Analyzing an Adaptive Reputation Metric for Anonymity Systems

Anupam Das
University of Illinois at Urbana-Champaign
Department of Computer Science
das17@illinois.edu

Nikita Borisov
University of Illinois at Urbana-Champaign
Department of Electrical and Computer Engineering
nikita@illinois.edu

Matthew Caesar
University of Illinois at Urbana-Champaign
Department of Computer Science
caesar@illinois.edu

ABSTRACT

Low-latency anonymity systems such as Tor rely on intermediate relays to forward user traffic; these relays, however, are often unreliable, resulting in a degraded user experience. Worse yet, malicious relays may introduce deliberate failures in a strategic manner in order to increase their chance of compromising anonymity. In this paper we propose using a reputation metric that can profile the reliability of relays in an anonymity system based on users' past experience. The two main challenges in building a reputation-based system for an anonymity system are: first, malicious participants can strategically oscillate between good and malicious nature to evade detection, and second, an observed failure in an anonymous communication cannot be uniquely attributed to a single relay. Our proposed framework addresses the former challenge by using a proportional-integral-derivative (PID) controller-based reputation metric that ensures malicious relays adopting time-varying strategic behavior obtain low reputation scores over time, and the latter by introducing a filtering scheme based on the evaluated reputation score to effectively discard relays mounting attacks. We collect data from the live Tor network and perform simulations to validate the proposed reputation-based filtering scheme. We show that an attacker does not gain any significant benefit by performing deliberate failures in the presence of the proposed reputation framework.

Categories and Subject Descriptors

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

HotSoS '14, April 08–09 2014, Raleigh, NC, USA.
Copyright 2014 ACM 978-1-4503-2907-1/14/04 ...$15.00.
http://dx.doi.org/10.1145/2600176.2600187.

General Terms

Security, Measurement

Keywords

Anonymity, Reputation Model, Tor Network, PID controller.

1. INTRODUCTION

Anonymous communication systems play an important role in protecting users against network surveillance and traffic analysis. Tor [19] is a widely used anonymity system with approximately 5 000 relays forwarding 300 TB and serving an estimated 300 000 unique users per day, as of March 2014[1]. The anonymity guarantees provided by Tor are directly linked with the reliability of the serving relays. Unreliable relays can impair anonymity guarantees in two ways: first, unreliable relays can degrade user experience leading to a certain fraction of the users abandoning the system and thus decreasing the anonymity set [32], and second, the remaining users end up retransmitting multiple messages, presenting further opportunities for observation. To make things worse malicious relays can strategically affect the reliability of Tor paths to increase their odds of compromising user anonymity. One such known active attack is called *selective denial-of-service* (DoS) [11, 12] where compromised relays drop any communication that they cannot compromise. There have been past instances of active attacks on Tor [1, 5, 6], as well as recent governmental endeavors [20, 22] to deanonymize Tor users, making identification of active attackers in anonymity systems an important problem.

Our goal is to analyze a reputation-based framework for detecting and defending users against relays mounting active attacks like selective DoS on the reliability of anonymous communications. There are two main challenges in building a reputation model for relays of an anonymity network like Tor. First, malicious relays can oscillate between good and bad behavior in order to evade detection, and second, an observed failure in an anonymous communication cannot be uniquely attributed to a single relay. To address the former challenge we adopt ideas from control theory and use a

[1] https://metrics.torproject.org

proportional-integral-derivative (PID) controller-based reputation metric that can capture the dynamic behavioral trend of a given Tor relay. Our PID controller-based reputation metric ensures that a malicious relay that oscillates between reliable and unreliable states obtains low reputation score over time. We address the latter challenge by a protocol to filtering relays based on their reputation score. We show that our proposed filtering scheme can effectively discard relays mounting active attacks when majority of the relays ($\approx 80\%$) are honest.

We analyze the effectiveness of our reputation based filtering protocol through simulation using data collected from the live Tor network. We conclude that in the presence of the PID controller-based reputation framework, attackers gain no significant advantage through active attacks like selective DoS.

Contributions. We offer the following contributions:

- We use a PID controller-based reputation framework that assigns quantitative scores to relays based on the reliability that they provide during anonymous communications. The reputation system captures dynamic behavioral change and penalizes any relay exhibiting such behavioral oscillation.

- We perform simulation on data collected from the live Tor network to demonstrate that our reputation based filtering scheme can effectively discard compromised relays.

Roadmap. The remainder of this paper is organized as follows. In the following section, we provide a science of security perspective on our work. Section 3 gives an overview of Tor, along with the threat model that we consider. We formally introduce the PID controller-based reputation metric in Section 4, and our filtering scheme in Section 5. To show the effectiveness of the proposed reputation-based filtering scheme, we present simulation results in Section 6. We describe some related work in Section 7 and some limitations and future work in Section 8. Section 9 concludes.

2. THE SCIENCE OF SECURITY

The three legs of science are theory, experiments, and simulation. Theory develops abstract models that describe behavior; a key element of theory is metrics that quantify that behavior in different ways. Theory can be used to predict metric values in different contexts. Experiments are used to discover behaviors and observe metrics that might be encoded in theoretical models, and to validate the predictions made by theoretical models. Simulation is different from both, and yet adds computational elements to both. More than physical experimentation, simulation can evaluate many different contexts and be used to scientifically understand behavior in terms of analyzing sensitivity of predicted system behavior to different system parameters. A carefully conducted simulation study of the behavior of security metrics is part of the science of security.

A framework like this is useful in helping to identify how our work contributes to a science of security. The centerpiece of this paper is a metric, one that is useful to distinguish between trustworthy Tor routers and ones that may be engaged in attacks on anonymity. As required for a science of security, this metric quantifies an attribute of security, and can be computed from field measurements. Our work uses models as well, models of network and network behavior, and uses simulation of those models to understand how this metric behaves. The simulation study is carefully done, both in terms of experimental methodology and in being driven by real connection information observed on the live Tor network. Thus we see that the topical study and research approach taken by this paper places it squarely in the context of the science of security.

3. BACKGROUND

In this paper we consider Tor, one of the most widely used low-latency anonymity system, as a case study to profile its participating relays/routers. We first present a brief overview of the Tor network, and then discuss how active attacks can lower anonymity in Tor. Next, we briefly discuss different types of reputation system.

