Denial of Service or Denial of Security?

How Attacks on Reliability can Compromise Anonymity

Nikita Borisov
University of Illinois at Urbana-Champaign
1308 West Main St.
Urbana, IL 61801
nikita@uiuc.edu

George Danezis
K.U. Leuven, ESAT/COSIC
Kasteelpark Arenberg 10
B-3001 Leuven-Heverlee, Belgium
George.Danezis@esat.kuleuven.be

Prateek Mittal
University of Illinois at Urbana-Champaign
1308 West Main St.
Urbana, IL 61801
mittal2@uiuc.edu

Parisa Tabriz
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
parisa.tabriz@gmail.com

ABSTRACT

We consider the effect attackers who disrupt anonymous communications have on the security of traditional high- and low-latency anonymous communication systems, as well as on the Hydra-Onion and Cashmere systems that aim to offer reliable mixing, and Salsa, a peer-to-peer anonymous communication network. We show that denial of service (DoS) lowers anonymity as messages need to get retransmitted to be delivered, presenting more opportunities for attack. We uncover a fundamental limit on the security of mix networks, showing that they cannot tolerate a majority of nodes being malicious. Cashmere, Hydra-Onion, and Salsa security is also badly affected by DoS attackers. Our results are backed by probabilistic modeling and extensive simulations and are of direct applicability to deployed anonymity systems.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems

General Terms

Reliability, Security

Keywords

Anonymity, reliability, denial of service, attacks

1. INTRODUCTION

Research into anonymous communication has had a long history, starting with Chaum's seminal paper on mix networks [4]. The research involved both design of new systems, some of which have enjoyed moderately successful deployment, and analysis of existing designs. In both cases, the evaluation of anonymity systems has largely focused on the security of the systems—that is, how likely it is that anonymity is compromised—with other metrics considered tangential. Recent work, however, has started to address metrics such as performance, usability, and reliability [8, 23, 15, 18, 30], due to the fact that these "secondary" characteristics are in fact of primary importance: a system that is inefficient, unreliable, or unusable in some other way will cause users to take their communication to other, non-anonymous channels.

Reliability, however, has a subtler and previously unexplored connection with security. Instead of a blanket denial-of-service (DoS) attack, an adversary may *selectively* affect reliability of the system in those states that are hardest to compromise, thereby causing the system to enter less secure states. In particular, we explore an attack where DoS is performed whenever the communication cannot be compromised. Such selective DoS is both easier to carry out than an attack on the entire system, and can be more effective; instead of driving users away from the system, they are presented with a less reliable, but still functional system. Faced with poor reliability, many users (and a lot of software) will naturally attempt the communication again, presenting more opportunities for attack.

We analyze the success of this attack as applied to conventional low and high-latency anonymous systems. The low-latency systems are exemplified by Tor, a popular anonymous communications network, and we model high-latency systems as a mix network such as the Mixminion [6] anonymous remailer network. In both cases, we show that as more nodes are compromised, systems under a selective DoS attack grow much more vulnerable than conventional security analysis would suggest. In particular, we show for the first time a *fundamental limit* on the security of the traditional mix architecture: messages routed in a network with a majority of compromised nodes can be de-anonymized with high probability by an adversary performing DoS attacks. Previous work [3] challenged the established wisdom that a

CCS'07, October 29–November 2, 2007, Alexandria, Virginia, USA.

single honest node is sufficient for security, but not to the extent our results demonstrate.

We also examine two proposed systems designed to improve the reliability of mix networks: Cashmere [30] and Hydra-Onions [18]. We find that though both systems do improve reliability, they do so at the price of reduced security, especially in the face of DoS. Cashmere is particularly susceptible to DoS because its authors erroneously believed that DoS was not a security concern and developed no mechanisms to deal with it. Hydra-Onions prove more resistant, but offer poor security when a significant number of nodes are compromised.

Finally, we look at how selective DoS affects Salsa [21]. Salsa is a low-latency anonymous communication system that is based on a peer-to-peer design, with the goal of surpassing the scaling limits that are faced by network such as Tor, and so we can think of it as an example of what "next generation" anonymous networks might be like. However, to deal with uncertainty of the p2p context, Salsa relies on redundant lookups, and is therefore more vulnerable to the selective DoS attack than conventional systems.

In the following section, we analyze selective denial of service against traditional high and low-latency mix systems. Section 3 examines our selective DoS attack against anonymous communication systems engineered specifically for reliability. We discuss the vulnerability of Salsa to the selective DoS attack in Section 4. Section 5 discusses potential countermeasures to selective DoS and in Section 6, we conclude.

2. DENIAL OF SERVICE AGAINST CONVENTIONAL ANONYMITY SYSTEMS

2.1 Tor

The Tor network [9] is a widely used system for low-latency anonymous Internet communication. The network has enjoyed quick growth since its initial deployment in 2003; as of August 2007, Tor is composed of nearly 1000 active routers supporting hundreds of thousands of users [24].

Communication over Tor happens through *tunnels* that are sent via multiple Tor routers. The tunnels are constructed in a telescoping manner and are protected by layered encryption, so that each router only knows the previous and next routers forwarding the tunnel. However, the low-latency nature of the communication allows the first and last router in a tunnel to collude and easily discover that they are forwarding the same stream by matching packet timings. Therefore, under conventional analysis, if $\bar{t}$ is the fraction of all Tor routers that are compromised, then $\bar{t}^2$ is the probability that any individual tunnel will be compromised [27].

This analysis assumes that routers are picked at random. In practice, Tor picks routers in proportion to their advertised bandwidth; this may be modeled by setting $\bar{t}$ to be the fraction of total bandwidth controlled by the attackers. Recent versions of Tor also use guard nodes in building tunnels. We present the analysis of Tor as originally described in [9] and discuss guard nodes in Section 2.1.2.

2.1.1 Reliability Analysis

The reliability of Tor tunnels is straightforward to determine. A Tor tunnel that goes through l routers (l is typically 3) will fail if any of the routers fail. Therefore, if f is the probability of a router being reliable, $R = f^l$ is the probability of the entire tunnel being reliable.[1]

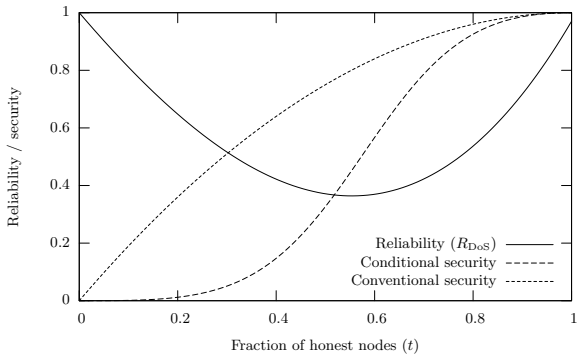


Figure 1: Reliability and security analysis of Tor under the selective DoS attack, with $f = 0.99$.

