Relay Recommendation System (RRS) and Selective Anonymity for Tor

Chenglong Li\textsuperscript{1,3}, Yibo Xue\textsuperscript{1,2}, Yingfei Dong\textsuperscript{4} and Dongsheng Wang\textsuperscript{1,2}
\textsuperscript{1}Tsinghua National Laboratory for Information Science and Technology (TNList), Beijing, 100084, China.
\textsuperscript{2}Research Institute of Information Technology (RIIT), Tsinghua University, Beijing, 100084, China.
\textsuperscript{3}Department of Computer Science & Technology, Tsinghua University, Beijing, 100084, China.
\textsuperscript{4}Department of Electrical Engineering, University of Hawaii, Honolulu, HI 96822, USA.

Abstract—As one of the most popular low-latency anonymous communication systems, Tor has been a great success but still faces some challenges, e.g., subject to low-resource attacks and no explicit mechanisms to explore tradeoffs between anonymity and performance. In this paper, we propose a Relay Recommendation System (RRS) for Tor: to provide reliable relay information for building paths with better performance; to mitigate low-resource attacks, to enable users to explore the tradeoffs between performance and anonymity based on their needs. We first present the design of RRS and corresponding performance improvement. RRS helps us defeat low-resource attacks. We further analyze the potential anonymity decrease, and propose new path selection schemes to enable selective anonymity based on user needs. We have evaluated RRS via both analysis and experiments on a Tor simulation platform. Our results show that RRS achieves a performance improvement of 25% with a small anonymity decrease.

Keywords- anonymous communication; reputation; anonymity; performance; attack

Submission paper category: regular paper

Word Count: about 8,000 words.

The material has been cleared through authors’ affiliations.

I. INTRODUCTION

Early anonymous communication systems are mainly designed for high-latency communications such as E-mail. Recent anonymous communication systems take advantage of the Internet and support low-latency transactions, such as web browsing or instant messaging. However, these systems (such as Tor) face several serious issues:

1. They usually achieve low latency by sacrificing anonymity. For example, the default path length of Tor is only 3 hops. Such an anonymity setting is fixed for the entire system and users cannot adjust it for their own anonymity needs.

2. They do not provide mechanisms to allow users to explore the tradeoffs between performance and anonymity based on their individual requirements. For example, a user may only need to circumvent network censorship for web surfing. It prefers fast responses over extreme anonymity. However, a government agent may prefer high anonymity over fast response.

3. Low-resource attacks (e.g., [22]) are often possible on current anonymous systems due to their P2P nature. Current large-scale anonymous systems are mostly built with third-party resources, without strict control. Tor is somewhat a hybrid P2P structure of volunteer routers. A peer can advertise false high bandwidth, which damages other users’ performance and allows attackers to perform low-resource attacks. Currently, Tor does not explicitly deal with these issues. Flags and bandwidth used in Tor path selection can be easily forged without supervision.

To address these challenges, we propose Relay Recommendation System (RRS) and corresponding new path selection schemes. RRS is embedded into the three-level Tor hierarchy (directory servers, relays, and clients, as shown in Fig.1) to evaluate relays based on client experiences. Furthermore, we propose to enhance path selection with multiple attributes instead of single bandwidth estimate [25] to improve user experienced performance. As RRS does not expose any historical communication relationships, no communication patterns are leaked. However, a naïve RRS may decrease anonymity while improving performance. By careful design choices, we can limit such loss to a small range while greatly improving performance, as shown by our simulation results. We further allow users to select anonymity and performance levels based on their needs. Slow responses have been one of main complains about anonymous networks. Hopefully, with performance improvement and flexible user choices, the proposed new system will attract more users and relays, and thus increases the size of anonymity set and anonymity. In the meantime, the proposed scheme also provides a natural solution to mitigate low-resource attacks. We also discuss how to use relay reputation provided by RRS to deal with other attacks.

The main contributions of the paper include as follows:

- Design and implement RRS, which provides reliable relay information for better path selection.
• Develop new path selection schemes based on relay reputation to improve path performance.
• Demonstrate the effectiveness of the proposed solution to deal with low-resource attacks.
• Provide selective mechanisms for users to tradeoff their performance and anonymity needs.
• Analyze the performance and security influences of RRS.

The remainder of the paper is organized as follows. We discuss the related work and introduce the proposed idea in Sec.2. We present our RRS implementation in Sec.3, demonstrate the performance improvement in Sec.4, and show the effect of mitigating low-resource attacks in Sec.5. We present the selective anonymity mechanism in Sec.6, and evaluate RRS to deal with other attacks in Sec.7. We conclude this paper and discuss future work in Sec.8.