3.1 Tor: A Low-latency Anonymity Network

Tor anonymizes user traffic by relaying the traffic through several intermediate Tor *relays* (also known as routers). A Tor user constructs a *circuit* comprised of several Tor relays forming a pipeline through which traffic is forwarded back and forth between the user and destination. A circuit typically involves three relays: the *entry*, *middle*, and *exit*. Tor protects the contents of the traffic by using a layered encryption scheme called onion encryption [33], where each relay decrypts a layer while forwarding. As a result, any individual router cannot reconstruct the whole circuit and link the source to the destination. Figure 1 summarizes how Tor circuits are constructed and used by clients. The relays in a circuit are chosen using specific constraints [17]. Each user selects the *entry* relay from a small, fixed number of relays that are flagged as "fast" and "stable". These relays are called *guard relays* [39]; their use is designed to defend from the predecessor attack [40]. To choose the exit relay, the user picks from among those relays that have an exit policy compatible with the desired destination. After these constraints, the relays for each position are chosen randomly, weighted by their bandwidth [2].

Tor aims to provide low-latency traffic forwarding for its users. As a result, as traffic is forwarded through a circuit, timing patterns remain observable, and an attacker who ob-

[2] This is a simplified description of the path selection; a detailed specification can be found at [18]. The omitted details do not significantly impact our analysis, and we use the full specification in our simulations.

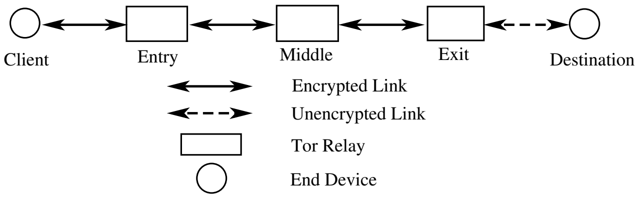


Figure 1: Schematic diagram of how Tor circuits are constructed and used. All links between the relays are encrypted, only the link between the exit relay and the destination device is unencrypted.

serves two different relays can perform timing correlation to determine whether they are participating in the same circuit [27, 35, 37, 43]. Thus, to compromise anonymity it suffices to observe the entry and the exit relays for a circuit. Figure 2 highlights the scenario pictorially. Standard security analysis of Tor [19, 37] shows that if c fraction of the relays are observed by an adversary then the adversary can violate anonymity on c^2 of all of the circuits. Due to bandwidth-weighted path selection in Tor, c is best thought of as the fraction of total Tor *bandwidth* that belongs to relays under observation[3]. The security of Tor, therefore, relies on the assumption that a typical adversary will not be able to observe a significant fraction of Tor relays. For most adversaries, the easiest way to observe relay traffic is to run their own relays. It should be noted that other forms of adversaries do exist, such as ISP-level adversaries, and Internet exchange-level adversaries [8, 21, 30], but these adversaries are typically assumed to be passive and are thus not the focus of this paper.

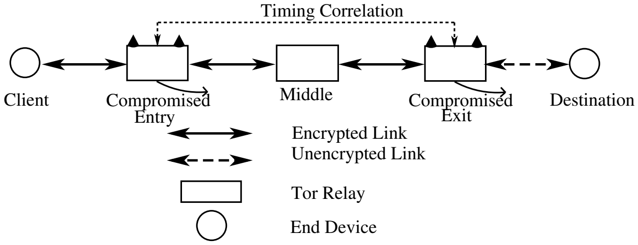


Figure 2: Schematic diagram showing how compromised relays can use timing pattern to determine if they are participating in a circuit that they can compromise.

3.2 Active Attack: Selective DoS in Tor

An adversary who controls a Tor relay can perform a number of active attacks to increase the odds of compromise [11, 12]. One approach is selective denial-of-service (DoS) [12]. A compromised relay that participates in a circuit can easily check whether both the entry and exit relays are under observation. If this is not the case, the relay can "break" the circuit by refusing to forward any traffic. This will cause a user to reformulate a circuit for the connection, giving the adversary another chance to compromise the circuit. Figure 3 highlights a selective DoS attack scenario pictorially.

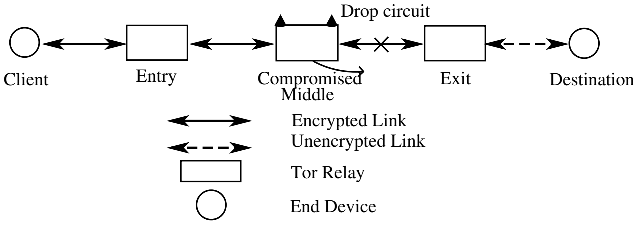


Figure 3: Schematic diagram of a scenario showing how compromised relays can carryout selective DoS attack. Compromised relays use timing pattern to determine if they are participating in a circuit that they can compromise.

A simple analysis shows that selective DoS increases the overall fraction of compromised circuits to:

$$\frac{c^2}{c^2 + (1-c)^3} > c^2 \tag{1}$$

because only circuits with compromised entry and exit relays (c^2) or circuits with no compromised relays ($(1-c)^3$) will be functional, and out of those c^2 will be compromised. For example, if 20% of the bandwidth is controlled by an adversary (i.e., $c = 0.2$) then the selective DoS attack nearly doubles the overall fraction of compromised circuits from 4% to 7.2%.

The use of guard relays changes the analysis somewhat. If none of a user's guards are compromised, then the user is effectively immune from the selective DoS attack, since the user will never use a compromised entry regardless of the attack. If, on the other hand, one or more of the guards are malicious then the user is significantly impacted, as the dishonest guard(s) chosen for a significant fraction of all circuits will break any circuit that does not use a compromised exit. For $c = 0.2$, if one of the guards is compromised then the selective DoS attack increases the overall fraction of compromised circuits from 6.6% to 13.5% and for two compromised guards this value increases from 13.3% to 38.5%. Therefore, guard relays mitigate the selective DoS attack in that it will affect fewer users if they choose honest guards, but amplify its effect for users who are unlucky enough to choose dishonest guards.

3.3 Reputation Models

A reputation model collects, aggregates, and distributes feedback about participants' past behavior [34]. Reputation models help users decide whom to trust, encourage trustworthy behavior, and discourage participation by users who are

[3] To be more precise, the correct fraction would be $c_g \cdot c_e$, where c_g and c_e are the fractions of the guard and exit bandwidth under observation, respectively. For simplicity of presentation, we will assume $c_g = c_e = c_m = c$ in the rest of the paper.

dishonest. Reputation models can be classified as *local* or *global*, based on the way information is aggregated [29]. In a local reputation model, feedback is derived only from direct encounters (first-hand experience) whereas in a global reputation model feedback is also derived indirectly (second-hand evidence) from other users. Hence, in the case of a global reputation model [25, 41, 42], a user aggregates feedback from all users who have ever interacted with a given participant, thus enabling it to quickly converge to a better decision. However, global reputation models are much more complex to manage than local approaches as malicious users have the opportunity to provide false feedback. Our focus is on using a local reputation model that accumulates only first-hand experience about Tor relays.