Though the Tor authors considered the selective denial of service attack in their threat model, they did not analyze its impact. Next, we analyze the effect of a selective DoS on Tor. We assume that dishonest routers will perform DoS on any tunnel they cannot compromise. This attack is easy to implement: if the adversary acts as a first or last router on a tunnel, the tunnel is observed for a brief period of time and matched against all other tunnels where a colluding router is the last or first router, respectively. If there is a match, the tunnel is compromised; otherwise, the adversary kills the tunnel by no longer forwarding traffic on it. The adversary also kills all tunnels where it is the middle node, unless both the previous and the next hop are also colluding. Alternately, Bauer *et al.* present an algorithm for linking tunnels even before traffic has been sent over them [1].

Under this attack, a tunnel is reliable only if the first and last nodes are compromised, or if it is composed of only reliable honest nodes. So the overall reliability of Tor in this case is:

$$R_{\mathrm{DoS}} = (1-t)^2 + (tf)^3$$

Figure 1 plots the reliability of Tor under the selective DoS attack as a function of t, with $f = 0.99$. The reliability decreases as the number of compromised nodes grows, until it reaches a minimum at $t = 0.55$, at which point it starts to rise again. This is because at that point, the $(1-t)^2$ component starts to dominate; that is, the dishonest nodes start to perform DoS on fewer tunnels because they can now compromise more of them.

Figure 1 also shows the number of secure tunnels, as a fraction of reliable ones; i.e. the conditional probability of a tunnel being secure given that it is reliable. This is a useful calculation, since the Tor software, faced with a non-functioning tunnel, will create a new one in its place, and will repeat this until a working tunnel is constructed; the conditional probability states the likelihood that this final tunnel will be secure. For high values of t, the line closely matches the conventional security figure of $\bar{t}^2$, but with higher numbers of compromised nodes it quickly diverges. For example, with $t = 0.5$, conventional analysis suggests that 75% of all

[1]This calculation simplifies away the detail that routers are picked without replacement, but with $l = 3$ and about 1000 routers, this is a suitable approximation.

paths should be secure, whereas under the selective-DoS attack, only 33% of the successful paths are uncompromised.

Of course, one hopes that fewer than 50% of Tor routers are dishonest—it would seem difficult for an adversary to compromise 400 out of 800 routers. However, the Tor network is run by volunteers and it accepts new routers with minimal verification; it is therefore not out of the question for some organization to contribute many new routers to the system, under different identities, and compromise a significant percentage of routers[2]. Additionally, attackers can misrepresent their bandwidth to gain a higher effective $\bar{t}$ value [1]. The important point is that the conventional analysis of Tor security significantly underestimates the vulnerability of Tor in this scenario.

2.1.2 Guard Nodes

The selective DoS attack is particularly troubling due to the predecessor attack [28]. As users pick random paths in a Tor network, one of these paths will be compromised with high probability after $O((1-t)^2 \ln n)$ path constructions, where n is the total number of nodes in the network. With a selective DoS attack, the attack will function much faster. To resist the predecessor attack, Tor has introduced a defense where each user picks a small fixed set of *guard nodes* that are always used as the first node in a tunnel [29]. Users who pick honest guard nodes will be immune to the predecessor attack.

Guard nodes similarly help defend against selective DoS; users with honest guards may suffer in terms of reliability but their anonymity will never be compromised. However, in another way, guard nodes actually make the selective DoS attack more powerful, since the probability that a single successful tunnel will be compromised is actually higher using guards. For example, it is easy to see that with a single guard node, $(1-t)$ of all tunnels will be compromised; our preliminary results show that using 3 guard nodes, as in the current Tor implementation, also results in a higher number of compromised tunnels. We leave a full investigation of guard nodes to future work, but we note that fixing both the first and last node, as studied by Wright *et al.* [29], is more likely to be an effective defense to selective DoS in Tor.

2.2 Mix Networks

We next discuss the application of the selective DoS attack to high-latency systems based on mix networks, such as the MixMaster [20] and MixMinion [6] networks used for sending anonymous email.

A mix is a router that hides the correspondence between its input and output messages. Mix-Net systems [4] consist of a series of such mixes and provide unlinkability between a sender and recipient. In the original proposal, each message is sent through a sequence of mixes that is chosen randomly from all available mixes. Messages are encrypted in layers with the public keys of the mixes and are then sent through them in series before reaching their eventual destination. Each mix decrypts a layer of the message using its private key, performs some batching strategy to reorder and delay messages, and then forwards it onward.

Chaum's original Mix-Net proposal was further developed in Babel [17], and the Mixmaster [20] and Mixminion [6] systems were later deployed. These systems can be classified as high-latency Mix-Nets since they introduce large and variable latencies during batching. Because of this, however, high-latency Mix-Nets are much more robust to timing attacks than systems such as Tor. Only when the adversary controls every mix in the forwarding path will the anonymity of a message be compromised.

[2]In fact, a smaller scale version of such a Sybil attack [12] has recently been observed on the Tor network.

2.2.1 Reliability

As mix-based systems were developed and deployed, the issue of reliability became apparent as volunteer-run nodes were often unavailable or offline. One approach was to introduce *pingers* to keep track of the reliability of nodes. Pingers attempt to relay traffic through all mixes and keep track of the messages that are eventually delivered. To avoid malicious nodes manipulating the rankings, the pinging traffic is forwarded through the anonymous network itself. Mix clients then select routes based on these reliability rankings. Such a strategy, however, biases the nodes used, may be manipulable by the adversary, and could reduce the anonymity of messages.

An alternative strategy to ensure reliability, supported by Mixminion and Mixmaster, is to send copies of the messages, or fragments thereof, through independent paths. There is no interaction between copies of the message travelling on the multiple paths, and the mixes on the different paths operate independently. Furthermore, messages on different paths are bitwise unlinkable amongst themselves, reducing the potential for traffic analysis.

2.2.2 Conventional Analysis

We first analyze the security and reliability of mix networks under the simple attacker strategy of forwarding all messages. Our results are straightforward and well-known; we present them here for comparison purposes only. We assume that reliability is achieved by sending multiple copies of the messages, as described above. We introduce several parameters for describing the mix network in Table 1; we will use them throughout the rest of the paper.

We first calculate the probability of security in a mix network with parameters (l, w, t, f). For a message to be compromised, at least one full route should be composed of dishonest mixes. A route has at least one honest mix with probability $1-\bar{t}^l$. The probability that all routes have at least one honest mix is:

$$\left(1-\bar{t}^l\right)^w \tag{1}$$

The security of mix networks is significantly higher than low-latency systems, since increasing l results in an exponential increase of security. For example, with $l=5$ (the default used by MixMinion), even if 50% of all mixes are compromised, only 3% of all messages can be read. Cautious users may choose even higher values of l, so that their messages remain secure even under the most pessimistic assumptions about the number of compromised mixes.

We can next compute reliability in a mix network with parameters (l, w, t, f), where adversary nodes simply relay all communications. For the messages to be delivered, there must be at least one full route with no unreliable mixes. The probability of one such route is $(\bar{t}+t\cdot f)^l$, and the probability that not all routes are unreliable is:

$$1-\left[1-(\bar{t}+t\cdot f)^l\right]^w \tag{2}$$

Variable	Description
l	The *length* of all paths. We assume all copies of the message travel over paths of the same length.
w	(for *width*) The number of independent paths over which a copy of the message is transmitted.
t	The probability a mix is honest. Its converse $\bar{t} = 1 - t$ is the probability a node is in the hands of the adversary. We assume that all nodes when chosen have the same probability of being corrupt, independently of the number of previously honest or corrupt nodes selected.
f	The probability an honest node is reliable. Its converse $\bar{f} = 1 - f$ is the probability it is unreliable. This does not apply to corrupt nodes, which are reliable or not depending on the attack strategy—a reliable node relays the message correctly, while an unreliable one is simply offline, and behaves as if it does not exist in the network.

