More Anonymous Onion Routing Through Trust

Aaron Johnson
Computer Science Department
Yale University
New Haven, CT 06520 USA
aaron.johnson@yale.edu

Paul Syverson
Center for High Assurance Computer Systems
U.S. Naval Research Laboratory
Washington, DC 20375 USA
syverson@itd.nrl.navy.mil

Abstract

We consider using trust information to improve the anonymity provided by onion-routing networks. In particular, we introduce a model of trust in network nodes and use it to design path-selection strategies that minimize the probability that the adversary can successfully control the entrance to and exit from the network. This minimizes the chance that the adversary can observe and correlate patterns in the data flowing over the path and thereby deanonymize the user. We first describe the general case in which onion routers can be assigned arbitrary levels of trust. Selecting a strategy can be formulated in a straightforward way as a linear program, but it is exponential in size. We thus analyze a natural simplification of path selection for this case. More importantly, however, when choosing routes in practice, only a very coarse assessment of trust in specific onion routers is likely to be feasible. Therefore, we focus next on the special case in which there are only two trust levels. For this more practical case we identify three optimal route-selection strategies such that at least one is optimal, depending on the trust levels of the two classes, their size, and the reach of the adversary. This can yield practical input into routing decisions. We set out the relevant parameters and choices for making such decisions.

Keywords

onion routing; anonymous communication; trust; minimax;

1. Introduction

When designing or analyzing anonymous communication networks, researchers generally assume that all nodes routing traffic are equally trusted. But this typically is incorrect. There is much information available to those selecting routes that can affect trust: information about who runs some components of the infrastructure, what computing platforms are used, how long and how reliably some components have been running, etc. And if routing designs were to begin taking trust into account, then even more extensive and diverse bases for trust might be available.

Onion routing is a type of anonymous communication that creates cryptographic circuits along an unpredictable route through a network of nodes called *onion routers* and passes traffic bidirectionally along those circuits with minimal latency [1], [2], [3]. An adversary observing an entry node and an exit node of an onion-routing network through which one is, e.g., browsing the web can easily link the two ends of the connection and correlate source to destination. This has been an acknowledged feature of the design since its inception [4]. Correlation is easily done with extremely high confidence by *passive timing*, that is, simply by observing the timing pattern of data entering the network and of data exiting the network and matching incoming and outgoing patterns. Correlation can also be done with *active timing*, where the adversary inserts unique patterns in incoming data and observes where they appear among outgoing data. It is this vulnerability of onion routing circuits to hostile pairs of entry and exit nodes that is our focus. There are many documented attacks that have some affect on onion routing—correlation, congestion, intersection, destination fingerprinting, latency, etc. None of the others have the efficiency or certainty that correlation does when an attacker owns so little of the network (i.e., just one entry node and one exit node) and observes so little traffic.

Correlation is, at least in this way, the most significant unaddressed problem for onion routing and one that can likely be improved with trust knowledge. (Correlation could be countered by mixing, padding, or other approaches; however, to date no proposed countermeasure has had both low enough overhead and high enough expectation of success against realistic attackers to be pursued in practice.) This introduces many questions, such as whether using more trusted nodes helps profile or identify clients and what to do about that, how to model diverse trust assumptions, etc. But even ignoring these, it is not obvious how to take advantage of trust as a criterion in route selection. In particular, using trusted nodes more often has the disadvantage of simultaneously providing a small set of nodes for the adversary attempt to monitor. This paper is specifically focused on whether there is a way to use trust to reduce the probability of a circuit compromise by endpoints.

Trust has many meanings and applications in computer security [5], [6], [7], [8], [9], [10], [11], [12]. Much of the literature is concerned in one way or another with propagation or transfer of trust from where it is to where it needs to be. Our concern is not with the transfer of trust information but with what it means in the context of onion routing and how to make use of it. We consider how trust associated with network nodes or links might be used to protect (or reveal) information that would undermine the anonymity of communicants.

Tor [13] is the current widely-deployed and used public onion-routing network, with an estimated quarter-million concurrent users and a few thousand network nodes. It is thus useful to consider trust issues that arise for this deployed network. For example, a correlating adversary could try to compromise nodes in the network. Because Tor nodes are run

by volunteers, however, an even easier attack is to simply set up hostile nodes and use those to attack traffic on the network. We have already noted that correlation attacks are strong and low cost. This shows us that they are also easy to deploy in practice.

One way Tor reduces the threat of linking exit activity to sources is by use of entry guards, a small number of nodes that a single client uses persistently to connect to the Tor network. If a client has chosen guard nodes that are not compromised, it can never be linked by correlation to its activity by a pair of compromised entry-exit nodes. When entry guards were introduced [14], there was a brief discussion of the relative merits of choosing guards randomly versus based on trust or other features of the guard nodes. So far, no one has analyzed the implications of choosing nodes based on trust. Entry guards are currently chosen randomly from the set of Tor nodes (subject to some performance and other criteria). Abusing entry-guard selection criteria can increase the chances of a node being chosen as an entry guard, especially if they are based on reliability, performance, etc. rather than based on any sort of trust. Many of the threats initially observed about this ([14], [15]) are not feasible in the current Tor network. Statistically, however, the percentage of all circuits compromised by hostile entry-exit pairs is not reduced by the use of randomly chosen entry guards, nor is the probability that any given client will have compromised guards; it only affects the distribution of compromised circuits over the client space. If one were able to choose not just guards but whole routes from a more trusted set of nodes, then one's threat of circuit compromise might be reduced. We hope through our analysis to show how best to add this protection to Tor and similar systems.

In this paper we first set out a simple model that should facilitate reasoning about using trust in routing. We define trust simply to be the probability that an attempt by the adversary to control a node fails. We include a roving adversary that can attempt to compromise a certain number of nodes. Route selection is modeled as a three-stage game in which the user first picks a distribution over paths, then the adversary chooses a set of nodes to attempt to compromise, and finally the user samples a path from his distribution. While we expect this model to bear further fruit, we use it in this paper to show a number of results of both theoretical and practical interest.

We consider various strategies for choosing first and last nodes in the network so as to minimize the maximum probability a correlating adversary has for linking source to destination. We first look at the general case, in which there is an arbitrary number of trust levels. We observe that a straightforward algorithm to calculate an optimal distribution runs in time exponential in the size of the adversary. We consider a natural simplification of looking at distributions on individual nodes rather than pairs of nodes and considering the product distribution as an approximation of the joint distribution on pairs. We find two optimal distributions over single nodes, but we then show that optimal distributions on pairs are arbitrarily better than products of those optimal distributions on single nodes.

In practice, it is unlikely that one can realistically assign many different levels of trust, and so we next consider restricting to the case where there are only two trust levels for nodes in the network. Here we find three distributions and prove that in every case one of them must be optimal. Lastly, we discuss determining in practice when one of the three distributions is optimal based on the values of the system variables: trust values, size of the trusted and untrusted sets, and the size of the adversary.

2. An uncompromising model of node trust

A user wants to use a network of onion routers for anonymous communication. He trusts some onion routers more than others in the sense that he trusts that they are less likely to attempt to compromise his anonymity. How should he take this trust into account when he selects his paths?

2.1. The model

To make this question concrete, we need to make the notions of trust, anonymity, and an adversary precise.