4. REPUTATION FRAMEWORK

Our goal is to use a local reputation model that can be used by a Tor user to filter out unreliable Tor relays. This section discusses the different components of the proposed PID controller-based reputation framework.

4.1 Reputation Score

To evaluate the reputation of a given Tor relay, a user keeps track of its own local experience with the relay through a reputation metric. We adopt the reputation metric proposed by Srivatsa et al. [36]. They propose a PID controller-based reputation framework which can handle strategic malicious behavior. A typical PID controller [31] is described by the following equation:

$$u(t) = \alpha \cdot e(t) + \beta \cdot \frac{1}{t}\int_0^t e(x)dx + \gamma \cdot e'(t) \qquad (2)$$

where $e(t)$ is the error (feedback) function, $u(t)$ is the control output, and $e'(t)$ is the first derivative of $e(x)$ at point t.

The first component of equation (2) (proportional) refers to the contribution of the current report. The second component (integral) represents the impact of past reports (historical aggregation). The third component (derivative) reflects the sudden change in the reputation score of a relay from the very recent past. Choosing a larger value for α means higher significance is given towards current feedback. A larger value of β gives heavier weight to the past performance. The averaging nature of the integral component enables the framework to tolerate errors in raw reputation scores and reflect consistent relay behavior. A larger value of γ amplifies sudden change in relay behavior from the recent past (as indicated by the derivative component) and handles sudden fluctuations in relay behavior.

In our case, feedback is obtained from discrete interactions with a relay and therefore we must adapt the equation into a discrete form. We follow the methodology of Srivasta et al. and incorporate each component of equation (2) in the following manner:

Incorporating Current Feedback: For simplicity we have used a binary rating system where a user rates a relay based on whether a circuit built through that relay was usable or not. One thing to remember, since it is hard to pin-point the relay responsible for the failure we update the reputation score of all the three relays participating in a Tor circuit.

$$R[i] = \begin{cases} 0.01, & \text{if circuit failed} \\ 1, & \text{otherwise} \end{cases} \qquad (3)$$

When a circuit successfully transmits user traffic each relay in the circuit obtains a high feedback score of 1, whereas if the circuit fails to transmit any user traffic each relay obtains a low feedback score of 0.01. Thus, for a bad/malicious performance relays obtain a score which is 100 folds worse than a productive performance.

Incorporating History: We now describe how we compute the integral (historical) component of the reputation score. Lets assume we are computing the historical reputation of relay n at interval i, denoted as $H[i]$ (representing a discrete version of the continuous integral in equation (2)). Suppose the system stores the reputation value of relay n over the last $maxH$ (maximum history) intervals, $H[i]$ could then be derived as a weighted sum over the last $maxH$ reputation scores of relay n using the following equation:

$$H[i] = \frac{\sum_{k=1}^{maxH} R[i-k] \cdot \omega_k}{\sum_{k=1}^{maxH} \omega_k} \qquad (4)$$

The weights ω_k are chosen as the reciprocal of the reputation score, $\omega_k = \frac{1}{R[i-k]}$. Such an evaluation assigns more importance to those interactions where the relay behaved particularly badly. We use Srivasta et al.'s fading memory optimization that summarizes the last $2^m - 1$ feedback values using only m variables (see their paper for the details of this technique [36]).

Incorporating Sudden Fluctuation: Once we have the current feedback-based reputation score (i.e., $R[i]$) for relay n at interval i and its past reputation score (i.e., $H[i]$), we can compute the derivative component ($D[i]$) as follows:

$$D[i] = R[i] - H[i] \qquad (5)$$

To penalize sudden drop compared to sudden rise in reputation, the positive and negative gradient of reputation score is assigned different weights. We do so to discourage oscillation in behavior and to ensure that relays exhibiting frequent oscillation obtain low reputation score over time. Hence, the final reputation score is computed using the following equation:

$$R_n[i] = \alpha \cdot R[i] + \beta \cdot H[i] + \gamma(D[i]) \cdot D[i] \qquad (6)$$

where $\gamma(x)$ is define as follows:

$$\gamma(x) = \begin{cases} \gamma_1, & \text{if } x >= 0 \\ \gamma_2, & \text{if } x < 0 \end{cases} \qquad (7)$$

with $\gamma_1 < \gamma_2$. In Section 6, we will explore the impact of α, β, and γ on the reputation score more elaborately.

5. FILTERING COMPROMISED RELAYS

In this section we describe how we use reputation score to filter out potentially compromised relays. If we assume a small fraction of all relays are compromised, we can show that the average reputation of honest relays will be higher than that of compromised ones. Thus, to filter out potentially compromised relays, we only need to find outliers in terms of reputation score. In this work we have looked at the following filtering scheme:

Mean-STD Filter: The client computes the average (μ) and standard deviation (σ) of the top (in terms of reputation score) $(1 - c)$ fraction of the relays he/she has interacted with (assuming c fraction of the relays are compromised). Next, filter out any relay n whose reputation score lies outside the range of ($\mu - k\sigma$, $\mu + k\sigma$). Here, k represents to what degree of deviation we are willing to tolerate from the expected reputation score. We filter outliers in both directions because when large fraction of the guards are compromised, compromised exits tend to obtain a higher reputation score (as majority of the circuits have a compromised guard in such scenario) compared to the other honest relays. Therefore, under our assumption a high reputation score does not always imply a trustworthy relay, rather a reputation score in the vicinity of the expected reputation score implies (probabilistically) a trustworthy relay.

We summarize the performance of the filtering technique in Section 6. From a security perspective, we are interested in cases when clients have some compromised and some honest guards. In such cases, we can adopt the following strategy with respect to selecting guards: *"Consider only the most reputable guard"*. The reason behind using the most reputable guard relay is that if one or two of the guards are compromised then their reputation score should be lower than that of the honest ones, so selecting the highest reputable guard helps to filter out potentially compromised guard(s).

Tor clients make use of the filtered list of Tor relays (after profiling a large set of Tor relays) for future circuit construction by following Tor's conventional bandwidth-proportional path construction protocol.

6. EXPERIMENTAL EVALUATION

In this section, we present a series of simulation results. First, we look at how the various parameters in the reputation metric impact the reputation score. Next, we investigate the false errors of the proposed filtering scheme. Finally, we evaluate the performance of both the reputation metric and filtering protocol by computing the probability of selecting compromised circuits once reputation-based filtering has been performed.