II. RELATED WORK AND PROPOSED IDEA

A. Tor

Tor is a popular low-latency anonymous communication system using a volunteer relay network to provide anonymous service. It has a hierarchical structure, as shown in Fig.1. Directory servers at the top level collect information of all relays in the network, and publish a consensus file of relays. They are administrated by trustworthy third parties. Relays and bridges [26] at the middle level are run by volunteers who are willing to provide resources for forwarding anonymous traffic for clients at the bottom level. The functions of relays and bridges are the same. The only difference between them is the release policy. All relays are published to all clients. However, a client can only obtain a few bridges at a time via email or web browsing. Bridges are used as the first hops to evade censorship. Because there are no need to differentiate relays and bridges in this paper, we use relays or nodes in the following to represent both relays and bridges. A client normally first contacts with directory servers to obtain relay information, and then choose three relays with a path selection algorithm to build an anonymous circuit. We consider Tor as a hybrid P2P network.

Tor extends the basic idea of Chaum’s MIXnet [1]. The first implementation is Onion Routing [2]. The latest implementation Tor [3] builds an anonymous path with multi-layer encryption to conceal communication relationships as shown in Fig.1. A client uses a Tor proxy to construct an anonymous path of three nodes to communicate with a target resource. Only the client knows all three nodes on the path. A relay on the path only knows about its predecessor and successor. The client encrypts a packet with relays’ public keys into three layers. A relay decrypts a packet with its private key and knows how to forward it to the next hop.

In addition to its great success, Tor still faces several issues. First, it does not have a mechanism to evaluate a relay. Directory servers gather relay information solely based on self-generated reports from relays. Therefore, a malicious relay can lie about its resources and enable low-resource attacks. This also damages user perceived performance because it may select a poor path based on the false claims. Second, it does not provide a mechanism for users to explore tradeoffs between anonymity and performance. All users and transactions are treated as the same in the entire system. To address these issues, we propose RRS and corresponding path selection schemes for providing more reliable relay information and exploring tradeoffs between performance and anonymity, as presented in the following.

B. Reputation Systems and Proposed RRS

We propose RRS to help us improve performance, explore tradeoffs between anonymity and performance, and mitigate attacks. RRS is based on a reputation system commonly used in P2P networks to encourage peers with high contributions, and then improve system performance and deterring liars [9-14, 27]. PeerTrust [11] separates feedback trust from service trust by comparing evaluation similarities of previous partners. Follow-up schemes [12-14] personalize reputation values from recommendations based on PeerTrust. Although these approaches are fairly effective on their own terms by considering peer feedbacks, they are not suitable to anonymous systems because they need track transaction histories, which expose past communication patterns.

For this reason, reputation systems are rarely considered in anonymous systems except [15, 16]. In [15], the authors developed on a low-resource attack via false relay reports and briefly mentioned that a reputation system may be a defense solution to the attack without further details. In [16], the authors presented a trust model with multiple trust levels to improve the path-selection strategies of Tor and minimize the probability of end-to-end correlation attacks. However, both papers primarily focused on attacks and did not provide detailed mechanisms to explore the tradeoffs between anonymity and performance.
To avoid saving transaction histories and leaking communication patterns, we use incremental reputation calculation and only keep aggregated information in a short sliding interval. In current Tor, relay information is collected and maintained by a group of directory servers. We comply with the same centralized structure of Tor, and assume that relay information is still administrated by directory servers. If attackers are able to compromise the directory servers, they have many straightforward methods to attack the system, e.g., only telling a client about malicious nodes, or making normal nodes less desirable. So, RRS’s dependence of the directory servers is not likely to increase attacker chances. Although extra loads are put on the directory servers, we can limit them to an acceptable range as demonstrate in the following.

With RRS, a straightforward way to select nodes for a path is solely based on node reputation. It helps improve user experienced performance at the expense of anonymity. However, such loss of anonymity can be limited to negligible in practice. This is one key argument for our approach. Users can obtain better performance without significant anonymity losses. RRS also helps us defeat other attacks on Tor.

Furthermore, we propose a selective mechanism which allows users to choose different performance and anonymity levels based on their needs. As such flexibility may attract more users and relays, we will have a larger anonymity set and thus achieve better anonymity. A similar idea was proposed in [25], which allows users to select different anonymity levels. However, their focus is on the Tor load-balancing algorithm, not on the influence of the whole system. We extend the tunable idea and give concrete levels based on RRS.

As RRS itself may be attacked, we have also considered potential issues and addressed them through various defense methods to minimize potential damage.

C. Definitions

For ease of discussion, we list main notions used in this paper in Table I. Other notations will be introduced when needed.