Table 1: Variables used in reliability and security analysis

2.2.3 Selective DoS Attack

Similar to the case in Tor, an adversary may choose to apply a selective DoS strategy to maximize the chances of compromising messages. Instead of relaying all messages, bad mixes only relay those messages that they can trace from the beginning to end: the mixes decrypt as much of the message as they can using the keys of all the colluding mixes and determine whether there is an honest mix somewhere in the chain. Messages that cannot be compromised in this way are either dropped or modified in a subtle way so that they are unrecoverable by the recipient. The sender then has to send more copies of the message to increase its chances of arriving, which in turn increases the chances that the adversary captures the message.

In a mix network with parameters (l, w, t, f), where adversaries drop all communications they cannot compromise, a message will only be delivered if some path is either fully compromised or fully honest and reliable. This occurs with probability $r = (1-t)^l + (t \cdot f)^l$. At least one such path within w must be picked which happens with probability $1-(1-r)^w$, so the reliability of mix networks under DoS is:

$$1 - \left(1 - \left[\bar{t}^l + (t \cdot f)^l\right]\right)^w \tag{3}$$

The DoS strategy does not affect the probability the message is secure; the results are the same as in (1). So what advantage does this strategy present? To achieve the same level of reliability, a sender must send the messages more times, which in turn provides more opportunities for the adversary to capture the message. How many more copies of the message should be sent, though?

NOTE 1. *Given a mix network with parameters (l, w_{pas}, t, f) with a passive adversary leading to messages having a probability of delivery p_{pas}, the number of copies of a message in a network with parameters (l, w_{DoS}, t, f) with a DoS adversary to achieve the same degree of reliability is:*

$$w_{DoS} = \frac{\log(1 - (\bar{t} + t \cdot f)^l)}{\log(1 - (\bar{t}^l + (t \cdot f)^l))} w_{pas} \tag{4}$$

PROOF SKETCH. Require the two probabilities of reliable delivery from (2) and (3) (with w equal to w_{pas} and w_{DoS}, respectively) be equal and solve for w_{DoS}. □

NOTE 2. *The probability of security for a particular target reliability c can easily be calculated:*

$$sec_{pas} = (1 - \bar{t}^l)^{\frac{\log(1-c)}{\log(1-(\bar{t}+t\cdot f)^l)}} \tag{5}$$

$$sec_{DoS} = (1 - \bar{t}^l)^{\frac{\log(1-c)}{\log(1-(\bar{t}^l+(t\cdot f)^l))}} \tag{6}$$

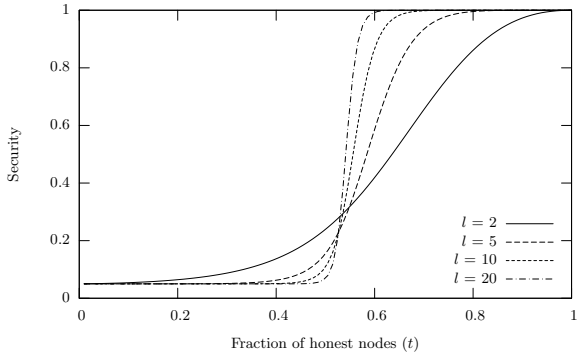


Figure 3: Security of mix networks for different choices of l under the DoS strategy.

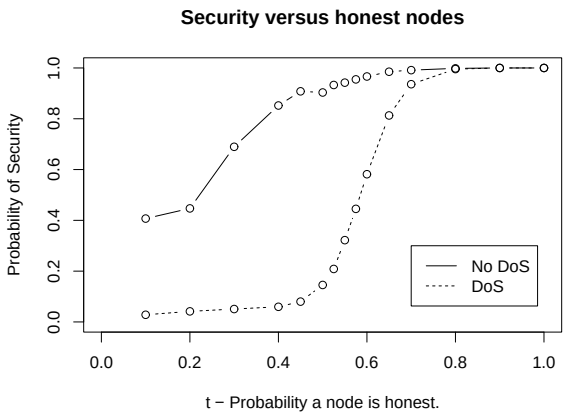
Figure 2 presents the results of simulations used to validate our calculations. Nodes in these experiments tuned their sending parameters w to achieve a reliability of 95%, with fixed $l = 5$ and $f = 0.90$. The parameters l and f are chosen to mimic observed behavior of Mixminion nodes.[3] Figure 2(a) presents the fraction of secure messages out of 5000 sample messages sent through the network, for both the passive and DoS attacker strategies (the sample size leads to an error of less than 1% for all our simulation results; the discontinuities in the graph are due to changing quantized values of w). It is clear that an attacker who denies service has an advantage, depicted as the gap between the two lines representing the probabilities of success of the two attacks.

Figure 2(b) depicts the replication factor w that honest nodes are forced to use to compensate for network unreliability. Under the DoS strategy, the number of copies of a message can become very large. The number of replicas peaks at about $t = 0.5$. For smaller fractions of honest nodes, reliability is guaranteed by the fact that the route is often compromised, and for larger fractions, reliability is restored by the honest nodes. This should act as a clear warning to mix system administrators: reliability is not a measure of security.

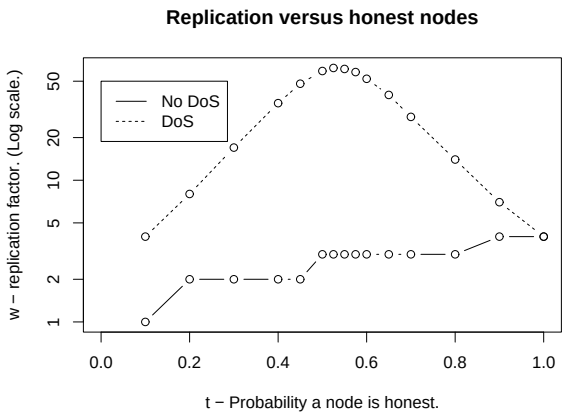
2.2.4 Increasing path lengths (l)

One response to increase security under the DoS attack may be to use longer paths. Conventional analysis suggests that higher values of l provide exponentially higher security, so arbitrary security levels can easily be achieved by increas-

[3]This is lower than the figure we used for reliability in Tor; Tor nodes are periodically probed by the directory server and ejected from the network if they are not reliable.



(a) Security for different fractions of honest nodes.



(b) Number of messages to be sent for different fractions of honest nodes.

Figure 2: The effect of different fractions of honest nodes t on the security of the routes, and the replication factor w, for a target reliability of 95%. Experimental results: 5000 samples per point, $l = 5$ (Mixminion), $f = 0.90$.

ing l. Can the same approach work under the selective DoS strategy?