6.1 Sensitivity of Parameters

First, we look at the impact of the three parameters (α,β,γ) on the reputation score. To obtain a better understanding of how the reputation metric reacts to oscillating behavior, we compute the reputation score of a compromised relay (with all the other relays being honest) that participates in a total of 100 circuits oscillating between honest and malicious nature every 20 interactions. We analyze the following three cases where in each case one of the parameter dominates over the other two:

1. $\alpha \gg \beta, \gamma$
2. $\beta \gg \alpha, \gamma$
3. $\gamma \gg \alpha, \beta$

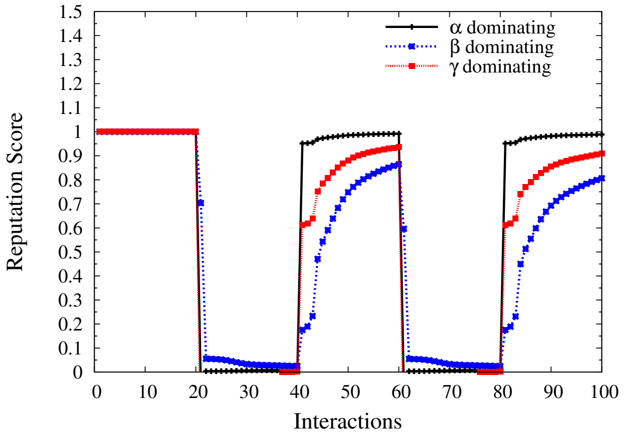


Figure 4: Analysis of parametric sensitivity. In our final model we set $\beta \gg \alpha$ and $\gamma_1 \ll \gamma_2$.

Figure 4 highlights the three scenarios. With α, dominating the reputation follows the actual behavior of the relay since it amounts to disregarding the history or the current fluctuations in the behavior (see equation (6)). With β, dominating the reputation metric gives more importance to the behavioral history of a relay and as a result the reputation score does not change very quickly. Instead it slowly and steadily adapts to its actual behavior. With γ, dominating the reputation score responds very swiftly to sudden changes in the behavior of a relay. Observe the steep jumps in the reputation score that correspond to the time instants when the relay changes its behavior.

For our final setting we want the reputation metric to assign low score to relays oscillating between good and malicious behavior, so we set β (historical component) to a higher value than α. We also want the reputation metric to react swiftly to sudden changes but more for sudden drop in performance than sudden rise, hence, we assign a higher value to γ_2 than γ_1 (i.e., $\gamma_2 >> \gamma_1$). The exact values used for the parameters are described in the following section.

6.2 Simulation Setup

We implemented a simulator that emulates the basic functionality of Tor circuit construction and active circuit dropping. Our simulator then profiles relays based on their circuit dropping characteristics.

Processing Input: We collected relay information (such as- IP address, advertised bandwidth and probability of the relay being selected for entry, middle and exit position) from the Tor compass project [2] and randomly assigned 20% of the bandwidth to be controlled by an adversary, thus setting c to 0.2. For our experimental setup we consider 3 guards, 23 middle relays and 23 exits. All the selected relays belong to a distinct /16 IP subnet and are chosen randomly proportional to their advertised bandwidth. The reason behind using 23 middle and exit relays is that we assume a user uses Tor for three hours continuously[4] and since a given circuit is alive for only 10 minutes, a user would need 18 circuits in a 3 hour period. With 23 middle and exit relays and a 20% of the bandwidth under control of the adversary, we would expect 18 relays in each category to be honest[5], allowing for the possibility of the client selecting different relays for each circuit.

We then create a total of $3 \cdot 23 \cdot 23 = 1587$ possible circuits and randomly select them one by one to determine the reputation of the selected relays. We randomly select circuits to hide temporal patterns during circuit construction. For example, if circuits are chosen sequentially then it might be easy for a malicious relay to realize that a user is constructing multiple circuits using the same entry and middle relay, and thus, behave strategically knowing that.

Experimental Design: Table 1 summarizes the parameters used in our simulations. We vary two environmental parameters (g,d) to analyze the robustness and effectiveness of the reputation metric against active selective DoS attacks. With $d = 100\%$, attackers always perform selective DoS and drop a circuit they cannot compromise, and when $d = 0\%$, circuits are never dropped (no attack). With $d = 50\%$, circuits that cannot be compromised are only dropped 50% of the time, in order to reduce the chances of acquiring a low reputation while still carrying out the attack. Circuits that *can* be compromised by attackers are never dropped. We only simulate the scenarios for $g = 1/3, 2/3$ (i.e., one or two out of the three guards are compromised), as $g = 0, 1$ are trivial scenarios. To approximate the circuit failure rate present in the current Tor network we use the TorFlow project [4]. The TorFlow project measures the performance of Tor network by creating Tor circuits and recording their failure rate. We run TorFlow's *buildtime.py* [4] python script to generate 10 000 Tor circuits and record their failure rate. We found the average failure rate over 10 runs to be approximately 21%. Thus, we set the circuit failure rate, f to 0.21 in all our simulations. All of our simulation results are averaged over 100 runs with 95% confidence interval.

Table 1: Simulation Parameters

	Parameter	Description	Value/Range
Computation Setting	α	Proportional gain	0.1
	β	Integral gain	0.9
	γ_1	Positive derivative gain	0.05
	γ_2	Negative derivative gain	0.2
Environment Setting	g	Fraction of compromised guards	$[0, 1/3, 2/3, 1]$
	d	Drop rate by compromised relay	$0 \leq d \leq 1$
	f	Transient network failure	0.21

6.3 Simulation Analysis

Evolution of Reputation Score: First, we look at how reputation score evolves for various drop date, d. Figure 5 shows the average reputation score with the 95% confidence interval for both honest and compromised relays. From figure 5(a) we see that as drop rate, d, increases the difference in reputation score between a honest and compromised relay increases for $g = 1/3$. This distinctive difference makes filtering easy. However, for $g = 2/3$ we see that both type of relays have similar reputation score. In the following section we will show that even for $g = 2/3$, we can successfully filter compromised relays if we assume honest relays dominate the total population. This suggests that the reputation metric can successfully capture the dropping characteristics of compromised relays.