<table>
<thead>
<tr>
<th>TABLE I. NOTIONS USED IN THIS PAPER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>N</td>
</tr>
<tr>
<td>M</td>
</tr>
<tr>
<td>R_l, ..., R_N</td>
</tr>
<tr>
<td>B_r</td>
</tr>
<tr>
<td>T_u</td>
</tr>
<tr>
<td>B_e</td>
</tr>
<tr>
<td>V &lt;&lt;= B, U&gt;</td>
</tr>
<tr>
<td>I</td>
</tr>
<tr>
<td>C</td>
</tr>
<tr>
<td>C_{R_i}</td>
</tr>
<tr>
<td>T_j(R_j)</td>
</tr>
<tr>
<td>α</td>
</tr>
<tr>
<td>K</td>
</tr>
<tr>
<td>Δt</td>
</tr>
<tr>
<td>N_{HR}</td>
</tr>
<tr>
<td>N_{LR}</td>
</tr>
</tbody>
</table>

D. Attack Model

In this paper, we mainly consider malicious node attack, in which an attacker tries to control both the first node and the third node of a Tor circuit for correlation attacks, e.g., based on timing. To increase its chances, an attacker will first try to control or insert as many as malicious relays as possible. Under the current Tor path selection, attackers are in favor of controlling high bandwidth nodes, or forging nodes with high bandwidth, in order to be selected in more paths.

III. PROPOSED RRS FOR TOR

A. Employing Multiple Attributes for Better Evaluation

Tor currently uses an estimate bandwidth as only factor in routing decision which is the lesser of the observed bandwidth and the bandwidth rate limit from the relay (router) descriptor [8]. However, this value is reported by a relay itself and may result in many problems such as low resource attacks. To deal with this issue, the system needs a better method to evaluate relays, in order to make the path selection more reliable and effective. We consider two classes of multi-attributes in RRS:

- Self-reported: In this class, a relay reports its attributes to directory servers by itself, including; (1) B_r, Reported Bandwidth (KBps), which is a bandwidth estimate of a relay. It is used to weight router/relay selection in the current Tor. We keep this attribute to compare with the effective bandwidth of a relay; (2) T_u, Current Uptime (second), which is obtained in server descriptors. The reason to choose this parameter is that relays with long uptime tend to be “super nodes” and generally provide more bandwidth as reported in [17].

- Client-graded: A client measures the effective bandwidth (B_e) of a relay for its transactions and the durations of transactions. A relay j may serve a total of C_j connections in the current measurement interval in the entire system. All clients report their measurements to directory servers and collect client measured effective bandwidth B_e(j) and its duration t(j) from all clients in a measurement interval. Then, we can determine the effective bandwidth B_e(j) of relay j in a measurement interval i as

\[ B_e(i,j) = \sum_{j=1}^{C_i} t_j \times B_e(j) / \sum_{j=1}^{C_i} t_j. \]

In the client grading process, to avoid clients submitting forged measurements of fake connections, directory servers need to check with relays if they have served the client at that time. When majority relays confirm, the grades of relays are accepted. Otherwise, discarded.

We limit the number of grades by one client in a time period. In Δt time, the maximum number of grades for a client IP address is set to a threshold K. Notice that Δt is shorter than the measurement interval length I of RRS. More
accurately, because Tor clients tear down and change circuit in ten minutes, we set $\Delta t$ equating ten minutes. This limit does not affect honesty relays and clients, but makes the cheating more difficult.

We combine these three attributes into a vector $V= <B, U>$, where $B$ is a bandwidth estimate and $U$ is uptime.

$$
B = \begin{cases} 
B_r, & \text{if } B_r \geq 1 \text{ and } B_e \geq B_r \\
B_r \frac{B_e}{B_r}, & \text{otherwise}
\end{cases}
$$

We can see $B$ is more close to the real bandwidth of relays. When $B_r < B_e$, there may be false reports and we set up a punishing mechanism as presented in the following.

Furthermore, we use linear normalization method to normalize attributes to $[0, 1]$. The normalization function is defined as $n(x)=(x-MinValue)/(MaxValue-MinValue)$. The version of vector $\mathbf{V}(\mathbf{j})$ of relay $j$ includes the normalized $B$ and $U$.

Although malicious users may lie, they need to commit more resources to achieve the effect as in the current Tor. We will discuss this in the following.

B. Evaluating Relay Reputation

After collecting client grades in a measurement interval, directory servers calculate the reputation values of all relays based on multi-attribute vectors and publish a reputation list. We will introduce the evaluation process here.

For a relay $R_i$ in the system, its reputation value in the first measurement interval is defined as:

$$
T_0(R_i) = w_1 * B_0(R_i) + w_2 * U_0(R_i)
$$

where $w_1$, $w_2$ are weights for $B$ and $U$. We have $w_1 + w_2 = 1$ and $w_1 > w_2$ in order to emphasize the effect of bandwidth. Based on our simulation experience, we set $w_1 = 0.75$ and $w_2 = 0.25$ for testing.

We then use an incremental method to evaluate relay reputations in the following intervals. In the $(k+1)^{th}$ interval, For relay $R_i$, assume $T_k = w_1 * B_k + w_2 * U_k(R_i)$.