Let R be the set of routers, $|R| = n$. Let there be an adversary that is trying to compromise the user's anonymity. The adversary selects k routers in R that he will attempt to compromise and use for deanonymization. If a router is not selected, it cannot be used by the adversary in an attack.

When an onion router i is selected, the adversary fails to compromise it with probability t_i. This represents the user's trust in the router. It will be convenient to define $c_i = 1-t_i$, the probability that the adversary *does* successfully compromise router i when he attempts to do so.

A user selects a path for a circuit from some probability distribution. If the adversary has selected and successfully compromised the first and last nodes on the chosen path, the user has no anonymity. Otherwise, the user's connection is anonymous. Therefore, to calculate anonymity, we need only look at the user's distribution over entry-and-exit-node pairs.

We would like to find the probability distribution over pairs of routers that minimizes the chance that both members of the pair are selected by the adversary and successfully compromised. More precisely, we want to find $p \in \Delta_{n(n-1)/2}$, that is, a probability distribution p over pairs in R, that minimizes

$$c(p) = \max_{K \subseteq R : |K| = k} \sum_{\{r,s\} \in \binom{K}{2}} p(r,s) c_r c_s$$

For a set S and $j \leq |S|$, we use $\binom{S}{j}$ to represent the collection of all subsets of S of size j. Also, for convenience, we write $p(\{r,s\})$ as $p(r,s)$.

2.2. The adversary

Attackers of limited size have long been countenanced in the security and fault-tolerance literature. While caution might

suggest designing against an adversary that can compromise the entire network as a worst case, usable results are often broken against such an adversary. And, especially for large diverse networks, it is typically unrealistic to assume that an adversary has such reach. System and protocol designs have been shown to provide a guarantee against various types of failure or compromise as long as no more than some fixed threshold of nodes is compromised at any time, e.g., Byzantine fault-tolerance.

The particular partial-network adversary from which our work derives is the roving adversary of Ostrovsky and Yung [16]. They introduced and were motivated by the concept of proactive security, in which an adversary could compromise arbitrary optimal sets of nodes given his current information. The roving adversary can potentially compromise every node in the network, but it can compromise no more than a fixed maximum number of nodes at any one time. Proactive security is concerned with properties that are resilient to such attacks. This can be useful for secret sharing and other distributed applications. The adversary model was applied to onion routing by Syverson et al. [4].

We alter the basic roving adversary model in two ways. First, to incorporate trust we add the idea that an adversary does not always succeed when attempting to compromise a node. Second, the adversary selects only one set to attack—there is no roving. It may be useful to bring roving back in for future work. Though likely of limited use for individual correlation attacks (given the typically short duration of onion-routing circuits), roving could allow the adversary to learn various communication and trust properties of the network and its users.

The adversary is assumed to have prior knowledge of the distribution that is used to pick a route, and he uses this knowledge to pick the set of nodes that he will attempt to compromise. It is realistic in many settings to assume the adversary has such knowledge. For example, the probability distributions may be set in some software or common system parameters given to a wide group in which there is at least one compromised member. The adversary may also be able to infer trust information from outside knowledge about the user.

2.3. Trust

Trust is captured in our model with the probability t_i that the adversary's attempt to compromise a node fails. This notion accommodates several different means by which users in the real world might trust an onion router.

The probability might represent the user's estimate of how likely it is that the operator of a given node is trying to provide, rather than break, anonymity. It might represent the user's faith in the security of a given node against outside attack.

To arrive at such conclusions, the users must rely on some outside knowledge. This might include knowledge of the organizations or individuals who run nodes, both knowledge of their technical competence and the likelihood of themselves harboring ill intent. It also includes knowledge of computing platforms on which a network node is running, geopolitical information about the node, knowledge about the hosting facility where a node might be housed or the service provider(s) for its access to the underlying communications network, etc.

Admittedly, it may not be the case that one can realistically assign specific probabilities to each node in the network separately. It is for this reason that we consider in sections 5 and 6 restriction to just two trust levels. Even if one cannot be certain of the probability of compromise to assign at one level or another, one may be in a position to know the divergence of those levels. This is particularly the case if one is considering nodes run by, e.g., security or law-enforcement agencies of friendly governments or their contractors vs. the rest of the nodes on the network. Alternatively one can imagine sets of nodes run by reputable human rights groups, NGOs, or human rights agencies of friendly governments.

Unlike many other areas, network performance or reliability reputation are not good bases for trust for anonymous communication. That is because an adversary that is focused on learning as much as possible about communication patterns has incentive to run the highest performing, most reliable nodes in the network. Thus, many of the usual metrics do not apply. The relation however is subtle because failure to consider performance at all would always result in the optimal choice being a secure brick [17].

2.4. Anonymity

We will consider a user to be anonymous unless the adversary has compromised the first and last routers on his path. This is motivated by the correlation attacks mentioned above. The model does not include some other methods the adversary can use, for example congestion attacks [18], [19], denial-of-service attacks [20], latency [21], or destination fingerprinting [22], [23]. It also does not take into account the total effect of an adversary's actions on a user's anonymity, such as the analysis performed in [24]. The attacks on which we focus are conceptually much simpler than these others, but more importantly, as noted in Section 1, none of these other attacks succeeds with as much certainty using as little resources as this one. Note that such entry-exit correlation attacks could also be done by the links from source to the entry onion router on the entry side and links from the exit onion router to the destination on the exit side (or by the destination itself). For example, an autonomous system or internet exchange on these links could participate in a correlation attack [25], [26]. We focus, however, on just the attack as it can be done by network nodes. Besides simplifying analysis, this is reasonable to model as a practical attack given the ease with which nodes can be added to the network.

Using this model, the user's selection of the pair constituting the first and last onion routers on his path is the only relevant factor in his anonymity. The user may make this selection using any probability distribution p over pairs of routers.

2.5. Objective function

We set as our objective function to find the distribution on pairs of routers that minimizes the probability of circuit compromise over all possible sets that the adversary could choose:

$$\min_{p \in \Delta_{n(n-1)/2}} \max_{K \subseteq R : |K| = k} \sum_{\{r,s\} \in \binom{K}{2}} p(r,s) c_r c_s.$$

This provides a worst-case guarantee, and if the user has a distribution with a low worst-case value, he is guaranteed anonymity with high probability regardless of the adversary's actions. As a worst-case criterion, however, it may direct the user to protect against adversarial actions that are unlikely. Indeed, while the adversary's goal is to find the subset $K \subseteq R$ that maximizes his chance of compromise, it is easy to see that this problem in general is equivalent to the **NP**-hard problem CLIQUE. Therefore the adversary may fail in many cases to actually select the worst-case set.

3. Strategies for the general case

Given arbitrary trust values $t_1, \ldots, t_n$, we would like to find a polynomial-time algorithm that takes as input the trust values and outputs an optimal or near-optimal distribution p^*.