Filtering Compromised Relays: To better understand how our filtering protocol sets cutoffs, we evaluate our proposed filtering scheme (as described in Section 5) where we first compute the average and standard deviation of the top 80% $(= 1 - c)$ of relays. For this simulation, we set $d = 1$ and compute the ranking score of all the relays by varying g. Figure 6 shows the ranking score of both honest and compromised relays for different numbers of compromised guards. As discussed in Section 5, we set cutoffs based on the average ranking score of the top 80% ranked relays. To filter outliers we exclude relays that are further than $k = 1.645$ standard deviations away from the average[6]. The dotted/dashed lines in the figure represent the boundaries of the acceptable region $(\mu - k\sigma, \mu + k\sigma)$. Figure 6 shows that as the number of compromised guards increases the distinction between honest and compromised relay shrinks. This is understandable because as the number of compromised guards increase, the ranking score for compromised relays also start to increase because more and more circuits with compromised guards are created. However, since honest relays dominate the total population, the average reputation score of the system lies close to the average reputation score of the honest relays. As a result, even with $g = 2/3$ we can successfully filter out a significant portion of the compromised relays.

[4] Tor users download the Tor consensus data every three hours, thus, it would make sense to refresh the relay list every three hours.

[5] Due to uneven bandwidth allocation, the actual number of honest relays could be significantly different.

[6] Due to space limitations we omitted the exploration of varying k. But intuitively as k is decreased more relays are filtered; making it harder for compromised relays to get accepted.

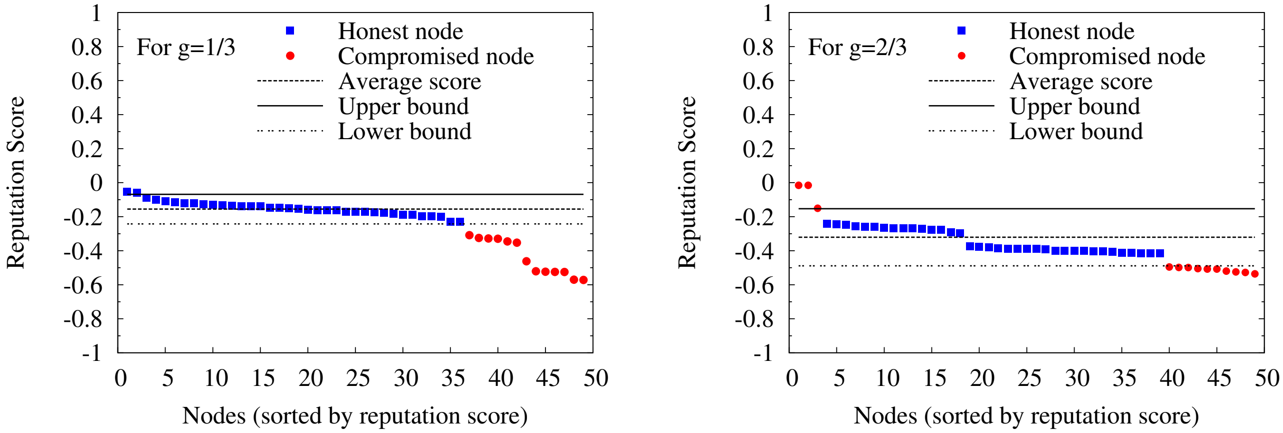


Figure 6: Ranking score of honest and compromised relays for various fractions of compromised guards. Relays that are $k = 1.645$ standard deviations away from the mean are considered outliers. We can see that even with $g = 2/3$ a client can successfully filter out a significant portion of the compromised relays.

Evaluating Robustness: In this section we present results that highlight the robustness of the reputation model in the presence of compromised relays. First, we look at false positive and false negative errors of our filtering protocol, and then we evaluate the probability of constructing compromised circuits under different drop rates.

6.3.1 *False Errors*

We define false negative (FN) and false positive (FP) error as follows:

- **FN:** Fraction of compromised relays in the accepted list.
- **FP:** Fraction of honest relays in the discarded list.

Figure 7 highlights the calculated FN and FP errors. Ideally you want both false errors to be low but since compromised relays are a minority and honest relays are plentiful, lowering FN is *more important* than lowering FP. Figure 7 shows that as the drop rate d increases, FN decreases. We see a similar trend for FP. This is expected because as the drop rate increases the distinction between compromised and honest relays becomes more clearer. Therefore, whether honest relays are heavily penalized (for $g \geq 2/3$) or rewarded (for $g \leq 1/3$), the average reputation score of the relays in the system shifts toward the ranking score of honest relays as the majority of the relays are assumed to be honest. That is why we see FP error reduce to almost 5%. These results indicate that carrying out active attacks like selective DoS is a losing proposal for an attacker.

6.3.2 *Probability of constructing compromised circuits*

Next, we evaluate the probability of constructing a compromised circuit once outliers have been discarded (we denote compromised circuits as CXC where both the entry and exit relays are compromised; C refers to a compromised relay and X refers to any type of relay). The probability of such an event is:

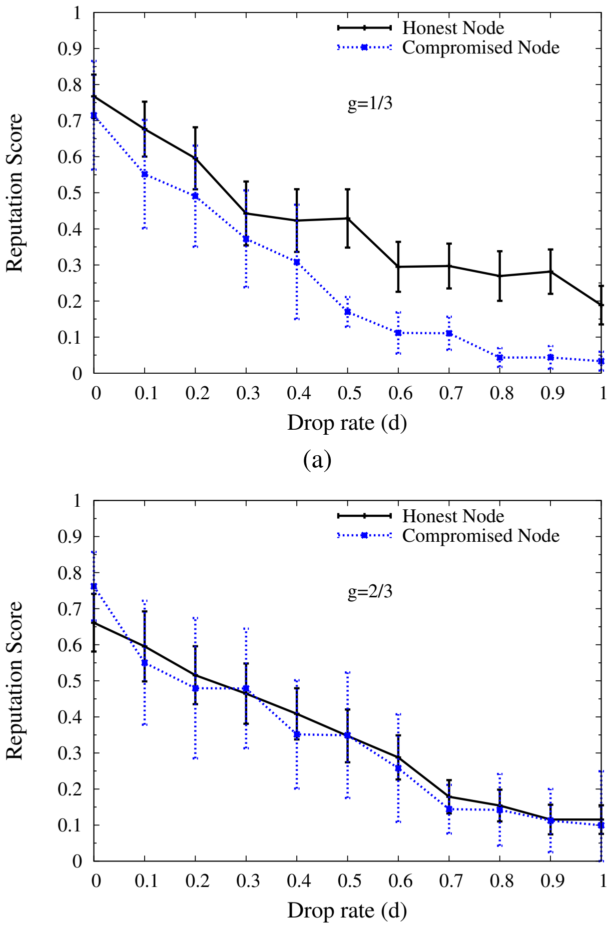
$$\frac{g_f c_f}{g_f c_f + (1-g_f)(1-c_f)^2 + (1-d)[1-g_f c_f - (1-g_f)(1-c_f)^2]} \tag{8}$$

