

Fig. 5. ROC plots for MD correlation of (a) middleman and (b) victim relays

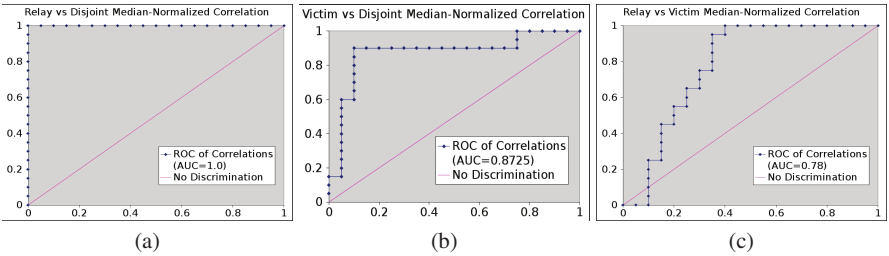
throughput capacity of morphmix instances on our planetlab nodes. Fig. 4(c) shows a "false positive", e.g. the probe latencies of a disjoint node that show high correlation to the burst signal. It seems clear that the high correlation here is due to an unrelated spike in the load of the "disjoint" planetlab node.

To determine whether a node was in a relay, Murdoch and Danezis computed the correlation of its normalized latency signal $L'(t)$ with the burst server's binary template signal $S(t)$, i.e. the correlation is calculated as $\frac{\sum_t S(t) \times L'(t)}{\sum_t S(t)}$, and nodes with "high" correlation are classified as intersecting the circuit. Setting different thresholds of classification will yield different rates of true and false positives; varying this threshold produces the receiver operating characteristic (ROC) curve for this correlation function. Figure 5 shows the resulting curves for the victim and middleman nodes. Following [9], we summarize an ROC curve by its AUC (Area Under Curve), where 0.5 indicates a non-distinguishing classifier and 1.0 indicates a perfect classifier. In our experiments, the MD correlation function had AUC 0.9675 when targeting a middleman node and 0.795 when targeting a victim node. This suggests that circuit clogging can be applied to effectively confirm suspected circuit initiators in MorphMix, and even more effectively to identify relays.

Motivated by several of the anomalies shown in figure 4, we also calculated the correlation when probe latencies are normalized with respect to the median probe latency, rather than the average. Figure 6 shows the impact of this change. The AUC for runs targeting the middleman node increases to 1.0, while the AUC for runs targeting the victim node increases to 0.87, with an equal error rate of 10%. Fig. 6 also shows that the magnitude of the median-normalized correlation is a good indicator of whether an intersecting node is a victim or middleman.

3.2 Node Discovery Attack

Although finding all nodes in a morphmix network is less critical given that suspected clients can be monitored directly, the full circuit clogging attack depends to some extent on building a list of currently participating nodes. Although it is not an explicit goal of MorphMix to hide the participant list, we performed a simple experiment to measure an attacker's ability to discover this information. In our experiment, we first start, and



(a) (b) (c)

Fig. 6. ROC plots for median-normalized correlation of (a) middleman vs disjoint,(b) victim vs. disjoint, and (c) victim vs middleman morphmix instances

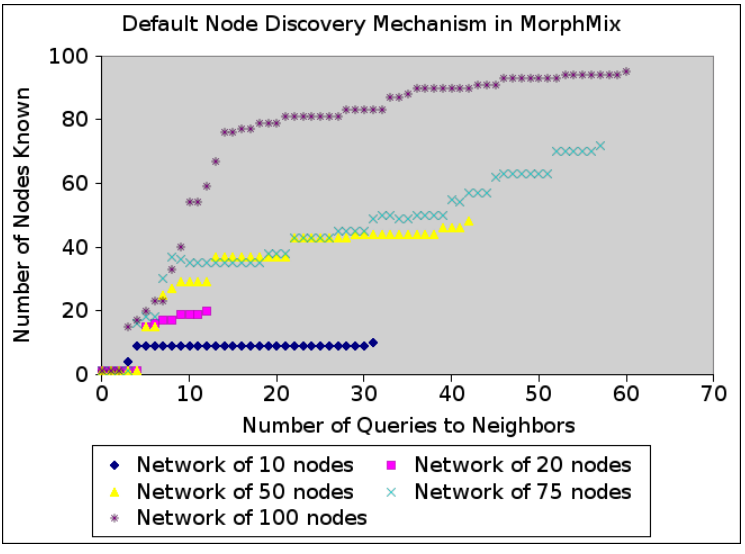


Fig. 7. Results of Node Discovery experiment for $N \in 10, 25, 50, 75, 100$: experiments halted when 95% of peers were found