Figure 3 shows the security achieved for varying values of l using the same parameters as in Figure 2 under the DoS strategy. For higher values of t, increased values of l have the expected effect of increasing security. However, for low values of t, longer paths not only do not help the security, but in fact are a detriment. This is because the chance of path compromise, $\bar{t}^l$, becomes higher than the chance of a successful honest path, $(tf)^l$, and the difference grows with higher l. The crossover point on the graph is when $\bar{t} = tf$.

The results show that, when reliability, and not just security, is taken into account, mix networks have a fundamental limit on the number of compromised mixes. When a majority of nodes are corrupt, mix networks are "unsafe at any path length."

3. DENIAL OF SERVICE AGAINST SYSTEMS FEATURING RELIABILITY

3.1 Cashmere

Cashmere [30] is an anonymous routing layer that uses *relay groups* instead of single-node mixes to provide increased connection reliability. Each relay group is composed of a set of nodes that share a common public/private key pair. This gives any member of the group the ability to decrypt a layer of the message and forward it to the next relay group. Each node in Cashmere is assigned a unique nodeID and each relay group a unique groupID such that a node is a member of a relay group if the groupID is a prefix of its nodeID.

Cashmere is implemented on top of the Pastry [26] structured overlay and makes use of its anycast mechanism to route a message towards any node with the correct groupID prefix; the node that receives the message is named the *relay group root*. The Pastry mechanisms for maintaining reliability ensure that such a node will be found as long as at least one member of the relay group is reliable. The root decrypts the message, broadcasts the payload to all members of his relay group, and then sends the message to the next relay group in the forwarding path. The actual destination may be located in any relay group, not necessarily the last one; a node recognizes itself as the destination when it can decrypt the message payload.

3.1.1 Security and Reliability of Cashmere

In the presence of a passive adversary, honest and reliable, as well as dishonest nodes will forward traffic appropriately. As long as there is at least one of these nodes acting as the relay group root for every relay until the destination, the connection will remain reliable. Since any of the nodes in a relay group can decrypt the current layer of the forwarding path, connections may be insecure whenever there is at least one dishonest mix present in every relay group leading up to the destination. Since the destination itself is not revealed in the message, but instead is chosen among the members of all relay groups, we also require the destination to be dishonest to consider a message compromised; in other words, we measure sender anonymity. (Unlinkability is more complex to model; see [30] for a conventional analysis. We leave the study of unlinkability under selective DoS to future work.)

We next compute the probability that a message is delivered reliably in Cashmere with parameters (l, w, t, f), where dishonest nodes simply relay all communications. The probability that a relay group is reliable, or has at least one honest, reliable or dishonest node present, is $1 - (t\bar{f})^w$. To ensure message reliability, each relay group before the one that contains the destination must be reliable, and the destination must itself be reliable. The destination is in each relay group with probability $1/l$. Therefore, the message is reliably delivered with probability:

$$\sum_{i=0}^{l-1} \frac{1}{l} \left(1 - (t\bar{f})^w\right)^i (\bar{t} + tf) \tag{7}$$

Next we consider security. For message anonymity to be compromised, each relay group leading up the destination

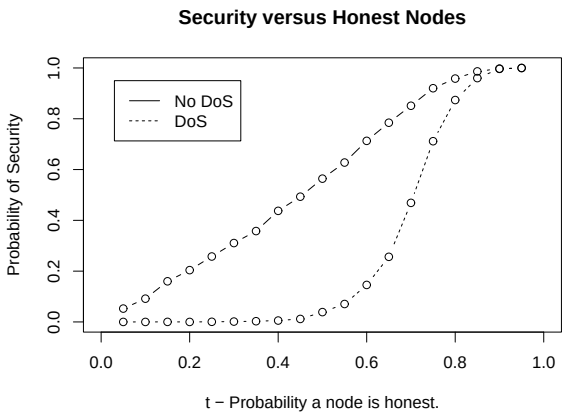


Figure 4: The security of Cashmere under different fractions of honest nodes with an expected reliability of 100%. Experimental results: 5000 samples per point, $l = 5$, and $f = .90$.

must be compromised. The probability that at least one node in the relay group is malicious is $1-t^w$, as must be the case for every relay group leading up the destination. Finally, the destination itself must be malicious for the route to be compromised. Therefore, the probability that the message will be delivered anonymously is:

$$1 - \sum_{i=0}^{l-1} \frac{1}{l} \left(1 - t^w\right)^i \bar{t} \tag{8}$$

3.1.2 Cashmere under a DoS adversary

Cashmere routing is affected by DoS attacks when any of the relay group roots are dishonest. Unless the adversary has compromised the entire forwarding path, he will drop any connection that goes through a relay root he controls.

For a message to be delivered reliably, every group until the destination must deliver the message reliably. This means that either every relay root and the destination are reliable and honest, or the entire path is compromised and thus remains reliable. A relay root is chosen out of only reliable nodes, so it is reliable and honest with probability $tf/(tf + \bar{t}$. So the probability of a message being reliably delivered is:

$$\sum_{i=0}^{l-1} \frac{1}{l} \left(\left(1 - t^w\right)^i \bar{t} + \left(\frac{tf}{tf + \bar{t}} \right)^i tf \right) \tag{9}$$

Figure 4 presents the results of Cashmere simulations under a passive and DoS attacker. In these experiments, honest nodes had $f = .90$, connection lengths were $l = 5$, and the group size $w = 5$, as recommended by the Cashmere authors. This setup produces nearly 100% reliability under a passive adversary. Note that it is impossible to increase reliability under the DoS strategy by increasing w, since the adversary need only to capture the root node to block the message. The graph depicts the fraction of successful connections that remain secure; the DoS strategy is very effective at reducing this number quickly.

These results highlight that under the DoS strategy, both the reliability and security of Cashmere are strictly worse than for mix networks with equivalent w. This is because to deny service, the adversary must capture only a single relay root that precedes the destination, whereas mix networks require an adversary on every path. Similarly, to violate security, the adversary needs only a single adversary in each group, rather than an entire compromised path in mix networks. By failing to consider denial of service as a security concern, the authors have created great potential for the selective DoS attack to succeed.

The graph also shows that, under the parameters we considered, mix networks offer substantially greater security even under the passive adversary strategy. Cashmere *is* useful when there are few compromised nodes and very frequent failures. With $t = 1.0$ and $f = 0.5$, the reliability of Cashmere with $w = 5$ is 94%, conditioned on the destination being reliable. To achieve the same reliability in mix networks, $w = 89$ would be necessary!

3.2 Hydra-Onions

The Hydra-Onion system was designed to resist active adversaries dropping onions during transmission [18]. Just as in mix networks, w copies[4] of a Hydra-Onion are sent in a cascade. However, at each step, a mix will forward two copies of the Hydra-Onion to two different mix servers at the next step.