where $g_f c_f$ refers to the fraction of circuits with a compromised guard and exit, while $(1-g_f)(1-c_f)^2$ refers to the fraction of circuits with all honest relays at each position in the circuit (g_f and c_f represent the fraction of guards and other relays that are compromised in the accepted list, respectively). Figure 8 shows the probability of constructing compromised circuits against different drop rate, d. For both $g = 1/3$ and $g = 2/3$ we see that our reputation based filtering protocol performs much better than what the conventional Tor guarantees (indicated by the dashed lines). This means that the PID controller-based reputation framework can effectively penalize compromised relays mounting active dropping and our proposed filtering scheme can then successfully filter such compromised relays. Thus, in the presence of the proposed reputation framework an attacker does not gain any significant benefit by performing deliberate circuit dropping.

6.4 Summary

A summary of our findings follow:

- The PID controller-based reputation metric can successfully penalize malicious oscillating-behavior.
- If majority of the Tor relays are honest (in our studies 80% of the relays were assumed to be honest) then even with two out the three guards being compromised our filtering protocol can effectively discard compromised relays.
- In the presence of our reputation-based filtering framework an attacker does not gain any significant benefit by performing deliberate circuit dropping.



(a)

(b)

Figure 5: Evolution of reputation score (with 95% confidence interval) for various drop rates. As drop rate increases the difference in reputation score between a honest and compromised relay increases for $g = 1/3$. But, for $g = 2/3$ the difference in much less.

7. RELATED WORK

Securing anonymity systems against active attacks is relatively a new research topic. Borisov et al. [12] first showed that a selective DoS attack can have devastating consequences for both high and low-latency anonymity systems.

More recently, Danner et al. [14, 15] proposed a detection algorithm for selective DoS attack in Tor. Their algorithm probes each individual Tor relay in the network and requires $O(n)$ probes to detect all compromised relays for a network comprising of n participants. However, to handle transient network failures they proposed repeating each probe r number of times, so their approach requires $O(nr)$ probes. Thus, at best their approach seems suitable for a centralized deployment. However, their approach assumes that compromised relays exhibit fixed characteristic of always dropping non-compromised circuits. They do not consider complex attack strategies where compromised relays may perform random dropping. Such dynamic malicious behavior could potentially increase the number of probes required to successfully identify the compromised relays.

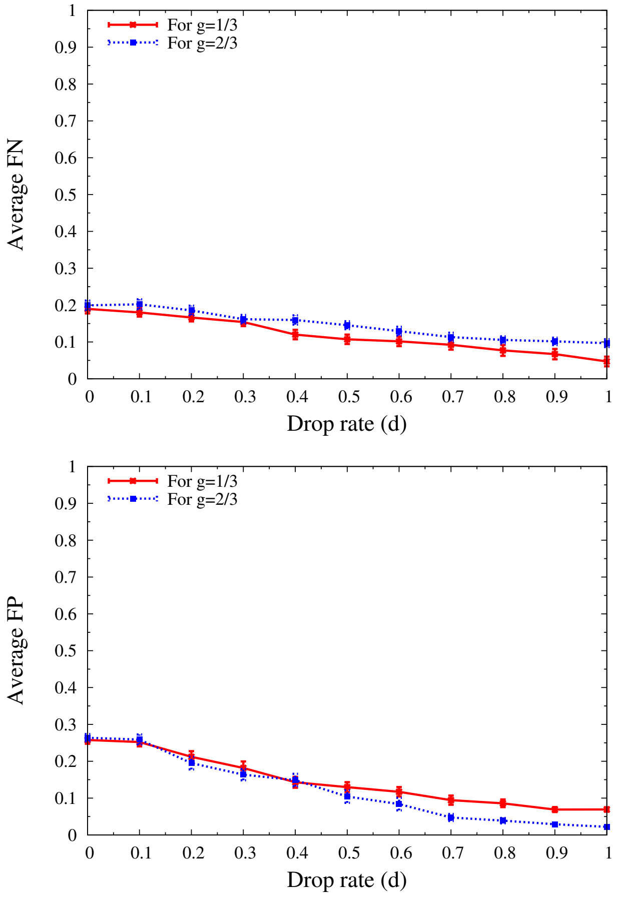


Figure 7: Average FN and FP with 95% confidence interval against drop rate d. Both FN and FP decrease as drop rate increases.

Mike Perry proposed a client-side accounting mechanism that tracks the circuit failure rate for each of the client's guards [3]. The goal is to avoid malicious guard relays that deliberately fail circuits extending to non-colluding exit relays. However, profiling only guards is not enough because it is less likely that an attacker will launch selective DoS at guard position, only to sacrifice the cost of obtaining a guard status (guards fulfill strong commitments like–minimum bandwidth, minimum uptime). Rather deploying a moderate number of cheap middle-only relays can boost the effect of the selective DoS [10].

Researchers have also leveraged incentive schemes [9,38] to encourage good behavior from Tor relays. All incentive schemes basically encourage participants to be cooperative by providing the cooperating participants with something that they care about; however, incentive schemes do not enforce malicious participants to behave properly.

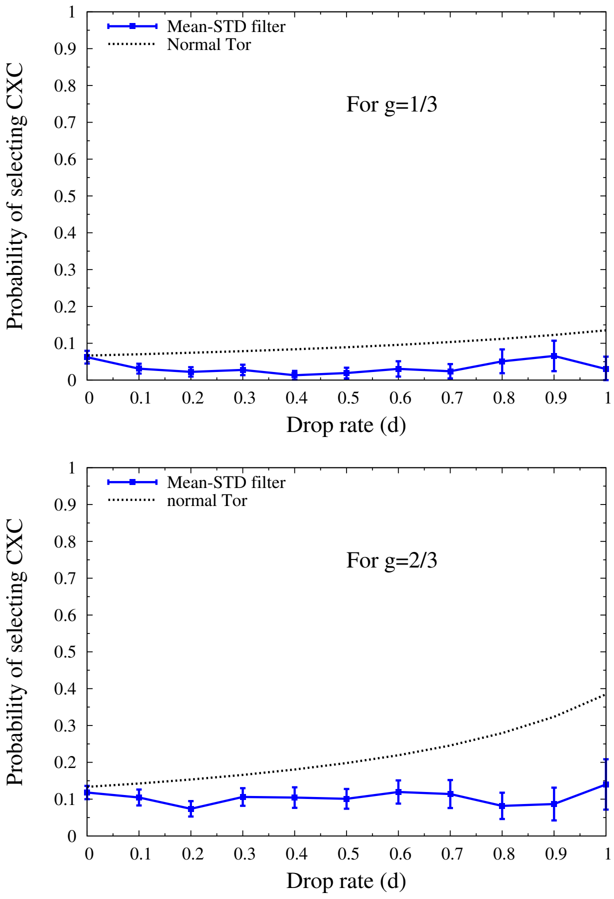


Figure 8: Probability of constructing compromised (CXC) circuits after filtering outliers. We see that our approach outperforms conventional Tor as drop rate increases.