3.1. Exact algorithm

There is a straightforward formulation of this problem as a linear program. Let the set of variables be p_{ij}, $i, j \in R$. The following constraints ensure that p is a probability distribution:

$$\sum_{\{r,s\} \in \binom{R}{2}} p_{rs} = 1$$
$$0 \le p_{rs} \le 1 \qquad \text{for all } \{r,s\} \in \binom{R}{2}.$$

We want to find the distribution that satisfies the minimax criterion

$$\min_p \max_{K \in \binom{R}{k}} \sum_{\{r,s\} \in \binom{K}{2}} c_r c_s p(r,s).$$

For any fixed K, the sum

$$c(p,K) = \sum_{\{r,s\} \in \binom{K}{2}} p(r,s) c_r c_s$$

is linear in p. Therefore the minimax criterion minimizes the maximum of linear functions. We can thus transform it into a simple minimization problem by adding a slack variable t and some linear constraints. We force t to be greater than the maximum of our linear functions:

$$t - c(p,K) \ge 0 \quad \text{for all } K \in \binom{R}{k}$$

Then the objective function is simply $\min t$. Unfortunately, this linear program is of exponential size $\big(O(n^k)\big)$ because of the constraints for each subset.

3.2. Choosing a simple distribution

A straightforward simplification is to consider restricting the output to be a distribution in which the first and last routers are chosen independently and identically at random and then minimizing the probability that they are individually compromised.

Let p_R be a distribution on R. We consider the distribution p_R^* that minimizes the probability that an adversary chooses and successfully compromises a single router:

$$\begin{aligned} c(p_R) &= \max_{K \in \binom{R}{k}} \sum_{r \in K} p_R(r) c_r \\ p_R^* &= \operatorname*{argmin}_{p_R} c(p_R) \end{aligned}$$

The following theorem states that it is always optimal either to put all the probability on the most trusted router or to set the probabilities such that the values $c_i p_R(r_i)$ are equal for all $r_i \in R$.

Theorem 1: Let $c_\mu = \min_j c_j$. Let p_R^1 put all the probability on the most trusted router:

$$p_R^1(r) = \begin{cases} 1 & \text{if } r = r_\mu \\ 0 & \text{otherwise} \end{cases}$$

Let p_R^2 set probability inversely proportional to c_i:

$$p_R^2(r_i) = \alpha / c_i$$

where $\alpha = \left(\sum_i 1/c_i\right)^{-1}$.

Then

$$c(p_R^*) = \begin{cases} c(p_R^1) & \text{if } c_\mu \le k\alpha \\ c(p_R^2) & \text{otherwise} \end{cases}$$

Proof: Suppose p_R is an optimal distribution. Sort the routers so that $c_1 p_R(r_1) \ge c_2 p_R(r_2) \ge \ldots \ge c_n p_R(r_n)$. The set K that maximizes $\sum_{r \in K} c_r p_R(r)$ is then $\{r_1, r_2, \ldots, r_k\}$, and the value of p_R is $c(p_R) = \sum_{i=1}^k c_i p_R(r_i)$.

Let l be the largest index such that $c_l p_R(r_l) = c_k p_R(r_k)$.

If $l < n$, we could decrease $c_i p_R(r_i), k \le i \le l$ by moving $\epsilon c_k / c_i$ probability from r_i to r_{l+1}. This decreases $c_i r_i$ by $c_k \epsilon$ and increases $c_{l+1} p_R(r_{l+1})$ by $\epsilon c_{l+1} c_k / c_i$. For small enough ϵ we maintain that if $i < j$ then $c_i p_R(r_i) \ge c_j p_R(r_j)$, and therefore we reduce the value $c(p_R)$. Therefore p_R cannot be optimal, contradicting our assumption.

Thus it must be that $l = n$. Let m be the smallest index such that $c_m p_R(r_m) = c_k p_R(r_k)$. Assume that p_R is an optimal distribution that has the smallest m possible.

If $m = 1$, we are in the case that $c_i p_R(r_i) = c_j p_R(r_j)$ for $1 \le i, j \le n$. This is the distribution p_R^2.

Suppose $m > 1$. If $p_R(r_m) = 0$, then $c(p_R) = \sum_{i=1}^{m-1} c_i p_R(r_i)$. Let $c_\mu = \min_i c_i$. Because all of the probability is contained in a set that the adversary can completely select, we do not increase $c(p_R)$ by moving all the probability

to r_μ:

$$\begin{aligned} c(p_R) &= \sum_{i=1}^{m-1} c_i p_R(r_i) \\ &\geq \sum_{i=1}^{m-1} c_\mu p_R(r_i) \\ &= c_\mu. \end{aligned}$$

c_μ is equal to $c(p_R^1)$.

Now consider the case that $p_R(r_m) > 0$. Recall that $c_i p_R(r_i) = c_j p_R(r_j)$ for all pairs r_i,r_j, in the set $S = \{r_i, m \leq i \leq n\}$. Consider moving probability between r_{m-1} and S in a way that maintains the equality of $c_i p_R(r_i)$ for $r_i \in S$. This can be achieved by setting the probability of r_{m-1} to

$$p'_R(r_{m-1}, t) = p_R(r_{m-1}) + t$$

and the probability of $r_i \in S$ to

$$p'_R(r_i, t) = p_R(r_i) - \frac{t}{c_i(\sum_{r_j \in S} 1/c_j)}.$$

For small enough values of t, this preserves the property that if $i > j$ then $c_i p'_R(r_i, t) \leq c_j p'_R(r_j, t)$. Therefore $c(p'_R) = \sum_{i=1}^{k} c_i p'_R(r_i, t)$. The fact that p'_R is linear in t makes $c(p'_R)$ also linear in t for small enough values of t.

If $D_t c(p'_R)|_{t=0} \geq 0$, then for $t < 0$ large enough $c(p'_R)$ doesn't increase. This corresponds to moving probability from r_{m-1} to S, and the smallest t that maintains the ordering by $c_i p'_R(r_i)$ results in $c_{m-1} p'_R(r_{m-1}) = c_m p'_R(r_m)$. This contradicts the assumption about the minimality of the index m.

If $D_t c(p'_R)|_{t=0} \leq 0$, then for $t > 0$ small enough $c(p'_R)$ doesn't increase. This corresponds to moving probability from S to r_{m-1}. In fact, no positive value of t increases $c(p'_R)$. This is because setting $t > 0$ decreases the probability of all r_i, $i > k$, and only increases the probability of $r_{m-1}, m \leq k$, and thus preserves the fact that $c(p'_R) = \sum_{i=1}^{k} c_i p'_R(r_i, t)$. Therefore we can increase t until $c_i p'_R(r_i) = 0$ for all $r_i \in S$. This puts us in the case where $p'_R(r_m) = 0$, which we have already shown implies that $c(p'_R) \geq c(p_R^1)$.

Thus we have shown that either p_R^1 or p_R^2 is an optimal distribution. $c(p_R^1) = c_1$ and $c(p_R^2) = k\alpha$. Therefore, if $c_1 \leq k\alpha$, $c(p_R^*) = c(p_R^1)$, and otherwise $c(p_R^*) = c(p_R^2)$. □

We might hope that the product distributions $p_R^1 \times p_R^1$ and $p_R^2 \times p_R^2$ over $R \times R$ are good approximations to an optimal distribution p^*. However, this is not the case, and we can find inputs such that $c(p_R^i)/c(p^*)$, $i \in \{1, 2\}$, is arbitrarily high. In fact, we can show this for slightly improved distributions p^1 and p^2 over $\binom{R}{2}$.