Each onion has the following format:

$$\begin{aligned} O_i \quad = \{ \quad & Enc_{J_{i,1}}(J_{i+1,1}, J_{i+1,a(1)}, k_{i+1}), \\ & Enc_{J_{i,2}}(J_{i+1,2}, J_{i+1,a(2)}, k_{i+1}), \\ & \cdots, \\ & SEnc_{k_{i+1}}(O_{i+1})\} \end{aligned}$$

where $J_{i,j}$ are the identities of the mixes at step i, and $a()$ is a permutation of nodes with $a(i) \neq i$ for all i. In this case, Enc_J is the hybrid asymmetric/symmetric encryption under the public key of J and $SEnc_k$ is the symmetric encryption under key k. Each mix server decrypts the piece of the onion encrypted under its key and learns the identities of two servers in the next step as well as the symmetric decryption key for the next layer of the onion. This communication pattern is displayed in Figure 5.

3.2.1 Security and Reliability of Hydra-Onions

Since any of the mixes at step i can decrypt the Hydra-Onion O_i, a Hydra-Onion is insecure whenever there is at least one dishonest mix at each step. Therefore, for a Hydra-Onion mix with parameter (l, w, t, f), the probability that a message is secure is:

$$1 - \left(1 - t^w\right)^l \tag{10}$$

The reliability of Hydra-Onions is somewhat harder to ascertain due to the randomized forwarding nature of the mixes. The intuition behind the design is that random graphs are expanders, and therefore, a single Hydra-Onion will quickly replicate to fill the $w - 1$ missing ones.

To evaluate Hydra-Onion reliability, we have developed a simulation of the scheme. We create a network of l by w mixes, and connect the nodes using randomly generated permutations a. Each of the nodes is assigned to be honest with

[4]The authors of the Hydra-Onion system call this parameter k; however, we use w for consistent presentation.

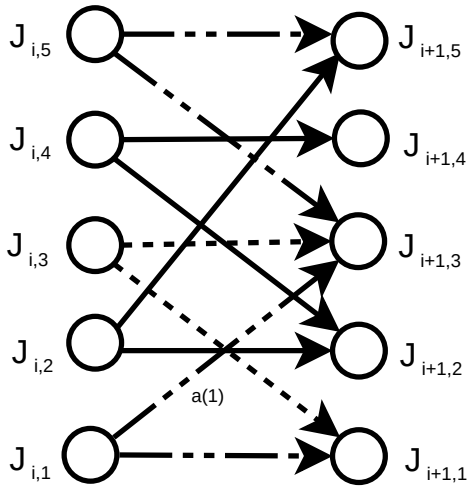


Figure 5: Hydra-Onion communication patterns

probability t and, if honest, to be reliable with probability f.

In the case of simple attacker strategy, we assume that dishonest nodes are always reliable. To simulate the reliability given the DoS attacker strategy, we first determine whether there is at least one dishonest mix at each step. In this case, the onion is compromised and the dishonest nodes are reliable and forward all messages. Otherwise, the dishonest nodes perform a denial of service and drop all traffic sent to them.

3.2.2 Analysis

Figure 6 shows the analysis of Hydra-Onions under both the simple adversary strategy and the DoS strategy. Setting $f = 0.9, l = 5$ and varying t, we increased w until we could obtain 95% reliability. The figures plot the w required to achieve this reliability, as well as the security at that w.

As designed, Hydra-Onions are effective at providing reliability in the face of denial of service: even under heavy denial of service, $w = 6$ suffices to achieve 95% reliability. However, this is done at the expense of security: increasing values of w very quickly decrease the security of Hydra-Onions, as a single compromised mix at each step suffices to compromise the entire onion. When 15% of nodes are malicious, 5% of all onions are compromised, and when the fraction of malicious nodes rises to 30%, over half of all paths are compromised. 30%, and even 15%, may sound like high fractions of attackers, but recall that the conventional analysis of a mix network with 5 hops that ignores DoS suggests that path compromise occurs with probability 0.25% and 0.008% respectively with these fractions of attackers. For comparison, mix networks with 30% attackers are able to achieve 93.6% security, albeit with a width of 28.

Therefore, Hydra-Onions are not a good tool when a significant number of mixes are compromised. As can be seen from the right limit of the graph, they are also inefficient when nearly all nodes are honest: with 95% honest nodes, Hydra-Onions use a width of 3 in the DoS strategy, which has a communications cost equivalent to a mix network width of 6, since each mix sends two onions. Mix networks under the same parameters require $w = 5$ to achieve 95% reliability, and provide better security. (Though the security advantage is slight, as both schemes achieve over 99.99% security with these parameters.)

The main advantage of Hydra-Onions seems to be when most nodes are honest, but not reliable (either due to inherent reliability problems or external DoS attacks.) For example, with $f = 0.5$, mix networks require $w = 95$ to achieve 95% reliability, even when no nodes are compromised, whereas Hydra-Onions only require $w = 19$ in the same situation. However, a simpler variant of Hydra-Onions proposed by the same authors, called DUO-Onions, may be more appropriate for this case. DUO-Onions [18] are designed to handle fail-stop failures, such as unreliable nodes or DoS, by iteratively picking the next mix in a list whenever the first choice is unreachable. DUO-Onions have the advantage of using dramatically less bandwidth in the case that nodes *are* reliable, and only sending extra onions as necessitated by failures. They are unable to address Byzantine faults of the forwarding mixes, but as our analysis shows, neither are Hydra-Onions.

4. DENIAL OF SERVICE AGAINST SALSA

Salsa [21] is an anonymous communication system designed to overcome the scalability problems in traditional mix systems. As in Tor, a tunnel is built between the initiator and the recipient via proxy routers (nodes) for anonymous communication. Layered encryption ensures that each node knows only its previous and next hop in the tunnel. The nodes used for the tunnels are randomly selected from the global pool of nodes, even though each node has only local knowledge of a small subset of the network.

Salsa is based on a distributed hash table (DHT) that maps nodes to a point in an ID space corresponding to the hash of their IP address. There are two basic mechanisms in the Salsa architecture: (1) a node lookup mechanism and (2) a tunnel building mechanism. The former returns the IP address and public key of node in the DHT closest to a given point in the ID space. The latter is used to build a Tor-like tunnel. Both schemes use redundancy to avoid attacks and both are susceptible to the selective DoS attack. Given the space constraints, we shall focus our discussion on the impact of selective DoS attack on the tunnel building mechanism and we will assume that the lookup mechanism is perfectly secure. Salsa is resistant to conventional attacks that target the lookup mechanism as long as the fraction of malicious nodes in the system ($\bar{t}$) is less than 0.2. Since Salsa does not provide adequate anonymity for high values of $\bar{t}$, we shall limit our analysis to low values.

To build a tunnel the initiator chooses r random IDs and looks up the corresponding nodes (called the first set of nodes). Keys are established with each of these nodes. Each of the first set of nodes does a lookup for r additional nodes (second set of nodes). A circuit is built to each of the nodes in the second group, relayed through one of the nodes in the first group. Again, the initiator instructs the second set of nodes (via the circuits) to do a redundant lookup for a final node. One of the paths created between the first and the second set of nodes is selected and the final node is added to the tunnel. We use the parameter l to refer to the number of stages in the tunnel. ([21] suggests $r = 3$ and $l = 3$).

4.1 Conventional Analysis

A tunnel in the Salsa system can be compromised if there is at least one attacker node in every stage of the tunnel.