There are reputation based routing protocols for wireless adhoc networks [28] that try to identify selfish/malicious routers with the intent to avoiding them during forward path setup. While these protocols have similar goal as ours there are different challenges in directly using them for anonymity systems. For example, in all of these protocols routers maintain reputation information about their neighbors which they share with other routers in the network. This information sharing could potentially introduce new attack vectors where an adversary could figure out which relays certain users are using. Moreover, to the best of our knowledge none of these protocols handle strategic malicious behavior.

There are many papers on reputation systems for P2P networks [25, 41, 42]. TrustGuard [36] proposes a reputation framework which is capable of handling strategic malicious behavior. But, most models focus on building distributed reputation systems, rather than worrying about privacy and anonymity as described in [34]. Dingledine et al. [16] described a reputation system for a mix-network environment [13]. But their approach relies on trusted witnesses which are hard to find in Tor network.

Control theory provides a systematic approach to designing closed loop systems that are stable and accurate even in the presence of oscillations. Control theory has been used in the design of many aspects of computing [7]. For example, in data networks control theory has been applied to flow control [26]; different versions of TCP/IP protocols have been designed using control theory [23]. Web-based recommender systems [24] have utilized control theory.

8. LIMITATIONS AND FUTURE WORK

Our work has a few limitations. First, in the absence of attacks, a small fraction of honest relays are classified as outliers due to random network failures. For anonymity systems, it is much more critical to blacklist malicious relays than to ensure that all honest relays are included. Moreover, these discarded honest relays should reflect either low performing or highly congested relays in absence of attack. Thus discarding them might actually help in shuffling the overall network load. Second, we only considered a specific type of circuit dropping, namely, selective DoS and its probabilistic variants. Other form of dropping could be employed by compromised relays; for example, they can target a specific set of relays to bring down their reputation score. We leave the analysis of such attack scenarios as future work. Finally, new users benefit from our reputation model only after a certain amount of usage.

9. CONCLUSION

Anonymity systems are vulnerable to active attacks like selective denial-of-service. Such attacks, however, can be detected by profiling relay behavior. We analyzed a generic reputation framework that profiles relays based on their historical behavior. Our PID controller-based reputation metric takes adaptive malicious behavior into consideration and penalizes any relay exhibiting such behavior. Our simulation results suggest that the proposed reputation framework can effectively filter out compromised relays mounting active attacks in the form of dropping non-compromised circuits. We conclude that with the PID controller-based reputation framework deployed an attacker does not gain any significant benefit by performing deliberate circuit dropping.

Acknowledgements

We would like to thank David M. Nicol for his valuable input and guidance in writing this paper. This paper reports on work that was supported in part by NSF CNS 0953655.

10. REFERENCES

[1] Known bad relays in Tor. https://trac.torproject.org/projects/tor/wiki/doc/badRelays.

[2] Tor Compass. https://compass.torproject.org/.

[3] Tor Proposal 209. https://gitweb.torproject.org/user/mikeperry/torspec.git/blob/path-bias-tuning:/proposals/209-path-bias-tuning.txt.

[4] TorFlow Project. https://gitweb.torproject.org/torflow.git.

[5] Trotsky IP addresses. https://trac.torproject.org/projects/tor/wiki/doc/badRelays/trotskyIps.

[6] Tor directory authorities compromised., 2010. https://blog.torproject.org/blog/tor-project-infrastructure-updates.

[7] ABDELZAHER, T., DIAO, Y., HELLERSTEIN, J., LU, C., AND ZHU, X. Introduction to Control Theory And Its Application to Computing Systems. In *Performance Modeling and Engineering*. Springer US, 2008, pp. 185–215.

[8] AKHOONDI, M., YU, C., AND MADHYASTHA, H. V. LASTor: A Low-Latency AS-Aware Tor Client. In *Proceedings of the 33rd IEEE Symposium on Security and Privacy* (2012), SP '12, IEEE Computer Society, pp. 476–490.

[9] ANDROULAKI, E., RAYKOVA, M., SRIVATSAN, S., STAVROU, A., AND BELLOVIN, S. PAR: Payment for Anonymous Routing. In *Proceedings of the 8th Symposium on Privacy Enhancing Technologies*, PETS'08. Springer Berlin Heidelberg, 2008, pp. 219–236.

[10] BAUER, K., JUEN, J., BORISOV, N., GRUNWALD, D., SICKER, D., AND MCCOY, D. On the Optimal Path Length for Tor, 2010. http://petsymposium.org/2010/papers/hotpets10-Bauer.pdf.

[11] BAUER, K., MCCOY, D., GRUNWALD, D., KOHNO, T., AND SICKER, D. Low-resource Routing Attacks Against Tor. In *Proceedings of the 6th ACM Workshop on Privacy in the Electronic Society* (2007), WPES '07, ACM, pp. 11–20.

[12] BORISOV, N., DANEZIS, G., MITTAL, P., AND TABRIZ, P. Denial of Service or Denial of Security? In *Proceedings of the 14th ACM Conference on Computer and Communications Security* (2007), CCS '07, ACM, pp. 92–102.

[13] CHAUM, D. L. Untraceable online mail, return addresses, and digital pseudonyms. *Commun. ACM 24*, 2 (1981), 84–90.

[14] DANNER, N., DEFABBIA-KANE, S., KRIZANC, D., AND LIBERATORE, M. Effectiveness and Detection of Denial-of-Service Attacks in Tor. *ACM Transactions on Information and System Security (TISSEC) 15*, 3 (November 2012), 11:1–11:25.

[15] DANNER, N., KRIZANC, D., AND LIBERATORE, M. Detecting Denial of Service Attacks in Tor. In *Proceedings of the 13th International Conference on Financial Cryptography and Data Security*, FC '09. Springer Berlin Heidelberg, 2009, pp. 273–284.