Notice that $p_R^i \times p_R^i$, $i \in \{1, 2\}$, puts positive probability on the user choosing the same router twice. The problem as formulated in Section 2 allows distributions only over distinct pairs in $\binom{R}{2}$. This doesn't affect the optimum, however. There is always an optimal distribution that puts zero probability on $(r, r) \in R \times R$. Let p be a distribution on $R \times R$. Then let

$$p'(r, s) = \begin{cases} 0 & \text{if } r = s \\ p(r, s) + q_{rs} & \text{otherwise} \end{cases}$$

where for all $r \in R$, $\sum_{s \neq r} q_{rs} = p(r, r)$.

Lemma 2: $c(p') \leq c(p)$ □

Now assume that $c_1 \leq c_2 \leq \ldots \leq c_n$ and consider two distributions over $\binom{R}{2}$:

$$p^1(r, s) = \begin{cases} 1 & \text{if } r = c_1 \wedge s = c_2 \\ 0 & \text{otherwise} \end{cases}$$

and

$$p^2(r, s) = \frac{\alpha}{c_r c_s}$$

where $\alpha = \left(\sum_{\{r,s\} \in \binom{R}{2}} 1/(c_r c_s)\right)^{-1}$. By Lemma 2 $c(p^1) \leq c(p_R^1)$ and $c(p^2) \leq c(p_R^2)$.

Now let $\mathcal{I}_n = (c_1, \ldots, c_n, k)$ be a problem instance that, as n grows, satisfies

1) $c_1 = O(1/n)$.
2) $c_2 > c$ for some constant $c \in (0, 1)$.
3) $k = o(n)$
4) $k = \omega(1)$

For large enough n, $\mathcal{I}_n$ has an optimal value that is arbitrarily smaller than the values achieved by p^1 and p^2. Let $c(\mathcal{I}_n, p)$ be the value of $\mathcal{I}_n$ under distribution p.

Theorem 3:

$$c(\mathcal{I}_n, p^1)/c(\mathcal{I}_n, p^*) = \Omega\left(\frac{n}{k}\right) \quad (1)$$

$$c(\mathcal{I}_n, p^2)/c(\mathcal{I}_n, p^*) = \Omega(k) \quad (2)$$

Proof: The following distribution achieves the ratios in Eqs. 1 and 2. Let

$$p^3(r, s) = \begin{cases} \frac{\alpha}{c_r c_s} & \text{if } r = r_1 \\ 0 & \text{otherwise} \end{cases}$$

where $\alpha = \left(\sum_{i>1} 1/(c_1 c_i)\right)^{-1}$. This distribution puts weight on all distinct pairs that include r_1. It represents a middle approach between putting all the probability on the lightest pair, as p^1 does, and spreading the probability over all pairs, as p^2 does. The optimal distribution for each $\mathcal{I}_n$ only has higher ratios with p^1 and p^2 than p^3 does.

The ratio between p^1 and p^3 is

$$\begin{aligned} \frac{c(\mathcal{I}_n, p^1)}{c(\mathcal{I}_n, p^3)} &= \frac{c_1 c_2}{(k-1)/\left(\sum_{i=2}^{n} 1/(c_1 c_i)\right)} \\ &\geq (1 + c_2(n-2)/c_n)/(k-1) \\ &= \Omega\left(\frac{n}{k}\right). \end{aligned}$$

The ratio between p^2 and p^3 is

$$\begin{aligned}
\frac{c(\mathcal{I}_n, p^2)}{c(\mathcal{I}_n, p^3)} &= \frac{\binom{k}{2}\left(\sum_{i\neq j} 1/(c_i c_j)\right)^{-1}}{(k-1)/\left(\sum_{i=2}^{n} 1/(c_1 c_i)\right)} && (3)\\
&= \frac{k}{2}\left(1 + c_1 \frac{\sum_{2\leq i<j\leq n} 1/(c_i c_j)}{\sum_{i=2}^{n} 1/c_i}\right)^{-1} && (4)\\
&\geq \frac{k}{2}\left(1 + \frac{c_1}{2}\left(\sum_{i=2}^{n} 1/c_i - 1\right)\right)^{-1} && (5)\\
&= \Omega(k). && (6)
\end{aligned}$$

In Eq. 5, $\sum_{i=2}^{n} 1/c_i$ is bounded by n because $c_i > c$, $i > 1$. The last line then follows because $c_1 = O(1/n)$. □

Intuitively, the reason p^1 does arbitrarily worse than p^3 is that it doesn't take advantage of an adversary of size $o(n)$ by putting probability on $\Omega(n)$ pairs, while p^2 does arbitrarily worse than p^3 because it puts probability on pairs $\{r_i, r_j\}$, $i, j > 1$, that have $\Omega(n)$ times higher probability of being successfully compromised than pairs including r_1.

4. When pairing off, trust is everything

Allowing arbitrary trust values may be unnecessarily general. Users are unlikely to have precise knowledge of the probability of compromise for each onion router in the network. Instead, they seem more likely to have a few classes of trust into which they can partition the routers, or to have detailed knowledge about only a small number of routers. This fact may help us deal with the apparent computational intractability of the general problem. Also, the potentially complicated optima that result from arbitrary trust values may not satisfy other criteria for path-selection strategies that our problem formulation does not include. For example, we may want the number of possible optimal strategies to be small so users share their behavior with many others, or we may want the strategies to be robust to small changes in trust values.

Therefore, we now consider the case that there are only two trust values. We refer to the nodes with higher trust as the *trusted* set, and nodes with lower trust as the *untrusted* set. This case is simple yet results in non-obvious conclusions, and also still provides practical advice to users.

In Section 5 we show that, when there are only two trust values, there are three strategies that are potentially optimal. But first we give here a lemma that allows us to consider only distributions that treat the routers within a trust set identically. Note that this lemma holds for general trust values.

Lemma 4: Let U be a set of routers with identical trust values c, where $|U| = m$. Let V be the rest of the routers, where $|V| = n$. Then the set of routers is $R = U \cup V$. There exists an optimal distribution p in which the following hold:

1) For all $\{u, v\}, \{w, x\} \in \binom{U}{2}$, $p(u, v) = p(w, x)$.
2) For all $v \in V$, $u, w \in U$, $p(v, u) = p(v, w)$.