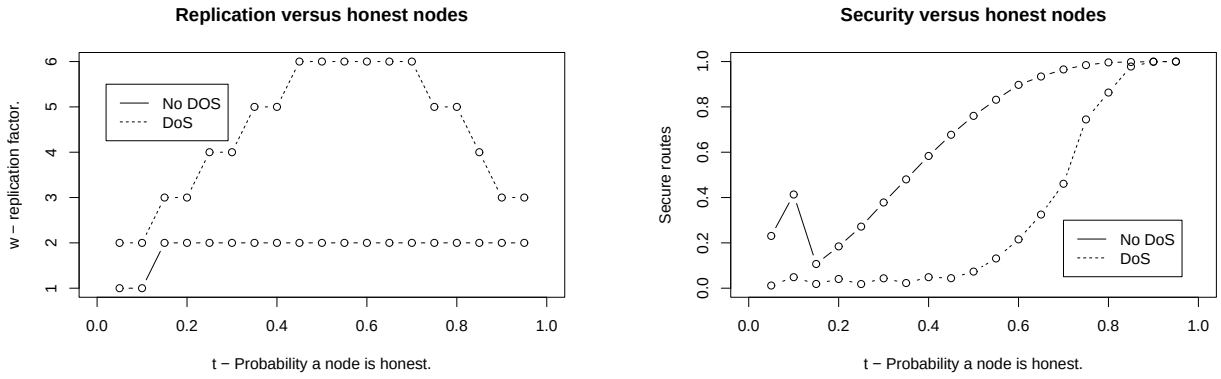


Figure 6: Replication factor w that achieves reliability of 95% for Hydra-Onions under different fraction of honest nodes t, and the corresponding security. Analytical results. $l = 5, f = 0.90$.

Also, by end-to-end timing analysis, the tunnel will be compromised if the first and last forwarding nodes in the tunnel are compromised. The conventional analysis in [21] shows that the latter attack dominates and the probability of compromise is not much larger than $\bar{t}^2$.

We note that the tunnel building process is subject to a public key modification attack. If all r nodes in a particular stage are compromised, they can modify the public keys of the next set of nodes being looked up. This attack defeats Salsa's bound check algorithm that ensures the IP address is within the right range, since it cannot detect an incorrect public key. Also, since the traffic toward the node whose public key has been modified is forwarded via corrupt nodes, the attackers are guaranteed to intercept the messages. They can then complete the tunnel building process by emulating all remaining stages (and hence, the last node). Thus, if the attackers have the initiator information and any stage is fully compromised, the tunnel is compromised.

4.2 Selective DoS attack

The idea of selective DoS attack is to deny service to trustworthy nodes so that user traffic moves toward compromised nodes. The compromised nodes will try to abort the tunnel building process whenever the tunnel cannot be compromised. A malicious node can easily launch a denial of service by returning an arbitrary result from a lookup. The Salsa tunnel building mechanism aborts if the lookup information provided by the redundant r nodes in any stage is inconsistent[5].

The attackers should deny service in two cases. First, if the last node is honest, and there is an attacker in the second last stage, that attacker will perform DoS, unless all r nodes in that stage are malicious. (This can be easily determined on the reception of r messages at attacker nodes containing lookup requests for the identical r nodes in the next stage.) Also, if the attacker nodes are selected to forward traffic in a tunnel, they can deny service if the tunnel has not been compromised. The nodes will perform traffic analysis on the first portion of the stream sent over a tunnel and correlate it with all other streams observed by other attackers. If the stream can be linked to both an initiator and a destination, the attackers continue forwarding traffic; otherwise, they terminate the tunnel as it cannot be compromised.

The attack algorithm is as follows:

if a stage is completely compromised **then**
  emulate remaining hops via public key modification attack.
else
  if the second-to-last stage has an attacker **and** the last node being looked is honest **then**
    return arbitrary information to DoS the tunnel
  else
    return correct results
  end if
end if
if attacker selected to forward traffic **then**
  perform traffic analysis
end if
if attackers cannot identify the source and destination of the tunnel after a timeout **then**
  stop forwarding traffic on that tunnel
end if

4.3 Analysis

We compare the performance of three attack methodologies on the Salsa tunnel building mechanism. The first one consists of conventional passive attacks in which the tunnel is compromised whenever there is an attacker in every stage or via end to end timing analysis. The second methodology involves active modification of the public key of nodes being looked up whenever the attackers have compromised an entire stage. This attack also includes the conventional attacks. In the third methodology, nodes try to selectively DoS the tunnels which are likely not to be compromised. The public key modification and conventional attacks are also included in this methodology. We have used $f = 1$ in our analysis. Our results have been computed by modeling the Salsa tunnel building mechanism as a stochastic activity network in the Möbius framework [5]. Figure 7 shows the fraction of compromised tunnels for varying attacker ratios under the three attacks. We find that the public key

[5]This behavior is not precisely specified in [21], but has been confirmed by the Salsa authors in a private communication.

modification attack does not yield a significant advantage over conventional attacks. This is because the probability of compromising an entire stage for $r = 3$ is very low.

Our analysis shows that the current Salsa design is extremely vulnerable to the selective DoS attack, especially for high attacker ratios. In fact, as compared to the conventional security analysis of 6.82% compromised tunnels for an attacker ratio of $\bar{t} = 0.2$, the selective DoS attack results in 19.14% compromised tunnels. Also, the fraction of compromised tunnels may even be higher (depending on the attacker ratio) than that of a system with a single intermediate proxy. This shows that the selective DoS attack has devastating effects on the security of Salsa.

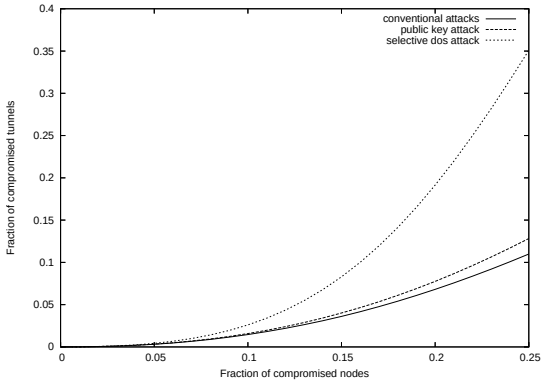


Figure 7: Effect of selective DoS on Salsa tunnel building

Given the massive reduction in anonymity made possible by the selective DoS attack, we study whether other choices of r and l could better resist this attack. We find that the choice of both the number of nodes in a stage (r) and the number of stages (l) has a considerable impact on system anonymity under selective DoS.

Figure 8(a) shows the effect of varying r on the system anonymity under selective DoS attack. We find that increasing redundancy above $r = 3$ clearly increases the fraction of compromised tunnels. Values of $r = 2$ or $r = 3$ provide better anonymity. Though a higher redundancy in r makes it harder for the attacker to launch a public key modification attack, it increases the probability of an attacker being present in any of the stages. This probability is equal to $1 - t^r$. An increase in this probability translates into more opportunities for the attacker to launch the selective DoS attack. It also helps the attacker goal of having a presence in every stage which compromises the tunnel via passive conventional attacks. We conclude that increasing redundancy in r does not buy us more anonymity, and $r = 2$ or $r = 3$ are good design choices.