So, the reputation value in the $(k+1)$ interval is defined as:

$$
T_{k+1}(R_i) = \begin{cases} 
T_k(R_i) + \alpha_1 * (T_k(R_i) - T_k(R_i)), & \text{if } T_k(R_i) > T_k(R_i) \\
T_k(R_i) + \alpha_2 * (T_k(R_i) - T_k(R_i)), & \text{if } T_k(R_i) < T_k(R_i)
\end{cases}
$$

where $0<\alpha_1 \leq \alpha_2<1$. We define $\alpha = \alpha_1/\alpha_2$, which is used as a punishing factor. When $T_k(R_i) > 1$ or $T_k(R_i) < 0$, it is set to 1 or 0 in order to make $0 \leq T_k(R_i) \leq 1$. The reason that $\alpha_1 \leq \alpha_2$ is to punish a reputation decrease more than a reputation increase. Depending on our simulation experience, the values of $\alpha_1$ and $\alpha_2$ are set as $\alpha_1 = 0.4$ and $\alpha_2 = 0.6$.

We use the above Formula (1) and (2) to calculate relay reputation values. These reputation values help us improve path selection effectiveness by combining effective bandwidth and relay up times. Therefore, we can make better decision in path selection by considering both relay performance (bandwidth) and reliability (up time).

We also propose three policies to protect RRS from abnormal behaviors as follows:

- Punishing Policy – In the reputation formula (2), we use punishing factor $\alpha = \alpha_1/\alpha_2$ to punish relays whose reputation drops in the current interval. With the punishing policy, assessments from honest clients become more significant.

- Relay Confirmation – Confirming the authenticity of connections prevents liars from submitting forged grades of fake connections. Although attackers can defeat this confirmation by controlling both clients and relays, they only collude with themselves. They cannot simply damage the reputation of randomly-selected honest relays.

- Grading Limitation – In the calculation of reputation, not all grades should be counted. When adversaries control both malicious clients and relays, they can affect the system by submitting forged grades numerous times in a short period. This policy will limit their impacts. It does not deter the attack, but it increases the resource requirement on the attacker side.

C. Recommendation Bandwidth Selection

We revise the current Tor path selection algorithm to take advantage of relay reputations provided by RRS. The current Tor path selection combines several flags, relay estimate bandwidth, and random selection together to figure out safe and fast service. It works fine to some degree on the current Tor. However, it only considers one piece of information about relay—reported bandwidth. We now have more information of relays to achieve better path selection.

We replace the current Tor path selection function $\text{Weighted_Bandwidth_Selection}(\text{node})$ with our proposed $\text{Recommendation_Bandwidth_Selection}(\text{node})$ function. In the original function, it selects a node from the list with the probability based on weighted bandwidth, i.e., $P(\text{node } j) = \frac{B_j}{\sum B_j}$. However, in our function, it selects a node from the list with the probability based on recommendations, i.e., $P(\text{node } j) = \frac{T(\text{node } j)}{\sum T}$. The new algorithm improves the original bandwidth selection with multi-attribute recommendation.

D. Cost of RRS

Similar to other reputation systems, RRS brings extra costs. We analyze the time and space complexity of RRS and then discuss its influence to the whole system.

We first discuss the computational complexity of. In the implementation of RRS, we have three groups of entities: clients, relays, and directory servers.

- Clients – There are two steps for a client in RRS. At the beginning of a communication, a client needs to choose nodes for a path based on the path selection algorithm. The new path selection algorithm is a little more complicated than the original one. After the communication, a client needs to grade all relays on the path. A client only needs to report $B_e$. Assume there are totally $C$ connections in the current interval, the total computational complexity of all clients should be $O(C)$.

- Relays – Relays do not participate the grading and calculation processes. They are only involved in confirmation process. So, the total cost is
proportional to the number of connections, i.e., $O(C)$, because each connection involves three relays.

- **Directory Servers** – Directory servers collect all grades and compute relay reputations. Assume there are total $N$ nodes in the system. The computational complexity is $O(N)$. Based on formula (1) and (2), the total computation cost is $O(N) + O(C)$. Therefore the total cost at directory servers is $O(N) + O(C)$.

Therefore, the extra computational cost of RRS in a measurement interval is $O(N) + O(C)$, which is related to the number of relays and connections. We test this extra load on a common PC with known $N$ and $C$. From the official Tor site, the number of users per day is about 400,000, i.e., about 20,000 users per hour. When we set $C = 1,000,000$ connections per measurement interval (one hour), the total calculation time is less than 30 seconds. So, clearly for the current system size, the cost of the proposed reputation calculation is acceptable.

We then consider the space and bandwidth usage of RRS. At clients, they need to record effective bandwidth and report to directory servers, which do not cost much. Directory servers have to store a vector list and a reputation list for all relays. Its space cost is linearly to the size of Tor. Currently, Tor has about 3,000 to 5,000 relays active per day, and is not a problem.