Proof: Consider some distribution over pairs $p : \binom{R}{2} \to [0, 1]$, $\sum_{\{r,s\}\in\binom{R}{2}} p(r, s) = 1$. Consider any subset $S \subseteq V$. Let X_S be a subset chosen randomly from all subsets X of size k such that $X \cap V = S$. Let $j = k - |S|$ be the size of $X_S \cap U$. Let $c(p, K)$ be the probability of compromise under p, given that set K is chosen by the adversary. That is,

$$c(p, K) = \sum_{\{r,s\}\in\binom{K}{2}} p(r, s) c_r c_s$$

We can calculate the expected probability of compromise of X_S as follows:

$$\begin{aligned}
&\mathbb{E}\left[c(p, X_S)\right]\\
&= \left\{\binom{m}{j}^{-1} \sum_{T\subseteq\binom{U}{j}} \left[\begin{array}{l} \sum_{\{t,u\}\in\binom{T}{2}} p(t, u)c^2 + \\ \sum_{u\in T, v\in S} p(u, v)c \cdot c_v + \\ \sum_{\{v,w\}\in\binom{S}{2}} p(v, w)c_v c_w \end{array}\right] \right. && (7)\\
&= \left\{\begin{array}{l} \binom{m}{j}^{-1}\binom{m-2}{j-2}c^2 \sum_{\{t,u\}\in\binom{U}{2}} p(t, u) + \\ \binom{m}{j}^{-1}\binom{m-1}{j-1}c \sum_{v\in S, u\in U} p(v, u)c_v + \\ \sum_{\{v,w\}\in\binom{S}{2}} p(v, w)c_v c_w \end{array}\right. && (8)\\
&= \left\{\begin{array}{l} \frac{j(j-1)c^2}{m(m-1)} \sum_{\{t,u\}\in\binom{U}{2}} p(t, u) + \\ \frac{j\cdot c}{m} \sum_{v\in S, u\in U} p(v, u)c_v + \\ \sum_{\{v,w\}\in\binom{S}{2}} p(v, w)c_v c_w \end{array}\right. && (9)
\end{aligned}$$

There must be some set $T \subseteq U$ of size j such that $c(p, S \cup T)$ is at least the expectation expressed in Eq. 9. If we modify p to treat all nodes in U the same, and thus satisfy the conditions in the statement of the lemma, every such T achieves the value in Eq. 9. Let p' be this modified distribution:

$$p'(r, s) = \begin{cases} \sum_{\{t,u\}\in\binom{U}{2}} p(t, u)/\binom{m}{2} & \text{if } \{r, s\} \in \binom{U}{2} \\ \sum_{u\in U} p(r, u)/m & \text{if } r \in V, s \in U \\ \sum_{u\in U} p(s, u)/m & \text{if } r \in U, s \in V \\ p(r, s) & \text{if } \{r, s\} \in \binom{V}{2} \end{cases}$$

The probability of compromise for any value $S \cup T$ of X_S is

$$\begin{aligned}
c(p', S \cup T) =& \binom{j}{2}\binom{m}{2}^{-1} \sum_{\{t,u\}\in\binom{U}{2}} p(t, u)c^2 + \\
& \frac{j}{m} \sum_{v\in S}\sum_{u\in U} p(v, u)c_v c + \\
& \sum_{\{v,w\}\in\binom{S}{2}} p(v, w)c_v c_w. && (10)
\end{aligned}$$

Equations 9 and 10 are equal, and therefore $\max_{T:|T|=j} c(p', S \cup T) \leq \max_{T:|T|=j} c(p, S \cup T)$. Because this holds for all $S \subseteq V$, $\max_{K:|K|=k} c(p', K) \leq \max_{K:|K|=k} c(p, K)$. □

5. Choosing pairs to avoid compromise

> "*Dear Abby, Dear Abby, Well I never thought, that me and my girlfriend would ever get caught.*"
>
> *John Prine* — Lyrics to "Dear Abby"

Now we analyze optimal distributions for selecting pairs when there are two trust values in the network, c_1 and c_2, with $c_1 \leq c_2$. We show that, in this case, one of the following strategies is always optimal: (*i*) choose a pair of trusted routers uniformly at random, (*ii*) choose pairs such that $p(r,s)c_r c_s$ is equal for all $\{r,s\} \in \binom{R}{2}$, or (*iii*) choose only fully-trusted or fully-untrusted pairs such that the adversary has no advantage in attacking either trusted or untrusted routers. Distribution (*i*), corresponds to distribution p^2, described in Section 3.2, with the difference that (*i*) spreads probability to all the most-trusted routers and not just two. Distribution (*ii*) corresponds to distribution p^1 of Section 3.2. Distribution (*iii*) shows that non-obvious distributions can exist even when the trust values are very restricted.

Let U be the trusted set, with trust value c_1, $|U| = m$. Let V be the untrusted set, with trust value c_2, $|V| = n$.

Theorem 5: Let $v_0 = \max(k-m, 0)$ and $v_1 = \max(k-n, 0)$. Then let $g_0 = \frac{v_0(v_0-1)}{n(n-1)}$ and $g_1 = \frac{v_1(v_1-1)}{m(m-1)}$. One of the following is an optimal distribution:

$$p(r,s) = \begin{cases} \frac{(c_2)^2}{\binom{m}{2}(c_2)^2 + (mn)(c_1c_2) + \binom{n}{2}(c_1)^2} \\ \qquad \text{if } \{r,s\} \in \binom{U}{2} \\ \frac{(c_1c_2)}{\binom{m}{2}(c_2)^2 + (mn)(c_1c_2) + \binom{n}{2}(c_1)^2} \\ \quad \text{if } (r,s) \in U \times V \cup V \times U \\ \frac{(c_1)^2}{\binom{m}{2}(c_2)^2 + (mn)(c_1c_2) + \binom{n}{2}(c_1)^2} \\ \qquad \text{if } \{r,s\} \in \binom{V}{2} \end{cases} \tag{11}$$

$$p(r,s) = \begin{cases} \binom{m}{2}^{-1} & \text{if } \{r,s\} \in \binom{U}{2} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

$$p(r,s) = \begin{cases} \binom{m}{2}^{-1} \frac{c_2^2(1-g_0)}{c_1^2(1-g_1) + c_2^2(1-g_0)} \\ \qquad \text{if } \{r,s\} \in \binom{U}{2} \\ \binom{n}{2}^{-1} \frac{c_1^2(1-g_1)}{c_1^2(1-g_1) + c_2^2(1-g_0)} \\ \qquad \text{if } \{r,s\} \in \binom{V}{2} \\ 0 \\ \text{if } (r,s) \in U \times V \cup V \times U \end{cases} \tag{13}$$

Proof: Let p be some distribution on $\binom{R}{2}$. By Lemma 4, we can assume that $p(t,u) = p(x,y)$, if $t,u,x,y \in U$. Similarly, $p(v,w) = p(x,y)$, if $v,w,x,y \in V$. Again using Lemma 4, $p(u,v) = p(u,y) = p(x,y)$, if $u,x \in U$ and $v,y \in V$. This shows that all pairs intersecting both U and V have equal probability.

If $k >= n+m$, the adversary can try to compromise all routers. Thus the best strategy is to only choose pairs from the trusted set U, as described in Eq. 12. From now on, assume that $k < n+m$.