allow to stabilize, a network of N nodes. Then we start our "attacker" node, which starts off with a list consisting of a single known node. The attacker node repeatedly picks a random node from its list, contacts that node with a link request, and adds any neighbors of that node to its list. Note that this search can be easily distributed to multiple nodes.

Using the default configuration of the MorphMix protocol, a stabilized client will always return at least four nodes in response to a link request. If we assume that the neighbor list of each MorphMix node is uniformly chosen, and that the initial node of the attacker is uniformly chosen, then the probability, over all choices of neighbors, initial nodes, and sequences of contacted nodes, of contacting a node at a given step is $1/N$. Thus, we heuristically estimate that the expected number of steps needed to contact all nodes is $N \ln N/4$. We note that the number of lookups needed could be further reduced by a factor of $\ln N/4$ by having the attacker implement a breadth-first search of the network, at the expense of a slight increase in the amount of coordination required among multiple attacker nodes.

Figure 7 shows the results of our experiments for various network sizes. In nearly all cases, our attacker was able to identify 95% of the peers in the network by sending fewer than N node request messages. This supports our contention that MorphMix does not effectively hide the list of peers from an adversarial node.

4 Stochastic Fair Queue Mitigation Mechanism

The main reason for the success of circuit clogging is the fact that in most low-latency anonymity schemes, circuits *interfere* with each other: the amount, and timing, of traffic

in one circuit can influence the service received by another. In particular, the available throughput of a node is typically divided between active circuits, in order to make good use of this throughput. The side effect of this decision is that the activity of on circuit influences the queueing time for other circuits. As Murdoch and Danezis argue, this interference is effectively a covert channel between the circuits in a relay.

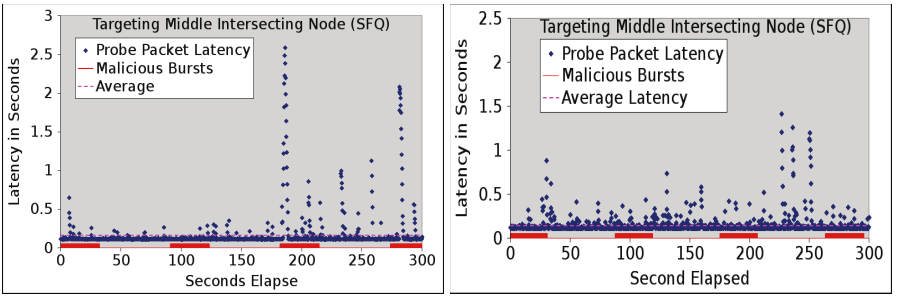
This covert channel can be mitigated to some extent by enforcing fairness among all circuits at the application layer - if each circuit has an equal share of resources regardless of activity, then from the application's point of view, the circuits do not interfere with each other. Unfortunately, this approach may not make efficient use of the resources available to the application: if we reserve service for circuits that do not use that service, we incur a performance penalty compared to servicing only active circuits. Furthermore, the use by the application of shared resources may results in exploitable "covert channels" at other levels. While we expect that the interference due to factors like shared operating system and processor time will be minimal, we note that relay clogging can exploit interference at the network level by denying service to a relay and checking whether service on the circuit improves or stays the same. This attack seems unavoidable at the application level, though we note that it is much more costly to mount this attack.

In the MorphMix prototype that we evaluated, messages for all circuits passing through a relay are processed through a single, central, first-in/first-out queue. Our mitigation technique is to simply change the scheduling policy for messages in this queue, reducing the extent to which circuits interfere. We replaced this FIFO outgoing queue with a Stochastic Fair Queue (SFQ) [16], a queueing discipline originally designed for routers to relieve network congestion due to ill-behaved UDP traffic. In the remainder of this section, we describe SFQ and its properties in more detail, give empirical evidence that SFQ reduces the effectiveness of circuit clogging while incurring only a small performance penalty, and consider alternative, more costly attacks on SFQ-enhanced morphmix.