We also consider bandwidth costs for transferring reputation information. For each connection, the client needs to report the effective bandwidth of relays. For every measurement interval, the reputation list is published within the Tor consensus file. It means that RRS adds one more item to the current 25 items in the consensus file [8]. Remember that Tor has multiple directory servers and mirrors. So, a client can report to one or more randomly selected servers to mitigate its bandwidth burden. From the Tor official metrics site, the average number of bytes spent on answering directory requests is 15MBps (both written and read). By our simulation and calculation, the extra bandwidth used for RSS is less than 1% to the existed cost.

In general, the costs of computation, space and bandwidth of RRS are acceptable.

### IV. PERFORMANCE IMPROVEMENT

We will first show the system performance improvement due to RRS. We use a Tor simulation platform to evaluate the improvement as introduced in the following. It is also used as the platform for other evaluations in this paper.

#### A. Tor Simulation Platform

Because it is impossible to test our methods on the real Tor, we have built a Tor simulation platform to evaluate the practical effect of RRS on both performance and anonymity. We extracted the key data structures and algorithms from Tor source code and built a simulator with several thousand nodes. It is implemented as a discrete-event simulator, following the exact Tor algorithms and protocols. It can simulate node joins, circuit lifetime, path selection, multiplexing of streams, various attacks, performance analysis, etc. A node in the platform behaves the same as a node in the real Tor. The only difference is that we have the complete control of each node to set up the testing environment and collect data. We simulate clients, relays and directory servers to perform real operations as Tor, initialize and observe attacks, and observe performance changes. This simulation platform gives us more confidence in investigating Tor related issues.

Because the distribution of relays is varied and unpredictable, we choose the real consensus data from Tor official site. In the research of [7], the authors found that 92% of exit TCP connections belong to HTTP making up 57% of exit traffic volume and 3% of exit connections belong to file sharing of BitTorrent making up 40% of Tor’s exit traffic volume. So for clients, we set characteristics of connections, such as transaction durations and bandwidth usage following the above data.

There are a few limitations for our simulation platform. First, we do not perform the actually data encryption or decryption or transmissions over the network. The nodes only perform key features such as join, leave, path selection, etc. This simplifies the computation load, while not reflecting the real network complexity. This is the common limitation in such a simulation. Second, the simulation system is driven by events such as the initializations or terminations of connections. The simulated results reflect the same trend to the real Tor algorithms and setting.
B. Default Experimental Settings

In our experiments, we use a set of default configuration to evaluate RRS based on the real dataset and simulation experience. All of the settings follow the design conception of RRS.

To obtain result closer to the real condition, we assign relays with realistic values of the estimate bandwidth and uptime obtained from Tor metrics site [6] (Oct, 2011). While the relay effective bandwidth is set to follow a normal distribution with a mean of 1500 Kbps and a standard deviation is 750. We have 3,000 relays and an arbitrary number of clients. The measurement interval \( I \) is set to one hour, and \( \Delta t \) is set to ten minutes.

C. Performance Improvement

Besides negligible overhead, RRS helps us improve the overall system performance. With the default setting, our experiment results show the improvement, as shown in Fig.2. With RRS, the average performance is improved about 25%. The experiments have two phases:

- In the first 10 intervals, all relays are honest and normal.
- In the next 20 intervals, we randomly select 5% of the total relays (60 out of 3000) and make them become abnormal, i.e., their reported bandwidth is set to a maximum value and their effective bandwidth drops to a minimal value.

From the results, we can see:

- In both phases, the proposed solution outperforms the original Tor.
- In the second phase, the original Tor system is easily fooled, resulting in a higher chance to select “liars” and a large decrease in performance. However, under the proposed scheme, the chance that a liar to be selected drops to near zero in about five intervals. The overall system performance quickly recovers after the occurrence of abnormal incidents.

In general, the average performance of the proposed scheme is 25% higher than that of original Tor in a stable state. “Liars” may due to many reasons, e.g., relays under DoS attacks or controlled by attackers. Tor has a hard time to deal with them, while the proposed scheme can adapt to the “attacks” caused by various reasons.

V. MITIGATING LOW-RESOURCE ATTACK

Tor uses a single bandwidth attribute to select nodes to build a path. Because the reported bandwidth is reported by a relay itself and can be easily tampered, low-resource attacks [22] can be initialized. Our proposed scheme can deal with this issue and mitigate low-resource attacks. We use a reputation preferential path selection algorithm, which uses multiple attributes including relay effective bandwidth reported by clients as well as the reported bandwidth and relay uptimes. It forces attackers to use more resources.