Let $K_j \subseteq R$ be of size k and have an intersection with U of size j. The value of j alone determines the probability of compromise for K_j, because it determines the number of pairs in $\binom{U}{2}$, $U \times V$, and $\binom{V}{2}$. As we have just shown, the exact pairs included do not matter because their probability is determined by their class. Let $p_1 = \sum_{\{t,u\} \in \binom{U}{2}} p(t,u)$, $p_2 = \sum_{(u,v) \in U \times V} p(u,v)$, and $p_3 = \sum_{\{v,w\} \in \binom{V}{2}} p(v,w)$. Then we can say that

$$\begin{aligned} c(p, K_j) = &\binom{j}{2}(c_1)^2 \frac{p_1}{\binom{m}{2}} + j(k-j)c_1c_2\frac{p_2}{mn} + \\ &\binom{k-j}{2}(c_2)^2 \frac{p_3}{\binom{n}{2}} \end{aligned} \tag{14}$$

To narrow the set of possible optimal assignments of p_1, p_2, and p_3, we will first consider the effect of varying p_2. The quantity we want to minimize is the maximum value of Eq. 14. Equation 14 is a quadratic function of j. Assume that the second derivative is non-zero. If it is zero it is easy to show that the distribution p is the distribution described in Eq. 11. Otherwise, we will show that we can improve the maximum by changing p_2. We can find the local extremum by taking the derivative of Eq. 14 and setting it to zero. Solving for j gives

$$j^* = \frac{\begin{array}{c} n(n-1)p_1c_1^2 - k(m-1)(n-1)p_2c_1c_2 + \\ (2k-1)m(m-1)p_3c_2^2 \end{array}}{\begin{array}{c} 2(n(n-1)p_1c_1^2 - (m-1)(n-1)p_2c_1c_2 + \\ m(m-1)p_3c_2^2) \end{array}}. \tag{15}$$

Unfortunately, j^* must be integral to represent a worst-case subset, and therefore we cannot just substitute the expression in Eq. 15 into Eq. 14 and solve for the optimal value of p_2. There may in fact be two values of j that are maxima, and varying p_2 could possibly increase the value at one while decreasing the value at other. Therefore, while varying p_2, we simultaneously vary p_1 and p_3 to maintain the local extremum of Eq. 14 at j^*. Then both possible maxima are changed in the same way.

By observing that $p_3 = 1 - p_1 - p_2$ in Eq. 15 we can see that p_1 and p_2 are linearly related. Solve this for p_1 and call the expression p_1'. Now let $j' \in \mathbb{N}$, $0 \leq j' \leq k$, be any value that maximizes $c(p, K_{j'})$. j' is either an endpoint of $[0,k]$ or a closest integer to a local maximum. Substitute p_1' for p_1 in $c(p, K_{j'})$, and the result is a linear function of p_2. Therefore either increasing or decreasing p_2 does not increase $c(p, K_{j'})$. Suppose we move p_2 in the direction that decreases $c(p, K_{j'})$. Because we vary p_1' (and p_3) with p_2 in such a way as to maintain the extremum of the parabola at the same value j^*, j' is maintained as a maximum of $c(p, K_j)$ as long as the second derivative of $c(p, K_{j'})$ remains non-zero.

The process of changing p_2 stops when (*i*) the second derivative of $c(p, K_{j'})$ becomes zero, (*ii*) p_2 reaches zero, (*iii*) p_3 reaches zero, or (*iv*) p_1 reaches zero.

Case (i): In this case, all sets have the same value. This is only satisfied when the distribution is that of Eq. 11.

Case (ii): In this case, all probability is in pairs of two trusted or two untrusted nodes. Therefore the maximizing value of j must be when it is as small as possible or as large as possible, i.e., at $\max(0, k-n)$ or $\max(k, m)$. If the former case is strictly larger, we can reduce it by decreasing p_3 and increasing p_1. If the latter case is strictly larger, we can do the reverse. Therefore the value in these two cases must be equal. To find the probabilities p_1 and p_3 that satisfy this, let $p_3 = 1 - p_1$, $v_0 = \max(k-m, 0)$, and $v_1 = \max(k-n, 0)$. Then setting them equal and solving for p_1 yields the condition

$$p_1 = \frac{c_2^2\left(1 - \frac{v_0(v_0-1)}{n(n-1)}\right)}{c_1^2\left(1 - \frac{v_1(v_1-1)}{m(m-1)}\right) + c_2^2\left(1 - \frac{v_0(v_0-1)}{n(n-1)}\right)}. \tag{16}$$

Equation 16 then gives us the probability for each pair in $\binom{U}{2}$ and $\binom{V}{2}$, and this is the same as the distribution in Eq. 13.

Case (iii): In this case, $p_3 = 0$. Then if $p_2 = 0$ also, we put all probability in the trusted nodes, which is the distribution described in Eq. 12.

Now suppose that $p_2 > 0$. We will consider moving probability between p_1, p_2, and p_3 to show that this case isn't possible. Let $p_2 = 1 - p_1$ in Eq. 14 and call this $c_3(p, K_j)$. Then use this to consider trading off p_1 and p_2 to find the optimal assignment. As p_1 varies, the change in the value of the set K_j is

$$D_{p_1}c_3(p, K_j) = \frac{jc_1}{m}\left[\frac{(j-1)c_1}{m-1} - \frac{(k-j)c_2}{n}\right]. \tag{17}$$

Next, let $p_2 = 1 - p_1 - p_3$ in Eq. 14 and call this $c_4(p, K_j)$. Moving p_2 to p_3 results in a change of

$$D_{p_3}c_4(p, K_j) = \frac{(k-j)c_2}{n}\left[\frac{(k-j-1)c_2}{n-1} - \frac{jc_1}{m}\right]. \tag{18}$$

Let $j^* \in \text{argmax}_j c(p, K_j)$ be the largest integer that is a maximum of $c(p, K_j)$.

We observe that $D_j^2 c(p, K_j) \leq 0$. If not, we would have $j^* = k$. Then Eq. 17 shows that decreasing p_1 would decrease the value at j^*, and p_1 is non-zero so we could do this because, at $p_1 = 0$, $c(p, K_j)$ is largest at $j^* = \lceil k/2 \rceil \neq k$. Such a decrease would contradict the optimality of j^*.

Now, because $D_j^2 c(p, K_j) \leq 0$, there may be some $\hat{j} \in \text{argmax}_j c(p, K_j)$ such that $\hat{j} < j^*$. There are four cases to consider here: (1) $D_{p_1}c_3(p, K_{j^*})$, $D_{p_1}c_3(p, K_{\hat{j}}) \leq 0$, (2) $D_{p_1}c_3(p, K_{j^*})$, $D_{p_1}c_3(p, K_{\hat{j}}) \geq 0$, (3) $D_{p_1}c_3(p, K_{j^*}) \geq 0$ and $D_{p_1}c_3(p, K_{\hat{j}}) \leq 0$, and (4) $D_{p_1}c_3(p, K_{j^*}) \leq 0$ and $D_{p_1}c_3(p, K_{\hat{j}}) \geq 0$.

In case (1), we could decrease c at j^* and $\hat{j}$ by moving probability from p_2 to p_1. This would contradict the optimality of p.

For case (2), we use the fact that

$$0 \leq a < b \Rightarrow \frac{a-1}{b-1} < \frac{a}{b}. \tag{19}$$

Inequality 19 implies that if $D_{p_1}c_3(p, K_j) \geq 0$, then $D_{p_3}c_4(p, K_j) \leq 0$. Therefore we could decrease c at j^* and $\hat{j}$ by moving probability from p_2 to p_3, contradicting the optimality of p.

For case (3), we show that we can still decrease both j^* and $\hat{j}$ while maintaining their equality, and hence maximality, by moving some probability from p_2 to p_1 and p_3. Moving probability from p_2 to p_1 increases the value at j^* and decreases the value at $\hat{j}$. This implies, by Inequality 19, that moving probability from p_2 to p_3 decreases the value at j^*. Furthermore, can assume that it increases it at $\hat{j}$ because otherwise we could decrease both j^* and $\hat{j}$ by moving probability directly from p_2 to p_3.