4.1 Basic Properties of SFQ

The SFQ policy probabilistically fairly distributes resources over a set M of circuits passing through a relay. It randomly maps each $m \in M$ to one of q queues for the entire life-span of the circuit m. Each of these q queues receives exactly $\left(\frac{1}{q}\right)$ fraction of the available throughput. If a particular queue has no active circuits, it still receives $\left(\frac{1}{q}\right)$ of the available throughput; in our implementation this was accomplished through busy waiting for a time period determined by a running average for all live message processing times. Within a queue, messages were still processed in a first-in/first-out fashion, so that circuits received service within a queue proportionally to their level of activity.

The parameter q determines how effectively a SFQ-enabled morphmix node utilizes its resources. If there are active circuits in some number $n < q$ of queues, then performance is degraded by a factor of $\frac{q}{q-n}$ while servicing the $q - n$ inactive queues. If we set q to be about half the expected number of active circuits passing through a relay, then we find that the probability that a queue is inactive is $(1 - \frac{1}{q})^{2q} \approx e^{-2}$, so that the expected performance degradation is $\frac{e^2}{e^2-1} \approx 1.16$.



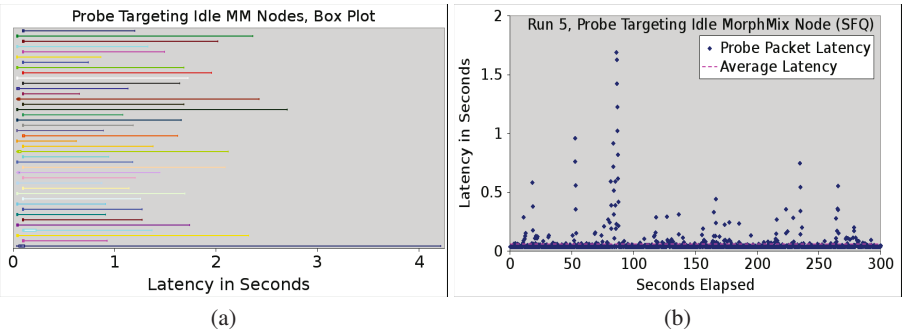
(a) (b)

Fig. 8. Circuit clogging results for SFQ-enabled MorphMix: (a) a true positive experiment (b) a false negative experiment

The number q of queues also impacts the extent to which SFQ mitigates the circuit clogging attack. In particular, an attacker-controlled circuit interferes with a given circuit with probability $1/q$. Thus, in the circuit clogging attack, there is a $1/q$ probability that the probe circuit and the burst circuit will interfere. This gives the probe server a $1/q$ advantage in distinguishing nodes that intersect the circuit from disjoint nodes. In our experiments with SFQ-enhanced morphmix, we set $q = 8$, so we expect that a probe server should be able to correctly identify an intersecting node 62.5% of the time.

4.2 Circuit Clogging Mitigation

To test the effectivness of SFQ in mitigating the circuit clogging attack, we performed another series of 20 runs on a planetlab deployment identical to the experiments described in Section 3.1, replacing every MorphMix instance with an instance of SFQ-enabled MorphMix. Figure 8 shows two representative experimental results. In Fig. 8(a), we see the result of an experiment where the probe circuit collided with the burst circuit, resulting in a high correlation, while Fig. 8(b) shows the more common result,



(a) (b)

Fig. 9. Probe latencies with no burst server present: (a) Box Plot of all probe times, 20 probes total. (b) Typical run showing several latency spikes.