A. Resistance to Low-Resource Attack

In the original Tor, attackers can cause severe damages to anonymity by controlling a number of low-resource malicious relays and conducting correlation attacks at the both end of a path [20, 22]. Such attacks exploit the current Tor path selection algorithm that uses the relay self-reported bandwidth. Low-resource malicious relays are able to forge arbitrary high attribute values (smaller than the threshold of 10MBps) without supervision. The proposed scheme can stop this quickly. As shown in Fig.2, when low-resource liars appear in the second phase, the chance that liars are selected is very low under our scheme while the original Tor has no effective way to deal with them.

B. Resistance to High-Resource Attack

Adversaries may also use relays with high resources attract more connections for launching correlation attacks. We conducted the following experiments with the assumption that the top 10% highest resource relays are malicious. The following results show that the proposed scheme provides better resistance to such a high-resource attack than the original Tor.

First, the proposed scheme reduces the effect of high-resource attacks to a lower level. As shown in Fig.3 (a), the
average reputation of malicious relays is higher than that of the system, but the gap between two curves shrinks as time goes. Fig.3 (b) shows the result with the default settings. With the proposed scheme, less than 2% of connections can be used by the attacker to perform end-to-end correlation. However, under the original Tor, close to 10% of connections can be used by the attacker for end-to-end correlations. Clearly, the proposed scheme provides good defense against both low-resource and high-resource attacks.

C. Increasing Attack Costs

We have to consider the worst case of resource-based attack with end-to-end correlations. Adversaries may choose to manipulate both malicious clients and relays in RRS. As long as they own sufficient resource of malicious relays and clients, they can compromise the system anonymity to certain degree. However, as shown in Fig.4, to obtain the same effect in end-to-end correlation attacks, the cost needed to control malicious relays under the proposed scheme is much higher than that in the original Tor in most cases. In addition, the attackers have control a large number of clients. For the results shown in Fig.4, the total number of relays in the system is 3,000. For example, to achieve the same effect of 30% (900) malicious relays in the original Tor, the attackers need to control 60% (1,800) of all relays. In addition, they also need control 750 clients. Assume all malicious clients give the same grades for malicious relays. The proposed scheme significantly increases the attack cost compared to the original Tor, and in turn make resource-based attacks much harder.

VI. SELECTIVE ANONYMITY

Users use anonymous communication systems for various purposes.

- Anti-censorship users: these users want to circumvent the network censorship by their companies, ISPs, or governments. They pay less attention to anonymity and expect high performance.
- Traditional anonymous users: these users want to hide their communications with reasonable performance.

- Highly-Confidential users: these users may be officers in governments or militaries, who need to transfer sensitive information through highly anonymous systems like Tor.

Users may also change their requirements at different times. Current anonymous communication systems like Tor provide no alternative options for users. All transactions and all users are treated as the same all the time. We believe a selective anonymity scheme which balance user needs in anonymity and performance can attract more users and further improve the system anonymity with a large anonymity set. The proposed scheme can help us achieve this smoothly on Tor.

Based on the level of node reputations, we can divide them into three subsets with roughly the same number of nodes: $N_{HR}$ (High Reputation nodes), $N_{MR}$ (Medium Reputation nodes) and $N_{LR}$ (Low Reputation nodes). These sets are readjusted at the end of each measurement interval. RRS have ensured that nodes with higher reputations have both higher real bandwidth and smaller gaps between their effective bandwidth and reported bandwidth, and are more reliable. Based on the levels of performance, we propose three options for users to select:

- $O_{HP}$ (High Performance): This option allows users to achieve high performance with a decrease of anonymity. When users choosing this option, we will select all three relays for a path from $N_{HR}$. We use the weighted reputation algorithm to choose nodes from $N_{HR}$.
- $O_{NM}$ (Normal): This option is intended for moderate performance and good anonymity. When users choose option, we choose nodes randomly among all nodes, regardless the division of three subsets.
- $O_{HC}$ (High Confidence): This option enables the highest level of anonymity at the cost of performance. With this option, we select one node from each subset to build a path. Within a subset, a node is selected with the weighted reputation algorithm.

We will show that such flexible mechanism works well and meet various user requirements well. It appropriately
matches the characteristic of Tor path selection. In the following, we mainly use the moderate option $O_{NM}$ to show the implementation and evaluation of the proposed scheme and compare it with the original Tor algorithms.

A. Performance of the Proposed Flexible Scheme

We now evaluate the performance of the proposed selective mechanism. Assume the total number of nodes in the system is 3,000, a typical number in the current Tor. All relays are first divided into three subsets with the same number of nodes (1000). For example, as shown in Fig.5, three ranges are: [0, 0.31] for $N_{LR}$ (Low Reputation nodes), [0.31, 0.57] for $N_{MR}$ (Medium Reputation nodes) and [0.57, 1] for $N_{HR}$ (High Reputation nodes). These ranges are adjusted at the end of each measurement interval. To verify the performance of the proposed mechanism, we evaluate its performance and anonymity with three options $O_{HP}$ (High Performance), $O_{NM}$ (Normal) and $O_{HC}$ (High Confidence).