[16] DINGLEDINE, R., FREEDMAN, M., HOPWOOD, D., AND MOLNAR, D. A Reputation System to Increase MIX-Net Reliability. In *Proceedings of the 4th International Workshop on Information Hiding*. Springer Berlin Heidelberg, 2001, pp. 126–141.

[17] DINGLEDINE, R., AND MATHEWSON, N. Tor path specification. https://gitweb.torproject.org/torspec.git/blob/HEAD:/path-spec.txt.

[18] DINGLEDINE, R., AND MATHEWSON, N. Tor path specification. https://gitweb.torproject.org/torspec.git?a=blob_plain;hb=HEAD;f=path-spec.txt.

[19] DINGLEDINE, R., MATHEWSON, N., AND SYVERSON, P. Tor: The Second-generation Onion Router. In *Proceedings of the 13th USENIX Security Symposium* (2004), SSYM'04, USENIX Association.

[20] ECKERSLEY, P., GALPERIN, E., AND RODRIGUEZ, K. Dutch government proposes cyberattacks against... everyone., 2012. https://www.eff.org/deeplinks/2012/10/dutch-government-proposes-cyberattacks-against-everyone.

[21] EDMAN, M., AND SYVERSON, P. AS-awareness in Tor Path Selection. In *Proceedings of the 16th ACM Conference on Computer and Communications Security* (2009), CCS '09, ACM, pp. 380–389.

[22] ESTES, A. C. NSA attacks Tor. http://gizmodo.com/the-nsas-been-trying-to-hack-into-tors-anonymous-inte-1441153819.

[23] HOLLOT, C., MISRA, V., TOWSLEY, D., AND GONG, W.-B. A control theoretic analysis of RED. In *Proceedings of the 20th IEEE International Conference on Computer Communications* (2001), vol. 3 of *INFOCOM '01*, pp. 1510–1519.

[24] JAMBOR, T., WANG, J., AND LATHIA, N. Using Control Theory for Stable and Efficient Recommender Systems. In *Proceedings of the 21st International Conference on World Wide Web* (2012), WWW '12, ACM, pp. 11–20.

[25] KAMVAR, S. D., SCHLOSSER, M. T., AND GARCIA-MOLINA, H. The Eigentrust Algorithm for Reputation Management in P2P Networks. In *Proceedings of the 12th International Conference on World Wide Web* (2003), WWW '03, ACM, pp. 640–651.

[26] KESHAV, S. A Control-theoretic Approach to Flow Control. *SIGCOMM Comput. Commun. Rev. 21*, 4 (Aug. 1991), 3–15.

[27] LEVINE, B., REITER, M., WANG, C., AND WRIGHT, M. Timing Attacks in Low-Latency Mix Systems. In *Proceedings of the 8th International Conference on Financial Cryptography*. Springer Berlin Heidelberg, 2004, pp. 251–265.

[28] MICHIARDI, P., AND MOLVA, R. Core: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks. In *Proceedings of the 6th IFIP TC6/TC11 Joint Working*

Conference on Communications and Multimedia Security. Springer US, 2002, pp. 107–121.

[29] MUI, L., MOHTASHEMI, M., AND HALBERSTADT, A. A computational model of trust and reputation for e-businesses. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences* (Washington, DC, USA, 2002), HICSS '02, IEEE Computer Society.

[30] MURDOCH, S. J., AND ZIELIŃSKI, P. Sampled Traffic Analysis by Internet-exchange-level Adversaries. In *Proceedings of the 7th Symposium on Privacy Enhancing Technologies* (2007), PETS'07, Springer-Verlag, pp. 167–183.

[31] OZBAY, H. *Introduction to Feedback Control Theory*, 1st ed. CRC Press, Inc., 1999.

[32] PFITZMANN, A., AND HANSEN, M. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf, Aug. 2010. v0.34.

[33] REED, M., SYVERSON, P., AND GOLDSCHLAG, D. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications 16*, 4 (May 1998), 482–494.

[34] RESNICK, P., KUWABARA, K., ZECKHAUSER, R., AND FRIEDMAN, E. Reputation systems. *Commun. ACM 43*, 12 (2000), 45–48.

[35] SHMATIKOV, V., AND WANG, M.-H. Timing Analysis in Low-Latency Mix Networks: Attacks and Defenses. In *Proceedings of the 11th European Symposium On Research In Computer Security*, ESORICS'06. Springer Berlin Heidelberg, 2006, pp. 18–33.

[36] SRIVATSA, M., XIONG, L., AND LIU, L. TrustGuard: Countering Vulnerabilities in Reputation Management for Decentralized Overlay Networks. In *Proceedings of the 14th International Conference on World Wide Web* (2005), WWW '05, ACM, pp. 422–431.

[37] SYVERSON, P., TSUDIK, G., REED, M., AND LANDWEHR, C. Towards an Analysis of Onion Routing Security. In *Proceedings of International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability*. Springer Berlin Heidelberg, 2001, pp. 96–114.

[38] TSUEN-WAN, DINGLEDINE, R., AND WALLACH, D. Building Incentives into Tor. In *Proceedings of the 14th International Conference on Financial Cryptography and Data Security*, FC'10. Springer Berlin Heidelberg, 2010, pp. 238–256.

[39] WRIGHT, M., ADLER, M., LEVINE, B., AND SHIELDS, C. Defending anonymous communications against passive logging attacks. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy* (2003), SP '03, pp. 28–41.

[40] WRIGHT, M. K., ADLER, M., LEVINE, B. N., AND SHIELDS, C. An Analysis of the Degradation of Anonymous Protocols. In *Proceedings of the 9th Network and Distributed System Security Symposium* (2002), NDSS '02.

[41] XIONG, L., AND LIU, L. PeerTrust: supporting reputation-based trust for peer-to-peer online communities. *IEEE Transactions on Knowledge and Data Engineering 16*, 7 (2004), 843–857.

[42] ZHOU, R., AND HWANG, K. PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing. *IEEE Transactions on Parallel and Distributed Systems 18*, 4 (April 2007), 460–473.

[43] ZHU, Y., FU, X., GRAHAM, B., BETTATI, R., AND ZHAO, W. On Flow Correlation Attacks and Countermeasures in Mix Networks. In *Proceedings of the 4th International Workshop on Privacy Enhancing Technologies*, PET'04. Springer Berlin Heidelberg, 2005, pp. 207–225.