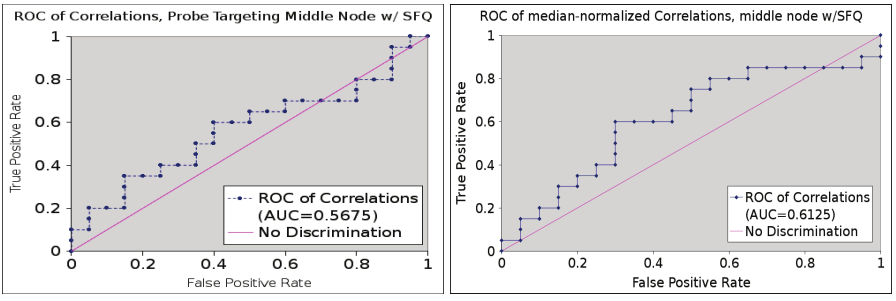


Fig. 10. SFQ circuit clogging: ROC plots for (a) MD correlation of relay vs disjoint nodes and (b) median-normalized correlation of relay vs disjoint nodes

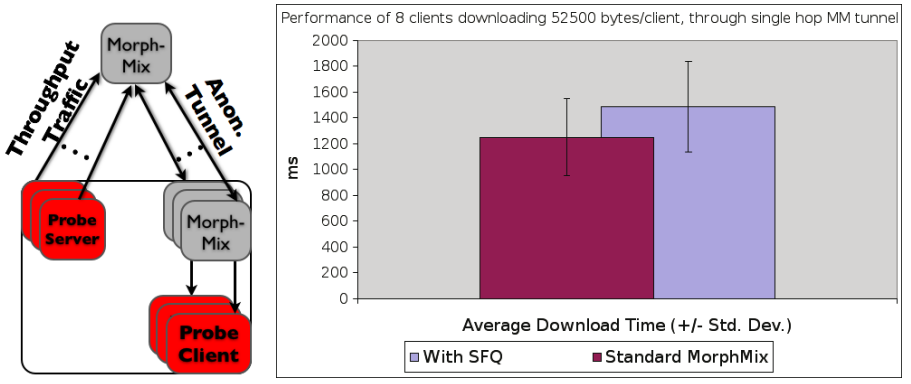
when the probe circuit was mapped to a different queue than the burst circuit and so had effectively no interference.

One observation about both of the runs shown in Figure 8 is the "noise" caused by large latency peaks during off periods. We speculated that these spikes resulted from the interaction of MorphMix's Java implementation with the PlanetLab shared execution environment. To highlight this, we ran an extra set of 20 experiments with the probes targeting only idle MorphMix node with SFQ. This batch was identical to the other SFQ batch, except that there was no victim node or burst server. Figure 9 summarizes the variability in probe latencies present even with no burst server. This illustrates the need for robust normalization.

Results. When all nodes composing the P2P network (except for the two controlled by the probe server) utilized SFQ, we notice a significant mitigation of the attack. The average MD correlation with the probe targeting an node intersecting the victim's tunnel was 1.0154 with standard deviation of 0.1071 and the average MD correlation with the probe targeting a disjoint node was 0.9789 with standard deviation 0.16241. Using median-normalized correlations, we see an average correlation of 1.246 targeting intersecting nodes and 1.191 for disjoint nodes, with standard deviations of 0.218 and 0.170, respectively. ROC plots for both correlation figures are shown in figure 10. As expected, the AUC for these measures is only slightly better than the line of no discrimination: 0.5675 in the case of MD correlations, and 0.6125 when using median-normalized correlation.

4.3 Efficiency of SFQ Mitigation

To test the performance penalty of running SFQ on a morphmix relay, we performed a set runs on a small planetlab deployment of MorphMix. In each run of the experiment, shown in Figure 11(a), the "client" machine started 8 standard MorphMix instances, and the "server" machine started either a SFQ-enhanced MorphMix instance or a standard MorphMix instance. The client instances each built a one-hop circuit through the server instance, and connected back to the client machine to download a 52KB file. The time required for all 8 instances to complete the download (not including circuit



(a) (b)

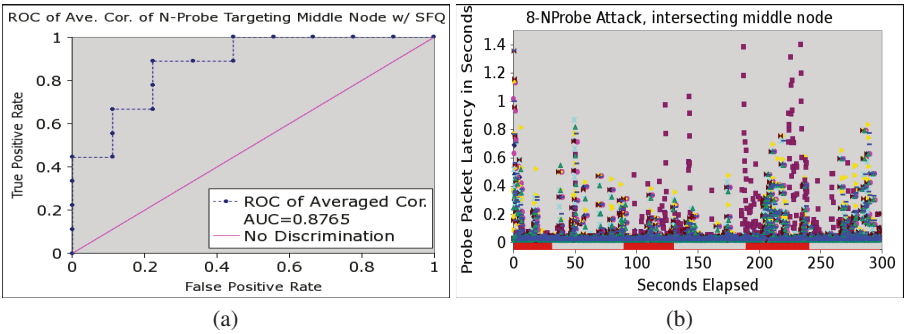
Fig. 11. Performance comparison for SFQ-enhanced and standard morphmix: (a) panetlab experimental setup, and (b) results for 3 runs of each type

construction) was recorded. In all, 3 runs with a standard MorphMix relay and 3 runs with a SFQ-enhanced relay were performed.