Three flexible options affect both system performance and node reputations, as shown in Fig.6. In the case of $O_{HP}$, relays with high reputation are often used to achieve the highest practical performance. For $O_{NM}$ relays are equally chosen to provide moderate performance. For $O_{HC}$, all classes are forcibly selected including low reputation class $N_{LR}$, so its performance is the lowest.

B. Anonymity Effect

The proposed options affect system anonymity. Intuitively, when we select relays with various reputations, more evenly, the system is more secure.

We define anonymity using information theory similar to [4, 5]. In the original Tor, an anonymity set include the set of honest relays in the system, denoted as $A = \{a_1, a_2, \ldots, a_n\}$, and $n = |A|$. Assume $X = \{x_1, x_2, \ldots, x_n\}$ is a discrete random variable and its probability density is denoted as $p_i = P(x_i)$, and $\sum_{i=1}^{n} p_i = 1$. Thus the entropy of $X$ is defined as $H(X) = -\sum_{i=1}^{n} p_i \log_2(p_i)$. Assume $H_M$ is the maximum entropy, and $H_M = \log_2(n)$. The information that an attacker can acquire is represented as $H_M - H(X)$. So the anonymity of a communication system is defined as

$$d = 1 - \frac{H_M - H(X)}{H_M} = \frac{H(X)}{H_M}$$

(3)

Clearly, $0 \leq d \leq 1$, and $d=0$ when there is only one element in the anonymity set. Or, when certain $p_i=1$, the anonymity of the system is minimal as $d=0$. While $p_i=1/N$, the system anonymity reach its maximum as $d=1$.

When considering the proposed mechanism, we revise formula (3) to formula (4) as:

$$d' = \frac{H(X)}{H_M} + \frac{H_2(X)}{H_M} + \frac{H_3(X)}{H_M}$$

(4)

where $H_1(X)$ stands for the entropy of subset $N_{NR}$, $H_2(X)$ for the entropy of subset $N_{MR}$ and $H_3(X)$ for the entropy of subset $N_{LR}$. We set three subsets with equal size. For option $O_{HP}$, only relays in $N_{NR}$ are used, thus $H_2(X) = 0$, and $H_3(X) = 0$. For option $O_{NM}$, relays are selected based on the weight of their reputation values. Assume an attacker consider relays from subset $N_{NR}$, $N_{MR}$, and $N_{LR}$ proportionally with weight $r_1$, $r_2$, and $r_3$, respectively. Assume $r_1+r_2+r_3=1$, and $r_1/r_2=10, r_2/r_3=10$. For option $O_{HC}$, three subsets are treated similarly. Assume an attacker may consider relays in three ways:

- Case 1: An attacker views all relays of the same equally.
- Case 2: An attacker considers relays proportional to their reputation values.

The results of $d'$ under these cases are shown in Fig.7. In each case, the highest anonymity is achieved with option $O_{HC}$, while the lowest is achieved with option $O_{HP}$. Fig.7 also shows that the system anonymity achieved with three options are as expected: the highest is with option $O_{HC}$; while the lowest is with option $O_{HP}$.

The experimental anonymity is not as obvious as theoretical analysis discussed in the above, because users with different requirements mix together in reality. Unless one type of users becomes the majority, attackers cannot know about the use of different relay classes.

Both performance and anonymity analysis show that the proposed scheme brings expected results to users to meet their requirements.
VII. RESISTANCE TO REPUTATION AND OTHER ATTACKS

A. Threat Analysis

We first consider the attacks aiming at the recommendation system. We categorize attacks on RRS to three levels based on their objectives and influences as follows.

- **Central level** – In this level, attackers mainly consider central servers, i.e., directory servers in Tor. Because in RRS, directory servers do not record extra communication relationships, damaging central servers will not expose historical patterns. Undoubtedly, after controlling central servers, attackers can forge the reputation list and then impact anonymity using subsequent attacks. However, there are many more direct and feasible attacks besides damaging reputation system, such as publishing a malicious consensus file. So attacks on this level with RRS will cause no more impact on anonymity than on the original Tor.

- **System level** – In this level, adversaries focus on the reputation system itself, such as slander attack, flatter attack, Sybil attack, newcomer attacker, and free riding attack. However, most attacks on this level mainly aim at damaging reputation incentives or evading reputation punishments, which are not directly related to anonymity. Some attacks may influence anonymity indirectly. For instance, a high reputation node passively working as an entry/exit will observe more connections. The observations can be used for correlation attacks. In general, threats on this level can amplify the effect of existed attacks indirectly but cause very limited damages on anonymity. We will show that RRS provides high resistance to system-level attacks.

- **Attribute level** – In this level, reputation values are the attacking target. This kind of attacks is common on the current Tor. Because the reported bandwidth in the original Tor plays a similar role as reputations in RRS, heavily affecting the path selection algorithm. These attacks will bring no more damages on RRS than on the original Tor.