For j^* and $\hat{j}$ to be integral maxima of Eq. 14, it must be that $j^* - 1 = \hat{j}$. Also, solving $D_{p_1}c_3 = D_{p_3}c_4$ for j, we find that at this point, $D_{p_1}c_3 \leq 0$ and $D_{p_3}c_4 \leq 0$. Therefore, j^* is at most one more than this point. We can observe by calculation that within this range the ratio $|D_{p_1}c_3/D_{p_3}c_4|$ is less than one. Similarly, $\hat{j}$ is at most one less than this point, and within this range the ratio $|D_{p_1}c_3/D_{p_3}c_4|$ is greater than one.

This shows that we can move probability from p_2 to p_1 and p_3 at rates that decrease the value at both j^* and $\hat{j}$. Because they were maximum, we have lowered the value of the worst-case subset K_{j^*}, contradicting the optimality of p.

Case (4) is not possible because $D_j^2[D_{p_1}c_2] \geq 0$ and $D_{p_1}c_3(p, K_0) = 0$.

Case (iv): In this case, if $p_2 > 0$, the case is symmetric to the case of $p_1, p_2 > 0$ and we can apply the same argument. Therefore assume that $p_2 = 0$, which implies that $p_3 = 1$. It must be that $m < n$ because otherwise we could set $p_1 = 1$ and $p_3 = 0$ and improve the worst case. But now consider moving some probability from p_3 to p_1. Let $p_1 = 1 - p_3$ in Eq. 14 and call this c_3. The change in the worst-case case subset, K_n, is

$$D_{p_3}c_3(p, K_n) = c_2^2 - \frac{(k-n)(k-n-1)c_1^2}{m(m-1)}.$$

This must be greater than zero because $c_2 \geq c_1$ and $k - n < m$. Therefore decreasing p_3 decreases $c(p, K_n)$, contradicting the optimality of p. □

6. Choosing a distribution

We have shown that there are three possibilities for an optimal strategy in choosing nodes that will minimize the best chances a fixed size adversary has to compromise both endpoints of an onion-routing circuit when a trusted set is available. To choose a distribution, a user can simply calculate the probability of compromise in each case and use the distribution with the smallest result. The optimal distribution depends on all the variables in the system: the trust values,

the size of the trusted set, the size of the untrusted set, and the size of the adversary.

In the first distribution, described in Eq. 11, the user chooses pairs $\{i, j\}$ to make $p(i, j)c_i c_j$ equal for all i, j. This is a random choice of pairs weighted by the trust in the pair. The probability of compromise under this strategy is

$$C_1 = \frac{k(k-1)c_1^2 c_2^2}{m(m-1)c_2^2 + 2mnc_1c_2 + n(n-1)c_1^2}. \quad (20)$$

This strategy is optimal when the network is large compared to the adversary, and so it benefits the user to spread out his distribution, even to less-trusted routers. It is also optimal when the trust values are close.

In the second distribution, described in Eq 12, the user randomly selects pairs from within the trusted set. This can only be optimal if the size k of the adversary is larger than the size m of the trusted set. Otherwise, the user could decrease the probability of compromise by putting some of the pair-selection distribution on pairs outside the trusted set. Doing so would not change the adversary's worst-case subset, which is entirely in the trusted set, but it would decrease the probability that those nodes are chose by the user. The probability of compromise, assuming $k > m$, is simply

$$C_2 = c_1^2. \quad (21)$$

We can compare this to Eq. 20 and observe that c_1 can always be made small enough to make this value less than the value of the first strategy. These equations also show that choosing only trusted nodes will be optimal when k is large relative to the network. When $k = m + n$, this case is always optimal.

The third distribution, given in Eq. 13, is perhaps the least obvious one, and arises as a result of the fact that users choose their distribution over pairs, while the adversary attacks individual routers. Let $v_0 = \max(k - m, 0)$ and $v_1 = \max(k - n, 0)$. Then let $g_0 = v_0(v_0 - 1)/(n(n - 1))$ and $g_1 = v_1(v_1 - 1)/(m(m - 1))$. In general, the probability of compromise under this distribution is

$$C_3 = \begin{cases} \frac{c_1^2 c_2^2 (1-g_0)}{c_1^2(1-g_1)+c_2^2(1-g_0)} + \\ \frac{v_0(v_0-1)c_1^2c_2^2(1-g_1)}{n(n-1)(c_1^2(1-g_1)+c_2^2(1-g_0))} \end{cases} \quad (22)$$

$$= \begin{cases} \frac{v_1(v_1-1)c_1^2c_2^2(1-g_0)}{m(m-1)(c_1^2(1-g_1)+c_2^2(1-g_0))} + \\ \frac{c_1^2c_2^2(1-g_1)}{c_1^2(1-g_1)+c_2^2(1-g_0)} \end{cases} \quad (23)$$

To make some sense of this, it is helpful to consider some special cases. When $n > k, m < k$, the probability of compromise is

$$C_3 = \frac{k(k-1)c_1^2c_2^2}{n(n-1)(c_1^2 + c_2^2(1-g_0))}$$

We can see that there is some large m such that C_3 is less than C_2 and C_1. What happens in this case is that there are large number of routers, and the user wants to spread his probability among them. However, because $k > n$, spreading the probability to all cross-pairs (one trusted and one untrusted router) means that an adversary selecting as many untrusted routers as possible gains $(k-n)n/(mn) = (k-n)/m$ of the probability on such pairs. On the other hand, when spreading to trusted pairs $(k-n)(k-n-1)/(m(m-1))$ of the shifted probability is captured by the adversary. The latter shrinks quadratically with m while the former shrinks only linearly. At some point it will be beneficial to spread probability to trusted pairs but not to cross-pairs. The case when $m > k, n < k$ is similar. This distribution is never optimal when $m > k$ and $n > k$, because the worst-case sets are contained within U and V, and so spreading probability to the cross-pairs some small amount will always decrease the probability of compromise.

7. Conclusion and future work

We have set out a simple model for reasoning about using trust for routing in onion-routing anonymity networks. This model modifies the traditional roving adversary by adding trust; so the success of the adversary in attacking nodes he chooses becomes probabilistic rather than certain. Trust is thus defined as the probability that the adversary fails in attempting to compromise a node. We used this model to look at end-to-end correlation attacks by nodes in onion-routing networks. We expect this model to be useful for future research by ourselves and others.

We used our model to show optimal strategies for choosing routes when trust information is available. The strategies are optimal in that they minimize the maximum probability a correlating adversary has for linking source to destination.

In the general case, where there is an arbitrary number of trust levels, we presented an algorithm to calculate an optimal distribution, an algorithm which runs in time exponential in the size of the adversary. We described a natural simplification and approximation of this, which permitted the calculation of optimal strategies on selection of individual nodes, but we also showed that the approximation based on this is arbitrarily worse than optimal distributions on pairs of nodes.

We then turned to consider a practical approach by limiting ourselves to two trust levels. In addition to being computationally tractable, users of deployed networks are more likely to be capable in practice of dividing routers into these levels. We described three distributions for this case and proved that one of them must be optimal. Lastly, we discussed determining in practice when one of the three distributions is optimal based on the values of the system variables: trust values, size of the trusted and untrusted sets, and the size of the adversary.