The results of the experiment are shown in Figure 11(b). With 8 queues and 8 active circuits, we expect to see performance degraded by a factor of $\frac{e}{e-1} \approx 1.6$. In our experiments, the performance was slightly better: average download time for standard MorphMix was 1250 ms with a standard error of 61ms, while average download time for SFQ was 1488ms, with a standard error of 71ms, giving an estimated performance penalty of 19%, rather than 60%. This suggests that, with the proper settings, SFQ can be an efficient mitigation technique for the basic circuit clogging attack.

4.4 N-Probe Attack

Given an anonymous network utilizing SFQ initialized to q slots, a clever adversary could initialize N probe circuits, rather than one, per relay. This increases the



(a) (b)

Fig. 12. Results of 8-probe attack on 8-queue SFQ: (a) ROC of maximum MD correlation, relay vs disjoint nodes; (b) typical result targetting a relay node, all 8 probes included.

probability that *some* probe circuit will intersect the queue supporting the the burst circuit. This way, the burst traffic and the probe traffic collide with each other as if they were sharing the same standard queue. Given q SFQ slots in neighboring MorphMix nodes, N probe connections into a single neighbor, and one "target" SFQ slot containing the modulating server's stream, the probability of an adversary intersecting the target SFQ slot with at least one of the N probe connections is $\rho(q, N) = 1 - \left(\frac{q-1}{q}\right)^N$. Thus an adversary running N probe circuits should ideally have advantage roughly $1/2 + \rho(q, N)/2$ at distinguishing relay intersecting from disjoint MorphMix nodes.[1]

To test the effectiveness of this attack, we performed another set of 9 runs each against our PlanetLab deployment of SFQ-enabled MorphMix. We used the same experimental setup as in section 4.2, including the use of 8 queues, except that the probe server ran 8 separate probing MorphMix instances (so that eight one-hop circuits ran through the targeted relay). Since the probe server had no way to determine a priori *which* of the 8 probe circuits intersected with the burst circuit, we calculated the MD correlation of the latencies of each circuit and took the maximum correlation as the summary statistic for a node. Figure 12(a) shows the ROC curve resulting from this experiment, while Fig 12(b) shows results of a typical run. The average max-MD correlation for an intersecting node was 2.99 with a standard error of 0.90, while the average max-MD correlation for a disjoint node was 1.16, with a standard error of 0.17. The AUC for the experiment was 0.8765.

These results suggest that the N-probe circuit clogging attack can potentially be effective against SFQ. We note, however, that the parameters $N = 8$, $q = 8$, were selected somewhat arbitrarily. In an active morphmix deployment, we might expect each node to initiate, e.g., 8 circuits (the default number of concurrent connections in Mozilla-based web browsers), each spanning 5 relays (the default circuit length), so that on average each MorphMix node would support 40 active circuits. In this case, using 20 queues would require the probe attacker to run 20 probes. In our experiments, the downstream bandwidth required per probe was 27Kbps, and the upstream bandwidth required per probe was 10Kbps. Thus the bandwidth cost of running 20 probes through a single relay would be 540Kbps downstream and 200Kbps upstream. Several measurement studies suggest that the average download capacity of an internet user is roughly 500Kbps [20,13]; thus enabling SFQ makes the cost of circuit clogging roughly comparable to the cost of relay clogging.

5 Discussion

In this paper, we have confirmed that the Murdoch and Danezis circuit clogging attack can be applied to identify circuit initiators in the MorphMix P2P anonymity scheme. This makes the attack much more powerful, for two reasons: first, by directly identifying victims, the attack obtains much more valuable information than when it is deployed

[1] This analysis assumes that an adversary cannot detect when two probes are in the same queue; an adversary who can do so can improve the probability $\rho(q, N)$ to N/q by ensuring that his probes are in disjoint buckets. Note that it is unclear how effectively this can be done in practice

against Tor; second, this enables confirmation attacks, where only a small set of suspect nodes need to be monitored, rather than all nodes in the network.