In practical, there are many potential attacks to reputation systems. So we categorize main attack methods to three levels, and then evaluate the typical cases of each level. Consistent with the analysis, we find that most attacks under the threat model can be limited.

B. Resistance to Reputation Attacks

We must prevent the RRS from attacks on reputations. We have discussed three levels of reputation attacks: central level, system level and attribute level. Central level attacks will cause most severe damages to the system. However, the damages of reputation-based attacks will not exceed the damages if direct attacks to central servers. For system level and attribute level attacks, adversaries may control only clients, only relays, or both, based on their ability. When attackers only control clients, they can reduce the reputation of specific relays. When controlling only relays, adversaries can forge their own uptime and reported bandwidth. However, they cannot control their effective bandwidth graded by honest clients. Only when manipulating both
reducing reputation of malicious relays and elevating reputation of honest relays. So we mainly focus on the evaluation of attacks controlling both relays and clients.

We define $atk_{ratio}$ as the ratio of the number of lying grades over the number of all grades in the current interval. Assume malicious relays provide the lowest effective bandwidth but report their uptime and reported bandwidth as max. Malicious clients give all honest relays the minimal value and give all malicious relays the max value. The results are shown in Fig.8. The average reputation will first fall a little but then keep increasing with the rise of $atk_{ratio}$ and the percentage of malicious nodes. While looking for the average performance, $atk_{ratio}$ impacts slightly under the same ratio of malicious nodes. The percentage of malicious relays is the key factor influencing the real performance of the system. When both $atk_{ratio}$ and the ratio of lying relays are small, the effective bandwidth curve decrease is slow and smooth. Only with high attack strength from relays, the average performance decrease evidently. However, when the power of attackers is sufficient high, they can certainly cause serious damages for any anonymous systems.

We also calculate costs of attackers in reputation-based attacks. The costs of such attacks mainly include client IP addresses and the number of malicious relays. The cost of client IP addresses increases linearly with $atk_{ratio}$, as $cost = atk_{ratio} \times (C \times \Delta t)/(K \times I))$. Malicious relays are also key resources in this kind of attacks, which heavily affect the effect of attacks. The implementation of RRS with proper policies restricts the effect of these attacks and increases their cost significantly.

C. Resistance to Other Attacks on Tor

Several categories of security issues and attacks to Tor have not been thoroughly addressed. We will discuss how the proposed scheme helps us deal with Tor attacks [17-25] as follows.

- Passive Traffic Analysis attacks: These attacks include website fingerprinting attack [18], end-to-end traffic confirmation attack [19], and traffic classification specific. Opponents observe and compare traffic patterns such as packet size, flow length, timing characteristic [20], between two ends, or compare with collected website fingerprints. RRS is unrelated with this level which can be mitigated by padding or increasing computational complexity of traffic correlation. Actually, RRS will not bring new identifiable traffic patterns to the current Tor. The communication of reputations is in encryption and no new communication relation is created.

- DoS attack: A DoS attack is still an open problem on the Internet. Making several directory servers or thousands of relays unavailable is not very hard for powerful attackers [17, 21]. The proposed scheme cannot deal with it neither. However, comparing with the original Tor, the proposed scheme learns the network status quickly. We can then respond and adjust based on the reputation values in time, similar to the effect shown in Fig.2.

- Routing zones attack: In real communications, a path traverses many autonomous systems (ASes). It is an unnoticed but critical security issue that less than two percent of ASes accounted for over half of the connections in Tor [24, 25]. It is verified in [24] that the distinct /16 subnet policy in the current path selection algorithm in Tor is not sufficient to solve this problem. Authors provide a new AS-aware path selection algorithm: first directory servers generate the AS topology snapshot, and then clients use the snapshot to avoid AS attacks. This level of attack does not directly affect RRS. However, AS-aware path selection is compatible with the proposed scheme, i.e. AS information can be added as a new attribute to our current reputation and AS-aware policies is able to be transferred to reputation calculation or be added to limitation policies directly. This will be our future work.

VIII. CONCLUSIONS

In this paper, we have proposed RRS and a corresponding new path selection scheme for Tor. The proposed reputation system is embedded into the existing Tor hierarchy, and helps us improve path performance. We have also demonstrated that the system is resistant to common reputation attacks. We have built a simulation platform of Tor and conducted experimented to evaluate the
proposed scheme. Our experiments have shown the effectiveness of the proposed scheme.

ACKNOWLEDGEMENT

Prof. Yingfei Dong’s current research is supported in part by US NSF Grants CNS-1041739 and CNS-1018971. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of National Science Foundation.

This work is funded by Tsinghua National Laboratory for Information Science and Technology (TNList) Cross-discipline Foundation.

REFERENCES