The results we have produced are more complicated than we expected, both to describe and to prove. It will be interesting to examine larger questions of trust in future work: What happens when a network is shared between entities that do not share trust levels placed on the nodes? What is the impact of trust on profiling in this case? What is the effect of learning if we add time to the model and allow the adversary to rove rather than conducting a one-off attack?

Though our motivation is onion routing, our analysis applies to any network where it would be beneficial to reduce the

chance of circuit-endpoint threats by choosing circuits with less vulnerable endpoints. It clearly generalizes to other low-latency anonymity designs, such as Crowds [27]. It also applies beyond networks for anonymity to other concerns. For example, network endpoints may be able to collaborate to cover up checksum or other errors that might flag data-integrity attacks. And, capturing internet traffic for any kind of analysis (cryptanalysis, textual analysis, traffic analysis, etc.) may be easier to do or harder to detect or both if pairs of nodes are collaborating for route capture. Alternatively they might collaborate for unfair resource sharing. Similar observations apply to ad-hoc and peer-to-peer networks and to sensor networks, for which vulnerability of cheap, low-power, and physically accessible nodes is a known concern. Going further, our results are not restricted in applicability to path endpoints. In any setting in which sets of principals can collaborate so that a successfully compromised pair can conduct an attack our results are potentially applicable. Examining larger numbers of nodes being attacked than just pairs is one possible generalization of this work that should apply in many settings.

References

[1] D. M. Goldschlag, M. G. Reed, and P. F. Syverson, "Hiding routing information," in *Information Hiding: First International Workshop, Proceedings*, 1996, pp. 137–150.

[2] M. G. Reed, P. F. Syverson, and D. M. Goldschlag, "Anonymous connections and onion routing," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 482–494, May 1998.

[3] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proceedings of the 13th USENIX Security Symposium*. USENIX Association, August 2004, pp. 303–319.

[4] P. Syverson, G. Tsudik, M. Reed, and C. Landwehr, "Towards an analysis of onion routing security," in *Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability*, H. Federrath, Ed. Springer-Verlag, LNCS 2009, July 2000, pp. 96–114.

[5] R. Fagin and J. Y. Halpern, "I'm OK if you're OK: On the notion of trusting communication," *Journal of Philosophical Logic*, vol. 17, no. 4, pp. 329–354, November 1988.

[6] G. J. Simmons and C. Meadows, "The role of trust in information integrity protocols," *Journal of Computer Security*, vol. 3, no. 1, pp. 71–84, 1994/1995.

[7] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized trust management," in *In Proceedings of the 1996 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, 1996, pp. 164–173.

[8] M. Abadi, "On SDSIs linked local name spaces," *Journal of Computer Security*, vol. 6, no. 1/2, pp. 2–22, 1998.

[9] A. Chander, J. C. Mitchell, and D. Dean, "A state-transition model of trust management and access control," in *Proceedings of the 14th IEEE Computer Security Foundations Workshop, CSFW '01*. IEEE Computer Society, 2001, pp. 27–43.

[10] V. Shmatikov and C. Talcott, "Reputation-based trust management," *Journal of Computer Security*, vol. 13, no. 1, pp. 167–190, 2005.

[11] A. J. sang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618 – 644, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/B6V8S-4GJK82P-1/2/a9a6e96414fa04641c1d31a57989618d

[12] S. De Capitani di Vimercati, S. Jajodia, S. Paraboschi, and P. Samarati, "Trust management services in relational databases," in *ASIACCS '07: Proceedings of the 2nd ACM symposium on Information, computer and communications security*. New York, NY, USA: ACM, 2007, pp. 149–160.

[13] "The Tor project home page," `https://www.torproject.org/`.

[14] L. Øverlier and P. Syverson, "Locating hidden servers," in *Proceedings of the 2006 IEEE Symposium on Security and Privacy*. IEEE CS, May 2006, pp. 100–114.

[15] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. Sicker, "Low-resource routing attacks against tor," in *WPES'07: Proceedings of the Workshop on Privacy in the Electronic Society*, T. Yu, Ed. ACM Press, October 2007, pp. 11–20.

[16] R. Ostrovsky and M. Yung, "How to withstand mobile virus attacks," in *Proceedings of the Tenth ACM Symposium on Principles of Distributed Computing (PODC '91)*. ACM Press, 1991, pp. 51–59.

[17] A. Acquisti, R. Dingledine, and P. Syverson, "On the economics of anonymity," in *Financial Cryptography, 7th International Conference, FC 2003*, R. N. Wright, Ed. Springer-Verlag, LNCS 2742, 2003, pp. 84–102.

[18] S. J. Murdoch and G. Danezis, "Low-cost traffic analysis of Tor," in *Proceedings of the 2005 IEEE Symposium on Security and Privacy*. IEEE CS, May 2005, pp. 183–195.

[19] N. S. Evans, C. Grothoff, and R. Dingledine, "A practical congestion attack on Tor using long paths," 2009, manuscript.

[20] N. Borisov, G. Danezis, P. Mittal, and P. Tabriz, "Denial of service or denial of security? How attacks on reliability can compromise anonymity," in *CCS'07: Proceedings of the 14th ACM Conference on Computer and Communications Security*, S. D. C. di Vimercati, P. Syverson, and D. Evans, Eds., October 2007, pp. 92–102.

[21] N. Hopper, E. Y. Vasserman, and E. Chan-Tin, "How much anonymity does network latency leak?" in *CCS'07: Proceedings of the 14th ACM Conference on Computer and Communications Security*, S. De Capitani di Vimercati, P. Syverson, and D. Evans, Eds. ACM Press, 2007, pp. 82–91.

[22] A. Hintz, "Fingerprinting websites using traffic analysis," in *Privacy Enhancing Technologies: Second International Workshop, PET 2002*, R. Dingledine and P. Syverson, Eds. San Francisco, CA, USA: Springer-Verlag, LNCS 2482, April 2002, pp. 171–178.

[23] M. Liberatore and B. N. Levine, "Inferring the source of encrypted HTTP connections," in *CCS'06: Proceedings of the 13th ACM Conference on Computer and Communications Security*, R. N. Wright, S. De Capitani di Vimercati, and V. Shmatikov, Eds. ACM Press, 2006, pp. 255–263.

[24] J. Feigenbaum, A. Johnson, and P. Syverson, "Probabilistic analysis of onion routing in a black-box model [extended abstract]," in *WPES'07: Proceedings of the Workshop on Privacy in the Electronic Society*, T. Yu, Ed. ACM Press, October 2007, pp. 1–10.

[25] N. Feamster and R. Dingledine, "Location diversity in anonymity networks," in *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2004)*, S. De Capitani di Vimercati and P. Syverson, Eds., 2004, pp. 66–76.

[26] S. J. Murdoch and P. Zieliński, "Sampled traffic analysis by internet-exchange-level adversaries," in *Proceedings of the Seventh Workshop on Privacy Enhancing Technologies (PET 2007)*, N. Borisov and P. Golle, Eds. Ottawa, Canada: Springer-Verlag, LNCS 4776, June 2007.

[27] M. Reiter and A. Rubin, "Crowds: Anonymity for web transactions," *ACM Transactions on Information and System Security*, vol. 1, no. 1, pp. 66–92, 1998.