In response to this observation, we proposed a novel application of an existing queueing policy, Stochastic Fair Queueing, to mitigate circuit clogging attacks. We have demonstrated empirically that enforcing SFQ on outgoing circuit links effectively mitigates the basic circuit clogging attack, and significantly increases the cost of a more sophisticated approach. Combined with recent results of Shmatikov and Wang [26], we believe that this indicates that understanding the security implications of different queueing policies in low-delay anonymity schemes may be an interesting direction for further research.

One interesting question that we have not fully addressed is how the N-probe circuit clogging attack will be affected by the presence of multiple active circuits on a relay. In our experiments, both the intersecting and disjoint relays supported no active circuits other than those created by the attack - probe circuits in both cases and the burst circuit in the case of the intersecting relay. In a sense, this is a best-case scenario for the N-probe attack, since other than random delay spikes, probes passing through a disjoint node will never intersect another circuit, causing all probe latencies to be close to the average and median. When another active circuit intersects with a probe circuit, probes may have higher than average latencies during some on-periods, leading to greater maximum correlations in disjoint nodes. This in turn would likely imply a greater false positive rate for a given correlation threshold.

Another interesting direction for future work is to apply the SFQ scheduling policy to circuits in Tor, mitigating the original circuit clogging attack as proposed by Murdoch and Danezis. Since each Tor node supports many clients, the expected number of active circuits in a Tor node may be quite high, possibly allowing a relay to support hundreds of queues. This leads us to expect that stochastic fair queueing may in fact make relay clogging – via network-level denial of service – a more attractive option than circuit clogging in the Tor network.

Acknowledgements. The authors wish to thank Roger Dingledine, Yongdae Kim, Marc Rennhard, Eugene Vasserman, and the anonymous FC reviewers for helpful discussions and comments about this work. This work was supported by the University of Minnesota's Undergraduate Research Opportunities Program and by the NSF under grant CNS-0546162.

References

1. AN.ON: Anonymity online, `http://anon.inf.tu-dresden.de/`
2. Back, A., Möller, U., Stiglic, A.: Traffic analysis attacks and trade-offs in anonymity providing systems. In: Moskowitz, I.S. (ed.) IH 2001. LNCS, vol. 2137, pp. 245–257. Springer, Heidelberg (2001)
3. Blum, A., Song, D., Venkataraman, S.: Detection of Interactive Stepping Stones: Algorithms and Confidence Bounds. In: Jonsson, E., Valdes, A., Almgren, M. (eds.) RAID 2004. LNCS, vol. 3224, pp. 258–277. Springer, Heidelberg (2004)
4. Chun, B., Culler, D., Roscoe, T., Bavier, A., Peterson, L., Wawrzoniak, M., Bowman, M.: Planetlab: an overlay testbed for broad-coverage services. SIGCOMM Comput. Commun. Rev. 33(3), 3–12 (2003)