Figure 8(b) shows the effect of varying the number of stages on the system anonymity with a fixed $r = 3$. We find that increasing the number of stages does not buy us more anonymity. In fact, increasing l above 3 proves counterproductive. The choice of parameter l has an interesting trade off. On one hand, increasing l decreases the probability that an attacker has a presence in every stage. However, it also gives more opportunities for an attacker to launch a selective DoS attack. Note that for $l = 2$, the effect of public key modification is dominant for small values of the attacker ratio, resulting in higher fraction of compromised tunnels. It becomes optimal (replacing $l = 3$) for higher attacker values. Under conventional analysis, a higher value of l can significantly improve anonymity at the cost of extra performance overhead. This is clearly not true with the selective DoS attack. We have shown that higher values of redundancy in l does not result in better anonymity. The values of $l = 2$ or 3 are optimal design choices.

Another simple change to Salsa design would be to use a majority, rather than consensus, decision when a conflict arises. In the current Salsa design, a single malicious node that returns an incorrect result will cause the tunnel to be aborted. An alternative is to use the result returned by the majority of the nodes, ignoring the dissenter. This will greatly reduce the ability to deny service in tunnel building, at the cost of making a public key modification attack easier. Our analysis of this method with $r = 3$ and $l = 2, 3$ shows that it does not perform better than the consensus algorithm; however, such a modification may still be useful since it takes fewer attempts to build a successful tunnel, improving usability.

5. COUNTERMEASURES AND RELATED WORK

In this section we briefly consider potential countermeasures to the selective DoS attack. Although applications of redundancy as seen in Cashmere and Hydra-Onions do not effectively address the attack, other measures might have a better chance.

As mentioned in Section 2.1.2, fixing the first and last nodes in a tunnel or mix path, may help defend against selective DoS attacks. Another straightforward approach would be to notice that a node is unreliable and exclude it from future path building. A challenge in this case is to detect which node is responsible for a packet getting dropped in an anonymous path. In particular, when a router A is supposed to send a packet to router B, it is difficult to tell which one was responsible for dropping a packet. One could design a reputation system that punishes both A and B, hoping that malicious routers will be punished more frequently, but notice that this approach breaks down when $t \leq 0.5$.

Dingledine *et al.* [7] propose the use of witness nodes to verify that communications were relayed correctly and design a reputation mechanism to avoid unreliable nodes. Dingledine and Syverson [11] further extend those ideas to construct reliable, longer term routes through networks of mixes. They present a variant of the denial of service attack that aims to reduce the reputation of honest nodes, so that they are less likely to be used.

Reliable anonymous communications have been the subject of a lot of research in the context of electronic election, starting with [25]. For this particular application, it is considered unacceptable to "lose" anonymously transmitted ballots, and mix systems that offer public verifiability of delivery have been devised. The state of the art in this field are the techniques by Neff [22] and Furukawa and Sako [14].

However, public verifiability in this case requires operating all of the mixes in a synchronous cascade [10], performing expensive cryptographic operations on each batch, and posting all of the intermediate batches on a public bulletin board. These requirements are appropriate in a voting context, but it would be difficult to design a general anonymous communication system that satisfies them.

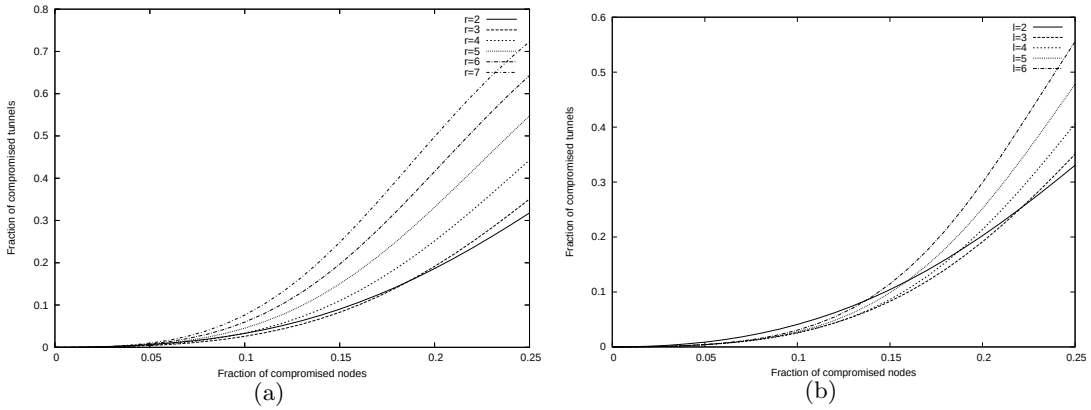


Figure 8: Effect of varying r (a) and l (b)

Jakobsson *et al.* [19] present an alternative for public verifiability of delivery they call *Randomized Partial Checking*. Mixes in a cascade or network, commit to their outputs, and are then required to reveal half of the correspondences between inputs and output. Using this technique, cheaters can be caught and excluded quickly. Gomułkiewicz *et al.* [16] show that the revealed information does not give any advantage to the adversary, even for short path lengths. Yet this scheme is not without dangers: users may be tempted to check each mix to determine whether their own messages have arrived, and naive mechanisms reveal users' paths. We note that no widely deployed system uses this mechanism, and that low-latency networks would not be secure if the traces necessary to implement it were publicly available (since they cannot easily be made secure against a global passive adversary).

6. CONCLUSION

We have shown that in anonymous communications systems, denial of service attacks reduce anonymity considerably. This shows that availability and anonymity are linked and reliability must be assured against adversaries and not just random failures.

Our results have a profound impact on the theory of mixes, but most importantly on deployed systems such as Mixminion [6] or Tor [9]: we show that traditional architectures, offering no protection against DoS attacks, are subject to complete compromise if the network is contains a majority of dishonest nodes. Traditional mixes aimed to protect a communication even if a single honest mix was on the path, and little previous work has questioned this security assumption [3]. We show that routes with few honest nodes will be subject to DoS, and only fully honest or fully compromised paths will survive. This intuition in an embryonic form was present in [11] in the context of reputation systems, but its deep repercussions for general mix systems, and the fundamental limit on the security it imposes, was not fully understood until now. For a long time designers believed that the security of mix systems could be brought arbitrarily high, as the path length increases. We now prove this to be wrong.

Mechanisms to prevent our denial of service based attacks, either by detecting maliciously unreliable nodes, or ensuring an honest majority, will have to be part of any future mix systems design and deployment. Sadly, designs that simply ensure reliability, such as Cashmere and Hydra-Onions, are curing the symptoms rather than the disease: they only focus on reliability while making the anonymity of the system even worse under DoS attacks.

Our work strongly demonstrates that mechanisms to address reliability, as well as preventing denial of service, must be designed and evaluated with criteria from *security engineering* and not merely network engineering. Our experience with Salsa suggests that this will be even more important for future designs that might be based on a peer-to-peer paradigm, since the extra complexity introduced by peer-to-peer networks can give attackers more chances for denial of service.

7. ACKNOWLEDGMENTS

We would like to thank Claudia Diaz and Emilia Käsper for their help with, and the proofreading of, some of the derivations, Paul Syverson for his comments about guard nodes, the anonymous reviewers for their helpful suggestions, and our shepherd, Roger Dingledine, for his direction in producing the final version of the paper.

8. REFERENCES

[1] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. Sicker. Low-resource routing attacks against Tor. In *ACM Workshop on Privacy in Electronic Society*, Alexandria, VA, Oct. 2007. ACM Press.

[2] S. M. Bellovin and D. A. Wagner, editors. *IEEE Symposium on Security and Privacy*, Berkeley, CA, May 2003. IEEE Computer Society.

[3] O. Berthold, A. Pfitzmann, and R. Standtke. The disadvantages of free MIX routes and how to overcome them. In Federrath [13], pages 30–45.

[4] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, February 1981.

[5] D. Daly, D. D. Deavours, J. M. Doyle, P. G. Webster, and W. H. Sanders. Möbius: An extensible tool for performance and dependability modeling. In B. R. Haverkort, H. C. Bohnenkamp, and C. U. Smith, editors, *Computer Performance Evaluation: Modelling Techniques and Tools*, volume 1786 of *Lecture Notes in*

Computer Science, pages 332–336, Schaumburg, IL, Mar. 2000. Springer.

[6] G. Danezis, R. Dingledine, and N. Mathewson. Mixminion: Design of a Type III Anonymous Remailer Protocol. In Bellovin and Wagner [2], pages 2–15.

[7] R. Dingledine, M. J. Freedman, D. Hopwood, and D. Molnar. A reputation system to increase MIX-net reliability. In I. S. Moskowitz, editor, *Information Hiding*, volume 2137 of *Lecture Notes in Computer Science*, pages 126–141, Pittsburgh, PA, 2001. Springer Berlin / Heidelberg.

[8] R. Dingledine and N. Mathewson. Anonymity loves company: Usability and the network effect. In R. Anderson, editor, *Fifth Workshop on the Economics of Information Security (WEIS)*, Cambridge, UK, June 2006.

[9] R. Dingledine, N. Mathewson, and P. F. Syverson. Tor: The Second-Generation Onion Router. In *The 13th USENIX Security Symposium*, pages 303–320, San Diego, CA, August 2004. USENIX Association.

[10] R. Dingledine, V. Shmatikov, and P. F. Syverson. Synchronous batching: From cascades to free routes. In D. Martin and A. Serjantov, editors, *4th Privacy Enhancing Technologies Workshop (PET)*, volume 3424 of *Lecture Notes in Computer Science*, pages 186–206, Toronto, Canada, May 2004. Springer Berlin / Heidelberg.

[11] R. Dingledine and P. F. Syverson. Reliable MIX cascade networks through reputation. In M. Blaze, editor, *Financial Cryptography*, volume 2357 of *Lecture Notes in Computer Science*, pages 253–268, Southampton, Bermuda, 2003. Springer Berlin / Heidelberg.

[12] J. Douceur. The Sybil Attack. In P. Druschel, M. F. Kaashoek, and A. I. T. Rowstron, editors, *International Workshop on Peer-to-Peer Systems (IPTPS)*, volume 2429 of *Lecture Notes in Computer Science*, pages 251–260, Cambridge, MA, Mar. 2002. Springer Berlin / Heidelberg.

[13] H. Federrath, editor. *Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, volume 2009 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, July 2000.

[14] J. Furukawa and K. Sako. An efficient scheme for proving a shuffle. In J. Kilian, editor, *Advances in Cryptology (CRYPTO)*, volume 2139 of *Lecture Notes in Computer Science*, pages 368–387, Santa Barbara, CA, USA, 19-23 August 2001. Springer Berlin / Heidelberg.

[15] P. Golle and A. Juels. Parallel mixing. In *ACM Conference on Computer and Communications Security*, pages 220–226, Washington, DC, Oct. 2005. ACM Press.

[16] M. Gomułkiewicz, M. Klonowski, and M. Kutyłowski. Provable unlinkability against traffic analysis already after $O(log(n))$ steps! In K. Zhang and Y. Zheng, editors, *Information Security Conference*, volume 3225 of *Lecture Notes in Computer Science*, pages 354–366, Palo Alto, CA, Sept. 2004. Springer Berlin / Heidelberg.

[17] C. Gülcü and G. Tsudik. Mixing E-mail with Babel. In *Network and Distributed Security Symposium (NDSS)*, pages 2–16, San Diego, CA, February 1996. Internet Society.

[18] J. Iwanik, M. Klonowski, and M. Kutyłowski. DUO-Onions and Hydra-Onions—failure and adversary resistant onion protocols. In *IFIP TC-6 TC-11 Conference on Communications and Multimedia Security*, pages 1–15, Windermere, United Kingdom, September 2004. Springer Boston.

[19] M. Jakobsson, A. Juels, and R. L. Rivest. Making mix nets robust for electronic voting by randomized partial checking. In D. Boneh, editor, *USENIX Security Symposium*, pages 339–353, San Francisco, CA, Aug. 2002. USENIX Association.

[20] U. Möller, L. Cottrell, P. Palfrader, and L. Sassaman. Mixmaster Protocol — Version 2. Draft, available at: `http://www.abditum.com/mixmaster-spec.txt`, July 2003.

[21] A. Nambiar and M. Wright. Salsa: a structured approach to large-scale anonymity. In *13th ACM conference on Computer and Communications Security*, pages 17–26, Alexandria, VA, Oct. 2006. ACM Press.

[22] C. A. Neff. A verifiable secret shuffle and its application to e-voting. In P. Samarati, editor, *8th ACM conference on Computer and Communications Security*, pages 116–125, Philadelphia, PA, Oct. 2001. ACM Press.

[23] L. Øverlier and P. Syverson. Valet services: Improving hidden servers with a personal touch. In *Sixth Workshop on Privacy Enhancing Technologies (PET)*, volume 4258 of *Lecture Notes in Computer Science*, pages 223–244, Cambridge, UK, June 2006. Springer Berlin / Heidelberg.

[24] P. Palfrader. Number of Running Tor Routers. `http://www.noreply.org/tor-running-routers/`, 2007.

[25] C. Park, K. Itoh, and K. Kurosawa. Efficient anonymous channel and all/nothing election scheme. In T. Helleseth, editor, *Advances in Cryptology (EUROCRYPT)*, volume 765 of *Lecture Notes in Computer Science*, pages 248–259, Lofthus, Norway, 23-27 May 1993. Springer Berlin / Heidelberg.

[26] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pages 329–350, Nov. 2001.

[27] P. Syverson, G. Tsudik, M. Reed, and C. Landwehr. Towards an analysis of onion routing security. In Federrath [13], pages 96–114.

[28] M. Wright, M. Adler, B. N. Levine, and C. Shields. An analysis of the degradation of anonymous protocols. In *Network and Distributed Security Symposium (NDSS)*. Internet Society, February 2002.

[29] M. Wright, M. Adler, B. N. Levine, and C. Shields. Defending anonymous communication against passive logging attacks. In Bellovin and Wagner [2].

[30] L. Zhuang, F. Zhou, B. Y. Zhao, and A. Rowstron. Cashmere: Resilient anonymous routing. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, Boston, MA, May 2005.