5. Dai, W.: Two attacks against freedom
6. Danezis, G.: Statistical disclosure attacks: Traffic confirmation in open environments. In: Gritzalis, Vimercati, Samarati, and Katsikas (eds.) Proceedings of Security and Privacy in the Age of Uncertainty (SEC 2003), IFIP TC11, pp. 421–426. Kluwer, Dordrecht (2003)
7. Dingledine, R., Mathewson, N., Syverson, P.F.: Tor: The second-generation onion router. In: 13th USENIX Security Symposium (August 2004)
8. Douceur, J.: The sybil attack (2002)
9. Fawcett, T.: An introduction to ROC analysis. Pattern Recognition Letters 27(8), 861–874 (2006)
10. Freedman, M.J., Morris, R.: Tarzan: a peer-to-peer anonymizing network layer. In: CCS 2002: Proceedings of the 9th ACM conference on Computer and communications security, pp. 193–206. ACM Press, New York (2002)
11. Hintz, A.: Fingerprinting websites using traffic analysis. In: Dingledine, R., Syverson, P.F. (eds.) PET 2002. LNCS, vol. 2482, pp. 171–178. Springer, Heidelberg (2003)
12. Hopper, N., Vasserman, E.Y., Chan-Tin, D.: How much anonymity does network latency leak? In: Proceedings of CCS 2007 (October 2007)
13. Izal, M., Urvoy-Keller, G., Biersack, E.W., Felber, P., Al Hamra, A., Garces-Erice, L.: Dissecting BitTorrent: Five Months in a Torrents Lifetime. Passive and Active Measurements 2004 (2004)
14. Levine, B.N., Reiter, M.K., Wang, C., Wright, M.K.: Timing attacks in low-latency mix-based systems. In: Juels, A. (ed.) FC 2004. LNCS, vol. 3110, pp. 251–265. Springer, Heidelberg (2004)
15. Mathewson, N., Dingledine, R.: Practical traffic analysis: Extending and resisting statistical disclosure. In: Martin, D., Serjantov, A. (eds.) PET 2004. LNCS, vol. 3424. Springer, Heidelberg (2005)
16. McKenney, P.: Stochastic fairness queuing (1990)
17. Murdoch, S.J., Danezis, G.: Low-cost traffic analysis of tor. IEEE SP 00, 183–195 (2005)
18. Murdoch, S.J., Zieliński, P.: Sampled traffic analysis by internet-exchange-level adversaries. In: Borisov, N., Golle, P. (eds.) PET 2007. LNCS, vol. 4776, pp. 167–183. Springer, Heidelberg (2007)
19. Nambiar, A., Wright, M.: Salsa: a structured approach to large-scale anonymity. In: Proceedings of the 13th ACM conference on Computer and communications security, pp. 17–26 (2006)
20. Pouwelse, J.A., Garbacki, P., Epema, D.H.J., Sips, H.J.: The bittorrent p2p file-sharing system: Measurements and analysis. In: Castro, M., van Renesse, R. (eds.) IPTPS 2005. LNCS, vol. 3640, pp. 205–216. Springer, Heidelberg (2005)
21. Raymond, J.-F.: Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems. In: Federrath, H. (ed.) Designing Privacy Enhancing Technologies. LNCS, vol. 2009, pp. 10–29. Springer, Heidelberg (2001)
22. Reiter, M., Rubin, A.: Crowds: Anonymity for web transactions. ACM Transactions on Information and System Security 1(1) (June 1998)
23. Rennhard, M., Plattner, B.: Introducing MorphMix: peer-to-peer based anonymous Internet usage with collusion detection. In: WPES 2002: Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society, pp. 91–102. ACM Press, New York (2002)
24. Serjantov, A., Dingledine, R., Syverson, P.: From a trickle to a flood: Active attacks on several mix types. In: Petitcolas, F.A.P. (ed.) IH 2002. LNCS, vol. 2578, pp. 36–52. Springer, Heidelberg (2003)
25. Serjantov, A., Sewell, P.: Passive attack analysis for connection-based anonymity systems. In: Snekkenes, E., Gollmann, D. (eds.) ESORICS 2003. LNCS, vol. 2808, pp. 116–131. Springer, Heidelberg (2003)

26. Shmatikov, V., Wang, M.-H.: Timing analysis in low-latency mix networks: Attacks and defenses. In: Gollmann, D., Meier, J., Sabelfeld, A. (eds.) ESORICS 2006. LNCS, vol. 4189, pp. 18–33. Springer, Heidelberg (2006)
27. Tabriz, P., Borisov, N.: Breaking the collusion detection mechanism of morphmix. In: Danezis, G., Golle, P. (eds.) PET 2006. LNCS, vol. 4258, pp. 368–383. Springer, Heidelberg (2006)
28. Wiangsripanawan, R., Susilo, W., Safavi-Naini, R.: Design principles for low latency anonymous network systems secure against timing attacks. In: Proceedings of the fifth Australasian symposium on ACSW frontiers (ACSW 2007), pp. 183–191. Australian Computer Society, Inc., Darlinghurst (2007)
29. Wright, M., Adler, M., Levine, B.N., Shields, C.: Defending anonymous communication against passive logging attacks. In: Proceedings of the 2003 IEEE Symposium on Security and Privacy (May 2003)